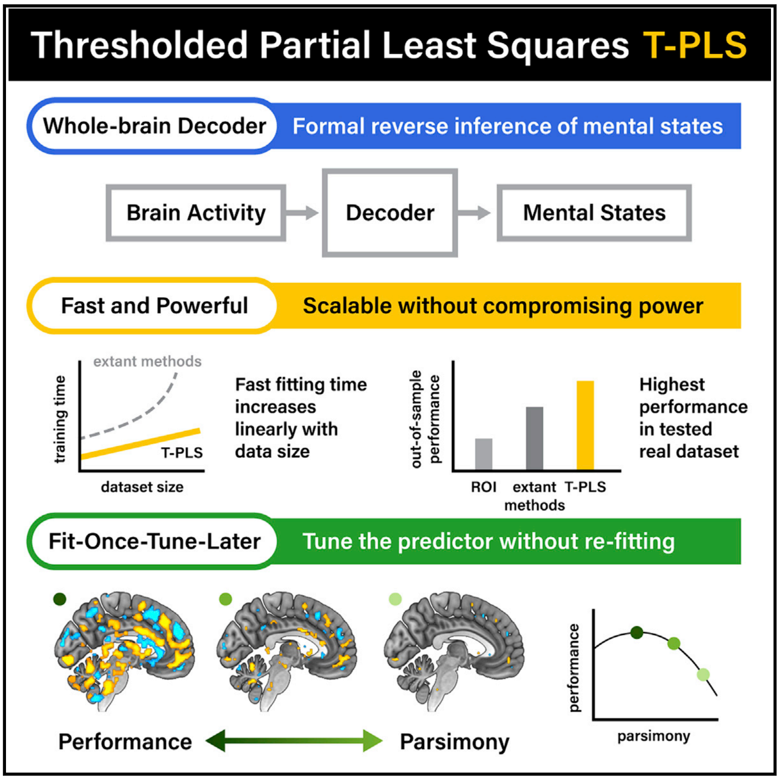
Cell Reports
Methods

# Fast construction of interpretable whole-brain decoders

## Graphical abstract


Thresholded Partial Least Squares T-PLS
Whole-brain Decoder
Formal reverse inference of mental states
Brain Activity
Decoder
Mental States
Fast and Powerful
Scalable without compromising power
extant methods
training time
T-PLS
dataset size
Fast fitting time increases linearly with data size
out-of-sample performance
ROI
extant
T-PLS
methods
Highest performance in tested real dataset
Fit-Once-Tune-Later
Tune the predictor without re-fitting
Performance
Parsimony
performance
parsimony

## Highlights

- T-PLS offers interpretable whole-brain multivariate decoders with minimal computation
- "Fit once, tune later" model tuning via cross-validation has nearly zero computation
- In real data, T-PLS shows highest predictive performance and fitting speed
- Users can decide post hoc the balance between predictive power and model parsimony

## Authors

Sangil Lee, Eric T. Bradlow,
Joseph W. Kable

## Correspondence

sangillee3rd@gmail.com

## In brief

Lee et al. propose a thresholded partial least squares (T-PLS) algorithm for building interpretable whole-brain decoders using high-dimensional fMRI data. T-PLS achieves fast, scalable computation times while offering variable selection via cross-validation to achieve interpretability. In real data, T-PLS shows higher predictive performance than extant methods.

Lee et al., 2022, Cell Reports Methods *2*, 100227
June 20, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.crmeth.2022.100227

CellPress

Cell Reports Methods

CellPress
OPEN ACCESS

Article

# Fast construction of interpretable whole-brain decoders

Sangil Lee,[1,2,3,4,*] Eric T. Bradlow,[2] and Joseph W. Kable[1,2]
[1]Department of Psychology, School of Arts and Sciences, University of Pennsylvania, Philadelphia, PA 19104, USA
[2]Marketing Department, Wharton School, University of Pennsylvania, PA 19104, USA
[3]Social Science Matrix, University of California, Berkeley, Berkeley, CA 94720, USA
[4]Lead contact
*Correspondence: sangillee3rd@gmail.com

https://doi.org/10.1016/j.crmeth.2022.100227

**MOTIVATION** Creating whole-brain predictors using functional MRI data can be challenging, especially in large datasets due to the computational burden of large number of features, large numbers of observations, and cross-validation. Our method exploits the analytical properties of the partial least squares algorithm to significantly reduce model fitting time as well as provide cross-validation-based tuning with nearly zero computational overhead.

## SUMMARY

Researchers often seek to decode mental states from brain activity measured with functional MRI. Rigorous decoding requires the use of formal neural prediction models, which are likely to be the most accurate if they use the whole brain. However, the computational burden and lack of interpretability of off-the-shelf statistical methods can make whole-brain decoding challenging. Here, we propose a method to build whole-brain neural decoders that are both interpretable and computationally efficient. We extend the partial least squares algorithm to build a regularized model with variable selection that offers a unique “fit once, tune later” approach: users need to fit the model only once and can choose the best tuning parameters post hoc. We show in real data that our method scales well with increasing data size and yields interpretable predictors. The algorithm is publicly available in multiple languages in the hope that interpretable whole-brain predictors can be implemented more widely in neuroimaging research.

## INTRODUCTION

Predicting mental states from brain activity can be immensely useful, especially for researchers who study cognitive or internal processes that are hard to measure via overt behavior. Such brain decoding is widely used in different areas of neuroscience from basic animal research to applied human research. For example, in basic research, predictive models from hippocampal place cell activity have shown that rats navigating a maze pre-play the maze ahead before deciding which way to turn (Johnson and Redish, 2007), and brain-wide electrical activity in mice can reveal their affective responses to stress (Hultman et al., 2018). In applied research, researchers have built brain-computer interfaces (BCIs) that read real-time neural activity to control robotic arms (Hochberg et al., 2012) or to type computer text (Willett et al., 2021) for patients who are paralyzed.

Brain decoding of mental states from human neuroimaging, however, has a somewhat fraught history. In early neuroimaging studies, researchers would often draw conclusions about mental states from brain images without actually building a formal predictive model. Typically, researchers would observe significant activity in a particular brain region and, given the previous (assumed or partially empirically understood) association of that region with a mental construct, conclude that their experimental paradigm involves said mental construct. This practice of informal reverse inference has been strenuously criticized, as observing brain activity in a single region usually provides very weak evidence regarding the engagement of specific mental processes (Poldrack, 2006). Instead, drawing rigorous conclusions about mental states from brain activity requires formal reverse inference, or brain decoding, as applied in other areas of neuroscience—training and testing a formal statistical (and typically multivariate) model that predicts mental states from brain activity measurements (Kohoutová et al., 2020; Poldrack, 2011, Poldrack et al., 2020).

Such statistical techniques have been applied widely in human neuroimaging to address questions about regional coding, by building multivariate predictors from activity in *a priori* regions of interest (ROI; Cox and Savoy, 2003) or a handful of local voxels at a time, as in searchlight multivoxel pattern analysis (MVPA)

(Etzel et al., 2013; Kriegeskorte et al., 2006). However, if the goal is formal reverse inference about mental states, using signals from the entire brain to build a decoder should improve specificity and sensitivity. While any single brain region can be active for multiple reasons, it is much less likely for two distinct cognitive processes to exhibit the exact same pattern of brain activity across the entire brain. Also, whole-brain predictors can provide higher decoding power than regional ones given that they have access to more potential (uncorrelated) signals. Accordingly, researchers have started to build whole-brain predictors for formal reverse inference, constructing successful whole-brain predictors of pain (Wager et al., 2013), valuation (Smith et al., 2014), negative affect when viewing pictures (Chang et al., 2015), and distinct emotional states (Kassam et al., 2013; Kragel and LaBar, 2014). There have also been models based on functional connectivity, albeit usually based on resting-state functional magnetic resonance imaging (fMRI) (Kucyi et al., 2021; Miranda-Dominguez et al., 2018; Rosenberg et al., 2016; Yamashita et al., 2018).

Unfortunately, there are two challenges in building whole-brain decoders with off-the-shelf statistical methods. A first difficulty is interpretability: A good whole-brain predictor should distinguish brain regions that are predictive from those that are not. Several methods incorporate no variable selection, such that the entire brain has non-zero coefficients. These include ridge regression (Grosenick et al., 2013), support vector machines (Whitehead and Armony, 2019), partial least squares (PLS) (McIntosh et al., 1996), and PCR-LASSO (principal component regression-least absolute shrinkage and selection operator), which uses principal-component analysis (PCA) for data reduction and then LASSO regression to select the most useful components (Wager et al., 2013). These approaches typically use the entire brain's coefficients for prediction, but later threshold the coefficients, at an arbitrary level or by bootstrap, to improve interpretability in inference (McIntosh et al., 1996; Wager et al., 2013). Alternatively, some methods do provide variable selection, but less helpfully: LASSO selects the most useful variables (voxels) for prediction, but does so without regard for spatial contiguity (Grosenick et al., 2013), resulting in predictive maps with scattered "sparkles of coefficients" across the brain rather than any interpretable clusters or regions. One method that does yield interpretably clustered coefficients is GraphNet, which combines the elastic net penalty with spatial contiguity information (Grosenick et al., 2013).

However, there is also a second difficulty, which is computational efficiency. This difficulty is particularly acute in regard to scaling for use in larger datasets, the collaborative collection of which is an increasing focus of human neuroimaging research (Allen et al., 2014; Bjork et al., 2017; Satterthwaite et al., 2014; Van Essen et al., 2012). Since neuroimaging data provide a substantial number of predictors (often >50,000), purely likelihood-based approaches often face the problem of computing gradients for a large number of variables. Adding to the computational burden, modern models often need to be fitted hundreds of times to find the best tuning parameters (i.e., hyperparameters). Parallelization, which commonly is a solution to such computational challenges, is also difficult for this class of predictive models. Mass-univariate approaches can be easily parallelized at the run level or subject level by carrying over the uncertainty of the run level estimates to higher-level analyses so that the final estimate at the group level accounts for the uncertainty of each run and subject (e.g., the approach taken in FSL by using Markov chain Monte Carlo sampling). Predictive models, however, typically do not use generalized linear model (GLM) approaches but instead opt for machine learning methods, penalized regressions, or data-reduction techniques that require all of the data to obtain the final predictor, hence limiting the possibility of parallelization.

Given these constraints, previous whole-brain predictors used as inputs either down-sampled images (i.e., coarser) with fewer voxels (e.g., Grosenick et al., 2013; Wager et al., 2013) or refined contrast maps that reduce the number of observations at the expense of foregoing trial-level predictions (Chang et al., 2015; Poldrack et al., 2009). As an alternative to purely likelihood-based methods, data-reduction approaches such as PCA can help by reducing the number of variables, thereby reducing the model fitting time. However, in larger datasets, PCA itself can become a bottleneck for computation time and memory usage, as it requires computation of the variance-covariance matrix of predictors. Also, PCA components are ordered in terms of variance explained in X, which is not necessarily relevant in predicting Y (Lever et al., 2017); hence, the most useful PCA component for prediction may be the 100th, 1,000th, or 10,000th one. These computational costs prevent the widespread use of whole-brain prediction methods in neuroimaging, especially for those without access to high-performance computing clusters.

Here, we propose our method, thresholded partial least squares (T-PLS, pronounced "tea, please"), which provides interpretable whole-brain predictors that are computationally efficient enough to run on personal laptops for most datasets. T-PLS extends PLS by providing an additional tuning parameter that selects the original variables, based on their importance, by cross-validation. More specifically, regular PLS uses cross-validation to decide on the optimal number of PLS components, while T-PLS uses cross-validation to decide on both the optimal number of PLS components and the number of original predictor variables to include in the model. This improves the interpretability of the final predictive model as well as its predictive performance. This additional variable selection step also has near-zero computational overhead, as it exploits the analytical properties of a modified PLS algorithm to offer a unique "fit once, tune later" approach in which the user fits the model only once and then evaluates the best tuning parameter as many times as needed without re-fitting the model. This is in stark contrast to most, if not all, modern methods that require re-fitting the model for every tuning parameter. Furthermore, it allows researchers to explore the variable importance ranking to make the trade-off decision between parsimony and predictive power. Here, we describe the algorithm and showcase its performance against other methods in a large neuroimaging dataset. In addition, we provide the T-PLS package online for MATLAB at github (https://github.com/sangillee/TPLSm), for R at CRAN (https://CRAN.R-project.org/package=TPLSr), and for Python at github (https://github.com/sangillee/TPLSp) as (we hope) this becomes a practical tool for others.

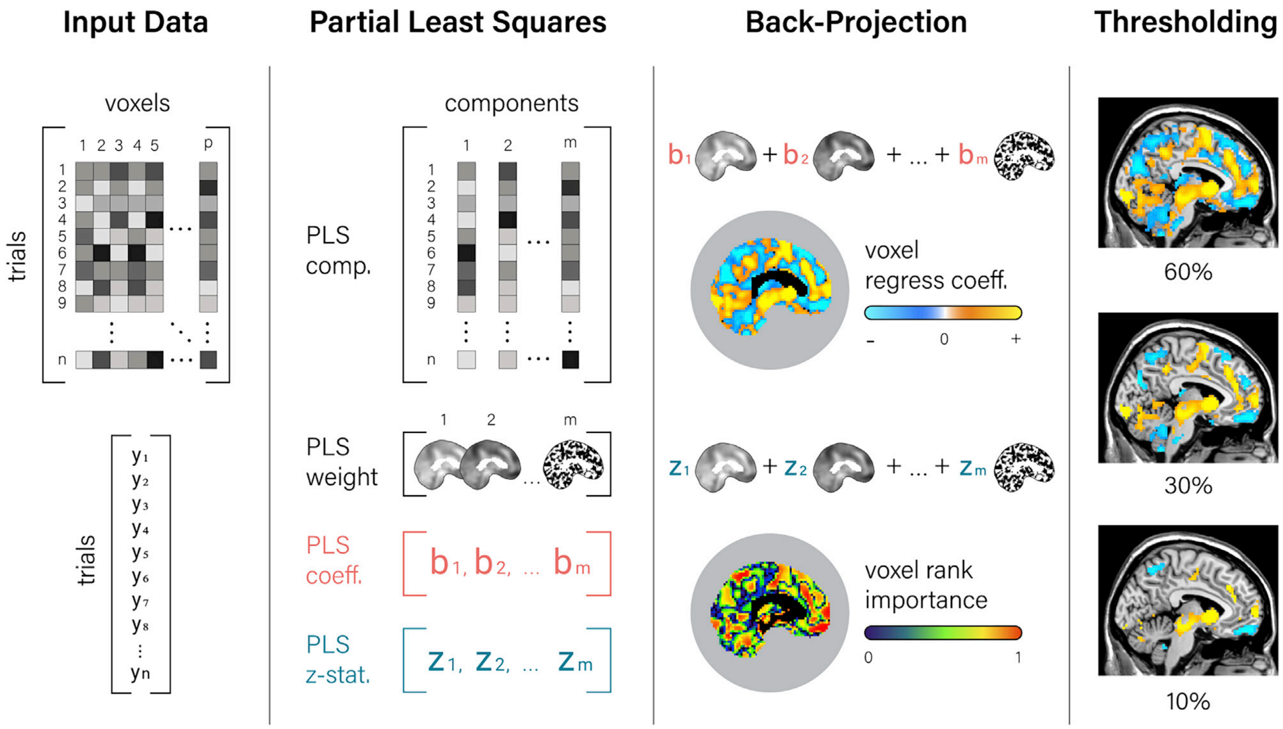


**Figure 1. Schematic of T-PLS fitting procedures**

T-PLS model fitting first requires extracting the partial least squares (PLS) components from the predictor matrix and obtaining the back projection maps of the components (PLS weight), their regression coefficients, and their z statistic. Next, the regression coefficients and the z statistics are back-projected into the voxel space using the weight maps, thus yielding a whole-brain coefficient map and a whole-brain voxel importance map, which is then ranked in absolute size between 0 and 1. Finally, the coefficient map is thresholded based on the voxel importance map to select voxels that are the most important.

## RESULTS

### Method overview

T-PLS, like many modern regression methods, requires two steps when training a model—fitting and parameter tuning. In the fitting step, the goal is to calculate the coefficient and the variable importance of each original voxel (Figure 1). In detail, the fitting step first extracts the PLS components that maximally explain the covariance between X (e.g., brain voxels) and Y (e.g., behavioral or cognitive state). The regression coefficients of these components are automatically calculated in the component extraction process. Then, the coefficients and the z statistics of the components are back-projected into the original variable space to calculate a coefficient and a variable importance measure for each voxel.

In the tuning step, two parameters are chosen—the number of PLS components used and the variable importance threshold that controls the number of voxels retained (Figure 2). For example, a T-PLS model that uses the first 21 PLS components and retains 50% of the original voxels may provide the highest out-of-sample cross-validation performance. Intuitively, the number of PLS components controls the degree of regularization, while the voxel threshold controls the level of parsimony. This is different from likelihood-based approaches such as LASSO, ridge, or elastic-net, in which the degree of regularization also controls the level of parsimony.

The key computational benefit of T-PLS comes from the fit once, tune later feature. The user can choose among an infinite number of tuning parameter combinations without having to re-fit the model, as all of the information required is already calculated in a one-time fitting. This is because once a T-PLS model with *m* components has been fit, all of the models with fewer components are also available (i.e., a 1-component, 2-component, …, *m*-component model). Since PLS components are orthogonal to one another, their regression coefficients do not change based on the number of components kept, thus allowing the user to choose the necessary components without re-fitting. While PCA components also have this feature, when Y is a vector, PLS computation only requires vector multiplications, which are fast and memory efficient, while PCA requires singular value decomposition of matrices, which has computation time that grows quadratically with data (PLS also uses singular value decompositions when Y is a matrix).

Using simulated data and a real neuroimaging dataset (involving an economic decision-making task), we compared the interpretability and computational efficiencies of ordinary least squares (OLS), PLS, LASSO, PCR-LASSO, and T-PLS. We focused on comparing T-PLS to PLS, LASSO, and PCR-LASSO because these methods have been used previously to build whole-brain predictors and provide interpretable linear models. We did not include a comparison with non-linear models, which have been used previously in partial-brain

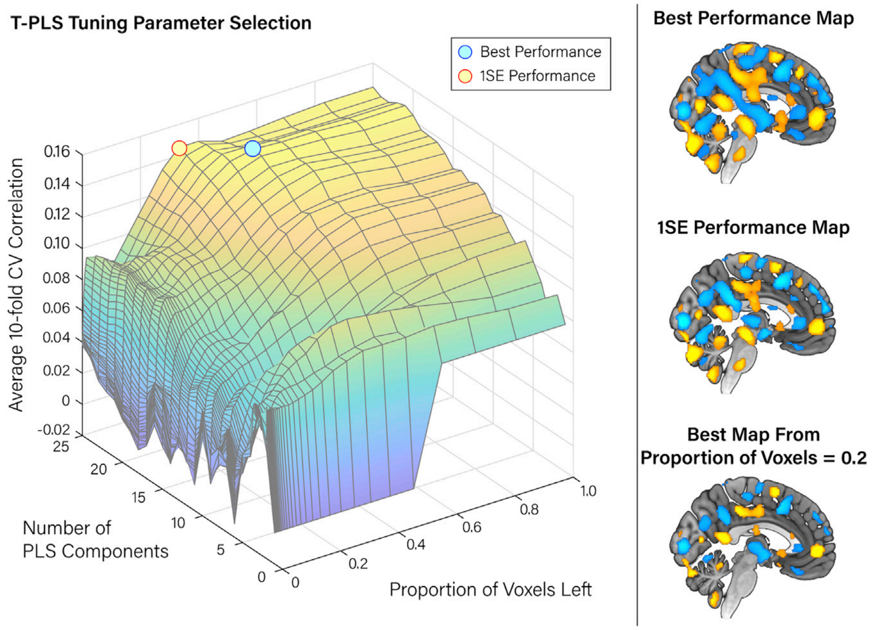


**Figure 2. Example T-PLS model tuning**

Left panel shows an example cross-validation performance surface as a function of the 2 tuning parameters of T-PLS—number of PLS components (1–25) and proportion of voxels left (0–1). The highest CV performance point is marked with a blue dot, with the corresponding whole-brain predictor shown on the right top panel. In addition, the model with the fewest voxels within 1 standard error of the performance of the best model is indicated with a yellow dot, with the corresponding map shown on the right center panel. The right bottom panel illustrates how the number of remaining voxels with coefficients reduce as the proportion of voxels left are reduced. The positive coefficients are marked with warm colors, the negative with cold colors, and the units are arbitrary because fMRI signals are in arbitrary units.

analysis (especially the visual cortex; e.g., Kay et al., 2008), because at the scale of the whole brain, non-linear models are harder to interpret and do not seem to predict better than linear models (Schulz et al., 2020; Thomas et al., 2020).

## Simulated data comparisons

We used simulated data to illustrate the interpretability of predictors from different methods. We simulated a 17-by-17 voxel grid in which only the center 5-by-5 voxels had signal that could be used for predicting a mental state (Y). The center 5-by-5 voxels were correlated with Y at r = 0.1 (a modest amount), while other voxels were uncorrelated with Y (Figure 3A). We smoothed this simulated map to account for the natural smoothness of brain images (Figure 3B). Then, we used different models (OLS, PLS, LASSO, PCR-LASSO, and T-PLS) to predict the mental state (Y) from the 17-by-17 voxel grid. The PLS map shows accurate capturing of the smoothed simulated signal, but does not offer variable selection, thus resulting in the entire map having non-zero coefficients (Figure 3C). T-PLS yields a predictor that most closely resembles the ground-truth signal, detecting the positively predictive 5-by-5 signal grid in the center and eliminating all of the other voxels (Figure 3D). These clustered coefficients distinguish the regions that are predictive from those that are not. The OLS predictor highlights the canonical problem with fMRI images—multicollinearity. The OLS predictor resulted in voxel coefficients that were tessellating in alternating signs, which makes it hard to determine the actual voxel-level patterns or even whether the signal is positively or negatively predictive (Figure 3E). PCR-LASSO uses PCA data reduction to create locally smooth predictors that reflect the smoothness of fMRI images. However, there is no built-in voxel selection, making it more difficult to distinguish regions that are predictive from those that are not, and the predictor contains PCA-based artifacts (Figure 3F; negative coefficients on the edges of the image). Finally, LASSO deals with multicollinearity by selecting only those variables that are the most useful in prediction and removing the rest. The resulting predictor, therefore, is very sparse, making it difficult to identify the regional pattern (Figure 3G).

## Real data comparisons

To compare these methods in real data, we used a large neuroimaging dataset from Kable et al. (2017) that involved two economic decision-making tasks—intertemporal choice and risky choice. In intertemporal choice, participants made choices between a smaller immediate monetary amount of $20 and a larger but delayed monetary amount (e.g., $40 in 30 days). In risky choice, subjects made choices between a smaller certain monetary amount of $20 and a larger but probabilistic monetary amount (e.g., $40 with 60% probability of winning). We combined data across the two tasks to create a whole-brain predictor of value-based choice.

### *Computation time*

We compared the model training times of LASSO, PCR-LASSO, PLS, and T-PLS across varying dataset sizes. OLS was not included as it cannot fit models with more predictors than observations. T-PLS showed exceptionally fast model-fitting time that scaled very easily to large datasets (Figure 4A). In the largest training dataset size of 512 people (256 sessions of ITC and 256 sessions of risky choice), T-PLS took 2 h 10 min on average to finish 10-fold cross-validation training. PCR-LASSO was 28 times slower than T-PLS, taking 2.3 days. LASSO was already taking close to 2 weeks for 256 participants and was too expensive to compute for larger dataset sizes. PLS, at least in its default off-the-shelf implementation, was approximately twice as slow as T-PLS. The difference in computation time between PLS and T-PLS is likely due to the singular value decomposition step of the default off-the-shelf PLS algorithm, which is unnecessary in this case, where the predicted variable Y is a vector and therefore omitted (among other steps) in T-PLS to optimize computational efficiency.

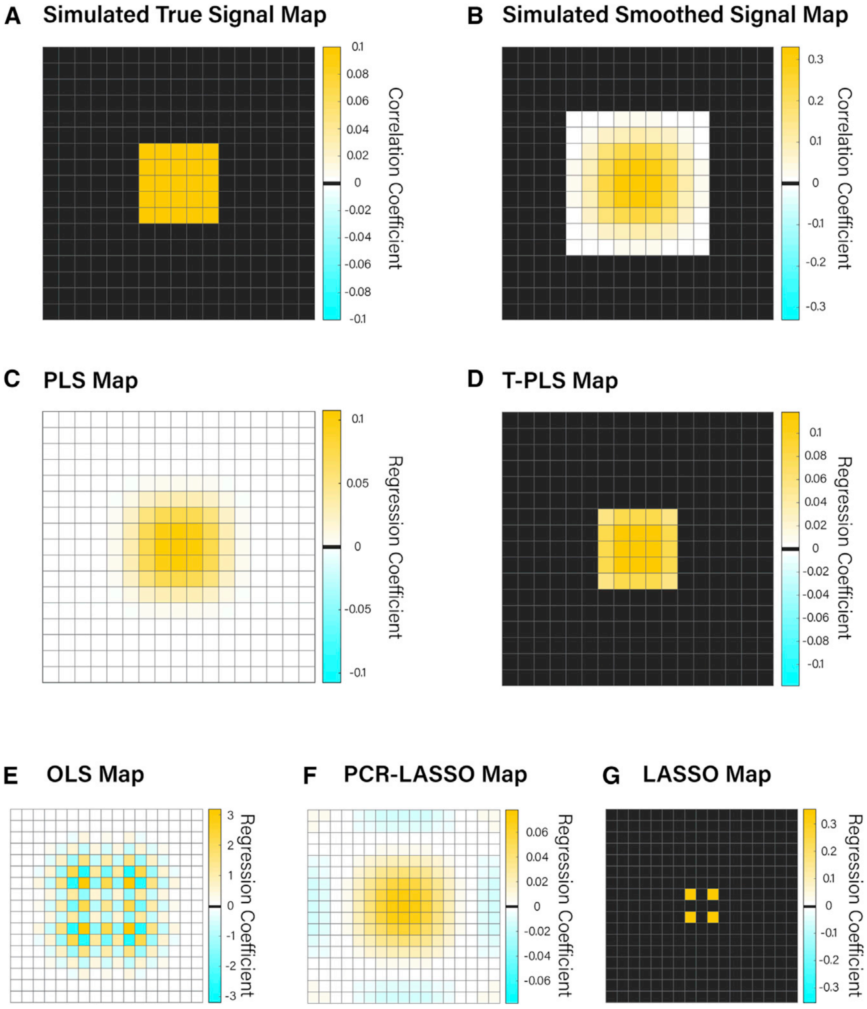


**Figure 3. Simulated neuroimaging signal and various predictor fits**

(A) A simulated 17-by-17 voxel grid in which only the center 5-by-5 grid has a signal that is predictive of Y at correlation $r = 0.1$.

(B) The result of a spatial smoothing filter to (A), meant to simulate fMRI image smoothness. Coefficients that are exactly 0 were marked as black, while those that are close to 0 were marked as near white.

(C)–(G) shows regression coefficients from PLS regression, T-PLS, OLS regression, PCR-LASSO regression, and LASSO regression, respectively. Of the prediction models, only (D) and (G) have variable selection, thereby making most of the voxels exactly 0 (marked black).

Importantly, when the fitting time per participant was assessed, T-PLS was the only algorithm that maintained a constant model fitting speed (∼15 s), whereas PCR-LASSO, LASSO, and PLS showed increasing fitting time per subject as dataset size increases (Figure 4B). For PCR-LASSO, this increase in fitting time per subject is likely because PCA requires inversion operations on the variance covariance matrix of X, which quadratically increases in size until the number of observations matches the number of variables. For LASSO, this increase in fitting time per subject is also likely due to the calculation of gradients based on matrix operations. T-PLS, in contrast, only requires vector calculations, for which the number of variables is the dominating factor. The speed of T-PLS will prove useful for large-scale neuroimaging studies as well as studies that may train many different predictors for different behaviors or mental constructs. In addition, since T-PLS only has to be fit once, this will provide an even greater benefit in computational time.

### *Memory usage*

T-PLS also used a very minimal amount of memory compared to other algorithms (Figure 4C). We broke down memory usage into three parts—default random-access memory (RAM) for loading the statistical program, RAM for loading the data, and RAM for computing the model from the data. The first two parts were the same across all of the algorithms because all of them needed to load the program and the data. The differences across algorithms came from the differences in RAM usage for model computation. T-PLS used the least amount of computation memory, followed by LASSO, PCR-LASSO, and PLS (although in larger datasets, PLS used less memory than PCR-LASSO). We again found that T-PLS was the most scalable out of all of the tested algorithms (Figure 4D). The computation RAM usage of T-PLS converges to approximately the same amount as the RAM needed for loading the data (1.75 to >0.95 times as dataset size increases). This is likely because T-PLS requires a mean-centered copy of the data matrix. Should researchers want, they can mean-center the data beforehand and use even less memory for T-PLS. In contrast, the RAM usage of LASSO for computation ranged from 2.6 to 2.2 times the data RAM size, while the RAM usage of PCR-LASSO for computation increased rapidly as the dataset size increased (3 to >4.9 times). PLS memory usage also decreased as the dataset size increased, which is likely due to the fact that PLS algorithms are generally scalable, but the off-the-shelf version likely has some overhead calculations that are not optimized for this application.

### *Prediction performance*

T-PLS showed the highest predictive performance across all of the tested dataset sizes (Figure 5), with the caveat that LASSO was too expensive to compute at the largest dataset size, since it was expected to take 1 year on average to fit. All three whole-brain predictors provided predictive performances that increased from the smallest dataset size to the largest dataset size. T-PLS showed the highest predictive performances across all of the dataset sizes. LASSO's performance was the worst of the three whole-brain methods in small dataset sizes, but

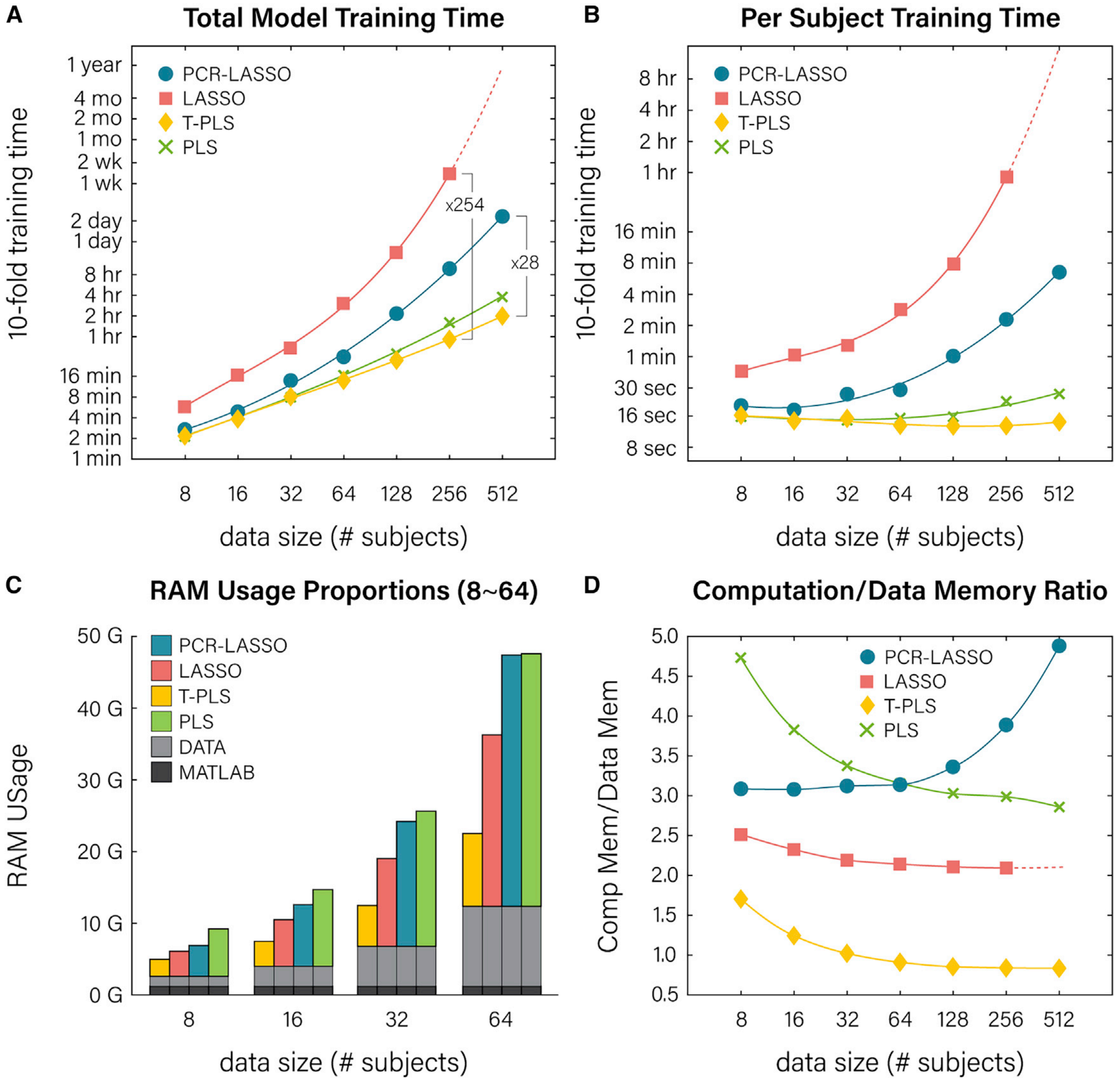


**Figure 4. Computation resource comparison**

(A and B) Ten-fold model training times for each algorithm at various dataset sizes (total time and total time divided by number of subjects, respectively). The lines show the best fitting cubic polynomial spline fit.

(C and D) RAM usage of each algorithm at various dataset sizes. (C) Memory decomposition of each algorithm (memory for turning on MATLAB, for loading data, and for computation). (D) The ratio between RAM usage for loading data and RAM usage for computation (colored versus light gray bar in C).

increased rapidly to be better than PCR-LASSO. PLS also followed a pattern of improvement similar to that of LASSO, being worse than PCR-LASSO in small dataset sizes, but overcoming it at larger dataset sizes. The performance difference between PLS and T-PLS likely highlights the benefit of the additional thresholding/voxel selection step present in T-PLS but not PLS.

We also compared whole-brain prediction methods against commonly used ROI-based methods by using a meta-analysis ROI from Bartra et al. (2013) to create ROI-average predictions (which do not require model training) and ROI-multivariate predictions (which use the training dataset to create multivariate predictors from the ROI). Predictors based on only the voxels within the ROI provided the worst performance across the board, with the ROI-multivariate method outperforming the ROI-average method. This demonstrates how whole-brain predictors can harness more signals from across the brain to provide greater predictive power than regional approaches.

### *Predictor interpretability*

T-PLS provided predictor maps that were easily interpretable and that differed in important ways from those of other approaches (Figure 6). T-PLS resulted in whole-brain predictors with regionally clustered coefficients and voxel selection (Figure 6A). PCR-LASSO also led to regionally clustered coefficients, but without voxel selection (Figure 6B). In contrast, LASSO predictors selected single voxels that were the most important for prediction; however, single voxels can be very difficult to

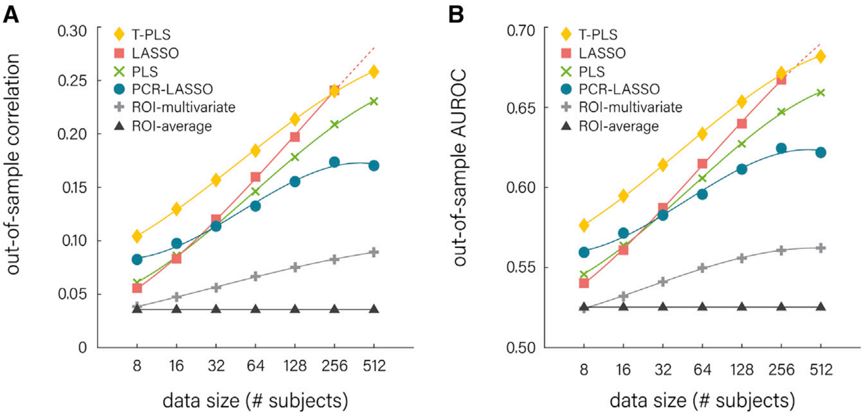


**Figure 5. Out-of-sample prediction accuracy of various algorithms**

Out-of-sample prediction performances of 5 algorithms measured via Pearson correlation (A) and AUROC (B) for predicting value-based accept/reject choices. The lines show best fitting cubic polynomial spline fit.

interpret because it is not always easy to pinpoint the region from which the voxels originate (Figure 6C). PLS, as expected, resulted in whole-brain predictors with no voxel selection, similar to PCR-LASSO (Figure 6D).

Apart from the interpretability of the finished predictor, T-PLS can also provide useful information on the relative importance of different brain regions by showing the trade off between additional thresholding and cross-validation performance (Figures 6E and 6F). Users can experiment with different thresholds to see how much predictive performance is sacrificed when fewer regions are "recruited" into the predictor. Figures 6G–6I show various predictors at different thresholds where stringent thresholds (e.g., Figure 6I) highlight the more important brain regions for prediction. This analysis is possible due to the fit once, tune later approach of T-PLS, which allows users to generate and compare as many predictor maps as they want without re-fitting the model.

## DISCUSSION

Brain decoding has become a powerful tool in linking neural activity to mental states. With fMRI in particular, decoding from the whole brain promises to be more specific and sensitive to the mental state of interest. In this paper, we introduced thresholded partial least squares (T-PLS) to address two major challenges of whole-brain prediction—computational load and interpretability. T-PLS exploits the analytical properties of PLS algorithms to dramatically reduce model fitting time, use less computational memory, and still provide high predictive performance. In a real neuroimaging dataset, T-PLS exhibited a per-subject fitting time that was fixed and hence scalable, unlike LASSO or PCR-LASSO. T-PLS also showed higher out-of-sample predictive performances than other whole-brain methods. Perhaps most important, T-PLS boasts a unique fit once, tune later feature, which not only leads to faster cross-validation but also allows researchers to explore various tuning parameters to choose the best level of sparsity given the trade off between parsimony and performance.

T-PLS builds upon previous uses of PLS in fMRI prediction (Kragel and LaBar, 2014; McIntosh et al., 1996) by introducing variable (voxel) selection that is based on fast analytical computation and cross-validation. T-PLS calculates the z statistics of the PLS components, which are usually not needed since PLS components are created to explain the most covariance to analytically compute the relative importance of each voxel by back-projection. This thresholding by variable (voxel) importance is an improvement from previous thresholding approaches, which either used arbitrary thresholds for the sake of interpretability (McIntosh et al., 1996) or time-consuming bootstrap measures to calculate each voxels' p values and create a thresholded map (which is different from the actual map used for prediction; Kragel and LaBar, 2014). Cross-validation can provide a principled data-driven method of thresholding that takes into account the amount of data, signal quality, and generalizability.

While T-PLS is similar to other methods such as PCR-LASSO that involve data reduction followed by regression, T-PLS has several beneficial features that aid in predictive performance and computational efficiency. For one, unlike PCA, PLS components are ordered in terms of covariance explained in X and Y, which ensures that the most useful components are the first few to be estimated, which in turn reduces the number of components to entertain. Also, the component selection method of T-PLS yields the same result as LASSO selection without having to fit LASSO. This is because in LASSO regression, if all of the predictor variables are orthogonal, the variable selection order follows the absolute size of the coefficients, which in the case of PLS, coincides with the original order of PLS components, since components are extracted in the order of most covariance explained.

As with all whole-brain prediction models, it is important to distinguish the maps obtained from prediction analyses from the maps obtained from a mass-univariate GLM analysis. To the former, the brain data are the predictor, while to the latter, it is the predicted. Researchers interested in making loci-specific inferences should consider using ROI prediction analyses in parallel or converting the T-PLS coefficient map to a map that can be interpreted like a GLM (Haufe et al., 2014). We see the relationship between whole-brain prediction methods and ROI-based prediction methods and GLM analyses as similar to that between multivariate regression and pairwise bivariate correlation. ROI-based methods and GLM analyses may yield insight about how one specific region is related to a mental process, but a whole-brain method can yield insight about how multiple ROIs combine with one another and contribute to prediction; they are both important analysis tools that every researcher must use to understand the whole picture. Concordantly, we see whole-brain prediction as an important analysis tool in the coming years in neuroimaging, and we hope that the method

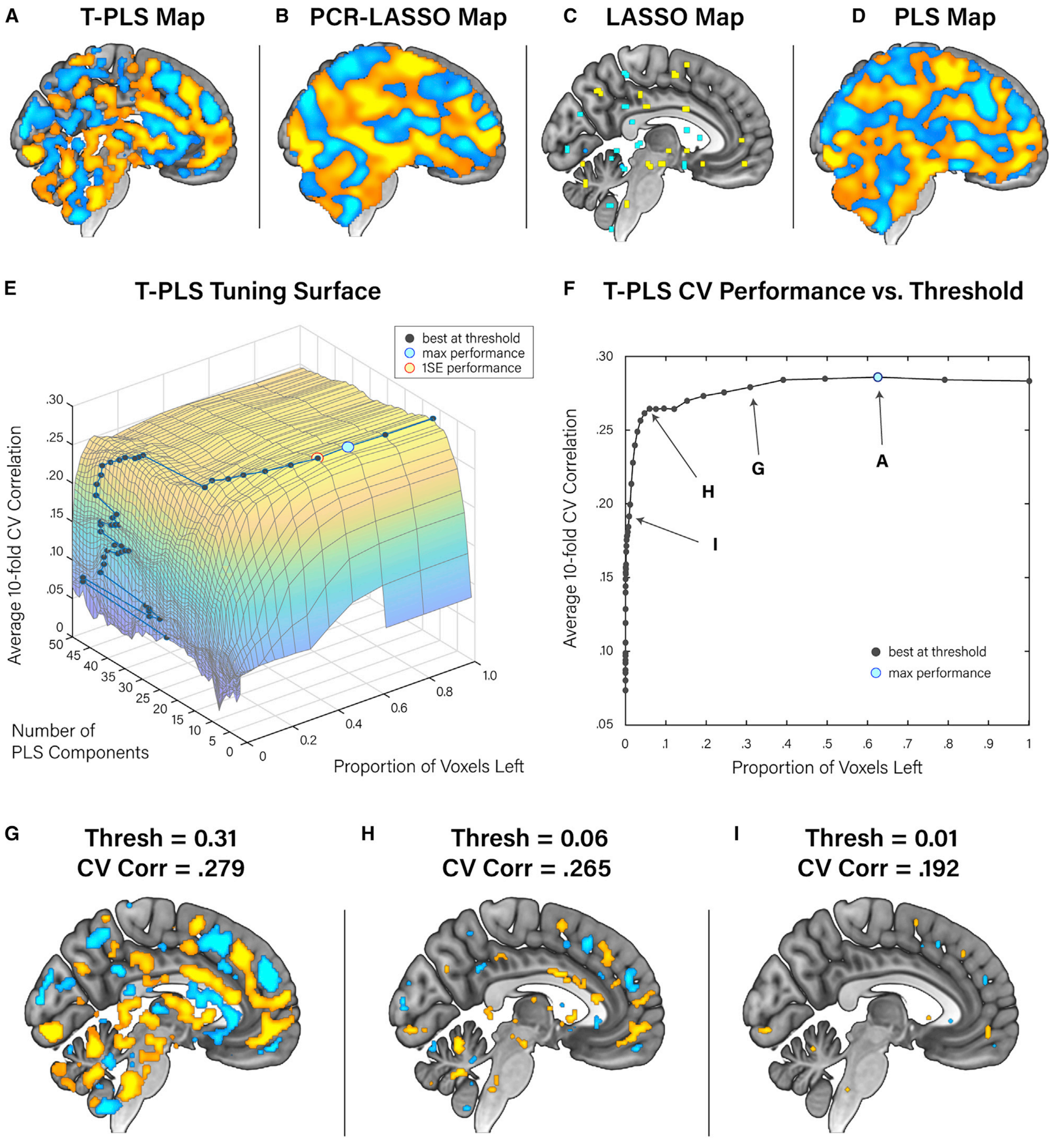


**Figure 6. Final predictors of value-based choice**

(A–D) The whole-brain predictor of value-based choice constructed via T-PLS, PCR-LASSO, LASSO, and PLS, respectively.

(E) Ten-fold cross-validation tuning curve for fitting the T-PLS predictor.

(F) Ten-fold cross-validation performance of the T-PLS model at various thresholding levels.

(G–I) Corresponding thresholded predictors from (F). The positive coefficients are marked with warm colors, the negative with cold colors, and the units are arbitrary because fMRI signals are in arbitrary units.

```
1. Inputs
    n-by-p matrix of predictors   $\mathbf{X}$
    n-by-1 vector to be predicted   $\mathbf{y}$
    n-by-1 vector of observation weights   $\mathbf{w}$

2. Weighted mean centering
    $\bar{\mathbf{X}}_{,j} = \mathbf{X}_{,j} - \mathbf{w}^{T}\mathbf{X}_{,j}$
    $\bar{\mathbf{y}} = \mathbf{y} - \mathbf{w}^{T}\mathbf{y}$

3. Initial weighted covariance
    $\mathbf{v}_1 = \bar{\mathbf{X}}^{T}(\mathbf{w} \odot \bar{\mathbf{y}})$

begin loop to calculate k^th PLS model. k = 1, 2, ...

    4. Calculate k^th PLS component, coefficient, and back-projection
        PLS component
            $\mathbf{C}_{,k} = \bar{\mathbf{X}}\mathbf{v}_k$
            $c_{norm} = \mathrm{sqrt}(\mathbf{w}^{T}\mathbf{C}_{,k}^{\circ 2})$
            $\mathbf{C}_{,k} = \mathbf{C}_{,k}/c_{norm}$
        PLS coefficient
            $\mathbf{b}_k = ||\mathbf{v}_k||^{2}/c_{norm}$
        Back-projection map
            $\mathbf{P}_{,k} = \mathbf{v}_k/c_{norm}$

    5. Update covariance by deflating
        weighted covariance of X and k^th component
            $\mathbf{h} = \bar{\mathbf{X}}^{T}(\mathbf{w} \odot \mathbf{C}_{,k})$
        deflating covariance
            $\mathbf{v}_{k+1} = \mathbf{v}_k - \mathbf{h}(\mathbf{h}^{T}\mathbf{v}_k)$

    6. Back-projection of coefficients
        k component T-PLS model coefficients   $\mathbf{B}_{,k} = \mathbf{P}_{,1:k}\mathbf{b}_{1:k}$

    7. Calculation and back-projection of z-statistics
        residuals from a k component PLS model   $\mathbf{r}_k = \mathbf{r}_{k-1} - \mathbf{C}_{,k}\mathbf{b}_k$
        standard error of k PLS coefficients   $\mathbf{se} = \mathrm{sqrt}[\,(\mathbf{C}_{,1:k}^{\circ 2})^{T}\,(\mathbf{w}^{\circ 2} \odot \mathbf{r}_k^{\circ 2})\,]$
        k component T-PLS model z-statistics   $\mathbf{Z}_{,k} = [\,\mathbf{P}_{,1:k}(\mathbf{b}_{1:k} \oslash \mathbf{se})\,] \oslash \mathrm{sqrt}(\mathrm{rowsum}(\mathbf{P}_{,1:k}^{\circ 2}))$

end of loop
```

**Figure 7. Summary of T-PLS fitting algorithm**

Matrices are denoted with boldface capital letters, vectors with boldface lowercase letters, and scalars with lightface lowercase letters. ⊙ denotes Hadamard product (element-wise multiplication), ⊘ denotes element-wise division, and ∘2 in the exponent denotes element-wise squaring.

that we propose here, along with the provided packages, can make this analysis a convenient and essential part of neuroimaging analysis pipelines.

## Limitations of the study

It is important to acknowledge that we did not compare all of the possible prediction methods. Even PCR and LASSO, which we have entertained here, have a large number of potential addendums and improvements (e.g., stochastic PCA that reduces RAM usage and computation time at the cost of accuracy [Halko et al., 2011]; stochastic gradient descent methods for LASSO that use subsamples of the data, which we explored in the analyses above and found were still much slower than T-PLS and less accurate than LASSO without stochastic gradient descent). One method we did not compare is GraphNet, which is a penalized regression method based on elastic-net that can yield interpretable whole-brain predictors (Grosenick et al., 2013). While we believe that GraphNet is a principled method of addressing whole-brain prediction, previous research has found that GraphNet yields lower predictive performance than LASSO (Mohr et al., 2015), which already showed lower performance than T-PLS in our analyses. Furthermore, given that the penalties are based on elastic-net, which has more tuning parameters than LASSO, we expect this algorithm to be slower than LASSO, which was already the slowest algorithm in our analyses. Finally, we also left out non-linear methods such as neural nets (Thomas et al., 2019), Gaussian processes (Marquand et al., 2010), or naive-Bayes (Kassam et al., 2013), as these have been used mostly in region-based decoding and would be expected to be considerably slower, given many more tuning parameters, as well as harder to interpret than linear models. Previous studies have also found that non-linear methods applied to the whole-brain fMRI data do not seem to yield higher performances than linear methods (Schulz et al., 2020; Thomas et al., 2020).

In addition, we tested only task-based prediction of value-based choice in this paper. A general caveat is that the highest performing algorithm can vary depending on the specifics of the dataset and the goal. For example, the predictive power of PCR-LASSO may depend heavily on whether the predictive neural signal explains a large amount of variance in the brain image; if so, then it is likely that the more prediction-pertinent components would be extracted earlier. However, the computational benefits of T-PLS over likelihood-based methods such as LASSO will be more apparent in high-dimensional data (in both the number of variables and number of observations). Based on the results in this paper, we expect T-PLS to be particularly useful in general high-dimensional prediction problems in which the relevant predictive signal constitutes only a small portion of the total variance of the predictors. We have made the T-PLS package available in multiple statistical languages so that researchers may try it easily and compare it with other methods in their research.

## STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - ○ Lead contact
  - ○ Materials availability
  - ○ Data and code availability
- METHOD DETAILS
  - ○ T-PLS algorithm - Fitting
  - ○ Neuroimaging dataset
- QUANTIFICATION AND STATISTICAL ANALYSIS
  - ○ Computation comparisons
  - ○ Predictive power comparisons
  - ○ Interpretability comparisons

## ACKNOWLEDGMENTS

The data were funded by grants from the National Cancer Institute R01 CA170297 to J.W.K. and Caryn Lerman. The high-performance Biomedical Image Computing and Informatics Cluster (https://www.med.upenn.edu/cbica/cubic.html) at the University of Pennsylvania Center for Biomedical Image Computing and Analytics was supported by the National Institutes of Health, grant no. 1S10OD023495-01 to Christos Davatzikos.

## AUTHOR CONTRIBUTIONS

Conceptualization, S.L. Methodology, S.L. Formal analysis, S.L. Investigation, S.L., E.T.B., and J.W.K. Writing – original draft, S.L. and E.T.B. Writing – review & editing, S.L., E.T.B., and J.W.K. Funding acquisition, J.W.K.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: October 29, 2021
Revised: April 11, 2022
Accepted: May 16, 2022
Published: June 6, 2022

## REFERENCES

Allen, N.E., Sudlow, C., Peakman, T., and Collins, R. (2014). UK biobank data: come and get it. Sci. Transl. Med. https://doi.org/10.1126/scitranslmed.3008601.

Bartra, O., McGuire, J.T., and Kable, J.W. (2013). The valuation system: a coordinate-based meta-analysis of BOLD fMRI experiments examining neural correlates of subjective value. Neuroimage *76*, 412–427. https://doi.org/10.1016/j.neuroimage.2013.02.063.

Bjork, J.M., Straub, L.K., Provost, R.G., and Neale, M.C. (2017). The ABCD study of neurodevelopment: identifying neurocircuit targets for prevention and treatment of adolescent substance abuse. Curr. Treat. Options Psychiatry. https://doi.org/10.1007/s40501-017-0108-y.

Chang, L.J., Gianaros, P.J., Manuck, S.B., Krishnan, A., and Wager, T.D. (2015). A sensitive and specific neural signature for picture-induced negative affect. PLoS Biol. *13*, 1–28. https://doi.org/10.1371/journal.pbio.1002180.

Cox, D.D., and Savoy, R.L. (2003). Functional magnetic resonance imaging (fMRI)"brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. Neuroimage *19*, 261–270. https://doi.org/10.1016/s1053-8119(03)00049-1.

de Jong, S. (1993). SIMPLS: an alternative approach to partial least squares regression. Chemometr. Intell. Lab. Syst. https://doi.org/10.1016/0169-7439(93)85002-X.

Etzel, J.A., Zacks, J.M., and Braver, T.S. (2013). Searchlight analysis: promise, pitfalls, and potential. NeuroImage. https://doi.org/10.1016/j.neuroimage.2013.03.041.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. *33*, 1–22. https://doi.org/10.18637/jss.v033.i01.

Green, L., and Myerson, J. (2004). A discounting framework for choice with delayed and probabilistic rewards. Psychol. Bull. https://doi.org/10.1037/0033-2909.130.5.769.

Grosenick, L., Klingenberg, B., Katovich, K., Knutson, B., and Taylor, J.E. (2013). Interpretable whole-brain prediction analysis with GraphNet. Neuroimage *72*, 304–321. https://doi.org/10.1016/j.neuroimage.2012.12.062.

Halko, N., Martinsson, P.G., and Tropp, J.A. (2011). Finding Structure with Randomness: Probabilistic Algorithms for Matrix Decompositions. SIAM Review *52*. https://doi.org/10.1137/090771806.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., and Bießmann, F. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. Neuroimage *87*, 96–110. https://doi.org/10.1016/j.neuroimage.2013.10.067.

Hochberg, L.R., Bacher, D., Jarosiewicz, B., Masse, N.Y., Simeral, J.D., Vogel, J., Haddadin, S., Liu, J., Cash, S.S., Van Der Smagt, P., and Donoghue, J.P. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature *485*, 372–375. https://doi.org/10.1038/nature11076.

Hultman, R., Ulrich, K., Sachs, B.D., Blount, C., Carlson, D.E., Ndubuizu, N., Bagot, R.C., Parise, E.M., Vu, M.-A.T., Gallagher, N.M., et al. (2018). Brain-wide electrical spatiotemporal dynamics encode depression vulnerability. Cell *173*, 166–180.e14. https://doi.org/10.1016/j.cell.2018.02.012.

Johnson, A., and Redish, A.D. (2007). Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point. J. Neurosci. *27*, 12176–12189. https://doi.org/10.1523/JNEUROSCI.3761-07.2007.

Jung, W.H., Lee, S., Lerman, C., and Kable, J.W. (2018). Amygdala functional and structural connectivity predicts individual risk tolerance. Neuron *98*, 394–404.e4. https://doi.org/10.1016/j.neuron.2018.03.019.

Kable, J.W., Caulfield, M.K., Falcone, M., McConnell, M., Bernardo, L., Parthasarathi, T., Cooper, N., Ashare, R., Audrain-McGovern, J., and Hornik, R. (2017). No effect of commercial cognitive training on neural activity during decision-making. J. Neurosci., 2816–2832.

Kable, J.W., and Glimcher, P.W. (2007). The neural correlates of subjective value during intertemporal choice. Nat. Neurosci. *10*, 1625–1633. https://doi.org/10.1038/nn2007.

Kahneman, D., and Tversky, A. (1979). Kahneman & tversky (1979) - prospect theory - an analysis of decision under risk. Econometrica. https://doi.org/10.2307/1914185.

Kassam, K.S., Markey, A.R., Cherkassky, V.L., Loewenstein, G., and Just, M.A. (2013). Identifying emotions on the basis of neural activation. PLoS One *8*, e66032. https://doi.org/10.1371/journal.pone.0066032.

Kay, K.N., Naselaris, T., Prenger, R.J., and Gallant, J.L. (2008). Identifying natural images from human brain activity. Nature *452*, 352–355. https://doi.org/10.1038/nature06713.

Kohoutová, L., Heo, J., Cha, S., Lee, S., Moon, T., Wager, T.D., and Woo, C.-W. (2020). Toward a unified framework for interpreting machine-learning models in neuroimaging. Nat. Protoc. *15*, 1399–1435. https://doi.org/10.1038/s41596-019-0289-5.

Kragel, P.A., and LaBar, K.S. (2014). Multivariate neural biomarkers of emotional states are categorically distinct. Soc. Cogn. Affect. Neurosci. *10*, 1437–1448. https://doi.org/10.1093/scan/nsv032.

Kriegeskorte, N., Goebel, R., and Bandettini, P. (2006). Information-based functional brain mapping. Proc. Natl. Acad. Sci. U S A *103*, 3863–3868. https://doi.org/10.1073/pnas.0600244103.

Kucyi, A., Esterman, M., Capella, J., Green, A., Uchida, M., Biederman, J., Gabrieli, J.D.E., Valera, E.M., and Whitfield-Gabrieli, S. (2021). Prediction of stimulus-independent and task-unrelated thought from functional brain networks. Nat. Commun. *12*, 1–17.

Lever, J., Krzywinski, M., and Altman, N. (2017). Points of significance: principal component analysis. Nat. Methods. https://doi.org/10.1038/nmeth.4346.

Marquand, A., Howard, M., Brammer, M., Chu, C., Coen, S., and Mourão-Miranda, J. (2010). Quantitative prediction of subjective pain intensity from whole-brain fMRI data using Gaussian processes. Neuroimage *49*, 2178–2189. https://doi.org/10.1016/j.neuroimage.2009.10.072.

McIntosh, A.R., Bookstein, F.L., Haxby, J.V., and Grady, C.L. (1996). Spatial pattern analysis of functional brain images using partial least squares. Neuroimage *3*, 143–157. https://doi.org/10.1006/nimg.1996.0016.

Miranda-Dominguez, O., Feczko, E., Grayson, D.S., Walum, H., Nigg, J.T., and Fair, D.A. (2018). Heritability of the human connectome: a connectotyping study. Netw. Neurosci. *2*, 175–199. https://doi.org/10.1162/netn_a_00029.

Mohr, H., Wolfensteller, U., Frimmel, S., and Ruge, H. (2015). Sparse regularization techniques provide novel insights into outcome integration processes. Neuroimage *104*, 163–176. https://doi.org/10.1016/j.neuroimage.2014.10.025.

Poldrack, R.A. (2006). Can cognitive processes be inferred from neuroimaging data? Trends. Cogn. Sci. *10*, 59–63. https://doi.org/10.1016/j.tics.2005.12.004.

Poldrack, R.A. (2011). Inferring mental states from neuroimaging data: from reverse inference to large-scale decoding. Neuron *72*, 692–697. https://doi.org/10.1016/j.neuron.2011.11.001.

Poldrack, R.A., Halchenko, Y.O., and Hanson, S.J. (2009). Decoding the large-scale structure of brain function by classifying mental states across individuals. Psychol. Sci. *20*, 1364–1372. https://doi.org/10.1111/j.1467-9280.2009.02460.x.

Poldrack, R.A., Huckins, G., and Varoquaux, G. (2020). Establishment of best practices for evidence for prediction: a review. JAMA Psychiatry *77*, 534–540. https://doi.org/10.1001/jamapsychiatry.2019.3671.

Qian, J., Hastie, T., Friedman, J., Tibshirani, R., and Simon, N. (2013). Glmnet for Matlab. http://www.stanford.edu/~hastie/glmnet_matlab.

Rissman, J., Gazzaley, A., and D'Esposito, M. (2004). Measuring functional connectivity during distinct stages of a cognitive task. Neuroimage *23*, 752–763.

Rosenberg, M.D., Finn, E.S., Scheinost, D., Papademetris, X., Shen, X., Constable, R.T., and Chun, M.M. (2016). A neuromarker of sustained attention from whole-brain functional connectivity. Nat. Neurosci. *19*, 165–171. https://doi.org/10.1038/nn.4179.

Samuelson, P.a. (1937). Note on measurement of utility. Rev. Econ. Stud. *4*, 155–161.

Satterthwaite, T.D., Elliott, M.A., Ruparel, K., Loughead, J., Prabhakaran, K., Calkins, M.E., Hopson, R., Jackson, C., Keefe, J., Riley, M., et al. (2014). Neuroimaging of the philadelphia neurodevelopmental cohort. NeuroImage. https://doi.org/10.1016/j.neuroimage.2013.07.064.

Schulz, M.-A., Yeo, B.T.T., Vogelstein, J.T., Mourao-Miranada, J., Kather, J.N., Kording, K., Richards, B., and Bzdok, D. (2020). Different scaling of linear models and deep learning in UKBiobank brain images versus machine-learning datasets. Nat. Commun. *11*, 4238–4315. https://doi.org/10.1038/s41467-020-18037-z.

Smith, A., Douglas Bernheim, B., Camerer, C.F., and Rangel, A. (2014). Neural activity reveals preferences without choices. Am. Econ. J. Microecon. *6*, 1–36. https://doi.org/10.1257/mic.6.2.1.

Thomas, A.W., Heekeren, H.R., Müller, K.R., and Samek, W. (2019). Analyzing neuroimaging data through recurrent deep learning models. Front. Neurosci. https://doi.org/10.3389/fnins.2019.01321.

Thomas, R.M., Gallo, S., Cerliani, L., Zhutovsky, P., El-Gazzar, A., and van Wingen, G. (2020). Classifying autism spectrum disorder using the temporal statistics of resting-state functional MRI data with 3D convolutional neural networks. Front. Psychiatry *11*, 440. https://doi.org/10.3389/fpsyt.2020.00440.

Van Essen, D.C., Ugurbil, K., Auerbach, E., Barch, D., Behrens, T.E.J., Bucholz, R., Chang, A., Chen, L., Corbetta, M., Curtiss, S.W., et al. (2012). The Human Connectome Project: a data acquisition perspective. NeuroImage. https://doi.org/10.1016/j.neuroimage.2012.02.018.

Wager, T.D., Atlas, L.Y., Lindquist, M.A., Roy, M., Woo, C.W., and Kross, E. (2013). An fMRI-based neurologic signature of physical pain. N. Engl. J. Med. *368*, 1388–1397. https://doi.org/10.1056/NEJMoa1204471.

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. Econometrica. https://doi.org/10.2307/1912934.

Whitehead, J.C., and Armony, J.L. (2019). Multivariate fMRI pattern analysis of fear perception across modalities. Eur. J. Neurosci. *49*, 1552–1563. https://doi.org/10.1111/ejn.14322.

Willett, F.R., Avansino, D.T., Hochberg, L.R., Henderson, J.M., and Shenoy, K.V. (2021). High-performance brain-to-text communication via handwriting. Nature *593*, 249–254. https://doi.org/10.1038/s41586-021-03506-2.

Yamashita, M., Yoshihara, Y., Hashimoto, R., Yahata, N., Ichikawa, N., Sakai, Y., Yamada, T., Matsukawa, N., Okada, G., Tanaka, S.C., et al. (2018). A prediction model of working memory across health and psychiatric disease using whole-brain functional connectivity. Elife *7*, e38844. https://doi.org/10.7554/eLife.38844.

# STAR★METHODS

## KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| Deposited data | | |
| Original dataset (raw fMRI, anatomical data) | Kable et al., (2017) | https://doi.org/10.18112/openneuro.ds002843.v1.0.1 |
| Analysis codes and interim data | This Paper | https://doi.org/10.17605/OSF.IO/JRTYU |
| Software and algorithms | | |
| Custom code | This paper | https://doi.org/10.17605/OSF.IO/JRTYU |

## RESOURCE AVAILABILITY

### Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Sangil Lee (sangillee@gmail.com).

### Materials availability

This study did not generate new unique reagents.

### Data and code availability

- This paper analyzes existing, publicly available data at openneuro (OpenNeuro: https://doi.org/10.18112/openneuro.ds002843.v1.0.1).
- All code and data are available online and are publicly available at open science framework (OSF: https://doi.org/10.17605/OSF.IO/JRTYU).
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

## METHOD DETAILS

### T-PLS algorithm - Fitting

The fitting algorithm for T-PLS is a combination of three parts: a modified SIMPLS algorithm for PLS (de Jong, 1993), back-projection, and calculation of z-statistics (Figure 7). The one modification that we make to the SIMPLS algorithm is in normalizing the PLS components to have weighted unit variance (step 4 in Figure 7). This facilitates computation of z-statistics later in the algorithm (step 7 in Figure 7). Below we detail the back-projection and z-statistic calculation steps of T-PLS as the remaining steps are typical procedures of a SIMPLS algorithm.

### Back-projection

After PLS components have been calculated (up to $k$ $^{th}$ component), we now have a $k$-component PLS regression model. To improve the interpretability of this PLS model, we can convert the PLS regression coefficients into coefficients for the original voxels. Since PLS components are created via weighted sums of original voxels (i.e., component = weight * voxels), one can simply multiply the PLS coefficient to the weights to create back-projected coefficients (i.e., coefficient * component = coefficient * weight * voxels = back-projected coefficient * voxels). This expresses the PLS regression in terms of each voxel's coefficients, which can make the predictor easier to interpret by identifying which regions are positively or negatively predictive of behavior or mental states. Back-projection is also used in the PCR-LASSO method applied by Wager et al. (2013), but with PCA components rather than PLS.

### Z-statistic calculation

We then calculate each voxel's measure of variable importance. We start by calculating the heteroscedasticity-consistent standard errors (also known as sandwich estimators; White, 1980):

$$Var(\mathbf{b}) = \left(\mathbf{C}^{\mathbf{T}}\mathbf{diag}(\mathbf{w})\mathbf{C}\right)^{-1}\mathbf{C}^{\mathbf{T}}diag(\mathbf{w})M\mathbf{diag}(\mathbf{w})^{\mathbf{T}}C\left(\mathbf{C}^{\mathbf{T}}\mathbf{diag}(\mathbf{w})\mathbf{C}\right)^{-1} \qquad \text{(Equation 1)}$$

where b denotes the coefficient estimates, w denotes the observation (trial) weights, M denotes the variance-covariance matrix for the observations, and C denotes the PLS components in a column-wise matrix. Here is where our modification to the SIMPLS algorithm becomes useful. Since the PLS components (matrix C) are all orthonormal (in weighted space), $\mathbf{C}^{\mathbf{T}}\mathbf{diag}(\mathbf{w})\mathbf{C}$ becomes an identity matrix, which cancels out the 'breads' of the sandwich and leaves us with $\mathbf{Var}(\mathbf{b}) = \mathbf{C}^{\mathbf{T}}\mathbf{diag}(\mathbf{w})\mathbf{Mdiag}(\mathbf{w})^{\mathbf{T}}\mathbf{C}$. Since we only need the diagonals of the variance-covariance matrix, we can express the standard error estimates concisely as the following:

$$se(\mathbf{b}) = \sqrt{\left(\left(\mathbf{C}^{\circ 2}\right)^{\mathbf{T}}\left(\mathbf{w}^{\circ 2} \odot \mathbf{r}^{\circ 2}\right)\right)} \qquad \text{(Equation 2)}$$

where $\mathbf{r}^{\circ 2}$ denotes the squared residual vector. The t-statistics (which are close to z-statistics with sufficient observations) can be then calculated by simple element-wise division of $b$ by $\mathbf{se}(\mathbf{b})$. Let this vector be denoted z. Then, we back-project the z statistic like the coefficients, and then normalize them so that they all have unit variance:

$$\mathbf{Z}_{,\mathbf{k}} = \left[\mathbf{P}_{,1:\mathbf{k}}\left(\frac{\mathbf{b}_{1:\mathbf{k}}}{\mathbf{se}}\right)\right] \Big/ \sqrt{\mathbf{rowsum}\left(\mathbf{P}^{\circ 2}_{,1:\mathbf{k}}\right)} \qquad \text{(Equation 3)}$$

where $\mathbf{Z}_{,\mathbf{k}}$ is the variable importance of each voxel calculated from a $k$ component T-PLS model, and **rowsum** denotes the vector that is the row sum of a matrix. This summarizes the fitting procedure of T-PLS. It is important to note that the back-projected z-statistics are *no longer* z-statistics as originally intended to test the significance of a single component given all other components. During back-projection, multiple z-statistics are weighted and combined which makes them unrelated to any null hypothesis. They do, however, provide a measure of signal to noise ratio (SNR) since z-statistics are calculated by dividing the coefficients by the standard error. It is this SNR aspect that we use here as a variable importance measure.

After both the fitting and tuning is complete (i.e., when number of PLS components and thresholding level has been decided), there is one more step that may be useful in some scenarios: post-fitting of bias (intercept). Since some variables are removed during the thresholding stage, the intercept should be re-fitted after thresholding. Let's say that we chose to evaluate a model with $j$ components, thresholded at 70% (removing 70% of variables). Then, the coefficients are $\mathbf{B}_{\cdot,\mathbf{j}}$ multiplied by index vector d where $\mathbf{d_i} = 1$ if the voxel's rank is in the top 30% and 0 otherwise. Then the new intercept is simply the difference between the weighted means of X and y:

$$\mathrm{b}_0 = \mathbf{w}^{\mathbf{T}} y - \mathbf{w}^{\mathbf{T}} X(\mathbf{B}_{\mathbf{j}} \circ \mathbf{d}). \qquad \text{(Equation 4)}$$

However, given the inherent unitless property of fMRI data, it may be best to test predictions at the run-level using correlation or AUC measures, neither of which depend on the intercept and the relative scaling of prediction scores and voxel activity level scaling.

## Neuroimaging dataset

We used a large neuroimaging dataset from Kable et al. (2017) to empirically compare T-PLS against two other whole-brain methods (PCR-LASSO and LASSO) as well as region-based (partial-brain) predictions. We chose this dataset as it was the most readily available large-scale dataset with whole-brain coverage that can be used to decode behavior from brain activity levels. Participants completed two experimental decision-making tasks, intertemporal choice and risky choice, which are both very common in the domain of social science (psychology, economics research, e.g., Green and Myerson, 2004; Kahneman and Tversky, 1979; Samuelson, 1937) and its interaction with neuroscience (e.g., Jung et al., 2018; Kable and Glimcher, 2007). In intertemporal choice, participants made choices between a smaller immediate monetary amount of $20 and a larger but delayed monetary amount (e.g., $40 in 30 days). In risky choice, subjects made choices between a smaller certain monetary amount of $20 and a larger but probabilistic monetary amount (e.g., $40 with 60% probability of winning). In both tasks the larger amount varied from trial to trial, as well as the associated delay or the risk, while the smaller monetary option was always fixed at $20. Only the larger monetary option was on the screen while the smaller $20 was not; participants made accept/reject choices based on whether they would prefer the larger monetary option on the screen or the smaller monetary option. Because the value of one of the options was always constant, Kable et al. (2017) were able to find signals in the brain that correlated with the subjective value of the varying option that was shown on the screen. Based on this result, we seek here to create a whole-brain predictor of choice that can use these signals to predict whether the participant will accept the option on the screen or reject it.

Some of the data from Kable et al. (2017) was removed from the analysis due to trivial reasons. To keep the number of observations per subject roughly similar, we excluded four pilot participants who had more trials than others. Counting both session 1 and session 2 data, we had 286 sessions worth of data, each with 120 binary choices. From here, we removed 4 intertemporal choice sessions and 6 risky choice sessions that had premature termination of scan due to technical issues. Additionally, 13 intertemporal choice sessions were removed for having extremely unbalanced choices (either accept or reject more than 95% of the time), and 6 sessions were removed for having too many missed responses (more than a quarter worth of session). For risky choice, 10 sessions were removed for unbalanced choices and 6 sessions were removed for too many missed responses. In total, we had 264 sessions worth of data for ITC and 267 sessions worth of data for RC. While most participants had both ITC and RC tasks, since several subjects only had one task, we decided to treat these two tasks' sessions as separate participants for our analyses. In total, the dataset gave us a total of 61,038 trials (observations) and 184,319 voxels (variables) across 531 task sessions (264 intertemporal choice sessions, 267 risky choice sessions), which we treat as 531 participants in this paper, as our goal is not in making substantive, or comparative, conclusions about the tasks.

The Kable et al. (2017) dataset was acquired with a Siemens 3T Trio scanner with a 32-channel head coil. High-resolution T1-weighted anatomical images were acquired using an MPRAGE sequence (T1 = 1100ms; 160 axial slices, 0.9375 x 0.9375 x 1.000 mm; 192 x 256 matrix). T2*-weighted functional images were acquired using an EPI sequence with 3mm isotropic voxels, 64 x 64 matrix, TR = 3,000ms, TE = 25ms, 53 axial slices, 104 volumes. B0 fieldmap images were collected for distortion correction (TR = 1270ms, TE = 5 and 7.46ms). The images were preprocessed via fMRIPrep 20.0.5. The preprocessing pipeline, in short, performed motion-correction, slice-time correction, and b0-map unwarping on all runs and registered and resampled to a MNI 2mm template. The authors of fMRIPrep has requested the automatically generated preprocessing info to be provided in its unaltered form. Given its length, we provide them with the rest of the analysis codes online at open science framework (https://doi.org/10.17605/OSF.IO/JRTYU).

For estimating the activity of each trial, we used beta-series regression (Rissman et al., 2004). The regressors were time-locked to the trial onset period with event duration of 0.1 seconds and convolved with a gamma HRF function. The last trial of each run was excluded from analysis because the BOLD activity of the last trial was often not observed due to the termination of the scan. This gave us 29 regressor of interest per 1 run of scan. Additionally, we included the following nuisance regressors which were generated from fmriprep: cosine components for high-pass filtering, CSF signal, white matter signal, global signal, standard 6 motion regressors, and 6 PCA components from an anatomical mask of white matter and CSF ('a_comp_cor'). After the single trial coefficients were estimated, all images were smoothed with a FWHM 5mm gaussian filter. To make analysis easy, we only used the voxels that were active for all subjects; this gave us a fairly conservative mask of the brain with 184,319 voxels.

## QUANTIFICATION AND STATISTICAL ANALYSIS

### Computation comparisons

We assessed the scalability of each whole-brain prediction method – LASSO, PCR-LASSO, PLS and T-PLS – by comparing their model fitting time and RAM usage at varying training dataset sizes (8, 16, 32, 64, 128, 256, and 512 participants). In each dataset size (e.g., 8 subjects), half of the data is drawn randomly from the risky choice dataset (i.e., 4 subjects) and the other is drawn randomly from intertemporal choice dataset. Each model is fitted using 10-fold cross-validation (CV). The training data is divided into 10 equal sized blocks and the model is fitted on 9 of the blocks and tested on the left-out block. This is repeated 10 times to assess cross-validation performance. Then, the tuning parameter that yields the highest CV performance is chosen and used to train the final predictor using all training data.

For LASSO, we used GLMNET for MATLAB (Friedman et al., 2010; Qian et al., 2013), which is arguably the fastest non-GPU package for fitting LASSO thanks to its use of regularized path and FORTRAN coding (The original GLMNET package for MATLAB could not import a dataset size of as large a magnitude as in this study because the FORTRAN API with MATLAB was written in 32-bit architecture; we have updated the FORTRAN code ourselves to 64-bit architecture to circumvent this issue; the updated package is provided here: https://github.com/sangillee/GLMNET64MATLAB). We used the default tuning parameter search, which uses 100 lambda values. For PCR-LASSO, we used the approach in the Wager et al. (2013) paper by extracting 200 components from all data, using 10-fold LASSO logistic regression to find the useful components, and subsequently running an unpenalized logistic regression using only the selected components. For PLS, we used the default PLS function in MATLAB ('plsregress') using 10-fold cross-validation to choose the best number of components. For T-PLS, in each of the 10 folds, we extract 25 PLS components and built the T-PLS model. Then, during cross-validation we choose the best-performing number of PLS components and threshold level. Each whole-brain method is fitted 400 times at each dataset size, each time randomly selecting the training data. All computations were performed on a large-scale computation cluster at the University of Pennsylvania (https://www.med.upenn.edu/cbica/cubic).

### Predictive power comparisons

We compared the out-of-sample predictive performances of the predictors built above. After the prediction model is fitted using 10-fold cross validation in the training data (e.g., 32 subjects), the remaining data (e.g., 531-32=499 subjects) is used as an out-of-sample testing dataset. Per-subject correlation and area under the ROC curve (AUC) are averaged across the out-of-sample participants to get an estimate of out-of-sample prediction performance. We also add two commonly used region-based prediction methods to the comparison of predictive power: region-average, and region-multivariate. Region-average is simply taking the average of all voxel activities within a designated region to make predictions; concordantly, region-average does not require fitting a model. Region-multivariate, on the other hand, uses the voxels in the region to build a predictor. While several methods can be used, here we use LASSO to make comparisons with our whole-brain methods easier. We use regions identified from a meta-analysis by Bartra et al. (2013), which examined around 150 neuroimaging studies and identified two regions that consistently showed correlated activity with valuation: ventral striatum and ventromedial prefrontal cortex (regions from Figure 9 of Bartra et al. 2013).

### Interpretability comparisons

We also compared the interpretability of the whole-brain methods (LASSO, PCR-LASSO, PLS, T-PLS). Using the same fitting procedures as before (10-fold cross-validation), T-PLS, PLS, and PCR-LASSO are fit using the entirety of the data (531 participants). LASSO, however, is only fit with a subsampled 256 participant dataset, as it is computationally too slow to fit using the entire dataset.

We visually compare the resulting whole-brain predictors and the associated areas of the brain to assess the scientific face validity of the identified brain regions.

Additionally, we used simulated data to provide further insight into differences in interpretability. We simulate a brain activity signal of a 17x17 voxel grid (total of 289 voxels), of which only a 5x5 grid in the center (25 voxels) carries signal that is predictive of Y, while all other voxels are completely orthogonal to Y (i.e., noise). We achieve this by first randomly generating 290 variables (289 voxels + 1 Y) each with 300 observations from a standard normal distribution. Then, we apply symmetric orthogonalization such that all 290 columns are orthogonal to each other. The first column of the new matrix is chosen as the predicted variable Y, while the other 289 variables became simulations of fMRI noise. To create 25 voxels of predictive voxel signal, we mix Y with 25 of the simulated fMRI noise variables to create 25 signals that are all exactly correlated with Y at r = 0.1. Each column is then z-scored to have unit variance. Finally, we place the 5x5 signal grid in the center of a 17x17 grid and apply 2D Gaussian smoothing (sd = 1 voxel) to simulate the inherent smoothness of fMRI signals. In sum, the resulting dataset is 300 observations of 17x17 voxel grid predictors with only the center 5x5 grid being predictive of Y. This simulated dataset is fit by OLS, LASSO, PCR-LASSO (10 components), PLS (10 components), and T-PLS models (10 components) to compare the resulting pattern of coefficients.