FReM – scalable and stable decoding with fast regularized ensemble of models

Andrés A Hoyos-Idrobo, Gaël Varoquaux, Yannick Schwartz, Bertrand Thirion

FReM – scalable and stable decoding with fast regularized ensemble of models

Andrés Hoyos-Idrobo[a,b,*], Gaël Varoquaux[a,b], Yannick Schwartz[a,b], Bertrand Thirion[a,b]

[a] *Parietal project-team, INRIA Saclay-île de France*
[b] *CEA/Neurospin bât 145, 91191 Gif-Sur-Yvette*

Abstract

Brain *decoding* relates behavior to brain activity through predictive models. These are also used to identify brain regions involved in the cognitive operations related to the observed behavior. Training such multivariate models is a high-dimensional statistical problem that calls for suitable priors. State of the art priors –eg small total-variation– enforce spatial structure on the maps to stabilize them and improve prediction. However, they come with a hefty computational cost. We build upon very fast dimension reduction with spatial structure and model ensembling to achieve decoders that are fast on large datasets and increase the stability of the predictions and the maps. Our approach, *fast regularized ensemble of models* (FReM), includes an implicit spatial regularization by using a voxel grouping with a fast clustering algorithm. In addition, it aggregates different estimators obtained across splits of a cross-validation loop, each time keeping the best possible model. Experiments on a large number of brain imaging datasets show that our combination of voxel clustering and model ensembling improves decoding maps stability and reduces the variance of prediction accuracy. Importantly, our method requires less samples than state-of-the-art methods to achieve a given level of prediction accuracy. Finally, FreM is highly parallelizable, and has lower computation cost than other spatially-regularized methods.

Keywords: fMRI; supervised learning; decoding; bagging; MVPA

1. Introduction: decoding needs stability

Decoding models predict stimuli or behavior from brain images. These models have become a standard tool in neuroimaging data analysis (Haynes and Rees, 2006; Norman et al., 2006; Varoquaux and Thirion, 2014). In clinical applications, they can be used to perform diagnosis or prognosis (Demirci et al., 2008; Fan et al., 2008). They are also used as evidence of the link between distributed activity patterns and an observed behavior (Haxby et al., 2001). Additionally, decoding used on a large variety of cognitive processes grounds a form of reverse inference (Poldrack, 2011; Schwartz et al., 2013). An appeal of decoding procedures is that they avoid multiple voxel-wise test and perform an omnibus test: "Can one predict the behavioral outcome from brain activity?"

Identifying the brain activity patterns that drive prediction of behavior is crucial for brain mapping and understanding (Gramfort et al., 2013; Mourão-Miranda et al., 2005). However achieving reliable and stable decoder maps is challenging due to the dimensionality of the problem: the number of samples is small –hundreds or less– whereas the number of features is typically the number of voxels in the brain –up to hundreds of thousands. Linear models, e.g. linear support vector machines (SVM), are often used (Pereira et al., 2009), as they have shown a good performance in a small-sample regime. In addition, their classification/regression weights form brain maps used for interpretation of the discriminative pattern (Mourão-Miranda et al., 2005).

However, the high dimensionality of the problem leads to multiple weight maps yielding the same predictive power, and some form of regularization has to be applied (Hastie et al., 2000). In across-subject settings, complex spatial and sparse penalties such as total-variation (TV) (Baldassarre et al., 2012; Michel et al., 2011) and Graph-net (Grosenick et al., 2013) help the decoder to capture the important brain regions shared across subjects. TV and its variants are considered as the state-of-the-art regularizers for brain images, as they handle local correlations present in the data. The main drawback of spatially-structured sparsity as in TV and related penalties is their computational cost.

A much cheaper alternative to these structured estimators is to use spatially-constrained clustering algorithms to perform voxel grouping. In decoding, voxel grouping is often used as part of the pipeline for stability selection of correlated voxels (Gramfort et al., 2012; Varoquaux et al., 2012; Wang et al., 2015). Additionally, it helps to improve the conditioning of the estimation problem. However, voxel grouping introduces high bias, as the patterns are constrained by the clusters shape.

One way to mitigate this bias is to use model aggregation or ensembling. These approaches have been

*Corresponding author

used to reduce the variability of the output of the decoder (Kuncheva and Rodríguez, 2010; Kuncheva et al., 2010a; Zhou, 2012). The central idea is to build a decoder by averaging the output of several "good" models. In particular, averaging linear models boils down to averaging weight maps. One way to estimate multiple models is to use bootstrap resampling to generate different training sets to fit the decoder, and then aggregate them. This approach is known as *Bagging*[1] (Breiman, 1996). It is easy to run in parallel, training each model independently. Yet, naive application of bagging to neuroimaging data induces high computational cost as the data are high dimensional, and parameters have to be set by internal cross-validation.

Decoding calls not only for hyperparameter selection, but also for model validation. Both tasks require a measure of the predictive power of the decoder. In practice, one runs two cross-validation loops –one inside the other– where each loop assesses prediction accuracy respectively for model selection and validation. Thus, investigators often train the decoder many times. These repeated calculations entail computational costs that limit day-to-day work on standard workstations. This is particularly problematic for more advanced decoders such as those with spatial regularizations that are beneficial to neuroimaging data (e.g. Grosenick et al., 2013; Michel et al., 2011; Mohr et al., 2015). In the face of growing data size, to enable good validation and ease of use on most hardware, a good decoder should be sparing on computation resources.

Contributions. Here, we propose a fast scheme to train regularized ensembles of models, FReM. It reduces the variance of the weight maps of the decoder, while ensuring high prediction accuracy. The core of this approach is to average the estimator with the best predictive power per loop inside the nested cross-validation. To benefit from spatial regularization while keeping fast run times, we show how an optional voxel-clustering can be included in the ensembling, bringing stable spatial patterns. We perform a series of classification experiments on several MRI datasets to demonstrate that ensembling regularized models gives state-of-the-art decoders. In particular, we show that they compare favorably to existing decoders in terms of prediction performance, weight-map stability, and computation time.

2. Background and prior art

2.1. Brain decoding

In neuroimaging, a decoder is a predictive model that, given n brain images, fits an external variable $\mathbf{y}$. In practice, we arrange n observed brain images composed of p voxels in a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. Linear predictive models, at the core of most decoders in neuroimaging, are then written (Hastie et al., 2009):

$$\mathbf{y} = f(\mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}), \tag{1}$$

where $\mathbf{y}$ denotes a target variable giving the experimental condition or health status of subjects, f represents the decision function in the classification; $\mathbf{w} \in \mathbb{R}^p$ denotes the weight vector/map, and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is a random error term.

In spite of a recently growing effort on the accumulation of neuroimaging data (Poldrack and Gorgolewski, 2015), the number n of samples per-class remains in the order of a few hundreds, whereas p can be hundreds of thousands of voxels ($p \gg n$). In this high-dimensional setting, there are many equivalent solutions and some form of regularization or prior is necessary to restrict model complexity. A standard approach relies on solving the following optimization problem:

$$\hat{\mathbf{w}}(\lambda) = \underset{\mathbf{w} \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \mathcal{L}(\mathbf{y}, \mathbf{X}; \mathbf{w}) + \lambda \Omega(\mathbf{w}) \right\}, \quad \lambda > 0, \tag{2}$$

where $\mathcal{L}$ is a data-fidelity term, a loss function that measures the quality of the estimator (e.g. logistic or hinge loss); Ω denotes the penalty/regularization term, and λ is the parameter that controls the amount of regularization. Two of the most often used penalties are: *1)* the ℓ_2-norm, that penalizes large $\mathbf{w}$ coefficients, and yields non-sparse solutions; *2)* the ℓ_1-norm, that promotes a small number of non-zero $\mathbf{w}$ coefficients, and yields sparse solutions (Tibshirani, 1994).

Nevertheless, as neuroimaging data exhibit strong correlations between the columns of $\mathbf{X}$, the ℓ_1-penalty yields unstable solutions as it tends to arbitrarily select only one among the correlated variables (Varoquaux et al., 2012; Yu, 2013). One way to tackle this is the use of additional spatially-informed penalties as Graphnet (Grosenick et al., 2013) or TV (total variation) (Eickenberg et al., 2015; Michel et al., 2011).

2.2. Model validation and selection

In high-dimensional settings, the number of candidate models is much larger than the number of samples. Therefore, we use regularization to constrain the complexity of the solution, and this penalization is controlled by the λ regularization parameter. The ensuing problem is then to find an optimal value for λ (i.e. finding the best bias-variance trade-off), yielding a model that exploits the richness of the data. One typically uses the predictive power of the decoder to choose the right amount of regularization.

Hyperparameters selection. In general, the setting of the hyperparameter is a data-specific choice, as it is governed by the amount of data and their signal-to-noise-ratio (SNR). The most common approach to set it is to use cross-validation to measure the predictive power for various amounts of regularization and retain the value that maximizes the predictive power across several cross-validation folds (Varoquaux et al., 2017). To assess predictive power in addition, the standard scheme is *nested*

[1] Bagging stands for Bootstrap aggregating

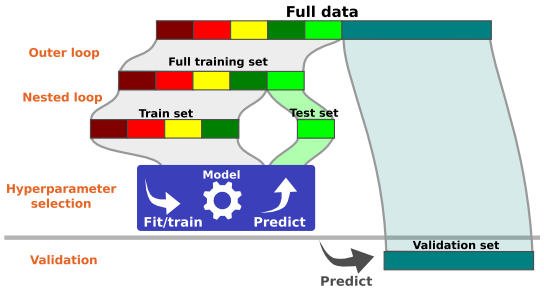

Figure 1: **Illustration of nested cross-validation:** Two cross-validation loops are run one inside the other. The inner loop is used to set the hyperparameters, whereas the outer loop is used to assess the predictive power of the decoder.

cross-validation, that consists of two cross-validation loops run one inside the other: an outer loop is used to assess the predictive power of the decoder, and an inner/nested loop is used to set the hyperparameter(s) (see Fig. 1): the train set is used to fit the decoder, while the test and validation sets are used to measure its ability to generalize to new data.

In most of the non-parametric approaches to select a regularization parameter, a suitable and finite set of l hyperparameters, $\lambda \in [\lambda_1, \ldots, \lambda_l]$ is first defined. For each cross-validation fold, one fits the decoder with all hyperparameters, and measures their prediction error on the test set. Then, one choses the λ_i value that maximizes the predictive power across folds.

2.3. Ensembling methods for better stability

Another desired characteristic of decoding algorithms is the stability to data perturbations. Stability is defined as the amount of change in the output of the decoder as a function of small changes in the data on which the decoder is trained (Shalev-Shwartz et al., 2010). These small changes or perturbations usually mean either deleting one example or replacing it with another one[2]. Stability implies that small variations in the data yield a commensurate variation in the prediction and the weight map of the decoder.

In neuroimaging, another approach to improve the stability of the estimators is to train the base estimator on several random partitions of the feature space, then select representative features according to a consensus. These partitions can be defined using various criteria, for instance: *i)* random voxels selection (Kuncheva et al., 2010b; Rondina et al., 2014); *ii)* using clustering (Varoquaux et al., 2012). Yet, these decoders need to fit more models, to accumulate selection statistics, and hence entail excessive computational costs given the number of models to fit.

[2]These perturbations are defined as sampling from the underlying distribution or replicating the experiment for a new set of data.

Model averaging. Model selection can run into some issues due to instability in the choice of the model, as any perturbation of the original data entails the selection of a completely different hyperparameter (Arlot and Celisse, 2010). Model averaging mitigates this problem by aggregating the output of several suitable models (Nemirovski, 2000). This method improves the predictive power of the base estimator, reducing the variance if the models are sufficiently uncorrelated (see Appendix A).

In particular, averaging linear models boils down to:

$$\hat{\mathbf{w}}_{\text{bagg}}(\lambda) = \frac{1}{b} \sum_{j=1}^{b} \hat{\mathbf{w}}^{(j)}(\lambda), \tag{3}$$

where b denotes the number of models to average; it is often chosen as 50 or 100, depending on the sample size and on the computation cost to train the estimator $\hat{\mathbf{w}}$ (Bühlmann and Yu, 2002). The weight map of the bagged estimator[3] displays less variance if the weight maps of the base estimators are sufficiently uncorrelated. Note that λ has to be set, requiring another nested loop of cross-validation (see Fig. 2a). Hence, choosing the parameter of the aggregated model is computationally expensive.

3. Materials and methods

3.1. Dimension reduction by feature agglomeration

In neuroimaging, dimension reduction is routinely used to alleviate problems due to high-dimensionality. A common way to select features is univariate feature screening, which uses a score (e.g. statistical test, correlation) to remove non-predictive variables. In particular, the method proposed by Dohmatob et al. (2015) operates as follows: The data $(\mathbf{X}, \mathbf{y})$ are standardized so that $\mathbf{y}$ has unit variance and zero mean, likewise each row of the design matrix $\mathbf{X}$. Additionally, $\mathbf{X}$ is smoothed with a Gaussian filter for the screening of voxels, but not during the fitting of the estimator. Then, for each voxel j one computes the absolute dot-product of $\mathbf{y}$ and the jth column of $\mathbf{X}$, $|\mathbf{X}_j^T \mathbf{y}|$. Finally, one selects the voxels with the highest $|\mathbf{X}_j^T \mathbf{y}|$ values.

However, this method does not take into account the spatial structure of brain images. Instead, we can reduce the dimension of the data by grouping similar neighboring voxels, moving from the voxel-space to a parcel-space. To do this, we can use anatomical/functional atlases or data-driven approaches.

Here we rely on a voxel grouping approach, where we use a fraction of the training data to train a clustering algorithm, finding suitable groups of features or parcellations. Then, we use these parcels on the remaining data to work at a parcel level. Formally, we define a feature-grouping matrix $\mathbf{\Phi} \in \mathbb{R}^{p \times k}$, where $k \ll p$, and each column has a constant value with support at each parcel[4]. We normalize each column to have unit ℓ_2-norm (Hoyos-Idrobo et al.,

[3]The bagged estimator is a Monte-Carlo approximation of $\mathbb{E}[\mathbf{w}]$.
[4]The feature-grouping matrix is orthogonal.

2016). To reduce the dimension, we multiply the data by the feature-grouping matrix, $\mathbf{X}_{\text{reduced}} = \mathbf{X}\,\mathbf{\Phi}$. We can also build an approximation[5] of the data, $\mathbf{X}_{\text{approx}} = \mathbf{X}\,\mathbf{\Phi}\,\mathbf{\Phi}^{\top}$.

This approach increases the SNR at the expense of spatial resolution without excluding potentially informative variables. It is often used in combination with sparse methods to alleviate their unstability when dealing with correlated variables (Bühlmann et al., 2013; Varoquaux et al., 2012).

3.2. Fast regularized ensemble of models (FReM)

Setting the hyperparameters of ensembles of models can be computationally expensive, as a single aggregated estimator requires fitting a base estimator $m \times l \times b$ times. This corresponds to three loops: *i)* m cross-validation loops to select the model, *ii)* l for the hyperparameters, and *iii)* b to find the base estimators to average (see Fig. 2a). To tackle this computational bottleneck, we average weight vectors of nested cross-validation folds at best performing hyperparameter values (in the sense of predictive power). By doing this, we can reduce the number of required fits to $l \times b$, where the number of estimators b is the number of cross-validation folds. Thus, this scheme uses one loop less than the standard bagging. In addition, we add an implicit spatial constraint using clustering of features, applying it at each fold to increase the randomness of the clusters shapes. For completeness we detail the proposed strategy in Algorithm 1 and Fig. 2b.

4. Empirical studies: stable brain decoding

In this section, we conduct a series of experiments to highlight the practical aspects of FReM in brain decoding. We use several MRI datasets to investigate their prediction performance, weight-map stability, and computation time.

4.1. Experiments on real neuroimaging data

To achieve reliable empirical conclusions, we consider a large number of different neuroimaging studies. We investigate FReM in several binary classification problems based on 8 fMRI datasets. We perform within-subject discrimination across sessions between various types of visual stimuli on the Haxby dataset (Haxby et al., 2001). In addition, we discriminate in an across subjects setting: *i)* different categories of visual stimuli from Duncan et al. (2009); *ii)* conditions with different levels of affective content with data from Wager et al. (2008); *iii)* mentalization with data from Moran et al. (2012); *iv)* famous, familiar, and scrambled faces from a visual-presentations dataset (Henson et al., 2002); *v)* left and right saccades in data from Knops et al. (2009); *vi)* relational and emotion processing, language, and gambling protocols from the human connectome project (HCP) (Essen et al., 2012); *vii)* response inhibition on openfMRI ds009 (Poldrack et al., 2013). We use the trial-by-trial (Z-score) maps computed in a first-level GLM to perform all across-subject predictions. Additionally, we predict the gender from VBM maps using the OASIS dataset (Marcus et al., 2007).

Standard preprocessing and first-level analysis were applied using SPM. The data were variance-normalized and spatially smoothed at 6 mm FWHM for fMRI data and 2 mm FWHM for VBM data.

Experimental setup. In all classification tasks, we use nested cross-validation for an accurate measure of the predictive power. We repeatedly split the data into a validation set and a decoding set. We choose validation sets of 20% the data, respecting the sample dependence structure (leaving out subjects or sessions). We set 10 folds for the outer cross validation loop.

As is standard practice in fMRI decoding (Pereira et al., 2009), we use univariate feature selection on the training set to select 20% of voxels and train the decoder on the selected features. We compare several decoders, split into two groups:

- *Non-ensembles:* Graph-net (Grosenick et al., 2013), TV-ℓ_1 (Michel et al., 2011), Log-enet (Zou and Hastie, 2005)[6], SVM-ℓ_1, and SVM-ℓ_2.

- *Ensembles:* FReM of SVM-ℓ_1, SVM-ℓ_2, both, with and without clustering. These estimators are fitted using the proposed scheme –see Algorithm 1.

We use scikit-learn (Pedregosa et al., 2011) for the Log-enet, and the SVM with ℓ_1 and ℓ_2 penalties. We use nilearn (Abraham et al., 2014) for Graph-net and TV-ℓ_1. When clustering is applied, we set the number k of clusters to 10% of the number p of voxels[7]. We rely on the fast agglomerative clustering presented in Hoyos-Idrobo et al. (2016). In brief, this algorithm iteratively performs 1-nearest neighbor grouping, reduces the graph at each iteration, then averages the input features and repeats the process until it reaches the desired number of clusters.

Regarding the ensembles of models, we use 50% of decoding sets to train the decoder[8].

In the first experiment, we empirically validate the performance of various decoders on different discrimination tasks. In particular, we measure the prediction score,

[5] This approximation can be seen as the application of an anisotropic smoothing.

[6] We don't use an SVM with elastic net penalty, as none of the solvers have an implementation, and using our own implementation will lead to unfair comparisons.

[7] We consider a useful dimension reduction range, $k \in \left[\frac{p}{20}, \frac{p}{10}\right]$. This regime gives a good trade-off between computational efficiency and data fidelity (Hoyos-Idrobo et al., 2016).

[8] In the standard bootstrap the whole dataset is resampled. However, it can be approximated with a subsampling of 50% of the data (Dümbgen et al., 2013; Praestgaard and Wellner, 1993; Shah and Samworth, 2013)

Algorithm 1 Fast regularized ensemble of models (FReM)

Require: Training data $\mathbf{X} \in \mathbb{R}^{n \times p}$, the desired number k of clusters, the sampling parameter m, the number b of estimators to aggregate, the set of l hyperparameters $[\lambda_1, \ldots, \lambda_l]$, the regularizer Ω and the loss $\mathcal{L}$.

Ensure: $\hat{\mathbf{w}}_{\text{bagg}}$

for $j = 1$ **to** b **do**

Build pseudo-dataset: $\{(\mathbf{X}^*, \mathbf{y}^*)\} \leftarrow \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^{m}$, where $\mathbf{X}^* \in \mathbb{R}^{m \times p}$, and $\mathbf{y}^* \in \mathbb{R}^m$. {draw m samples from $(\mathbf{X}, \mathbf{y})$ at random.}

Split into a training set and a testing set: $(\mathbf{X}^*_{\text{train}}, \mathbf{y}^*_{\text{train}}), (\mathbf{X}^*_{\text{test}}, \mathbf{y}^*_{\text{test}})$. {Select $\lfloor \frac{m}{2} \rfloor$ samples at random (without replacement).}

Build feature-grouping matrix: $\mathbf{\Phi}^{(j)} \in \mathbb{R}^{p \times k}$ {use Hoyos-Idrobo et al. (2016).}

Dimension reduction: $\tilde{\mathbf{X}}_{\text{red}} \leftarrow \mathbf{X}^*_{\text{train}} \mathbf{\Phi}^{(j)}$, where $\tilde{\mathbf{X}}_{\text{red}} \in \mathbb{R}^{\lfloor \frac{m}{2} \rfloor \times k}$.

Univariate feature selection: {use Dohmatob et al. (2015).}

for $i = 1$ **to** l **do**

Estimate weight map: $\hat{\mathbf{w}}^{(i)}_{\text{red}} = \underset{\mathbf{w} \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \mathcal{L}(\mathbf{y}^*, \tilde{\mathbf{X}}_{\text{red}}; \mathbf{w}) + \lambda_i \Omega(\mathbf{w}) \right\}$, $\hat{\mathbf{w}}^{(i)}_{\text{red}} \in \mathbb{R}^k$.

end for

Select the best model: $\hat{\mathbf{w}}^{(j)}_{\text{best}} \leftarrow$ the $\hat{\mathbf{w}}^{(i)}_{\text{red}}$, $i \in [0, \ldots, l]$ with the best performance on the test set.

Return to voxel-space: $\hat{\mathbf{w}}^{(j)}_{\text{approx}} = \hat{\mathbf{w}}^{(j)}_{\text{best}} \mathbf{\Phi}^{(j)\mathsf{T}}$, where $\hat{\mathbf{w}}_{\text{approx}} \in \mathbb{R}^p$.

end for

return $\hat{\mathbf{w}}_{\text{bagg}} \leftarrow \frac{1}{b} \sum_{j=1}^{b} \hat{\mathbf{w}}^{(j)}_{\text{approx}}$

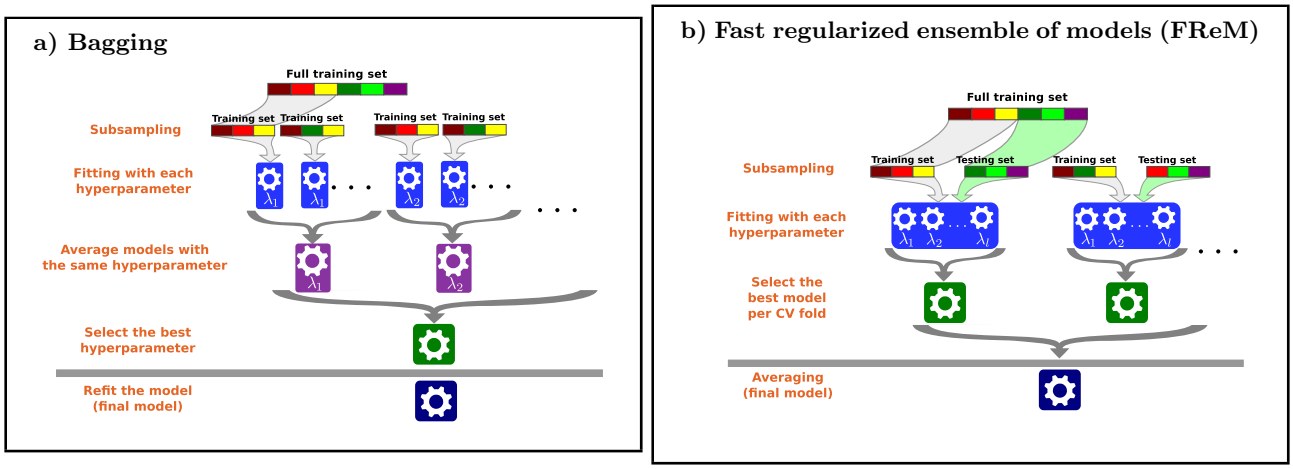

Figure 2: **Regularized ensemble of models:** Fast regularized ensemble of models uses one loop less than bagging.

computation time, and the correlation across folds (stability). In a second experiment, we explore the training speedup of decoders in a multi-core setting. Then we evaluate the similarity between weight maps obtained using all the data and the ones obtained for different sample sizes (small-sample recovery).

5. Results: evaluating decoder performance

5.1. Comparing FReM to bagging

We explore the performance of FReM and bagging on two discriminative tasks: discrimination of famous and scrambled faces from the Henson (2006) dataset, and discrimination of response inhibition on openfMRI ds009 (Poldrack et al., 2013). The regularization parameters of bagging are set by 10-fold inner cross validation.

Fig. 3 summarizes the relative performance of FReM and bagging with respect their base estimators across discriminative tasks. FReM of ℓ_2-penalized models with and without clustering display better weight map stability than bagging of SVM-ℓ_2. FReM of ℓ_1-penalized models without clustering has worst prediction accuracy, with a loss of 2% with respect to bagging. Regarding FReM with clustering, it has the same prediction accuracy as bagging, and it displays better weight map stability. For both penalties, FReM yields computation speedups, > 5 times faster, while preserving the gains in prediction accuracy.

5.2. Benchmarking decoders

For all discriminative conditions, we measure the prediction error on several left-out validation sets to assess the predictive power of the decoders. Additionally, we measure the correlation between the weight maps obtained

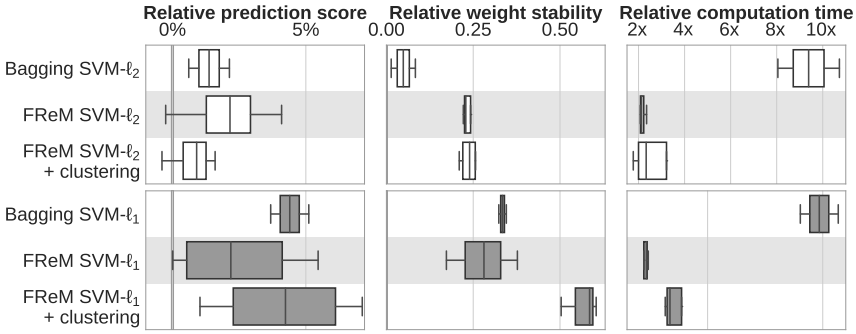

Figure 3: **Comparison of the performance of FReM and bagging:** Comparison on two discriminative tasks, discrimination of famous and scrambled faces from the Henson (2006) dataset, and discrimination of response inhibition on openfMRI ds009 (Poldrack et al., 2013). *(top)* Relative performance to SVM-ℓ_2. *(bottom)* Relative performance to SVM-ℓ_1. For both penalties, FReM improves the stability of weight maps, and it is > 5 times faster than bagging.

in each cross-validation fold, and the computation time required to train the decoder. To perform this analysis, we separate the datasets into two types: within-subject and across-subject. Throughout this experiment, we set the number b of estimators used in FReM to 50. This choice is discussed in Fig. 8.

Fig. 4 summarizes the relative performance with respect to the mean across decoders per discriminative task. In within-subject settings, all sparse methods have good prediction performance. Decoding using the standard SVM with both ℓ_1 and ℓ_2 penalty is fast, but the weight maps are less stable than the ones found by sparse structured methods –i.e. Graph-net and TV-ℓ_1. However, these complex penalties come with higher computation costs. As expected, using FReM reduces the variance of the prediction, while increasing the stability of the weight maps. This effect is enhanced when including a clustering step. The computation time of FReM with or without clustering is less than that of structured sparse classifiers.

For the discriminative task across subjects, FReM consistently improves prediction accuracy as well as the stability of the weight maps, while keeping a computation cost less than structured sparse classifiers. In addition, the use of spatial clustering has a beneficial impact on the spatial stability. In all the presented cases, FReM improves stability of the weight maps of the base estimator, while preserving the prediction accuracy. Note however that, for the combination of the SVM-ℓ_2 and clustering, it does not display any additional benefit.

Table 1 shows the comparison between each decoder and the decoder displaying the best prediction accuracy, namely FReM of SVM-ℓ_1 for within-subject problems, and FReM of SVM-ℓ_2 with clustering for across-subjects problems. These results confirm the above observations.

Experiments on simulated data. Unlike brain imaging datasets, simulations open the door to measuring the actual support of a decoder, as well as its prediction accuracy. Briefly, we generate data with 2 classes and 1728 voxels with temporally auto-correlated Gaussian noise (more details in Appendix B). The dataset contains a decoding set of 200 samples. We choose validation sets of 20% the data. We set 10 folds for the outer cross validation loop. We assess the support recovery of each decoder by building the precision-recall curve. This curve is generated by comparing the ground truth weight maps with the coefficients of the decoder after applying different thresholds.

Fig. 5 shows the prediction accuracy of each decoder across cross-validation folds on simulated data. FReM SVM-ℓ_1 with and without clustering have the best prediction accuracy on simulated data, followed by SVM-ℓ_1, and Graph-net. TV-ℓ_1 display slightly worst performance. Log-enet, SVM-ℓ_2, FReM SVM-ℓ_2 with and without clustering have a low predictive performance. FReM of ℓ_1 models increase the prediction accuracy, whereas for ℓ_2 models this remains similar.

Regarding the computation time, SVM-ℓ_2, FReM of SVM-ℓ_2 with and without clustering are the fastest methods to train, followed by SVM-ℓ_1, FReM of SVM-ℓ_1 with and without clustering. Log-enet, and Graph-net are slightly slower, whereas TV-ℓ_1 is at least 4 times slower than Graph-enet. On simulated data, FReM requires the same computation time than its base estimator.

Fig. 6 displays the precision-recall curve, which serves as an indicator of support recovery of the underlying spatial activation map. FReM of SVM-ℓ_1 with clustering, FReM of SVM-ℓ_2 with clustering, and TV-ℓ_1 display have the best support recovery. They are followed by Graph-net, SVM-ℓ_2, and FReM SVM-ℓ_2. In this experiment, Log-enet and SVM-ℓ_1 both fail to recover the support of the activation signal. We can see that FReM consistently improves the support recovery of base estimators. In particular, FReM of SVM-ℓ_1 with clustering displays the best trade-off between prediction accuracy and support recovery. It has a predictive performance similar to Graph-net, and a support recovery close to TV-ℓ_1.

5.3. Delineating brain regions

An important question regarding brain decoders is whether they segment well the brain regions that support the decoding. The validation of this question is hard, yet there is evidence that relying on ensembles of models is a good approach (Leung and Barron, 2006; Zhou, 2012). Fig. 7 displays the decoder maps for the face-recognition tasks. For these tasks, we expect prediction to be driven

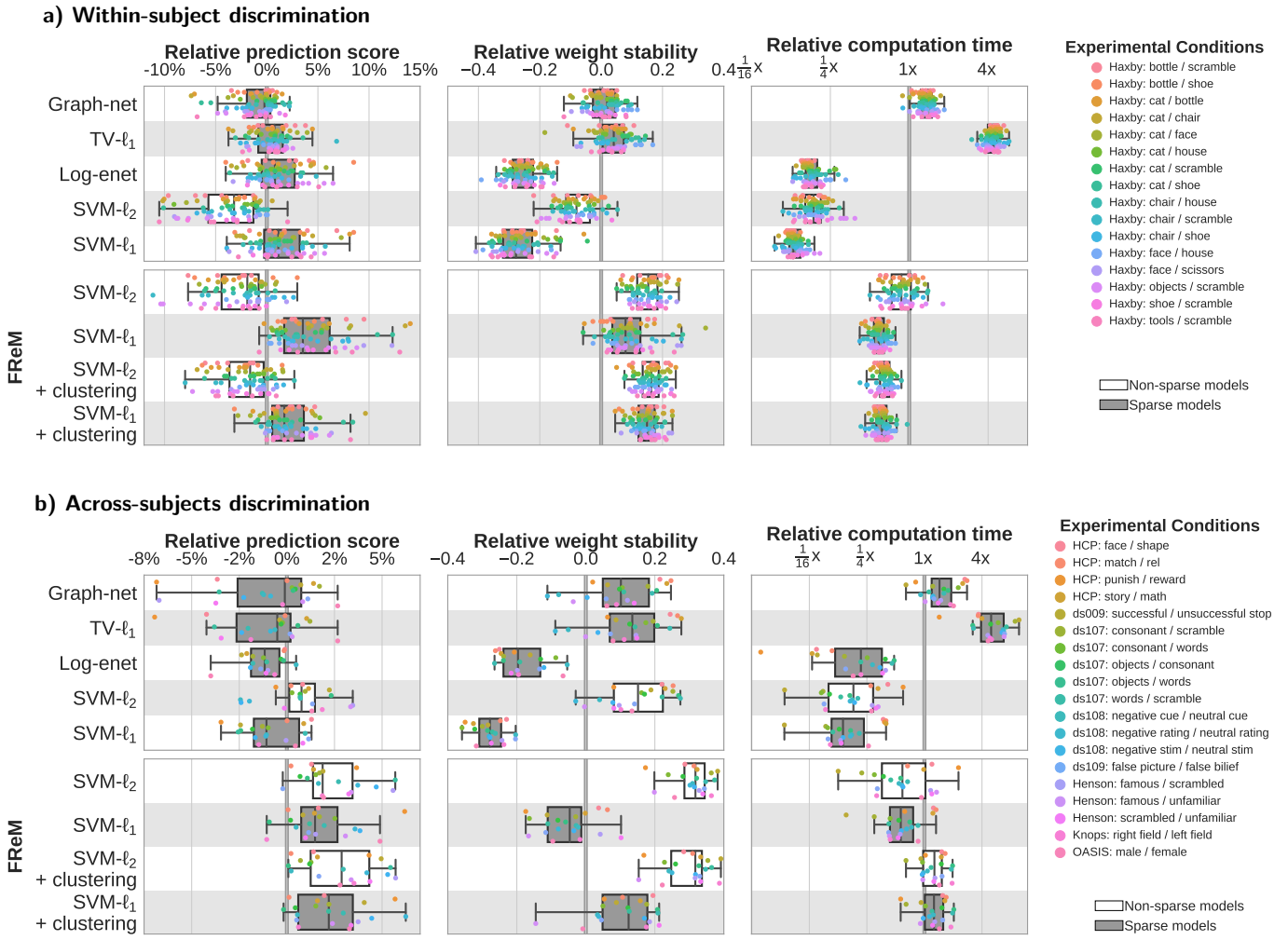

Figure 4: **Relative performance:** Relative prediction accuracy, weight stability and computation time for different classification tasks. Values are displayed relative to the mean over all the classifiers. *a)* FReM improves prediction accuracy, and when applied with clustering it also reduces the variability. The ensembles of models consistently improves the weights stability, with a computation time smaller than TV-ℓ_1 and Graph-net. *b)* FReM with and without clustering slightly improves the prediction accuracy, while significantly improving the stability. Note that the computation time is obtained using a single CPU, see Figure 9 for parallel-computing timings. Results without smoothing are presented in Figure C.2.

a) Within-subject discrimination

	Classifier	prediction score	weight stability	computation time
	Graph-net	1.4e-3 >	1.6e-4 >	**7.8e-14 <**
	TV-ℓ_1	6.3e-5 >	8.3e-3 >	**7.8e-14 <**
	Log-enet	5.9e-3 >	**7.8e-14 >**	**7.8e-14 >**
	SVM-ℓ_2	1.4e-9 >	**8.9e-12 >**	**7.8e-14 >**
	SVM-ℓ_1	2.4e-3 >	**7.8e-14 >**	**7.8e-14 >**
FReM	SVM-ℓ_2	1.5e-8 >	1.7e-4 <	**2.1e-13 <**
FReM	SVM-ℓ_1		**Reference**	
FReM	SVM-ℓ_2 + clustering	2.8e-7 >	3.1e-6 <	1.1e-8 <
FReM	SVM-ℓ_1 + clustering	9.3e-3 >	1.3e-3 <	5.2e-6 <

b) Across-subjects discrimination

	Classifier	prediction score	weight stability	computation time
	Graph-net	**5.2e-14 >**	**1.4e-33 >**	2.1e-5 <
	TV-ℓ_1	**8.8e-17 >**	**1.4e-33 >**	**1.4e-33 <**
	Log-enet	**1.3e-15 >**	**2.7e-30 >**	**2.7e-30 >**
	SVM-ℓ_2	1.1e-7 >	**1.5e-33 >**	**1.4e-33 >**
	SVM-ℓ_1	**6.3e-14 >**	**6.3e-32 >**	**6.3e-32 >**
FReM	SVM-ℓ_2	0.7 >	1.2e-6 <	**1.4e-24 >**
FReM	SVM-ℓ_1	0.5 >	**1.4e-33 >**	**1.4e-33 >**
FReM	SVM-ℓ_2 + clustering		**Reference**	
FReM	SVM-ℓ_1 + clustering	0.7 >	**1.4e-33 >**	1.6e-3 >

Table 1: **Comparison of performance:** Each decoder is compared with a reference. The values correspond to Bonferroni-corrected p-values obtained by paired Wilcoxon rank test. The direction in the parenthesis denotes the sign of the mean difference, and bold text denotes a significant results ($p < 10^{-10}$). Results without smoothing are presented in Table C.2.

by the functional areas of the visual cortex (Grill-Spector and Malach, 2004). Indeed, the maps outline regions in known visual areas –e.g. the fusiform face area (PPA).

In both within-subject and across-subject datasets, the

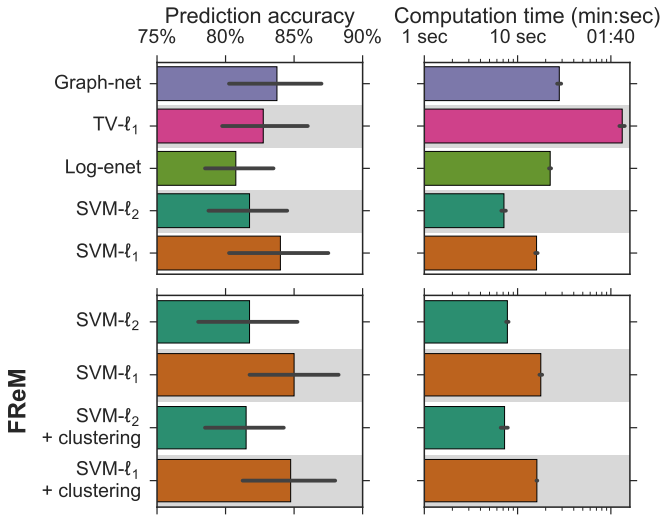

Figure 5: **Behavior on simulated data:** FReM of SVM-ℓ_1 with and without clustering display the best predictive performance, followed by SVM-ℓ_1, Graph-net, and TV-ℓ_1. Log-enet and ℓ_2 penalized methods display low performance. ℓ_2 methods are the fastest to compute, followed by SVM-ℓ_1, FReM of SVM-ℓ_1 with and without clustering, Log-enet, and Graph-net. TV-ℓ_1 is the slowest method.

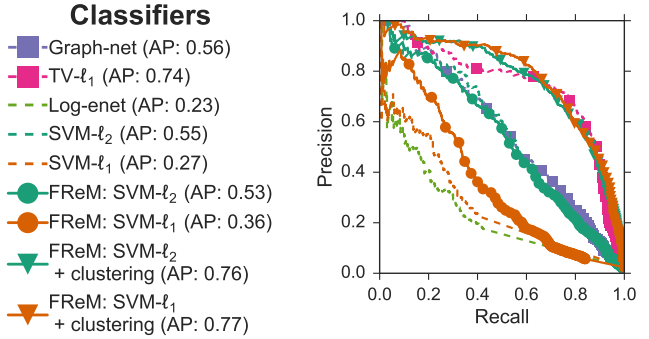

Figure 6: **Support recovery on simulated data:** Evaluation of support recovery for different decoders. The average precision (AP) is presented in parenthesis. FReM of SVM-ℓ_1 with clustering, FReM of SVM-ℓ_2 with clustering, and TV-ℓ_1 have the best recovery performance. They are followed by Graph-net, SVM-ℓ_2, and FReM of SVM-ℓ_2 without clustering. Log-enet and SVM-ℓ_1 fail to recover the ground truth. FReM generally improves the performance of base estimators.

SVM-ℓ_1 maps are unstructured, and even if using FReM of this model improves the stability of the weight maps, these maps remain scattered with a large number of small clusters. However, the use of clustering yields less and larger clusters, with maps that are qualitatively similar to TV-ℓ_1 maps. Graph-net and SVM-ℓ_2 display similar behavior, yielding various small clusters around large clusters of activation. In the case of SVM, the use of FReM can reduce the number of small clusters. The combination with clustering enhances this effect. Note that setting the threshold to visualize regions is difficult task as the noise level is unknown.

5.4. Setting the number of estimators to ensemble

The choice of the number b of estimators to combine also affects the stability of the weight maps and the running time. The number b can be interpreted as a "smoothing" parameter (Bühlmann and Yu, 2002), and the computation time that one is willing to pay to train a decoder. We measure the performance of ensembles of classifiers on three different datasets and across 10 folds of cross-validation. In practice, we often use a number of estimators between 50 and 100, but to verify if the model converges, we consider here a range from 10 to 640 estimators.

Fig. 8 shows that for ensembles of classifiers, prediction accuracy does not depend on the number of estimators, whereas the computation time is almost linear ($t \propto b^\gamma$, where $\gamma \approx 1$). We use the running time as a constraint to finally set the number of estimators to 50, as the weight stability of non-sparse classifiers are at least 95% of the asymptotic optimum. In addition, this is a good compromise between stability and computation cost. Hence, we use this number of estimators throughout all experiments.

6. Results: parallel computing of brain decoders

One important feature of ensembling models is scalability, as these methods can be trained in parallel in a multicore, shared-memory environment. This corresponds to current standard workstations, which frequently have a large number of CPUs[9]. Here, we measure the training time of various decoders across 5 folds of cross-validation. We perform face-discrimination tasks on two datasets with different sizes.

Fig. 9 shows that, in general, there is not an ideal decrease in the computation time as more CPUs are added. The SVM with ℓ_1 and ℓ_2 penalty are the fastest. In contrast, TV-ℓ_1 is the slowest, followed by Graph-net. In both datasets, FReM displays most of the speed up at 10 CPUs, and reaches a minimum at 20. These methods are much

[9]Parallel computation was run using joblib `https://pythonhosted.org/joblib/` to use multiple cores on a single computer with Python "multiprocessing". Benchmarks were done on Intel Xeon E5-2697 CPUs, clocked at 2.7GHz, with 12 cores per CPUs, on a single Linux (Ubuntu 16.04) computer.

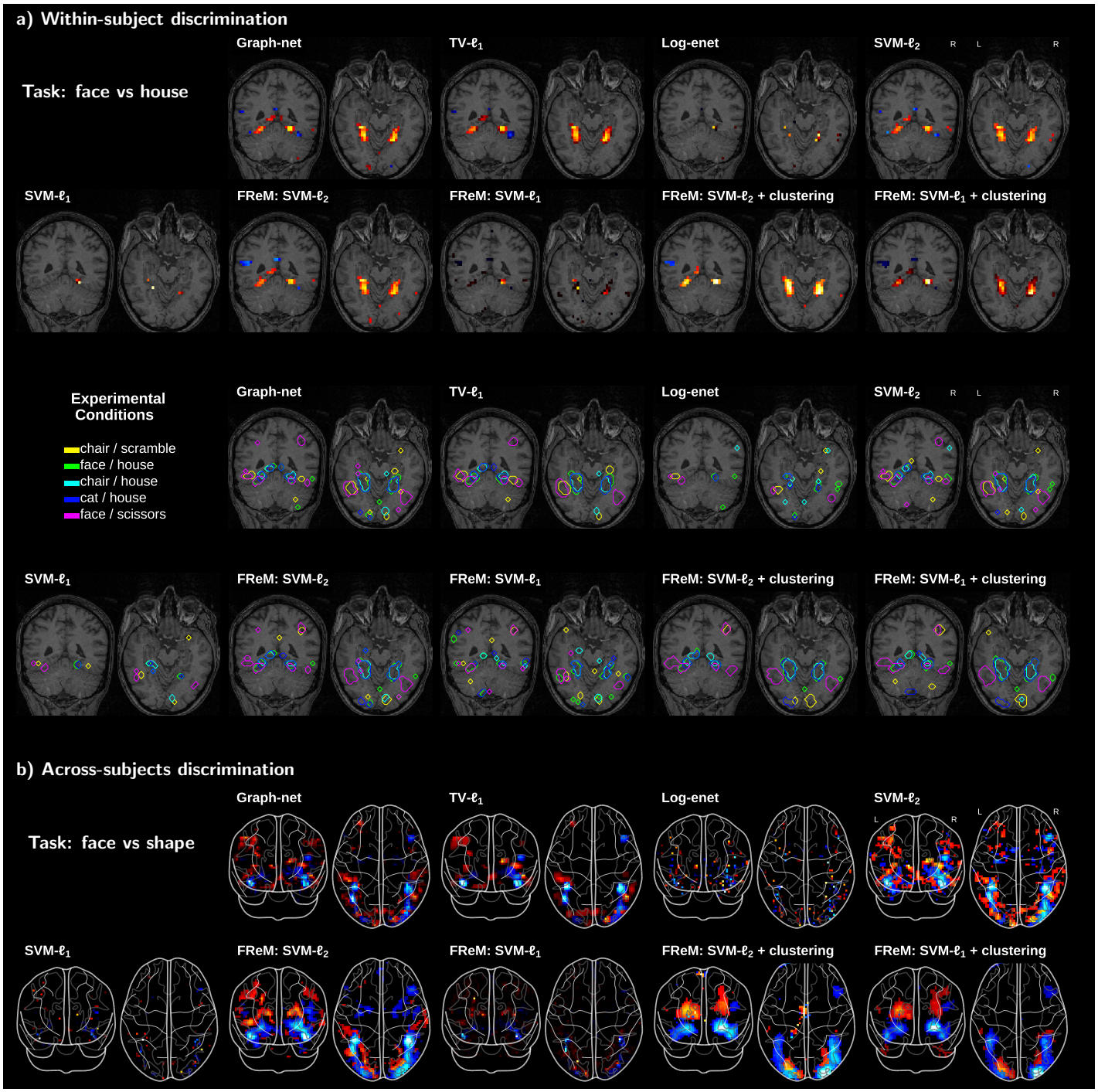

Figure 7: **Qualitative comparison of decoder weight maps:** Weight maps for different discriminative tasks on the Haxby and HCP datasets. The maps are thresholded at the 99 percentile for visualization purposes. In both dataset, *(top)* illustration of weight maps for the face-recognition task; *(bottom)* outlines of the other tasks. The weight maps obtained with TV-L1 and FReM methods with clustering display a prediction driven by the functional areas of the visual mosaic, such as: primary visual areas, lateral occipital complex, the face and place specific regions in the fusiform gyrus. An example of the difference of outlines between FReM and its base estimator see Figure D.3.

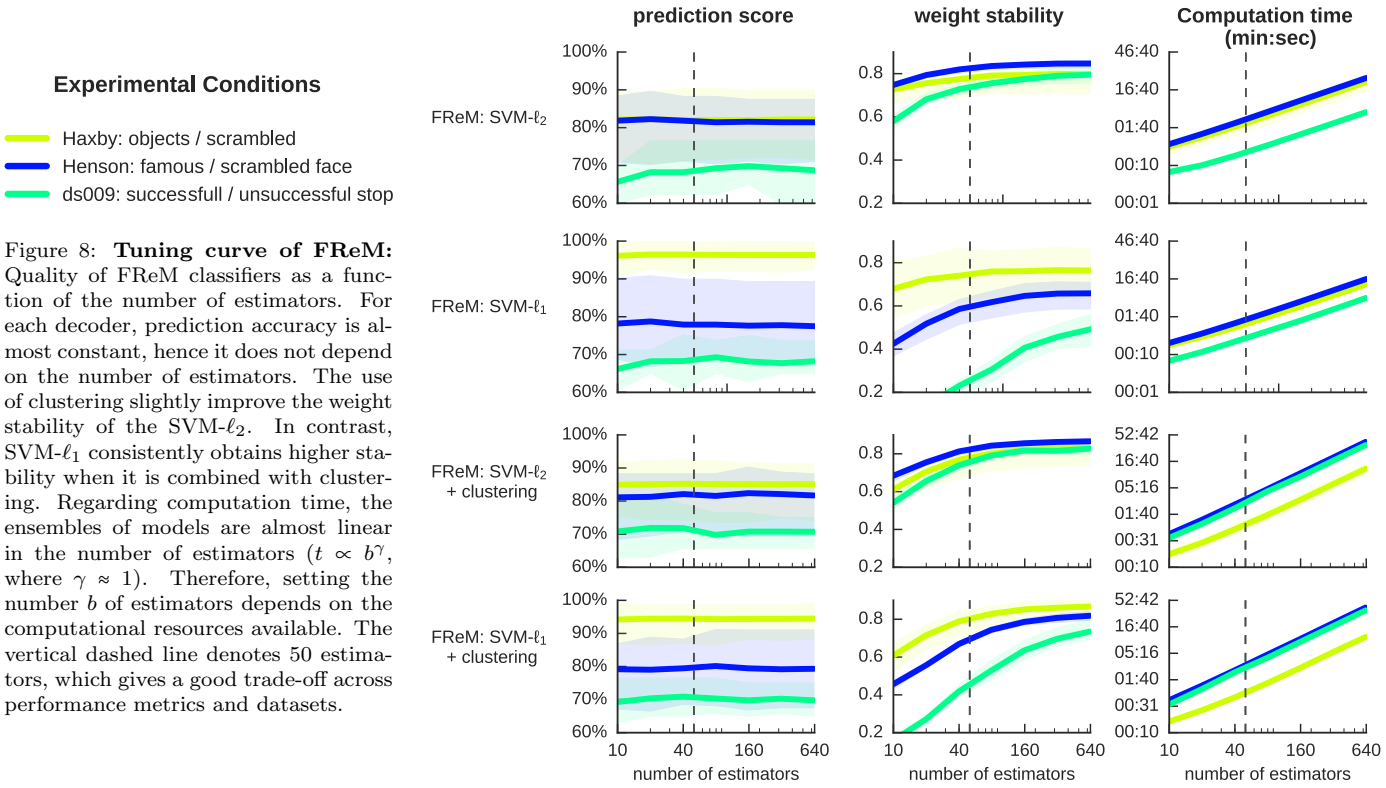

Figure 8: **Tuning curve of FReM:** Quality of FReM classifiers as a function of the number of estimators. For each decoder, prediction accuracy is almost constant, hence it does not depend on the number of estimators. The use of clustering slightly improve the weight stability of the SVM-ℓ_2. In contrast, SVM-ℓ_1 consistently obtains higher stability when it is combined with clustering. Regarding computation time, the ensembles of models are almost linear in the number of estimators ($t \propto b^\gamma$, where $\gamma \approx 1$). Therefore, setting the number b of estimators depends on the computational resources available. The vertical dashed line denotes 50 estimators, which gives a good trade-off across performance metrics and datasets.

faster that Graph-net. The combination of FReM with clustering does not increase computation cost as we use a fast clustering algorithm.

7. Results: small-sample recovery behavior of decoders

In fMRI, despite growing efforts in data accumulation (Essen et al., 2012; Poldrack et al., 2013), the sample size remains small in comparison with the number of voxels. Therefore, an important aspect of the brain decoders is their sample complexity –i.e. the number of samples required to bound the estimation error. Yet, assessing the recovery of weight maps is difficult, as we do not have access to the asymptotic result. To bypass this problem, we measure the similarity between the weight maps obtained with different sample sizes and the ones obtained using the whole dataset. This gives us an intuition on the small-sample recovery capacity.

Fig. 10 shows that across datasets, FReM of SVM-ℓ_2 with and without clustering, and FReM of SVM-ℓ_1 with clustering are consistently the best. In contrast, the SVM-ℓ_1 and Log-enet fail to recover the final weight maps. TV-ℓ_1, Graph-net, and SVM-ℓ_2 have a good performance on both datasets. FReM of SVM-ℓ_1 outperforms these methods on the within-subject discrimination, as the weight similarity rapidly increases. On across-subject datasets, FReM of SVM-ℓ_1 with clustering has almost the same performance as TV-ℓ_1, Graph-net, and SVM-ℓ_2, differing only after using 80% of the data for training.

8. Discussion and conclusion: using FReM

We have introduced a fast strategy to train regularized ensembles of models, FReM, that improves the stability of brain decoders. This scheme is summarized as follows: *i)* For each fold of the nested cross-validation loop, we select the estimator with the best predictive power; *ii)* we build an estimator by storing the models for all folds of cross-validation and averaging them. This approach differs from the stability selection methods (Meinshausen and Bühlmann, 2010; Varoquaux et al., 2012), as here we use the amplitude of predictive weight maps for the model aggregation, and not only their support.

Using FReM. The predictive power of FReM is not very dependent on the number of estimators used during the aggregation step. On the other hand, the stability of the resulting decoder improves as more estimators are used. On across-subjects datasets, the use of clustering improves the stability of sparse methods. In terms of computation time, this scheme displays an almost linear complexity in the number of estimators. Thus, setting the number of estimators is an arbitrary choice, it depends only on the computation resources available.

Comparing decoders. In both within and across subjects datasets, FReM has shown an improvement of the performance of the base estimator. This strategy reduces the variance of predictive power and increases the stability of weight maps of the base estimator. It also improves the small-sample behavior of the base estimators, boosting the

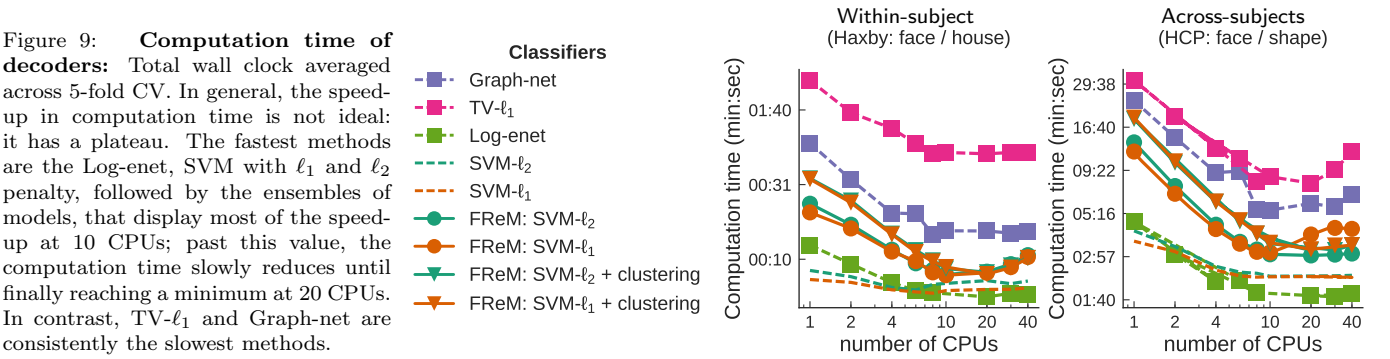

Figure 9: **Computation time of decoders:** Total wall clock averaged across 5-fold CV. In general, the speed-up in computation time is not ideal: it has a plateau. The fastest methods are the Log-enet, SVM with ℓ_1 and ℓ_2 penalty, followed by the ensembles of models, that display most of the speed-up at 10 CPUs; past this value, the computation time slowly reduces until finally reaching a minimum at 20 CPUs. In contrast, TV-ℓ_1 and Graph-net are consistently the slowest methods.

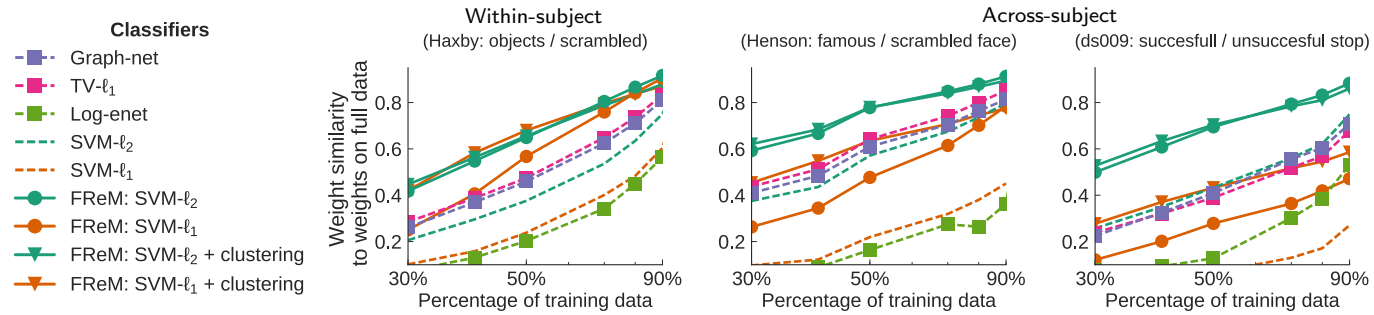

Figure 10: **Small-sample recovery behavior of decoders:** Evaluation of the correlation between decoder weight maps for each sample size and the ones obtained using the full dataset. FReM of SVM-ℓ_2 with and without clustering have consistently the best small-sample recovery performance, followed by TV-ℓ_1, Graph-net, and SVM-ℓ_2. *Within-subject)* FReM of SVM-ℓ_1 rapidly increase their weights similarity, outperforming TV-ℓ_1 and Graph-net. *Across-subjects)* the combination of clustering and FReM of SVM-ℓ_1 has a performance as good as TV-ℓ_1. In both discrimination tasks, SVM-ℓ_1 and Log-enet fail to recover the final decoder weight maps.

consistency of weight maps. In addition, this scheme leads to qualitatively good brain regions delineation.

In terms of computation time, the use of FReM yields decoders that are slower than the base estimators. But they are faster than state-of-the-art decoders, namely TV-ℓ_1 and Graph-net. Nevertheless, the speed-up of FReM can be enhanced by parallelizing the training of each estimator to aggregate. Thus, the training time is dominated by the fitting of each decoder. However, this gain is not ideal, and there is a plateau in the speed-up when the number of CPUs increases.

Regarding the combination of FReM and clustering, it has a spatial denoising effect on the resulting weight maps. This is reflected in the reduction of the variability and an increase in the prediction power. But, when the base estimator is a sparse method, the averaging step reduces the sparsity, and yields weight maps with many small values instead.

Our extensive empirical validation (36 decoding tasks, taken from 9 datasets) shows that the FReM, in particular using a SVM-ℓ_2 with clustering, gives the best stability-prediction trade-off, with a good qualitative delineation of brain regions. Averaging several "good" estimators yields a model that can adapt to the properties of the noise present in the data. Hence, it is more robust to violations of modeling assumptions. The application of this scheme with clustering benefits to the spatial stability of weight maps, a key requirement of any cross-population study of functional imaging signals.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Grant Agreement No 720270 (Human Brain Project SGA1).

References

Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Muller, A., Kossaifi, J., Gramfort, A., Thirion, B., Varoquaux, G., 2014. Machine learning for neuroimaging with scikit-learn. Frontiers in neuroinformatics 8.

Arlot, S., Celisse, A., 2010. A survey of cross-validation procedures for model selection. Statistics Surveys 4, 40–79.

Baldassarre, L., Mourao-Miranda, J., Pontil, M., 2012. Structured sparsity models for brain decoding from fMRI data, in: PRNI, p. 5.

Breiman, L., 1996. Bagging predictors. Machine learning 24, 123–140.

Bühlmann, P., Rütimann, P., van de Geer, S., Zhang, C.H., 2013. Correlated variables in regression: clustering and sparse estimation. Journal of Statistical Planning and Inference 143, 1835–1871.

Bühlmann, P., Yu, B., 2002. Analyzing bagging. Annals of Statistics 30, 927–961.

Demirci, O., Clark, V.P., Magnotta, V.A., Andreasen, N.C., Lauriello, J., Kiehl, K.A., Pearlson, G.D., Calhoun, V.D., 2008. A review of challenges in the use of fMRI for disease classification/characterization and a projection pursuit application from a multi-site fMRI schizophrenia study. Brain imaging and behavior 2, 207–226.

Dietterich, T.G., 2000. Ensemble Methods in Machine Learning. "Springer Berlin Heidelberg".

Dohmatob, E., Eickenberg, M., Thirion, B., Varoquaux, G., 2015. Speeding-up model-selection in GraphNet via early-stopping and univariate feature-screening. IEEE PRNI , 17–20.

Dümbgen, L., Samworth, R.J., Schuhmacher, D., 2013. Stochastic search for semiparametric linear regression models. From Probability to Statistics and Back: High-Dimensional Models and Processes – A Festschrift in Honor of Jon A. Wellner 9, 78–90.

Duncan, K., Pattamadilok, C., Knierim, I., Devlin, J., 2009. Consistency and variability in functional localisers. Neuroimage 46, 1018–1026.

Eickenberg, M., Dohmatob, E., Thirion, B., Varoquaux, G., 2015. Total variation meets sparsity: statistical learning with segmenting penalties, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer Berlin Heidelberg.

Essen, D.V., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T., Bucholz, R., Chang, A., Chen, L., Corbetta, M., Curtiss, S., Penna, S.D., Feinberg, D., Glasser, M., Harel, N., Heath, A., Larson-Prior, L., Marcus, D., Michalareas, G., Moeller, S., Oostenveld, R., Petersen, S., Prior, F., Schlaggar, B., Smith, S., Snyder, A., Xu, J., Yacoub, E., 2012. The human connectome project: A data acquisition perspective. NeuroImage 62, 2222–2231.

Fan, Y., Batmanghelich, N., Clark, C.M., Davatzikos, C., 2008. Spatial patterns of brain atrophy in MCI patients, identified via high-dimensional pattern classification, predict subsequent cognitive decline. NeuroImage 39, 1731–1743.

Gramfort, A., Thirion, B., Varoquaux, G., 2013. Identifying predictive regions from fMRI with TV-L1 prior. PRNI , 17.

Gramfort, A., Varoquaux, G., Thirion, B., 2012. Beyond Brain Reading: Randomized Sparsity and Clustering to Simultaneously Predict and Identify. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 9–16.

Grill-Spector, K., Malach, R., 2004. The human visual cortex. Annu. Rev. Neurosci. 27, 649–677.

Grosenick, L., Klingenberg, B., Katovich, K., Knutson, B., Taylor, J.E., 2013. Interpretable whole-brain prediction analysis with GraphNet. Neuroimage 72, 304–321.

Hastie, T., Tibshirani, R., Eisen, M.B., Alizadeh, A., Levy, R., Staudt, L., Chan, W.C., Botstein, D., Brown, P., 2000. 'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. Genome Biology 1.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA. 2 edition.

Haxby, J.V., Gobbini, M.I., Furey, M.L., Ishai, A., Schouten, J.L., Pietrini, P., 2001. Distributed and overlapping representations of faces and objects in ventral temporal cortex. Science 293, 2425–2430.

Haynes, J.D., Rees, G., 2006. Decoding mental states from brain activity in humans. Nat Rev Neurosci 7, 523–534.

Henson, R., 2006. Forward inference using functional neuroimaging: Dissociations versus associations. Trends in cognitive sciences 10, 64–69.

Henson, R., Shallice, T., Gorno-Tempini, M., Dolan, R., 2002. Face repetition effects in implicit and explicit memory tests as measured by fMRI. Cerebral Cortex 12, 178.

Hoyos-Idrobo, A., Varoquaux, G., Kahn, J., Thirion, B., 2016. Recursive nearest agglomeration (ReNA): fast clustering for approximation of structured signals. arXiv:1609.04608 .

Knops, A., Thirion, B., Hubbard, E.M., Michel, V., Dehaene, S., 2009. Recruitment of an area involved in eye movements during mental arithmetic. Science 324, 1583.

Kuncheva, L.I., Rodríguez, J.J., 2010. Classifier ensembles for fmri data analysis: an experiment. Magnetic resonance imaging 28, 583.

Kuncheva, L.I., Rodríguez, J.J., Plumpton, C.O., Linden, D.E., Johnston, S.J., 2010a. Random subspace ensembles for fmri classification. IEEE transactions on medical imaging 29, 531–542.

Kuncheva, L.I., Rodriguez, J.J., Plumpton, C.O., Linden, D.E.J., Johnston, S.J., 2010b. Random subspace ensembles for fMRI classification. IEEE Transactions on Medical Imaging 29, 531–542.

Leung, G., Barron, A.R., 2006. Information theory and mixing least-squares regressions. IEEE Transactions on information theory 52.

Marcus, D.S., Wang, T.H., Parker, J., et al., 2007. Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. J Cogn Neurosci 19, 1498.

Meinshausen, N., Bühlmann, P., 2010. Stability selection. J Roy Stat Soc B 72, 417–473.

Michel, V., Gramfort, A., Varoquaux, G., Eger, E., Thirion, B., 2011. Total variation regularization for fMRI-based prediction of behavior. IEEE transactions on medical imaging 30, 1328–1340.

Mohr, H., Wolfensteller, U., Frimmel, S., Ruge, H., 2015. Sparse regularization techniques provide novel insights into outcome integration processes. Neuroimage 104.

Moran, J.M., Jolly, E., Mitchell, J.P., 2012. Social-cognitive deficits in normal aging. J. Neurosci 32, 5553.

Mourão-Miranda, J., Bokde, A.L., Born, C., Hampel, H., Stetter, M., 2005. Classifying brain states and determining the discriminating activation patterns: Support vector machine on functional MRI data. NeuroImage 28, 980–995.

Nemirovski, A., 2000. Topics in non-parametric statistics. Lectures on Probability Theory and Statistics: Ecole d'Ete de Probabilites de Saint-Flour XXVIII-1998 28, 85.

Norman, K.A., Polyn, S.M., Detre, G.J., Haxby, J.V., 2006. Beyond mind-reading: multi-voxel pattern analysis of fMRI data. Trends in Cognitive Sciences 10.

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

Pereira, F., Mitchell, T., Botvinick, M., 2009. Machine learning classifiers and fMRI: A tutorial overview. NeuroImage 45, S199–S209.

Poldrack, R.A., 2011. Inferring mental states from neuroimaging data: from reverse inference to large-scale decoding. Neuron 72, 692–697.

Poldrack, R.A., Barch, D.M., Mitchell, J.P., Wager, T.D., Wagner, A.D., Devlin, J.T., Cumba, C., Koyejo, O., Milham, M.P., 2013. Toward open sharing of task-based fMRI data: the OpenfMRI project. Frontiers in Neuroinformatics 7.

Poldrack, R.A., Gorgolewski, K.J., 2015. OpenfMRI: Open sharing of task fMRI data. NeuroImage .

Praestgaard, J., Wellner, J.A., 1993. Exchangeably weighted bootstraps of the general empirical process. The Annals of Probability 21, 2053–2086.

Rondina, J.M., Hahn, T., de Oliveira, L., Marquand, A., Dresler, T., Leitner, T., Fallgatter, A.J., Shawe-Taylor, J., Mourão-Miranda, J.J., 2014. SCoRS-A method based on stability for feature selection and mapping in neuroimaging. IEEE Transactions on Medical Imaging 33, 85–98.

Schwartz, Y., Thirion, B., Varoquaux, G., 2013. Mapping cognitive ontologies to and from the brain, in: Advances in neural information processing systems.

Shah, R.D., Samworth, R.J., 2013. Variable selection with error control: another look at stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75, 55–80.

Shalev-Shwartz, S., Shamir, O., Srebro, N., Sridharan, K., 2010. Learnability, stability and uniform convergence. Journal of Machine Learning Research 11, 2635–2670.

Tibshirani, R., 1994. Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 58.

Varoquaux, G., Gramfort, A., Thirion, B., 2012. Small-sample brain mapping: sparse recovery on spatially correlated designs with randomization and clustering, in: International conference on machine learning, p. 1375.

Varoquaux, G., Raamana, P.R., Engemann, D.A., Hoyos-Idrobo, A., Schwartz, Y., Thirion, B., 2017. Assessing and tuning brain decoders: cross-validation, caveats, and guidelines. Neuroimage 145, Part B, 166–179.

Varoquaux, G., Thirion, B., 2014. How machine learning is shaping cognitive neuroimaging. GigaScience 3.

Wager, T., Davidson, M., Hughes, B., Lindquist, M., Ochsner, K.,

2008. Neural mechanisms of emotion regulation: evidence for two independent prefrontal-subcortical pathways. Neuron 59, 1037–1050.

Wang, Y., Zheng, J., Zhang, S., Duan, X., Chen, H., 2015. Randomized structural sparsity via constrained block subsampling for improved sensitivity of discriminative voxel identification. Neuroimage 117, 170–183.

Yu, B., 2013. Stability. Bernoulli 19, 1484–1500.

Zhou, Z.H., 2012. Ensemble Methods: Foundations and Algorithms. Chapman & Hall/CRC. 1st edition.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67, 301–320.

Appendix A. Bagging reduces the variance of the base model

A learning problem can be cast as identifying the best hypothesis in a space of hypotheses. Statistical issues arise when the sample size is small compared to the size of the hypothesis space. In this setting, there are several hypothesis that give the same prediction accuracy on training data. One can reduce the risk of choosing a "wrong" estimator by averaging the output of several of these estimators (Dietterich, 2000).

Bagging in regression (Breiman, 1996): We present Breiman's proof of bagging regressors to demonstrate its benefits. Let D be a training set which contains a sample of independent (x, y) drawn from the distribution P. Define $h(x, D)$ to be prediction function based on the sample D. Let $h_{\text{aggr}}(x) = \mathbb{E}_D[h(x, D)]$ be an aggregate of prediction functions.

Take x to be a fixed predictor and y an output value. Then

$$\begin{aligned}\mathbb{E}_D[(y - h(x,D))^2] &\geq \mathbb{E}_D[y - h(x,D)]^2 \quad \text{(by Jensen's inequality)}\\ &= y^2 - 2y\mathbb{E}_D[h(x,D)] + \mathbb{E}_D[h(x,D)]^2\\ &= (y - h_{\text{aggr}}(x))^2.\end{aligned}$$

Hence, the averaged predictor has lower mean-squared error than the base predictor. This improvement depends on how unequal $\mathbb{E}_D[h(x,D)^2] \geq \mathbb{E}_D[h(x,D)]^2$ are. The more the $h(x, D)$ vary with respect to each other, the more improvement the aggregation may produce.

To understand the effect of averaging, let us assume the extreme case where the models created by sampling are i.i.d. Let the aggregation be the mean of the predicted values $h_{\text{aggr}}(x) = \frac{1}{b}\sum_{i=1}^{b} h_i(x, D)$, where $h_i(x, D)$ denotes the prediction function based on the sample D. The predictions are i.i.d., and the variance of each of them is defined by $\mathbb{E}_D[(y - h_i(x,D))^2] = \sigma^2$. Hence, the variance of the aggregated estimator is:

$$\begin{aligned}\mathbb{E}_D\big[(y - h_{\text{aggr}}(x))^2\big] &= \frac{1}{b^2}\mathbb{E}_D\left[(\sum_{i=1}^{b} y - h_i(x,D))^2\right]\\ &= \frac{1}{b^2}\sum_{i=1}^{b}\mathbb{E}_D\big[(y - h_i(x,D))^2\big] \text{(by independence)}\\ &= \frac{1}{b}\sigma^2.\end{aligned}$$

We can see that averaging decreases the error as $\sqrt{b}$. Note that the i.i.d. case studied here is the most favorable case.

Appendix B. Experiments on Simulated data

Appendix B.1. Dataset simulation

We use the same approach presented in Michel et al. (2011) to simulate data satisfying the Eq. 1. The matrix $\mathbf{X}$ consists of $n = 100$ images, and $p = 1728$ voxels (size $12 \times 12 \times 12$). Each image contains a set of five square Regions of Interest (ROIs) (size $2 \times 2 \times 2$), and each of the four ROIs has a fixed weight in $\{-0.6, 0.5, -0.6, 0.5, 0.5\}$. Let us denote S the support of the ROIs (i.e. the 40 resulting voxels of interest), and $\mathbf{w}_{i,j,k}$ denotes the weights of the (i, j, k) voxel. The resulting images are smoothed with a Gaussian kernel with a standard deviation of 2 voxels, to mimic the correlation structure observed in real fMRI data. To simulate the spatial variability between images (inter-subject variability, movement artifacts in intra-subject variability), we define a new support of the ROIs, $\hat{S}$ such as, for each image l-th, 50% (randomly chosen) of the weights $\mathbf{w}$ are set to zero. Thus, we have $\hat{S} \subset S$. We simulate the target $\mathbf{y}$ for the l-th image as:

$$\mathbf{y}_l = \sum_{(i,j,k)\in\hat{S}} \mathbf{w}_{i,j,k}\mathbf{X}_{i,j,k,l} + \boldsymbol{\epsilon}_l, \tag{B.1}$$

with the signal in the (i, j, k) voxel of the l-th image simulated as:

$$\mathbf{X}_{i,j,k,l} \sim N(0, 1). \tag{B.2}$$

$\boldsymbol{\epsilon} \sim N(0, \gamma)$, We choose γ in order to have a SNR of 5 dB. Finally, we apply a sign function to Eq. B.1 to obtain a binary targets.

Appendix C. Benchmarking results without smoothing

Results without spatial smoothing of the voxels are given in Fig. C.2. The results obtained with and without spatial smoothing of the voxels are consistent. In within subject settings, structured sparse methods display good prediction accuracy and stability of weight maps. However, the computation time is slow. FReM of SVM-ℓ_1 yields to a better performance, increasing prediction accuracy and weight map stability, while reducing the computation time.

In between subject settings, FReM over performs other methods. In particular, FReM of SVM-ℓ_2 with clustering.

Table C.2 shows the comparison between each decoder and the mean performance across models. These results confirm the above observations.

Appendix D. FReM improves brain regions delineation

Fig. D.3 displays the difference between the outlines obtained with FReM and its base model. The difference between the outlined brain maps is characterized by small clusters. As expected, this effect is higher for SVM-ℓ_1. FReM of SVM-ℓ_1 with clustering leads to greater brain activation areas that are not found by the base estimator. On the other hand, FReM of SVM-ℓ_2 with clustering displays a marginal difference with its non-clustered counterpart.

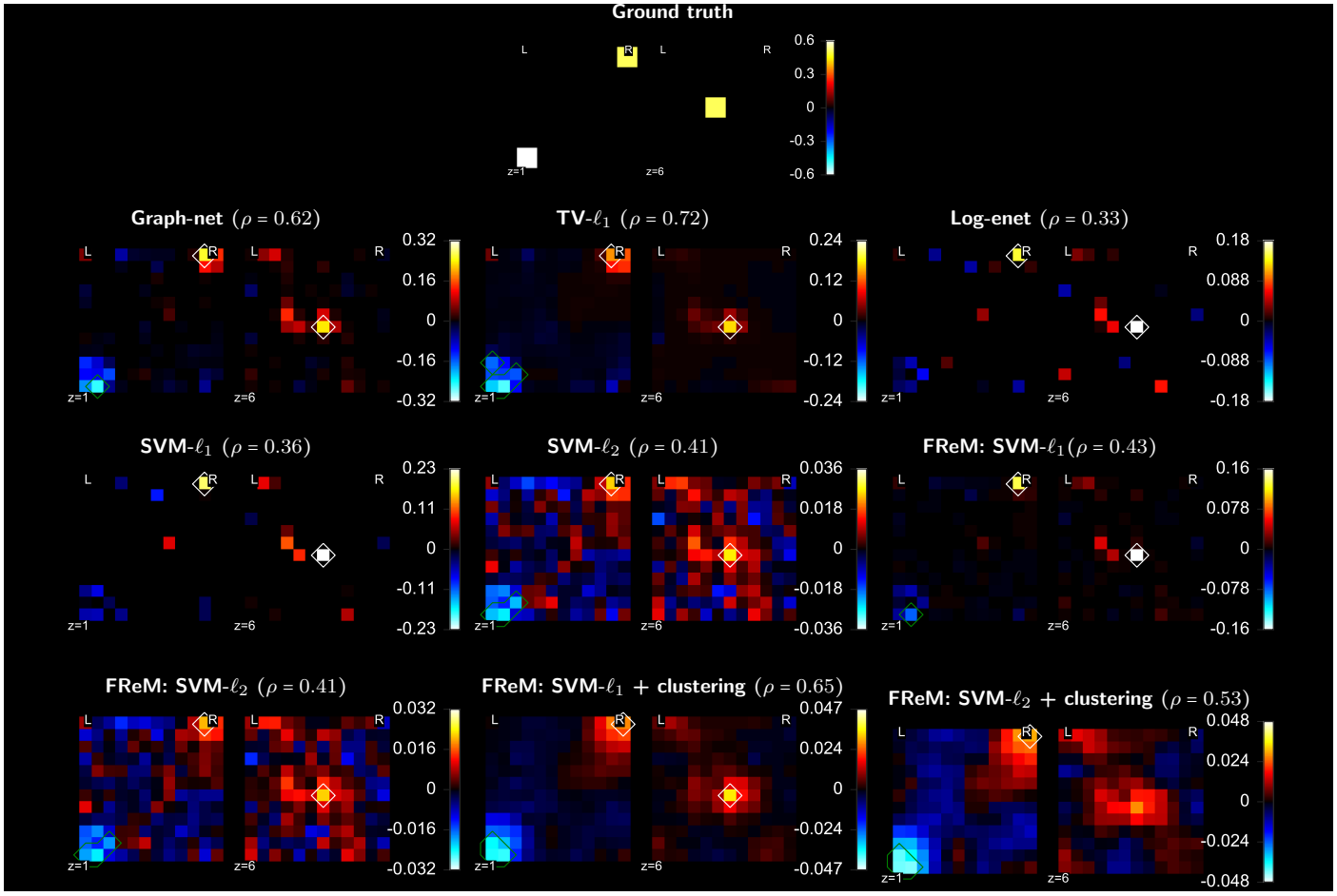

Figure B.1: **Weight maps of the decoder:** Weight maps for different decoders on simulated data. The values in parenthesis correspond to the Pearson correlation to ground truth. Note that the maps are unthresholded.

a) Within-subject discrimination

	Classifier	prediction score	weight stability	computation time
	Graph-net	2e-6 >	8.1e-8 >	**7.8e-14 <**
	TV-ℓ_1	1	5.8e-2 >	**7.8e-14 <**
	Log-enet	1	**7.8e-14 >**	**7.8e-14 >**
	SVM-ℓ_2	4.5e-10 >	**1.8e-12 >**	**7.8e-14 >**
	SVM-ℓ_1	1.7e-2 >	**8.4e-14 >**	**7.8e-14 >**
FReM	SVM-ℓ_2	**1.8e-11 >**	1	1
FReM	SVM-ℓ_1	**Reference**		
FReM	SVM-ℓ_2 + clustering	4.6e-6 >	9.5e-3 <	1
FReM	SVM-ℓ_1 + clustering	1	0.2 <	0.5 >

b) Across-subjects discrimination

	Classifier	prediction score	weight stability	computation time
	Graph-net	**5.4e-17 >**	**3.4e-22 >**	9.9e-4 >
	TV-ℓ_1	**7.9e-19 >**	**4.7e-17 >**	**6.4e-32 <**
	Log-enet	1.3e-8 >	**1.2e-28 >**	**1.2e-28 >**
	SVM-ℓ_2	3.6e-4 >	**2.8e-14 >**	**6.3e-32 >**
	SVM-ℓ_1	3.5e-4 >	**1.2e-28 >**	**1.2e-28 >**
FReM	SVM-ℓ_2	1	**1.8e-13 <**	**1.4e-22 >**
FReM	SVM-ℓ_1	1	**6.3e-32 >**	**6.3e-32 >**
FReM	SVM-ℓ_2 + clustering	**Reference**		
FReM	SVM-ℓ_1 + clustering	1	**3.6e-28 >**	0.2 >

Table C.2: **Comparison of performance:** Each decoder is compared with a reference. The values correspond to Bonferroni-corrected p-values obtained by paired Wilcoxon rank test. The direction in the parenthesis denotes the sign of the mean difference, and bold text denotes a significant results ($p < 10^{-10}$).

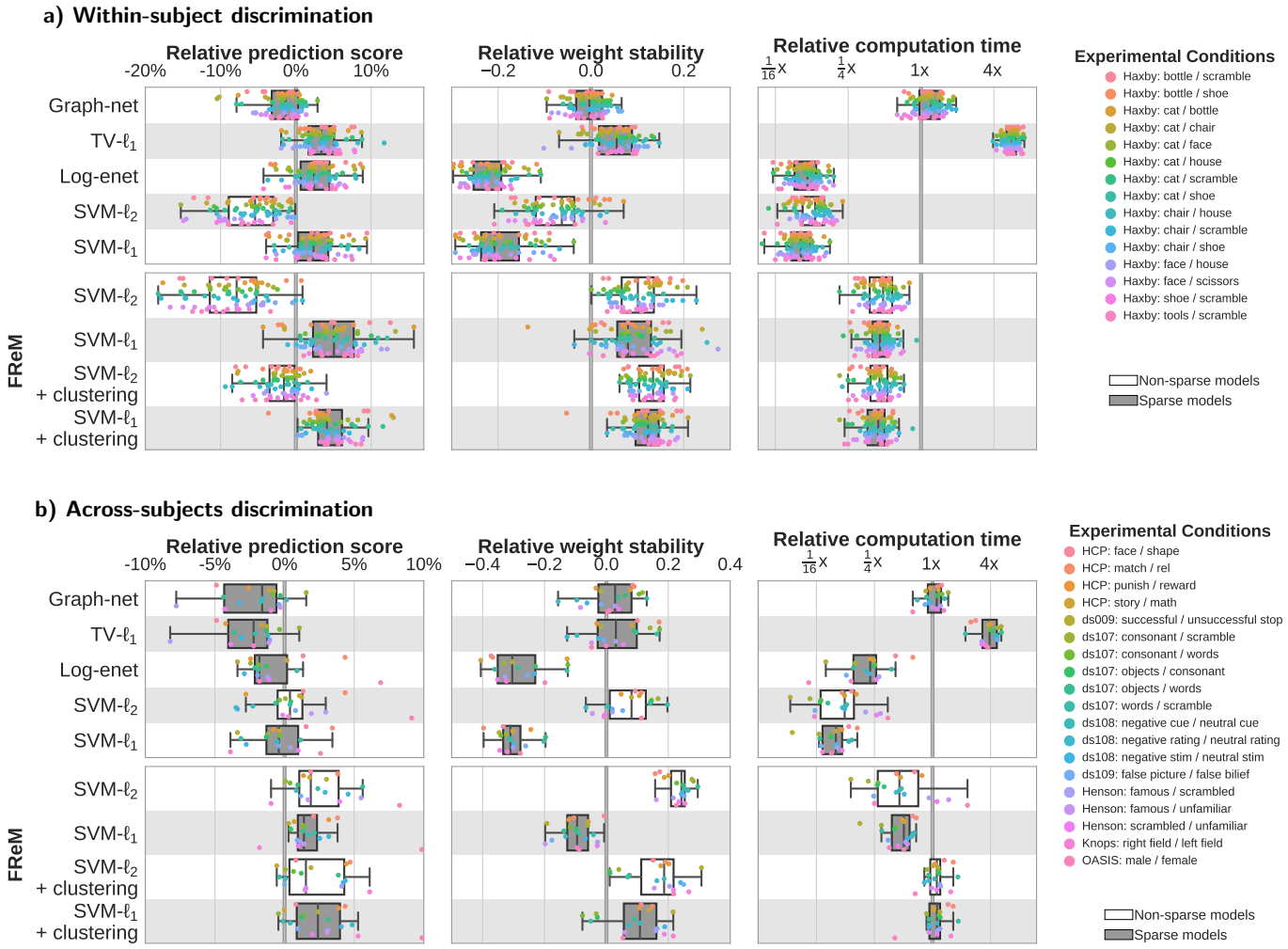

Figure C.2: **Relative performance:** Relative prediction accuracy, weight stability and computation time for different classification tasks. Values are displayed relative to the mean over all the classifiers.

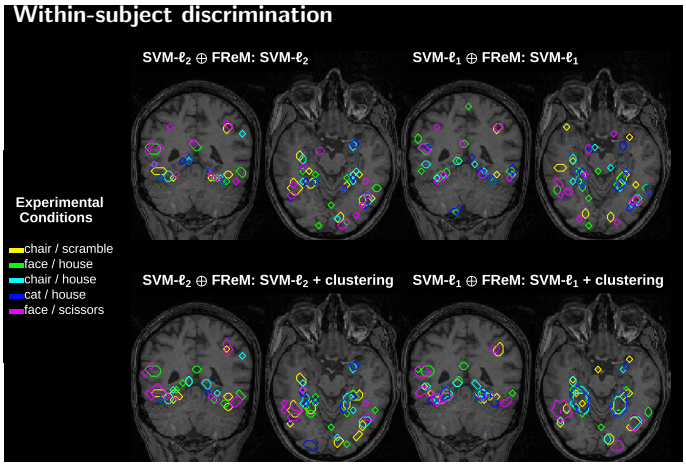

Figure D.3: **Difference on delineated brain regions:** Difference on weight maps of various decoders.