Interpretable Visualization and Higher-Order Dimension Reduction for ECoG Data

Kelly Geyer*
Dept. of Mathematics & Statistics
Boston University
Boston, MA
klgeyer@bu.edu

Frederick Campbell*
Microsoft
Redmond, WA
frcampbe@microsoft.com

Andersen Chang
Dept. of Statistics
Rice University
Houston, TX
atc7@rice.edu

John Magnotti
Dept. of Neurosurgery
Baylor College of Medicine
Houston, TX
magnotti@bcm.edu

Michael Beauchamp
Dept. of Neurosurgery
Baylor College of Medicine
Houston, TX
michael.beauchamp@bcm.edu

Genevera I. Allen[1,2,3]
Dept. of Electrical and Computer Engineering
Rice University
Houston, TX
gallen@rice.edu

***Abstract*—ElectroCOrticoGraphy (ECoG) technology measures electrical activity in the human brain via electrodes placed directly on the cortical surface during neurosurgery. Through its capability to record activity at an extremely fast temporal resolution, ECoG experiments have allowed scientists to better understand how the human brain processes speech. By its nature, ECoG data is extremely difficult for neuroscientists to directly interpret for two major reasons. Firstly, ECoG data tends to be extremely large in size, as each individual experiment yields data up to several GB. Secondly, ECoG data has a complex, higher-order nature; after signal processing, this type of data is typically organized as a 4-way tensor consisting of trials by electrodes by frequency by time. In this paper, we develop an interpretable dimension reduction approach called Regularized Higher Order Principal Components Analysis, as well as an extension to Regularized Higher Order Partial Least Squares, that allows neuroscientists to explore and visualize ECoG data. Our approach employs a sparse and functional Candecomp-Parafac (CP) decomposition that incorporates sparsity to select relevant electrodes and frequency bands, as well as smoothness over time and frequency, yielding directly interpretable factors. We demonstrate both the performance and interpretability of our method with an ECoG case study on audio and visual processing of human speech.**

***Index Terms*—Electrocorticography, tensor decomposition, higher-order PCA, sparse and functional PCA, CP-decomposition**

I. Introduction

ElectroCOrticoGraphy (ECoG) is a technique for measuring the electrical activity of a brain over time, among several locations in the brain. Because it can record brain activity at a higher spatiotemporal frequency and with less noise than previous technologies, ECoG studies have helped scientists understand better than before how the human brain functions for tasks such as speech recognition and processing. In the typical experimental setting involving ECoG, a stimulus such as an image or noise is presented to the subject and the responding activity in their brain is measured. Within each trial, a subject is exposed to a single stimulus, and the electrical activity of the subject's brain is measured by electrodes placed on the surface of the brain. Fourier transforms are then used to obtain spectrograms of brain activity over time for each electrode in each trial. ECoG data naturally give rise to a complex, high-dimensional, higher-order structure which has dependent measurements; the resulting data structures have thousands of covariates, and their files sizes usually exceeds 3-4 GB. Thus, without proper techniques, analyzing and interpreting ECoG data from the large amount of temporal-spatial information collected can be challenging from both a computational and analytical standpoint.

Several approaches have been used in the neuroscience literature to analyze and interpret ECoG data. Researchers may typically analyze a small number of electrodes [21], omitting or condensing down the number of variables and observations substantially. Additionally, they may process the data using procedures that condense a large number of frequencies into several frequency bands [30]. Omission makes an unwieldy data set manageable, but there is work that suggests condensing the data to frequency bands is an oversimplification [14]. Another strategy for decoding ECoG data is the usage of classical multivariate statistical methods such as principal component analysis (PCA) [17] or forward selection [23]. In general, classical multivariate analysis methods are commonly used in neuroimaging

∗ denotes equal contribution. 1-3 denote additional affiliations with 1. Dept. of Statistics, Rice University, 2. Dept. of Computer Science, Rice University, and 3. Jan and Dan Duncan Neurological Research Institute, Baylor College of Medicine.

studies [11, 20]. The primary disadvantage of these methods is that they require flattening of the higher-order structure of ECoG data in to matrix form, as the merging of different dimensions causes the results to lose much of their scientific interpretability. Computational feasibility can be a challenge for these methods as well, since representing higher-order ECoG data as a 2 dimensional matrix necessitates the estimation of an astronomically large number of parameters; this can make several popular dimension reduction and classification techniques, such as linear discriminant analysis (LDA), partial least squares (PLS), and logistic regression, computationally infeasible.

A more sophisticated approach for analyzing ECoG data is the use of tensor decomposition methods. Despite the complex nature of ECoG data, its representation can be simplified and ultimately be described by four modalities: $i.$) trial for each stimulus presented, ii) electrodes, iii) frequency resulting from measurement of electrical activity, and iv) recording time. The modality measurements of ECoG are consistent, despite potential differences between various experimental designs and settings; this means that in general ECoG data may be represented by a four-dimensional tensor, i.e. in the form $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times p \times q \times r}$. Tensor decompositions greatly reduce the number of estimated parameters, by returning one or more factor vectors for each mode. Cichocki [9] and Cichocki et al. [10] suggest that tensors decomposition can improve the analysis of data with spatial-temporal structure, by finding patterns in each of the modes separately. Tensor decomposition methods have been used to analyze ECoG in many previous studies [12, 13, 34, 35, 36, 37]. While these methods preserve the higher-order nature of ECoG data, the results can still be difficult for scientists to interpret, as one can not easily perform feature selection using the results from a basic tensor decomposition.

In this paper, we introduce Regularized Higher-Order Principal Component Analysis (ρ-PCA) as a method of dimension reduction for ECoG data, as well as an extension to Regularized Higher Order Partial Least Squares (ρ-PLS). Our method aims to improve the interpretability of the results of tensor decompositions on ECoG data. To do this, we use regularized tensor factorization techniques, adopting the work done in previous papers for sparse higher-order PCA [1, 3, 5] and building on the methodology presented in those papers to be specifically applied to ECoG data. In particular, ρ-PCA returns sparse tensor estimates along the electrode and frequency modes, which in turn helps to automatically identify the important features along those modalities by detecting the non-zero elements of the tensors. The framework of ρ-PCA also allows for other structures to be imposed on the estimated tensors, such as smoothness over time. We apply this new methodology to ECoG data from a speech perception experiment, in which we use ρ-PCA to make predictions about the stimuli the patient is experiencing though the observed brain response. We show that our method is competitive with other approaches currently used in the literature for analyzing ECoG data, both in terms of classification accuracy and in terms of computational performance. We also demonstrate how one can easily interpret and visualize the results of ρ-PCA through analyzing a specific trial from the aforementioned ECoG data set.

II. Tensor Decomposition for ECoG Analysis

In this section, we describe ρ-PCA mathematically. We refer to the dimensions of a tensor as its modes, and adopt the notation from Kolda and Bader [19]. We denote tensors as $\boldsymbol{\mathcal{X}}$, matrices as $\boldsymbol{X}$, vectors as $\boldsymbol{x}$, and scalars as x. A tensor may be multiplied by a matrix or vector by the d-th mode, using the operation $\times_d$. We can flatten a tensor into a matrix along the d-th mode, denoted by $\boldsymbol{X}_{(d)}$. We define vector norms $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{N} |x_i|$, $\|\boldsymbol{x}\|_2^2 = \sum_{i=1}^{N} |x_i|^2$, and $\|\boldsymbol{x}\|_{\boldsymbol{\Omega}}^2 = \boldsymbol{x}^T \boldsymbol{\Omega} \boldsymbol{x}$, where $\boldsymbol{\Omega} \in \mathbb{R}^{N \times N}$. The Frobenius norm for a matrix is $\|\boldsymbol{X}\|_F^2 = \sum_{i=1}^{N} \sum_{j=1}^{M} |x_{ij}|^2$; this may be generalized for a tensor. The trace operation is $\mathrm{Tr}\left(\boldsymbol{X}\right) = \sum_{i=1}^{n} x_{ii}$ for $X \in \mathbb{R}^{n \times n}$. The soft-thresholding function is defined as $S(x, \lambda) = sign(x) \cdot \left(|x| - \lambda\right)_+$.

A. Building the Metholodology of ρ-PCA

Here, we describe how we derive ρ-PCA from the tensor decomposition called sparse higher-order principal components analysis [1], as well the motivation behind our choice of regularization for each modality. As previously stated, an ECoG recording can be represented as a 4-way tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times p \times q \times r}$. In this tensor, there are n trials, p electrodes, q frequency bands, and r time stamps. ρ-PCA decomposes the ECoG tensor into K sets of sparse, nearly independent vectors for each modality, that sequentially correspond to the strongest patterns in each mode.

Sparse higher-order principal components analysis is based upon CP-decomposition [1]. The CP-decomposition of this tensor results in the sum of K outer products of the component factors:

$$\boldsymbol{\mathcal{X}} \approx \sum_{k=1}^{K} d_k \cdot \boldsymbol{u}_k \circ \boldsymbol{v}_k \circ \boldsymbol{w}_k \circ \boldsymbol{t}_k \tag{1}$$

where $\{\boldsymbol{u}_k\}_{k=1}^K \in \mathbb{R}^n$, $\{\boldsymbol{v}_k\}_{k=1}^K \in \mathbb{R}^p$, $\{\boldsymbol{w}_k\}_{k=1}^K \in \mathbb{R}^q$, and $\{\boldsymbol{t}_k\}_{k=1}^K \in \mathbb{R}^r$ are factor vectors for trials, electrodes, frequencies and time, respectively. All of these vectors are standardized to have a ℓ_2-norm of one, and the weights are absorbed into the constants $\{d_k\}_{k=1}^K \in \mathbb{R}$. These factor vectors can be estimated using the optimization problem for CP-decomposition, which seeks a low-rank

solution. The optimization problem below, for example, is used to find a rank $K = 1$ solution [1]:

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{t}}{\text{minimize}} \quad & \|\boldsymbol{\mathcal{X}} - d \cdot \boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{w} \circ \boldsymbol{t}\|_F^2 \\ \text{subject to} \quad & \boldsymbol{u}^T\boldsymbol{u} \leq 1, \boldsymbol{v}^T\boldsymbol{v} \leq 1, \\ & \boldsymbol{w}^T\boldsymbol{w} \leq 1, \boldsymbol{t}^T\boldsymbol{t} \leq 1. \end{aligned} \tag{2}$$

The solution of the CP-decomposition may be found using the Alternating Least Squares (ALS) algorithm. However, this can take many iterations to converge, and it is not guaranteed to converge to a global minimum or stationary point [19]. Instead, we use the Tensor Power Algorithm (TPA) for ρ-PCA, proposed by Allen [1]. There are several benefits to using a TPA-based approach over ALS. Firstly, TPA always converges to a point that is stationary, while ALS may not converge to one. Secondly, TPA is a greedy approach in that the first factors computed will explain the most variance in the data, thus yielding component vectors that explain sequentially decreasing amounts of variance within the data. TPA enables one to computer fewer components overall, while potentially capturing the most variance. Lastly, the deflation step of TPA enforces pseudo-independence among each set of component vectors. To proceed with the TPA approach, we recast Eq. (2) using results from Kolda [18], and obtain Eq. (3):

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{t}}{\text{maximize}} \quad & \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t} \\ \text{subject to} \quad & \boldsymbol{u}^T\boldsymbol{u} \leq 1, \boldsymbol{v}^T\boldsymbol{v} \leq 1, \\ & \boldsymbol{w}^T\boldsymbol{w} \leq 1, \boldsymbol{t}^T\boldsymbol{t} \leq 1. \end{aligned} \tag{3}$$

It is simple to verify that Eqns. (2) and (3) are equivalent [1]. Notice that Eq. (3) is separable in the factors, and enables iterative block-wise optimization. That is, the objective function is maximized with respect to one of the variables $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ or $\boldsymbol{t}$, while holding all others fixed for each iteration.

To enhance the interpretability of results, we add regularization penalties and constraints to the optimization problem in Eq. (3). Our methodology is an extension of regularized tensor factorization by Allen and Maletić-Savatić [4], with penalties specific to each mode of the ECoG data. This tensor factorization decomposes an N-way tensor into N sets of component factors. General convex penalties and constraints can then be applied to each of these component vectors separately [4]. We propose to use the following types of regularization for each ECoG mode based on their known structures:

i) **Electrodes:** We propose to use an ℓ_1-norm penalty, denoted as $\|x\|_1$, to select the most relevant electrodes for specific tasks.

ii) **Frequency:** We propose to select relevant frequencies, and smooth them to yield more interpretable results. To this end, we employ the sparse and functional penalties from Allen and Weylandt [5]. The frequency mode is penalized by both an ℓ_1-norm penalty to select the most relevant frequencies, and a difference matrix $\boldsymbol{\Omega_w}$ to smooths over frequency values. Smoothing over frequencies removes noise observed between similar frequencies.

iii) **Time:** We propose a smoothing penalty over time to yield more interpretable time factors by removing noise from observations close in time. We penalize the time factors with a difference matrix $\boldsymbol{\Omega}_t$.

[1]We use ρ-PCA to find $K \geq 1$ sets of factor vectors for each modality.

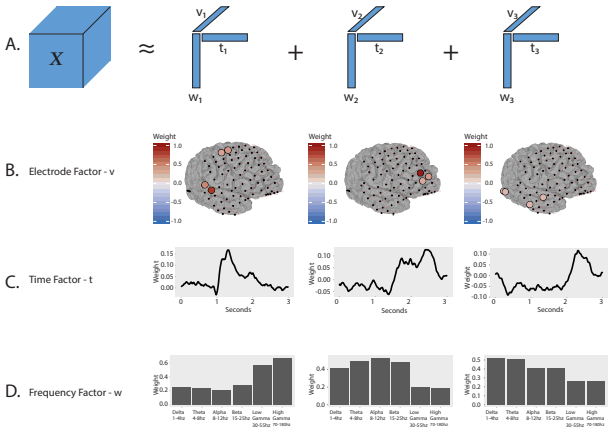

Figure 1: Clear & noisy audio classification for patient YAK using ρ-PLS + LDA. A.) ECoG tensor $\mathcal{X}$ is decomposed into three components. Each component has an electrode factor, a time factor and a frequency factor. B.) The electrode factors highlight which electrodes exhibit a differential response to the two stimuli classes. C.) The time factors show when there is a differential response to the two stimuli classes. D.) The frequency factors communicate which frequency bands have a differential response to the stimuli.

Eq. (3) is the result of applying these penalities to the rank-one tensor model, in order to yield the optimization problem that defines ρ-PCA:

$$\begin{aligned} \underset{\boldsymbol{u},\boldsymbol{v},\boldsymbol{w},\boldsymbol{t}}{\text{maximize}} \quad & \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t} \\ & -\lambda_v \|\boldsymbol{v}\|_1 - \lambda_w \|\boldsymbol{w}\|_1 \\ \text{subject to} \quad & \boldsymbol{u}^T\boldsymbol{u} \leq 1, \boldsymbol{v}^T\boldsymbol{v} \leq 1, \\ & \boldsymbol{w}^T(\boldsymbol{I} + \alpha_w\boldsymbol{\Omega_w})\boldsymbol{w} \leq 1, \\ & \boldsymbol{t}^T(\boldsymbol{I} + \alpha_t\boldsymbol{\Omega_t})\boldsymbol{t} \leq 1. \end{aligned} \tag{4}$$

The matrices $\boldsymbol{\Omega}_w$ and $\boldsymbol{\Omega}_t$ are 2nd-order difference matrices, and ultimately smooth over notable changes in signal over time and frequencies values. The constants $\lambda_v, \lambda_w \geq 0$ control the amount of sparsity applied to the electrode and frequency factors. Increasing these values will greatly shrink the factors, as well as set some to 0,

making them less significant. The constants $\alpha_w, \alpha_t \geq 0$ control the amount of smoothing over frequency and time. Increasing these values makes the elements of the factor vectors smoother and more homogeneous. Additionally, we define $\boldsymbol{S}_w = \boldsymbol{I}_q + \alpha_w \boldsymbol{\Omega}_w \succ 0$, L_w as the largest eigenvalue of $\boldsymbol{S}_w$, and $\boldsymbol{S}_t = \boldsymbol{I}_r + \alpha_t \boldsymbol{\Omega}_t \succ 0$. Notably, the ℓ_1-norm penalties in Eq. (4) enable us to perform automatic data-driven feature selection along the electrode and frequency modalities while estimating the factors in ρ-PCA. They interpretable results, as one can easily identify the most important electrodes and frequencies in an ECoG data set simply by finding the non-zero elements of the factors.

Figure 1 contains ρ-PCA factors for patient YAK, and shows the estimated tensor decomposition along three modalities. The sparsity penalty of the electrode mode allow us to identify 3-4 influential electrodes for each tensor component, as well as evaluate the corresponding time and frequency factors associated with those particular electrodes.

Alg. 1 is a computationally efficient algorithm for computing ρ-PCA, and it is an extension of TPA [1]. It utilizes a coordinate-wise update schema in which each mode is updated individually, while holding the other ones constant, in an iterative manner until convergence is reached. As with TPA in general, Alg. 1 is guaranteed to converge to a local maximum:

Theorem 1. *Each factor-wise block coordinate-wise update of ρ-PCA monotonically increases the objective that when iterated, converge to a local maximum of the optimization problem* (4)*.*

In addition to Theorem 1, there are two more significant advantages of using TPA for ρ-PCA. Firstly, the ℓ_2-norm and $\|\cdot\|_{\boldsymbol{\Omega}}$ are equal to one or zero to avoid degenerate solutions [2]. Secondly, the block coordinate solutions are simple and closed-form, making each update is inexpensive to compute [2].

Hyperparameter selection for ρ-PCA can be performed using a nested Bayesian Information Criterion schema [4, 22]. A full description of hyperparameter selection for ρ-PCA is presented in Alg. 3 in Appendix B.

B. Supervised Setting: ρ-PLS

ρ-PCA is easily be extended for supervised learning via Partial Least Squares (PLS) regression. Traditional PLS finds the direction of the covariates that explains the maximum variance in the response [15]. Recall the optimization problem for traditional PLS:

$$\begin{aligned} \underset{\boldsymbol{v},\boldsymbol{w}}{\text{maximize}} \quad & \text{Cov}\left(\boldsymbol{v}^T\boldsymbol{X}, \boldsymbol{w}^T\boldsymbol{Y}\right) := \boldsymbol{v}^T\underline{\boldsymbol{X}^T\boldsymbol{Y}}\boldsymbol{w} \\ \text{subject to} \quad & \|\boldsymbol{v}\|_2 = \|\boldsymbol{w}\|_2 = 1, \\ & \boldsymbol{v}^T\boldsymbol{v} = \boldsymbol{I}_p, \boldsymbol{w}^T\boldsymbol{w} = \boldsymbol{I}_q \end{aligned} \tag{10}$$

where $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, $\boldsymbol{Y} \in \mathbb{R}^{n\times q}$, $\boldsymbol{v} \in \mathbb{R}^p$, and $\boldsymbol{w} \in \mathbb{R}^q$.

Algorithm 1 ρ-PCA Algorithm via the Tensor Power Method

1) Initialization:
   a) Set $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}$.
   b) Pre-compute $\boldsymbol{S}_w$ and $\boldsymbol{S}_t$.
   c) Initialize $\left\{\left(\hat{d}_k, \hat{\boldsymbol{u}}_k, \hat{\boldsymbol{v}}_k, \hat{\boldsymbol{w}}_k, \hat{\boldsymbol{t}}_k\right)\right\}_{k=1}^K$.
2) For $k = 1, \ldots, K$:
   (a) Repeat until convergence:
      i. Update trial component estimate $\hat{\boldsymbol{u}}_k$,

$$\begin{aligned} \tilde{\boldsymbol{u}}_k &= \hat{\boldsymbol{\mathcal{X}}} \times_2 \hat{\boldsymbol{v}}_{k-1} \times_3 \hat{\boldsymbol{w}}_{k-1} \times_4 \hat{\boldsymbol{t}}_{k-1}, \\ \hat{\boldsymbol{u}}_k &= \begin{cases} \frac{\tilde{\boldsymbol{u}}_k}{\|\tilde{\boldsymbol{u}}_k\|_2}, & \|\tilde{\boldsymbol{u}}_k\|_2 > 0, \\ 0, & otherwise \end{cases} \end{aligned} \tag{5}$$

      ii. Update electrode component estimate $\hat{\boldsymbol{v}}_k$,

$$\begin{aligned} \tilde{\boldsymbol{v}}_k &= S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_{k-1} \times_3 \hat{\boldsymbol{w}}_{k-1} \times_4 \hat{\boldsymbol{t}}_{k-1}, \lambda_v), \\ \hat{\boldsymbol{v}}_k &= \begin{cases} \frac{\tilde{\boldsymbol{v}}_k}{\|\tilde{\boldsymbol{v}}_k\|_2}, & \|\tilde{\boldsymbol{v}}_k\|_2 > 0, \\ 0, & otherwise \end{cases} \end{aligned} \tag{6}$$

      iii. Update frequency component estimate $\hat{\boldsymbol{w}}_k$,

$$\begin{aligned} \tilde{\boldsymbol{w}}_k &= \text{prox}_{\frac{\lambda_w}{L_w}\|\cdot\|_1}(\hat{\boldsymbol{w}}_{k-1} + \\ &\quad L_w^{-1}\left(\hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_{k-1} \times_2 \hat{\boldsymbol{v}}_{k-1} \times_4 \hat{\boldsymbol{t}}_{k-1} - \right. \\ &\quad \left. \boldsymbol{S}_w\hat{\boldsymbol{w}}_{k-1}\right)) \\ \hat{\boldsymbol{w}}_k &= \begin{cases} \frac{\tilde{\boldsymbol{w}}_k}{\|\tilde{\boldsymbol{w}}_k\|_{\boldsymbol{S}_w}}, & \|\tilde{\boldsymbol{w}}_k\|_{\boldsymbol{S}_w} > 0, \\ 0, & otherwise \end{cases} \end{aligned} \tag{7}$$

      iv. Update time component estimate $\hat{\boldsymbol{t}}_k$,

$$\begin{aligned} \tilde{\boldsymbol{t}}_k &= \boldsymbol{S}_t^{-1}\left(\hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_{k-1} \times_2 \hat{\boldsymbol{v}}_{k-1} \times_3 \hat{\boldsymbol{w}}_{k-1}\right), \\ \hat{\boldsymbol{t}}_k &= \begin{cases} \frac{\tilde{\boldsymbol{t}}_k}{\|\tilde{\boldsymbol{t}}_k\|_{\boldsymbol{S}_t}}, & \|\tilde{\boldsymbol{t}}_k\|_{\boldsymbol{S}_t} > 0, \\ 0, & otherwise \end{cases} \end{aligned} \tag{8}$$

   (b) Deflate to encourage pseudo-independence between sets of K component vectors,

$$\begin{aligned} \hat{d}_k &= \hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_k \times_2 \hat{\boldsymbol{v}}_k \times_3 \hat{\boldsymbol{w}}_k \times_4 \hat{\boldsymbol{t}}_k, \\ \hat{\boldsymbol{\mathcal{X}}} &= \hat{\boldsymbol{\mathcal{X}}} - \hat{d}_k \cdot \hat{\boldsymbol{u}}_k \circ \hat{\boldsymbol{v}}_k \circ \hat{\boldsymbol{w}}_k \circ \hat{\boldsymbol{t}}_k \end{aligned} \tag{9}$$

3) Return factors $\left\{\left(\hat{d}_k, \hat{\boldsymbol{u}}_k, \hat{\boldsymbol{v}}_k, \hat{\boldsymbol{w}}_k, \hat{\boldsymbol{t}}_k\right)\right\}_{k=1}^K$.

The coefficient vectors have a 2-norm of one, and are constrained to be orthogonal. We can easily extend Eq. (10) for higher-order arrays, as well as include regularization, while dropping the orthogonality constraints typical in CP decompositions. In turn, the traditional covariate tensor $\boldsymbol{\mathcal{X}}$ is replaced by the covariance tensor $\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{y}$, where $\boldsymbol{y} \in \{0,1\}^n$, is the response. The optimization problem in Eq. (11) defines the ρ-PLS solution:

$$\begin{aligned} \underset{\boldsymbol{v},\boldsymbol{w},\boldsymbol{t}}{\text{maximize}} \quad & \left(\underline{\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{y}}\right) \times_1 \boldsymbol{v} \times_2 \boldsymbol{w} \times_3 \boldsymbol{t} \\ & -\lambda_v \|\boldsymbol{v}\|_1 - \lambda_w \|\boldsymbol{w}\|_1 \\ \text{subject to} \quad & \boldsymbol{w}^T(\boldsymbol{I}_q + \alpha_w\boldsymbol{\Omega}_{\boldsymbol{w}})\boldsymbol{w} \leq 1, \\ & \boldsymbol{v}^T\boldsymbol{v} \leq 1, \ \& \ \boldsymbol{t}^T(\boldsymbol{I}_r + \alpha_t\boldsymbol{\Omega}_{\boldsymbol{t}})\boldsymbol{t} \leq 1 \end{aligned} \tag{11}$$

Algorithm 2 ρ-PLS Algorithm: Extension of ρ-PCA for supervised dimension reduction and prediction.

1) Initialization:
   (a) Define the covariates $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times p \times q \times r}$, and response $\boldsymbol{y} \in \{0,1\}^n$
   (b) Center each column of $\boldsymbol{y}$, denoted as $\bar{\boldsymbol{y}}$.
2) Compute the covariance tensor,
$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} \times_1 \bar{\boldsymbol{y}} \in \mathbb{R}^{p \times q \times r}$$
3) Estimate the component vectors,
$$\left[\{\hat{\boldsymbol{m}}_k\}_{k=1}^K, \{\hat{\boldsymbol{v}}_k\}_{k=1}^K, \{\hat{\boldsymbol{w}}_k\}_{k=1}^K, \{\hat{\boldsymbol{t}}_k\}_{k=1}^K\right] = \rho\text{-PCA}\left(\boldsymbol{\mathcal{Z}}, K, \boldsymbol{\lambda}\right)$$
where $\boldsymbol{\lambda} = \{(\lambda_v, \lambda_w, \alpha_w, \alpha_t)_k\}_{k=1}^K$ are the regularization parameters.
4) Reduce the dimensionality of the covariates for each $k \in \{1, \ldots, K\}$,
$$\tilde{\boldsymbol{x}}_k = \boldsymbol{\mathcal{X}} \times_2 \hat{\boldsymbol{v}}_k \times_3 \hat{\boldsymbol{w}}_k \times_4 \hat{\boldsymbol{t}}_k \in \mathbb{R}^n$$
Concatenate the $\tilde{\boldsymbol{x}}_k$'s to form $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_K] \in \mathbb{R}^{n \times K}$
5) Return the reduced data $\tilde{\boldsymbol{X}}$ and component vectors $\left[\{\hat{\boldsymbol{m}}_k\}_{k=1}^K, \{\hat{\boldsymbol{v}}_k\}_{k=1}^K, \{\hat{\boldsymbol{w}}_k\}_{k=1}^K, \{\hat{\boldsymbol{t}}_k\}_{k=1}^K\right]$.
   (a) Build model to predict response $\boldsymbol{Y}$ from $\tilde{\boldsymbol{X}}$.
   (b) Visualize and interpret data, as described in Table I.

Alg. 2 illustrates the implementation of ρ-PLS. Overall, ρ-PLS reduces the ECoG tensor data into a reduced form, based on the class separation. Following this, the reduced data may be plugged into any supervised learning model. We use a nested K-fold cross-validation scheme to find optimal hyperparameters for ρ-PLS, explained in Alg. 4 in Appendix B.

III. Case Study: Neural-decoding with ECoG

We compare ρ-PCA and ρ-PLS to other popular techniques, on a real electrocorticography (ECoG) data set; we use data first analyzed by Ozker et al. [28]. Our analysis differs significantly, but the data collection and preprocessing procedures are the same; we describe these steps below.

A. Data and Experimental Design

We seek to make accurate predictions about the stimuli given a patient's ECoG measurements. Patients have subdural electrodes surgically implanted, and agree to participate in an experiment in which each they are shown a movie of a woman saying either the word "Rock" or "Rain". The conditions of the audio-visual speech stimuli are varied by either adding noise to the audio or visual, in all possible combinations. This results in a total of eight classes of stimuli movies. For each trial, a patient watches a single video and tries to identify the word stated.

Table I: Summary of visualizations that can potentially be created from the modes of ρ-PLS.

Factor	Formula	Dimension	Visualization
Node	$\hat{\boldsymbol{V}} := \{\hat{\boldsymbol{v}}_k\}_{k=1}^K$	$(p \times K)$	Sparsity plots display relevant nodes.
Frequency	$\hat{\boldsymbol{W}} := \{\hat{\boldsymbol{w}}_k\}_{k=1}^K$	$(w \times K)$	Bar charts summarize the frequencies.
Time	$\hat{\boldsymbol{T}} := \{\hat{\boldsymbol{t}}_k\}_{k=1}^K$	$(r \times K)$	A time series plot shows the shape of the average activity over time.
Observation	$\boldsymbol{\mathcal{X}} \times_2 \hat{\boldsymbol{v}} \times_3 \hat{\boldsymbol{w}} \times_4 \hat{\boldsymbol{t}}$	$(n \times 1)$	Scatterplots communicate how well the trials separate.
Nodes by Time	$\boldsymbol{\mathcal{Z}} \times_2 \hat{\boldsymbol{w}}$	$(p \times t)$	Heatmaps show node activity over time.
Frequency by Time	$\boldsymbol{\mathcal{Z}} \times_1 \hat{\boldsymbol{v}}$	$(q \times t)$	A time series plot shows the frequency intensity over time.
Node by Freq.	$\boldsymbol{\mathcal{Z}} \times_3 \hat{\boldsymbol{t}}$	$(p \times q)$	Bar charts show important node-frequency combinations.

1) Preprocessing Data: The data was processed using the FieldTrip Toolbox in MATLAB 8.5.0 [27, MATLAB-8.5.0] and RAVE [25]. A common average reference [24] was used to remove artifacts from the electrodes. The data was epoched into trials lasting three seconds. Line noise was removed at 60, 120 and 180 Hz. We transformed the data to time-frequency space using the multi-taper method with three Slepian tapers, a frequency window from 10 to 200 Hz, frequency steps of 2 Hz, time steps of 10 ms, temporal smoothing of 200 ms, and frequency smoothing of 10 Hz. After processing, the resulting data set measures signal power as a function of frequency and time at each electrode.

2) Formatting ECoG Tensors: We concatenate trials, and form an ECoG tensor for each patient. Each patient's brain is different and, as a result, it is ill-advised to perform classification across patients. A typical patient participates in around 150 trials and has about 100 electrodes, 100 frequencies and a duration of 3 seconds (measured as 301 discrete points) yielding a 4D tensor. These data sets are between 3-5 GB for each patient.

Table II: This table compares the average accuracy, average audio run time, and standard deviations for ten trials, of various classification methods for patient YAH.

	Model	Word	Visual	Audio	Run Time (sec)
Unsupervised Methods	matrix PCA(X) + LDA	0.55 (0.09)	0.91 (0.08)	0.67 (0.08)	159.63 (31.27)
	matrix ICA(X) + LDA	0.55 (0.10)	0.73 (0.15)	0.60 (0.11)	173.76 (10.70)
	CP-ALS(X) + LDA	0.60 (0.07)	0.85 (0.07)	0.65 (0.09)	143.56 (36.04)
	Tucker-ALS(X) + LDA	0.47 (0.10)	0.62 (0.15)	0.50 (0.16)	78.82 (17.50)
	ρ-PCA(X) + LDA	0.59 (0.12)	0.88 (0.09)	0.66 (0.11)	1168.54 (424.44)
Supervised Methods	matrix SVM(X,y)	0.71 (0.10)	0.94 (0.05)	0.90 (0.07)	175.31 (42.75)
	matrix PLS (X) + LDA	0.65 (0.12)	0.91 (0.06)	0.66 (0.11)	14.90 (1.59)
	CP-ALS-PLS (X) + LDA	0.67 (0.08)	0.91 (0.05)	0.85 (0.10)	45.61 (3.21)
	Tucker-ALS-PLS(X) + LDA	0.50 (0.10)	0.89 (0.06)	0.76 (0.06)	46.21 (5.10)
	N-PLS + LDA	0.57 (0.09)	0.93 (0.05)	0.84 (0.04)	2254.42 (292.17)
	HOPLS + LDA	0.48 (0.07)	0.91 (0.06)	0.62 (0.10)	709.08 (51.91)
	ρ-PLS + LDA	0.65 (0.06)	0.89 (0.05)	0.89 (0.08)	48.21 (67.86)

Table III: This table compares the average accuracy and standard deviation of ρ-PLS + LDA, across each patient.

Subject	Word	Visual	Audio
YAK	0.61 (0.08)	0.96 (0.06)	0.58 (0.14)
YAH	0.65 (0.06)	0.89 (0.05)	0.89 (0.08)
YBA	0.59 (0.17)	0.99 (0.02)	0.49 (0.08)

B. Pipeline for Analysis

Our analysis consists of several steps that process, analyze and interpret the data:

1) Center and scale the data by the baseline mean and standard deviation, respectively; these are calculated using the measurements from the first half second of the trial, before any stimuli are presented.
2) Select regularization parameters in a component-wise fashion. This is done in a data-driven manner, following either of the procedures Alg. 3 or Alg. 4 in Appendix B.
3) Run either ρ-PCA or ρ-PLS for supervised or unsupervised learning.
4) Reduce the dimensionality of the data for decoding, and use the resulting components to visualize the results. (See Table I.)
5) Components from the ρ-PCA and ρ-PLS methods also be plugged into any general statistical learning model as covariates.

C. Comparative Study of Multi-Sensory Speech Decoding

We perform a comparative empirical study to demonstrate that ρ-PCA and ρ-PLS produce factors that can be used to accurately classify aspects of the stimuli. Our goal is to predict the stimuli conditions for trials, given a patient's observed ECoG data. We pose this problem as a set of binary classification problems for clean & noisy visuals, clean & noisy audio, and the words "rock" & "rain". While ρ-PCA and ρ-PLS can be used for a multi-class classification problem, we choose to simplify the problems into binary cases because this is more interesting in the context of neuroscience.

We compare ρ-PCA and ρ-PLS to several different methods that are common in the neuroscience field for analysis. This includes several unsupervised methods such as principal component analysis (matrix PCA) [32], independent component analysis (matrix ICA) [8], Alternating Least Squares for the Candecomp-Parafac decomposition [6] (CP-ALS), and Alternating Least Squares for the Tucker decomposition (Tucker-ALS) [36, 37]. We also compare our method to supervised methods, including support vector machines (matrix SVM), ordinary partial least squares (matrix PLS) [20], CP-ALS with partial least squares (CP-ALS-PLS), Tucker-ALS with partial least squares (Tucker-ALS-PLS), multiway partial least squares [7] (N-PLS), and Higher Order Partial Least Squares [34] (HOPLS); for all of the supervised methods, we treat the stimuli conditions as known. To fairly compare each dimension reduction approach, we combine each method with LDA, understanding that other approaches (e.g. SVM) might perform better. Also, to be fair to all methods for computational timing purposes, we first select oracle hyperparameters for the ρ-PCA, ρ-PLS, and SVM approaches. To do this, we assign 80% of trials to a training set, while the remaining 20% of trials are held out to be used as a test set. The selected hyperparameters are then fixed for the rest of the simulation study.

To compare the different methods of classification, we perform 10 replications of model fitting. For each replication, we randomly assign 90% of the trials to the training set, and 10% to the testing set. We train each of the different models using the training set, then estimate the classification accuracy of the methods using the test set. The results are our study are shown in Table II, which contains the average accuracy for each method, average run time (seconds) for the audio classification problem, and the standard deviations in parentheses. The accuracy for Word classification is noticeably poor. Classifying words is challenging with ECoG data since specific parts of the brain process different phonemes. It is possible there may not be electrodes on the exact spot in the brain that processes rock vs. rain, and hence the difficulty.

First, we compare the unsupervised methods in Table II. In general, prediction accuracy for Word and Audio are poor, yet more promising for Visual. In terms for Visual, the methods with the strongest accuracy are matrix PCA, CP-ALS, and ρ-PCA. The first two of these methods perform similarly, while ρ-PCA is notably slower. We hypothesize that this may be in part due to the fact that the algorithms for both matrix PCA and CP-ALS have been optimized in MATLAB with regards to the singular value decomposition and tensor operations, while our implementation of ρ-PCA has not been. Further, because ρ-PCA induces sparsity and smoothness, it requires more factors to achieve the same level of classification accuracy (a result well-known in the sparse PCA literature [4]), thus further leading to longer run times. Lastly, while ρ-PCA has a slower runtime, we gain interpretability as a trade-off; CP-ALS and ρ-PCA can provide the interpretations from Table I, and only ρ-PCA yields directly interpretable factors. It is worthy to note that ρ-PCA is much slower than ρ-PLS in Table II. This is largely due to the dimension reduction while calculating the covariace $\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{y}$, before running the ρ-PLS method.

Next, we compare the supervised methods in Table II, which produce promising prediction accuracy across Visual and Audio. Several methods perform strongly in one or two of the categories, yet do poorly in the other. Methods that appear to have reasonable prediction accuracy across the Visual and Audio tasks are the matrix SVM, CP-ALS-PLS, and ρ-PLS. Matrix SVM has the highest prediction accuracy across all categories. However, unlike the other front-runners, it doesn't offer interpretable solutions as it flattens the tensor. Also, Matrix SVM is much slower than CP-ALS-PLS, and ρ-PLS as well. On the other hand, CP-ALS-PLS and ρ-PLS have similar prediction accuracy and run time. However, our ρ-PLS method selects and smooths features so that each factor is neuroscientifically interpretable, unlike the CP-ALS-PLS approach; we demonstrate this interpretability aspect of our approach in the following section.

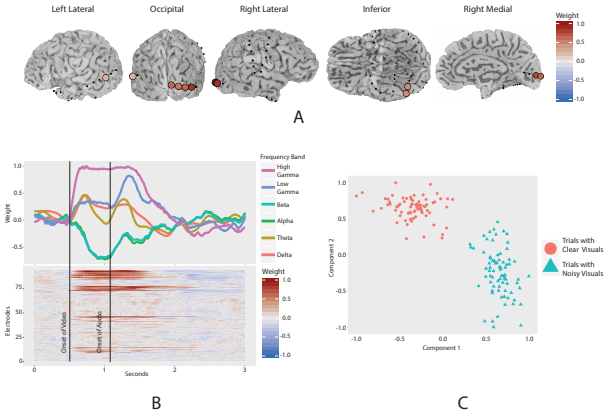

Figure 2: Visualizations of the ρ-PLS component for patient YBA for clear & noisy visuals classification. A.) Locations of the influential electrodes identified by ρ-PLS, and all are in the occipital region. B.) Projections of ρ-PLS factors communicate the behavior of the frequency bands and electrodes over time. Large values in the plots correspond to large differences between signal power classes. C.) Projection of the ρ-PLS components onto the ECoG tensor. The trials separate allowing for accurate classification.

D. Case Study: Patient YBA

We now show an in-depth study on patient YBA to demonstrate how ρ-PLS yields neuroscientifically interpretable results for brain decoding. For ease of understanding, we group the full set of frequency bands observed in the original data together into pre-defined neuroscientific bands of interest [28]. Additionally, we limit our attention to looking at the sparsity in the electrodes and smoothing over time with respect to the frequency bands.

We first analyze the ρ-PLS factors on the noisy & clean visual classification problem to demonstrate how ρ-PLS can identify important areas of interest in large ECoG tensors. Figure 2 displays interpretable visualizations of our results.

In Figure 2(A), the sparse electrode factor reveals 5 electrodes in the brain which exhibit large differences in activity behaviors between the clear & noisy visual trials. All of these electrodes are in the occipital region, a part of the visual cortex, which gives us confidence of the validity of our feature selection approach [28, 29]. In Figure 2(B), we visualize the corresponding components in the time and frequency modes in order to observe how the behaviors of the electrodes identified in Figure 2(A) change over time with respect to frequency bands. The

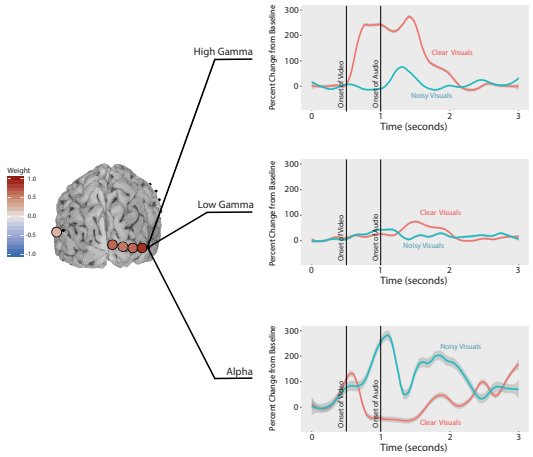

Figure 3: Average percent change from baseline for patient YBA at the High Gamma, Low Gamma and Alpha frequency bands for the clear & noisy visual classification problem. A locally weighted regression is fit to scaled trials. The grey areas around the estimate are the 95% confidence intervals. ρ-PLS identifies frequency bands, such as High Gamm and Alpha, that have a large difference in percent change from baseline.

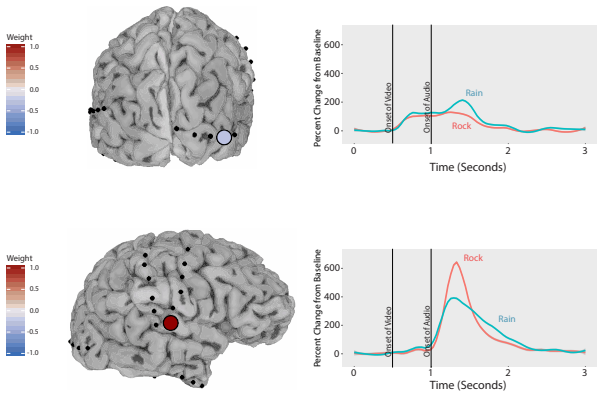

Figure 4: Average percent change from baseline for patient YBA at the High Gamma frequency band for the "rock" & "rain" problem. The magnitude of the ρ-PLS weight reflects the magnitude of the difference in average signal power between classes. Electrode 77 (top) has a small but noticeable difference in percent change from baseline, while Electrode 34 (bottom) has a large difference in percent change from baseline. The sign of the weight indicates which class has a stronger positive response.

largest changes in response are observed immediately after the onset of the video that occurs at 0.5 seconds; this is most apparent in the High Gamma, Beta, and Alpha frequency bands. Figure 2(C) shows how the trials separate allowing for accurate classification. It show the first two projects of the ρ-PLS components onto the ECoG tensor: $\boldsymbol{\mathcal{X}} \times_2 \hat{\boldsymbol{v}}_1 \times_3 \hat{\boldsymbol{w}}_1 \times_4 \hat{\boldsymbol{t}}_1$ and $\boldsymbol{\mathcal{X}} \times_2 \hat{\boldsymbol{v}}_2 \times_3 \hat{\boldsymbol{w}}_2 \times_4 \hat{\boldsymbol{t}}_2$.

In Figure 3, we compare the spectrograms for the High Gamma, Low Gamma, and Alpha frequency bands associated with electrode 77, one of the electrodes identified by ρ-PLS as influential in the noisy & clean visual classification problem. The activity in the High Gamma and Alpha frequency bands vary greatly. In particular, the activity in the High Gamma frequency band increases greatly for clear visuals, while the activity in the Alpha frequency decreases significantly. These are common observations [16, 31] that ρ-PLS is able to identify and highlight, also shown in Figure 2(B).

The last example investigates how the brain responds differently to the words "rock" and "rain" with audio-visual speech stimuli. Figure 4 depicts the ρ-PLS electrode components for the "rock" &"rain" classification problem, along with their corresponding spectrograms in the High Gamma frequency band over time. The electrode component has two non-zero weights, one in the posterior superior temporal gyrus in the audio cortex (bottom figures), and one in the occipital region of the visual cortex (top figures). There is a substantial difference in the High Gamma frequency band for the electrode in the posterior superior temporal gyrus; which supports previous work that connects the audio and visual cortex to speech perception [28, 31, 33].

This exploratory analysis highlights existing findings, but also allows scientists to quickly investigate new findings. The method ρ-PCA is able to identify influential electrodes for predicting stimulus conditions, along with corresponding frequency bands, and our findings match what has been shown in previous work. Thus, ρ-PLS offers a fast data driven method for sifting large data sets to identify regions, time points, and frequencies that warrant further study.

IV. Discussion

Brain decoding with ECoG data offers a unique opportunity to study the cortical processes associated with speech perception. However, ECoG data sets are large and complex, making them exceptionally difficult to analyze, especially for neuroscientists. In this paper, we introduce ρ-PCA and ρ-PLS, tensor decomposition approaches with sparse penalties and smoothing constraints that lead to directly interpretable tensor factors tailored to attributes of ECoG data. ρ-PCA is the first method we know of that considers the entirety of an ECoG data set and selects relevant features in the time points, frequencies and locations in an automated data-driven manner.

We evaluated the classification accuracy and computational performance of both ρ-PCA and ρ-PLS and compared them to methods commonly used in neuroimaging, namely other tensor decomposition methods and classical matrix-based methods. During our analysis, we found that the performance of matrix methods were very dependent on implementation and some were not computationally efficient enough for us to investigate. On the other hand, the performance of ρ-PLS was on par with or better than the other tensor decomposition approaches, while also providing more interpretable solutions and visualizations.

While we have demonstrated the effectiveness of ρ-PCA and ρ-PLS specifically on ECoG data, our work can potentially be used in other neuroimaging modalities that have spatio-temporal characteristics, such as EEGs, fMRI, among others. Although dimension reduction approaches have been an extremely important tool for analysis of neuroimaging data, we find that methods developed specifically for tensor data can be much more effective when the data has a natural representation as a tensor. Further, using constraints and regularization tailored to the attributes of the specific data set, yield more interpretable factors. Overall, we have developed a computationally efficient and neuroscientifically interpretable method for higher-order dimension reduction of ECoG data, with many potential uses beyond what we considered in this paper.

Supplementary Material

The appendices of this paper can be found on ????. The code and data are available at https://github.com/DataSlingers/rho-PCA.

Acknowledgements

KG was partially supported by NSF award 1547433. AC is supported by NSF NeuroNex-1707400. JM and MB are supported by NIH grant R01NS065395. GA is supported by NIH 1R01GM140468, NSF DMS-1554821, and NSF NeuroNex-1707400.

References

[1] Allen, G. (2012a). Sparse higher-order principal components analysis. In *Artificial Intelligence and Statistics*, pages 27–36.

[2] Allen, G. I. (2012b). Regularized tensor factorizations and higher-order principal components analysis. *arXiv preprint arXiv:1202.2476*.

[3] Allen, G. I. (2013). Multi-way functional principal components analysis. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 220–223. IEEE.

[4] Allen, G. I. and Maletić-Savatić, M. (2011). Sparse non-negative generalized pca with applications to metabolomics. *Bioinformatics*, 27(21):3029–3035.

[5] Allen, G. I. and Weylandt, M. (2019). Sparse and functional principal components analysis. In *2019 IEEE Data Science Workshop (DSW)*, pages 11–16.

[6] Bader, B. W., Kolda, T. G., et al. (2019). Matlab tensor toolbox version 3.1. Available online.

[7] Bro, R. (1996). Multiway calibration. multilinear pls. *Journal of chemometrics*, 10(1):47–61.

[8] Chao, Z. C. and Fujii, N. (2013). Mining spatio-spectro-temporal cortical dynamics: a guideline for offline and online electrocorticographic analyses. In *Methods in Neuroethological Research*, pages 39–55. Springer.

[9] Cichocki, A. (2013). Tensor decompositions: a new concept in brain data analysis? *arXiv preprint arXiv:1305.0395*.

[10] Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., and Phan, H. A. (2015). Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163.

[11] Delorme, A. and Makeig, S. (2004). Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of neuroscience methods*, 134(1):9–21.

[12] Eliseyev, A. and Aksenova, T. (2013). Recursive n-way partial least squares for brain-computer interface. *PloS one*, 8(7):e69962.

[13] Eliseyev, A., Moro, C., Faber, J., Wyss, A., Torres, N., Mestais, C., Benabid, A. L., and Aksenova, T. (2012). L1-penalized n-way pls for subset of electrodes selection in bci experiments. *Journal of neural engineering*, 9(4):045010.

[14] Gaona, C. M., Sharma, M., Freudenburg, Z. V., Breshears, J. D., Bundy, D. T., Roland, J., Barbour, D. L., Schalk, G., and Leuthardt, E. C. (2011). Nonuniform high-gamma (60–500 hz) power changes dissociate cognitive task and anatomy in human cortex. *The Journal of Neuroscience*, 31(6):2091–2100.

[15] Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17.

[16] Hipp, J. F., Engel, A. K., and Siegel, M. (2011). Oscillatory synchronization in large-scale cortical networks predicts perception. *Neuron*, 69(2):387–396.

[17] Kellis, S., Miller, K., Thomson, K., Brown, R., House, P., and Greger, B. (2010). Decoding spoken words using local field potentials recorded from the cortical surface. *Journal of neural engineering*, 7(5):056007.

[18] Kolda, T. G. (2006). Multilinear operators for higher-order decompositions. Technical report, Sandia

National Laboratories.

[19] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500.

[20] Krishnan, A., Williams, L. J., McIntosh, A. R., and Abdi, H. (2011). Partial least squares (pls) methods for neuroimaging: a tutorial and review. *Neuroimage*, 56(2):455–475.

[21] Kubanek, J., Miller, K., Ojemann, J., Wolpaw, J., and Schalk, G. (2009). Decoding flexion of individual fingers using electrocorticographic signals in humans. *Journal of neural engineering*, 6(6):066001.

[22] Lee, M., Shen, H., Huang, J. Z., and Marron, J. (2010). Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095.

[23] Liang, N. and Bougrain, L. (2012). Decoding finger flexion from band-specific ecog signals in humans. *Frontiers in neuroscience*, 6:91.

[24] Ludwig, K. A., Miriani, R. M., Langhals, N. B., Joseph, M. D., Anderson, D. J., and Kipke, D. R. (2009). Using a common average reference to improve cortical neuron recordings from microelectrode arrays. *Journal of neurophysiology*, 101(3):1679–1689.

[25] Magnotti, J. F., Wang, Z., and Beauchamp, M. S. (2020). Rave: comprehensive open-source software for reproducible analysis and visualization of intracranial eeg data. *NeuroImage*, 223:117341.

[MATLAB-8.5.0] MATLAB-8.5.0. The mathworks. *Inc., Natick, Massachusetts, United States*.

[27] Oostenveld, R., Fries, P., Maris, E., and Schoffelen, J.-M. (2011). Fieldtrip: open source software for advanced analysis of meg, eeg, and invasive electrophysiological data. *Computational intelligence and neuroscience*, 2011:1.

[28] Ozker, M., Schepers, I. M., Magnotti, J. F., Yoshor, D., and Beauchamp, M. S. (2017). A double dissociation between anterior and posterior superior temporal gyrus for processing audiovisual speech demonstrated by electrocorticography. *Journal of Cognitive Neuroscience*.

[29] Ozker, M., Yoshor, D., and Beauchamp, M. S. (2018). Frontal cortex selects representations of the talker's mouth to aid in speech perception. *eLife*, 7:e30387.

[30] Pei, X., Barbour, D. L., Leuthardt, E. C., and Schalk, G. (2011). Decoding vowels and consonants in spoken and imagined words using electrocorticographic signals in humans. *Journal of neural engineering*, 8(4):046028.

[31] Schepers, I. M., Yoshor, D., and Beauchamp, M. S. (2014). Electrocorticography reveals enhanced visual cortex responses to visual speech. *Cerebral Cortex*, 25(11):4103–4110.

[32] Shimada, S., Kunii, N., Kawai, K., Matsuo, T., Ishishita, Y., Ibayashi, K., and Saito, N. (2017). Impact of volume-conducted potential in interpretation of cortico-cortical evoked potential: Detailed analysis of high-resolution electrocorticography using two mathematical approaches. *Clinical Neurophysiology*, 128(4):549–557.

[33] Stevenson, R. A. and James, T. W. (2009). Audiovisual integration in human superior temporal sulcus: Inverse effectiveness and the neural processing of speech and object recognition. *Neuroimage*, 44(3):1210–1223.

[34] Zhao, Q., Caiafa, C. F., Mandic, D. P., Chao, Z. C., Nagasaka, Y., Fujii, N., Zhang, L., and Cichocki, A. (2013a). Higher order partial least squares (hopls): a generalized multilinear regression method. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1660–1673.

[35] Zhao, Q., Zhang, L., and Cichocki, A. (2014). Multilinear and nonlinear generalizations of partial least squares: an overview of recent advances. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(2):104–115.

[36] Zhao, Q., Zhou, G., Adali, T., Zhang, L., and Cichocki, A. (2013b). Kernel-based tensor partial least squares for reconstruction of limb movements. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3577–3581. IEEE.

[37] Zhao, Q., Zhou, G., Adali, T., Zhang, L., and Cichocki, A. (2013c). Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data. *IEEE Signal Processing Magazine*, 30(4):137–148.

APPENDIX

A. Convergence of Algorithm 1

This section provides a proof for Theorem 1, which guarantees the convergence of Algorithm 1 to a local solution. Proposition 1 explains the regulatory conditions for convergence.

Proposition 1. *Suppose that $(\boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*, \boldsymbol{t}^*)$ are the global maximum of Eq.* (4)*, holding all other factors fixed. It follows that $(\boldsymbol{u}^*, \boldsymbol{v}^*, \boldsymbol{w}^*, \boldsymbol{t}^*)$ must satisfy the Karush-Kuhn-Tucker (KKT) conditions, listed below. The variables γ_u, γ_v, γ_w, and γ_t are the dual variables associated with Eq.* (4)*. The notation $\tilde{\nabla}_x f(\boldsymbol{x})$ represents a general sub-gradient of f with respect to x.*

$$
\begin{aligned}
&(\boldsymbol{\mathcal{X}} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t}) \\
&-2\gamma_u \boldsymbol{u}^* = 0 \quad (\boldsymbol{u}\text{-stationarity}) \\
&\gamma_u \left(\|\boldsymbol{u}^*\|_2^2 - 1 \right) = 0 \quad (\boldsymbol{u}\text{-complementary slackness}) \\
&\gamma_u \geq 0 \quad (\boldsymbol{u}\text{-dual feasibility}) \\
&\|\boldsymbol{u}^*\|_2^2 \leq 1 \quad (\boldsymbol{u}\text{-primal feasibility}) \\
&(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t}) \\
&-\lambda_v \tilde{\nabla}_v \left(\|\boldsymbol{v}^*\|_1 \right) - 2\gamma_v \boldsymbol{v}^* = 0 \quad (\boldsymbol{v}\text{-stationarity}) \\
&\gamma_v \left(\|\boldsymbol{v}^*\|_2^2 - 1 \right) = 0 \quad (\boldsymbol{v}\text{-complementary slackness}) \\
&\gamma_v \geq 0 \quad (\boldsymbol{v}\text{-dual feasibility}) \\
&\|\boldsymbol{v}^*\|_2^2 \leq 1 \quad (\boldsymbol{v}\text{-primal feasibility}) \\
&(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_4 \boldsymbol{t}) \\
&-\lambda_w \tilde{\nabla}_w \left(\|\boldsymbol{w}^*\|_1 \right) - 2\gamma_w \boldsymbol{S}_w = 0 \quad (\boldsymbol{w}\text{-stationarity}) \\
&\gamma_w \left(\|\boldsymbol{w}^*\|_{\boldsymbol{S}_w}^2 - 1 \right) = 0 \quad (\boldsymbol{w}\text{-complementary slackness}) \\
&\gamma_w \geq 0 \quad (\boldsymbol{w}\text{-dual feasibility}) \\
&\|\boldsymbol{w}^*\|_{\boldsymbol{S}_w}^2 \leq 1 \quad (\boldsymbol{w}\text{-primal feasibility}) \\
&(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w}) \\
&-2\gamma_t \boldsymbol{S}_t \boldsymbol{t}^* = 0 \quad (\boldsymbol{t}\text{-stationarity}) \\
&\gamma_t \left(\|\boldsymbol{t}^*\|_{\boldsymbol{S}_t}^2 - 1 \right) = 0 \quad (\boldsymbol{t}\text{-complementary slackness}) \\
&\gamma_t \geq 0 \quad (\boldsymbol{t}\text{-dual feasibility}) \\
&\|\boldsymbol{t}^*\|_{\boldsymbol{S}_t}^2 \leq 1 \quad (\boldsymbol{t}\text{-primal feasibility})
\end{aligned}
$$

Proof of Theorem 1: We follow the approach used by Allen [2], and rely on results from Tseng [10]. Allen and Maletić-Savatić [4] states that we can solve for one factor at a time, holding the others fixed at feasible values. In order for this to hold, we show that each factor is a unique global solution of their respective regression problems. Additionally, we demonstrate that each factor is a local solution to Eq. (4), holding all others fixed. Lastly, we will derive the updates for each factor, shown in Equations (5)-(8) of Alg. 1.

Notice that $\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t}$, is continuously differentiable with respect to all factors. Each of the optimization problems for $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{t}$ are strictly concave. Alg. 1 requires that $\boldsymbol{\Omega} \succeq 0$ so that the matrix $I + \alpha_t \boldsymbol{\Omega}_t$ is positive definite by construction. Computing the Hessian of the Lagrangian with respect to $\boldsymbol{t}$ yields $-(\boldsymbol{I} + \alpha_t \boldsymbol{\Omega}_t)$, a negative definite matrix making the problem strictly concave. This implies that the solution $\boldsymbol{t}^*$ is unique. If we consider the Lagrangian for the other vectors, we can show that the respective optimization problems are strictly concave. The Lagrangian of the optimization problem with respect to $\boldsymbol{v}$, where we take the square root of the constraint, is,

$$
\begin{aligned}
\underset{\boldsymbol{v}}{\text{maximize}} \quad & \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t} \\
& - \lambda \|\boldsymbol{v}\|_1 - \gamma(\|\boldsymbol{v}\|_2^2 - 1)
\end{aligned}
$$

Using the Triangle Inequality and the definition of concavity, it follows that,

$$
\begin{aligned}
&\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 (c\boldsymbol{x} + (1-c)\boldsymbol{y}) \times_3 \boldsymbol{w} \times_4 \boldsymbol{t} \\
&-\lambda_v \|c\boldsymbol{x} + (1-c)\boldsymbol{y}\|_1 - (\|c\boldsymbol{x} + (1-c)\boldsymbol{y}\|_2^2 - 1) \\
&> c\,(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{x} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t}) \\
&-c\lambda_v \|\boldsymbol{x}\|_1 - c(\|\boldsymbol{x}\|_2^2 - 1) \\
&+(1-c)\,(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{y} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t}) \\
&-(1-c)\lambda_v \|\boldsymbol{y}\|_1 - (1-c)(\|\boldsymbol{y}\|_2^2 - 1),
\end{aligned}
$$

where the problem is strictly concave, which makes $\boldsymbol{v}^*$ unique. Similar arguments hold for $\boldsymbol{u}$ [1, 2] and $\boldsymbol{w}$ [5]. Together these properties imply that our problem converges to a stationary point of the joint optimization problem. Next, we will derive the block coordinate updates for Alg. 1.

(i.) **Derive the coordinate block solution for the trial mode.** First, we solve $\boldsymbol{u}$-stationarity to find the optimal $\boldsymbol{u}^*$,

$$
\boldsymbol{u}^* = \frac{1}{2\gamma_u} (\boldsymbol{\mathcal{X}} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t})
$$

Define $\tilde{\boldsymbol{u}} = \boldsymbol{\mathcal{X}} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t}$. We will use this result with $\boldsymbol{u}$-complementary slackness to find γ_u^*,

$$
\gamma_u \left(\|\boldsymbol{u}^*\|_2^2 - 1 \right) = \gamma_u \left(\frac{1}{4\gamma_u^2} \tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}} - 1 \right) = 0
$$

$$
\gamma_u^{*2} = \frac{1}{4} \tilde{\boldsymbol{u}}^T \tilde{\boldsymbol{u}} = \frac{1}{4} \|\tilde{\boldsymbol{u}}\|_2^2
$$

Plugging γ_u^* into $\boldsymbol{u}^*$ yields the coordinate update Eq. (5). By Proposition 1 and Proposition 1 from Allen [1], $\boldsymbol{u}^*$ is a global optimum of its sub-problem, and local optimum of Eq. (4) holding all other factors fixed.

(ii.) **Derive the coordinate block solution for the electrode mode.** Our approach to finding the

optimal $(\boldsymbol{v}^*, \gamma_v^*)$ follows those by Allen [1] and Witten et al. [11]. First, we will solve $\boldsymbol{v}$-stationarity to find the optimal $\boldsymbol{v}^*$. Define $\tilde{\boldsymbol{v}} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_3 \boldsymbol{w} \times_4 \boldsymbol{t}$. We can determine the value of each v_i^* element of $\boldsymbol{v}^*$ as follows,

$$2\gamma_v v_i^* = \begin{cases} (\tilde{\boldsymbol{v}})_i - \lambda_v & v_i > 0, \\ (\tilde{\boldsymbol{v}})_i + \lambda_v & v_i < 0, \\ (\tilde{\boldsymbol{v}})_i - [-\lambda_v, \lambda_v], & v_i = 0 \end{cases}$$

We can alternatively express $\boldsymbol{v}^*$ as a function of the soft-thresholding function,

$$\boldsymbol{v}^* = \frac{1}{2\gamma_v} S(\tilde{\boldsymbol{v}}, \lambda_v)$$

Now, we can use $\boldsymbol{v}$-complementary slackness to find γ_v^*,

$$\begin{aligned} \gamma_v \left(\|\boldsymbol{v}^*\|_2^2 - 1 \right) &= \gamma_v \left(\frac{1}{4\gamma_v^2} S(\tilde{\boldsymbol{v}}, \lambda_v)^T S(\tilde{\boldsymbol{v}}, \lambda_v) - 1 \right) \\ &= 0 \end{aligned}$$

$$\gamma_v^{*2} = \frac{1}{4} S(\tilde{\boldsymbol{v}}, \lambda_v)^T S(\tilde{\boldsymbol{v}}, \lambda_v) = \frac{1}{4} \|S(\tilde{\boldsymbol{v}}, \lambda_v)\|_2^2$$

Plugging γ_v^* into $\boldsymbol{v}^*$ results in the coordinate update Eq. (6). By Proposition 1 and Theorem 1 of Allen [1], $\boldsymbol{v}^*$ is a global optimum of its sub-problem, and local optimum of Eq. (4), holding all other factors fixed.

(iii.) **Derive the coordinate block solution for the frequency mode.** Our approach to finding the optimal $(\boldsymbol{w}^*, \gamma_w^*)$ follows those from Allen and Weylandt [5] and Allen et al. [3]. Notice that,

$$\underset{\boldsymbol{w} \in \mathbb{R}^q}{\operatorname{argmin}} \underbrace{\frac{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}}{2} - \boldsymbol{w}^T \left(\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_4 \boldsymbol{t} \right)}_{\text{smooth}} + \underbrace{\lambda_w \|\boldsymbol{w}\|_1 + i_{\bar{B}^q_{\boldsymbol{S}_w}}(\boldsymbol{w})}_{\text{non-differentiable}} \tag{12}$$

We define $\bar{\mathbb{B}}^q_{\boldsymbol{S}_w}$ as the unit ellipse of the $\boldsymbol{S}_w$-norm, $\bar{\mathbb{B}}^q_{\boldsymbol{S}_w} = \{\boldsymbol{w} \in \mathbb{R}^q | \boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w} \leq 1\}$. The term $i_{\bar{\mathbb{B}}^q_{\boldsymbol{S}_w}}$ is the so called infinite indicator of the feasible set of $\boldsymbol{w}$; defined by

$$i_{\mathcal{X}}(x) = \begin{cases} 0, & x \text{ is an element of } \mathcal{X}, \\ +\infty, & \text{otherwise} \end{cases}$$

Due to the fact that Eq. (12) is a separable sum of convex functions, we can use the proximal operator to find the coordinate update for $\boldsymbol{w}^*$ [8]. The proximal operator is defined as,

$$\operatorname{prox}_{\lambda f}(\boldsymbol{y}) = \arg\min_{\boldsymbol{x}} \left(f(\boldsymbol{x}) + \frac{1}{2\lambda} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 \right),$$

where $\lambda > 0$ is a constant and f is a function. In the context of Eq. (12), we take $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ and find that,

$$\begin{aligned} \operatorname{prox}_{f + i_{\bar{\mathbb{B}}^q_{\boldsymbol{S}_w}}}(\boldsymbol{x}) &= \arg\min_{\boldsymbol{w}} f(\boldsymbol{x}) + \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{w}\|_2^2 \\ &\quad + i_{\bar{\mathbb{B}}^q_{\boldsymbol{S}_w}}(\boldsymbol{w}) \\ &= \arg\min_{\boldsymbol{w} \in i_{\bar{\mathbb{B}}^q_{\boldsymbol{S}_w}}} f(\boldsymbol{x}) + \frac{1}{2} \|\boldsymbol{x} - \boldsymbol{w}\|_2^2 \\ &= \operatorname{proj}_{i_{\bar{\mathbb{B}}^q_{\boldsymbol{S}_w}}} \left(\operatorname{prox}_f(\boldsymbol{x}) \right) \end{aligned}$$

where $\operatorname{proj}_{\mathcal{X}}(\boldsymbol{x})$ denotes the projection of $\boldsymbol{x}$ onto $\mathcal{X}$. This result immediately follows from Allen and Weylandt [5] and Yu [12]; Eq. (7) is the coordinate update for the frequency mode. By Proposition 1 and Lemma 3 of Allen and Weylandt [5], $\boldsymbol{w}^*$ is a global optimum of its sub-problem, and local optimum of Eq. (4), holding all other factors fixed.

(iv.) **Derive the coordinate block solution for the time mode.** Our approach to finding the optimal $(\boldsymbol{t}^*, \gamma_t^*)$ that from Allen [2]. Define $\tilde{\boldsymbol{t}} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{u} \times_2 \boldsymbol{v} \times_3 \boldsymbol{w}$. We will solve $\boldsymbol{t}$-stationarity to find the optimal $\boldsymbol{t}^*$,

$$\boldsymbol{t}^* = \frac{1}{2\gamma_t} \boldsymbol{S}_t^{-1} \tilde{\boldsymbol{t}}$$

Next, we will use this results with $\boldsymbol{t}$-complementary slackness to find γ_t^*:

$$\begin{aligned} 0 &= \gamma_t \left(\|\boldsymbol{t}^*\|_{\boldsymbol{S}_t}^2 - 1 \right) \\ &= \gamma_t \left(\frac{1}{4\gamma_t^2} \tilde{\boldsymbol{t}}^T \boldsymbol{S}_t^{-T} \boldsymbol{S}_t \boldsymbol{S}_t^{-1} \tilde{\boldsymbol{t}} - 1 \right) \\ &= \gamma_t \left(\frac{1}{4\gamma_t^2} \tilde{\boldsymbol{t}}^T \boldsymbol{S}_t^{-1} \tilde{\boldsymbol{t}} - 1 \right) \\ &= \gamma_t \left(\frac{1}{4\gamma_t^2} \|\tilde{\boldsymbol{t}}\|_{\boldsymbol{S}_t^{-1}}^2 - 1 \right) \end{aligned}$$

$$\gamma_t^{*2} = \frac{1}{4} \|\tilde{\boldsymbol{t}}\|_{\boldsymbol{S}_t^{-1}}^2$$

Plugging γ_t^* into $\boldsymbol{t}^*$ yields the coordinate update Eq. (8). By Proposition 1, $\boldsymbol{t}^*$ is a global optimum of its sub-problem, and local optimum of Eq. (4), holding all other factors fixed.

We conclude that ρ-PCA converges to a local optimum, as well as find the block coordinate updates. □

B. Model Selection: Tuning ρ-PCA and ρ-PLS

1) Tuning ρ-PCA: Nested Bayesian Information Criterion.: The performance of ρ-PCA can be improved by selecting values of parameters that yield the best model fit criteria. There are several unknown parameters in Algorithm 1: λ_v, λ_w, α_w, and α_t. We suggest the tuning ρ-PCA using the nested Bayesian Information Criterion (BIC) schema [4, 7]. The BIC identifies the most promising candidate parameters based, in part, on the likelihood of the model [9]. In the case of the Algorithm 1, we cannot base our BIC on the likelihood, as the penalty selection is different for each mode. In response,

Proposition 2 defines the BIC for each mode separately, based on their respective regression problem.

Proposition 2. *We can calculate BIC for each mode as follows,*

$$BIC(\tilde{\boldsymbol{v}}) = \log\left(\frac{1}{p}\left\|\left(\boldsymbol{\mathcal{X}} \times_1 \hat{\boldsymbol{u}} \times_3 \hat{\boldsymbol{w}} \times_4 \hat{\boldsymbol{t}}\right) - \tilde{\boldsymbol{v}}\right\|_2^2\right) + \frac{1}{p}\log(p) \cdot \|\tilde{\boldsymbol{v}}\|_0, \tag{13}$$

$$BIC(\tilde{\boldsymbol{w}}) = \log\left(\frac{1}{q}\left\|\left(\boldsymbol{\mathcal{X}} \times_1 \hat{\boldsymbol{u}} \times_2 \hat{\boldsymbol{v}} \times_4 \hat{\boldsymbol{t}}\right) - \tilde{\boldsymbol{w}}\right\|_2^2\right) + \frac{1}{q}\log(q) \cdot Tr\left[\left(\boldsymbol{I}_{|\mathcal{A}|} + \alpha_w \boldsymbol{\Omega}_w^{\mathcal{A}}\right)^{-1}\right], \tag{14}$$

where $\mathcal{A}$ represents the indices of the active set, or estimated non-zero elements of $\hat{\boldsymbol{w}}$, and $\boldsymbol{\Omega}_w^{\mathcal{A}}$ is the corresponding submatrix of $\boldsymbol{\Omega}_w$,

$$BIC(\tilde{\boldsymbol{t}}) = \log\left(\frac{1}{r}\left\|\left(\boldsymbol{\mathcal{X}} \times_1 \hat{\boldsymbol{u}} \times_2 \hat{\boldsymbol{v}} \times_3 \hat{\boldsymbol{w}}\right) - \tilde{\boldsymbol{t}}\right\|_2^2\right) + \frac{1}{r}\log(r) \cdot Tr\left[\boldsymbol{S}_t\right] \tag{15}$$

Notice that the BIC formulations (13)-(15) are of similar form, with the exception of degrees of freedom in the second additive term. Each penalty we use in ρ-PCA is has a different calculation for its associated degrees of freedom. The degrees of freedom for electrodes ($\boldsymbol{v}$), frequency ($\boldsymbol{w}$), and time ($\boldsymbol{t}$) are derived from Allen [2], Allen and Weylandt [5], and Goutte and Amini [6], respectively. The mode trial ($\boldsymbol{u}$) is excluded, since it has no associated parameter. Algorithm 3 describes model selection for ρ-PCA in detail.

2) Tuning ρ-PLS: Nested Cross-validation.: Just like ρ-PCA, ρ-PLS has unknown hyperparameters λ_v, λ_w, α_w, and α_t. To select an optimal ρ-PLS model given the data at hand, we implement a nested N-fold cross-validation scheme within each factor K. This approach provides control over the regularization in each component. For each factor $k = 1, \ldots, K$, we select the set of regularization parameters that maximize prediction accuracy for all of the currently computed components. We cache the results of each run so that we can deflate the covariance tensor appropriately. This procedure described in Algorithm 4.

C. Cumulative Amount of Variance Explained

Principal Components Analysis computes a set of new variables called *Principal Components (PCs)* or *factor loadings*. The PCs describe the amount of variance explained by the covariates. Similarly, tensor PCs can be computed from a tensor decompositions [1]. In the context of ρ-PCA, the estimated K component vectors for each mode are:

$$\hat{\boldsymbol{U}} := \{\hat{\boldsymbol{u}}_k\}_{k=1}^K, \hat{\boldsymbol{V}} := \{\hat{\boldsymbol{v}}\}_{k=1}^K,$$

$$\hat{\boldsymbol{W}}_k := \{\hat{\boldsymbol{w}}_k\}_{k=1}^K, \text{ and } \hat{\boldsymbol{T}}_k := \{\hat{\boldsymbol{t}}_k\}_{k=1}^K$$

In turn, these projections can be used to find the cumulative proportion of variance explained and evaluate the estimation of ρ-PCA, noted by Proposition 3 [4]:

Proposition 3 (Allen and Maletić-Savatić [4])**.** *Define $\hat{\boldsymbol{U}}_k = \{\hat{\boldsymbol{u}}_i\}_{i=1}^k \in \mathbb{R}^{n \times k}$, and the projection matrix $\boldsymbol{P}_k^{(U)} = \hat{\boldsymbol{U}}_k \left(\hat{\boldsymbol{U}}_k^T \hat{\boldsymbol{U}}_k\right)^{-1} \hat{\boldsymbol{U}}_k^T$. The matrices $\boldsymbol{P}_k^{(V)}$, $\boldsymbol{P}_k^{(W)}$, and $\boldsymbol{P}_k^{(T)}$ can be defined analogously. The cumulative amount of variance explained (CAVE) by the first k higher-order principal components is given by*

$$CAVE_k = \frac{\left\|\boldsymbol{\mathcal{X}} \times_1 \boldsymbol{P}_k^{(U)} \times_2 \boldsymbol{P}_k^{(V)} \times_3 \boldsymbol{P}_k^{(W)} \times_4 \boldsymbol{P}_k^{(T)}\right\|_F^2}{\|\boldsymbol{\mathcal{X}}\|_F^2}$$

Algorithm 3 Model selection for ρ-PCA: Nested BIC

1) Initialization:
   (A) Set the maximum iterations B, $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}$, candidate parameters $\{\lambda_v\}_{i=1}^{P_v}$, $\{(\lambda_w, \alpha_w)\}_{i=1}^{P_w}$, and $\{\alpha_t\}_{i=1}^{P_t}$.
   (B) Initialize $\left\{\left(\hat{d}_k, \hat{\boldsymbol{u}}_k, \hat{\boldsymbol{v}}_k, \hat{\boldsymbol{w}}_k, \hat{\boldsymbol{t}}_k\right)\right\}_{k=1}^{K}$ using CP-decomposition.
2) For $k = 1, \ldots, K$:
   (A) Repeat until convergence, or for a maximum of B iterations:
      i. Update $\hat{\boldsymbol{u}}_k$ using Eq. (5) from Algorithm 1.
      ii. Update $\hat{\boldsymbol{v}}_k$,
         (a) For each candidate parameter λ_{v_i}, $i = 1, \ldots P_v$.
            – Calculate, $\breve{\boldsymbol{v}}_k(\lambda_{v_i}) = S(\hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_{k-1} \times_3 \hat{\boldsymbol{w}}_{k-1} \times_4 \hat{\boldsymbol{t}}_{k-1}, \lambda_{v_i})$
            – Find the BIC using $\breve{\boldsymbol{v}}_k(\lambda_{v_i})$ and Eq. (13), denoted as BIC$(\breve{\boldsymbol{v}}_k(\lambda_{v_i}))$.
            – Select the optimal parameter $\lambda_{v_i}^*$ and factor, $\tilde{\boldsymbol{v}}_k = \arg\min_{\lambda_{v_i}} \text{BIC}(\breve{\boldsymbol{v}}_k(\lambda_{v_i}))$.
            – Re-scale the factor to obtain $\hat{\boldsymbol{v}}_k$, $\hat{\boldsymbol{v}}_k = \begin{cases} \frac{\tilde{\boldsymbol{v}}_k}{\|\tilde{\boldsymbol{v}}_k\|_2}, & \|\tilde{\boldsymbol{v}}_k\|_2 > 0, \\ 0, & otherwise \end{cases}$
      iii. Update $\hat{\boldsymbol{w}}_k$ using a single proximal update
         (a) For each set of candidate parameters $(\lambda_w, \alpha_w)_i$, $i = 1, \ldots, P_w$,
            – Calculate $\breve{\boldsymbol{w}}_k(\lambda_{w_i}, \alpha_{w_i})$, $\boldsymbol{S}_{w_i} = \boldsymbol{I}_q + \alpha_{w_i}\boldsymbol{\Omega}_w$, and
$$\breve{\boldsymbol{w}}_k(\lambda_{w_i}, \alpha_{w_i}) = S\left(\hat{\boldsymbol{w}}_{k-1} + L_{w_i}^{-1}\left(\hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_{k-1} \times_2 \hat{\boldsymbol{v}}_{k-1} \times_4 \hat{\boldsymbol{t}}_{k-1} - \boldsymbol{S}_{w_i}\hat{\boldsymbol{w}}_{k-1}\right), \frac{\lambda_{w_i}}{L_{w_i}}\right)$$
            – Find the BIC using $\breve{\boldsymbol{w}}_k(\lambda_{w_i}, \alpha_{w_i})$ and Eq. (14), denoted as BIC$(\breve{\boldsymbol{w}}_k(\lambda_{w_i}, \alpha_{w_i}))$.
            – Select the optimal parameters $(\lambda_{w_i}^*, \alpha_{w_i}^*)$, and factor,
$$\tilde{\boldsymbol{w}}_k = \underset{(\lambda_{w_i}, \alpha_{w_i})}{\arg\min}\ \text{BIC}(\breve{\boldsymbol{w}}_k(\lambda_{w_i}, \alpha_{w_i}))$$
            – Re-scale the factor to get $\hat{\boldsymbol{w}}_k$, $\hat{\boldsymbol{w}} = \begin{cases} \frac{\tilde{\boldsymbol{w}}_k}{\|\tilde{\boldsymbol{w}}_k\|_{\boldsymbol{S}_{w_i}^*}}, & \|\tilde{\boldsymbol{w}}_k\|_{\boldsymbol{S}_{w_i}^*} > 0, \\ 0, & otherwise \end{cases}$
      iv. Update $\hat{\boldsymbol{t}}_k$,
         (a) For each parameter α_{t_i}, $i = 1, \ldots, P_t$,
            – Calculate $\breve{\boldsymbol{t}}_k(\alpha_{t_i})$, $\boldsymbol{S}_{t_i} = \boldsymbol{I}_r + \alpha_{t_i}\boldsymbol{\Omega}_t$, and $\breve{\boldsymbol{t}}_k(\alpha_{t_i}) = \boldsymbol{S}_t^{-1}\left(\hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_{k-1} \times_2 \hat{\boldsymbol{v}}_{k-1} \times_3 \hat{\boldsymbol{w}}_{k-1}\right)$.
            – Find the BIC using $\breve{\boldsymbol{t}}_k(\alpha_{t_i})$ and Eq. (15), denoted as BIC$(\breve{\boldsymbol{t}}_k(\alpha_{t_i}))$
            – Select the optimal parameter α_{t_i}, and factor $\tilde{\boldsymbol{t}}_k = \arg\min_{\alpha_{t_i}} \text{BIC}(\breve{\boldsymbol{t}}_k(\alpha_{t_i}))$.
            – Re-scale the factor for $\hat{\boldsymbol{t}}_k$, $\hat{\boldsymbol{t}} = \begin{cases} \frac{\tilde{\boldsymbol{t}}_k}{\|\tilde{\boldsymbol{t}}_k\|_{\boldsymbol{S}_{t_i}^*}}, & \|\tilde{\boldsymbol{t}}_k\|_{\boldsymbol{S}_{t_i}^*} > 0, \\ 0, & otherwise \end{cases}$
   (B) Estimate $\hat{d}_k$ and deflate $\hat{\boldsymbol{\mathcal{X}}}$,
$$\hat{d}_k = \hat{\boldsymbol{\mathcal{X}}} \times_1 \hat{\boldsymbol{u}}_k \times_2 \hat{\boldsymbol{v}}_k \times_3 \hat{\boldsymbol{w}}_k \times_4 \hat{\boldsymbol{t}}_k,$$
$$\hat{\boldsymbol{\mathcal{X}}} = \hat{\boldsymbol{\mathcal{X}}} - \hat{d}_k \cdot \hat{\boldsymbol{u}}_k \circ \hat{\boldsymbol{v}}_k \circ \hat{\boldsymbol{w}}_k \circ \hat{\boldsymbol{t}}_k$$
   (C) Cache the optimal tuning parameters $(\lambda_v^*, \lambda_w^*, \alpha_w^*, \alpha_t^*)$ for k.
3) Return the optimal parameters $\{(\lambda_v^*, \lambda_w^*, \alpha_w^*, \alpha_t^*)_k\}_{k=1}^K$.

Algorithm 4 Model selection for ρ-PLS: Nested N-fold cross validation

1) Initialization:
   (A) Define the number of folds B, and sets of candidate parameters $\boldsymbol{\lambda} = \left\{ (\lambda_v, \lambda_w, \alpha_w, \alpha_t)_j \right\}_{j=1}^{J}$.
   (B) Split data into B parts to create training $(\boldsymbol{\mathcal{X}}_b^{tr}, \boldsymbol{y}_b^{tr})$ and testing $(\boldsymbol{\mathcal{X}}_b^{ts}, \boldsymbol{y}_b^{ts})$ sets. Center all responses to obtain $\bar{\boldsymbol{y}}_b^{tr}$ and $\bar{\boldsymbol{y}}_b^{ts}$ for each fold.
2) For each component $k = 1, \ldots, K$,
   (A) For each fold $b = 1, \ldots, B$,
      i. For each set of candidate parameters $\boldsymbol{\lambda}_j = (\lambda_v, \lambda_w, \alpha_w, \alpha_t)_j$, $j = 1, \ldots, J$,
         (a) Compute the covariance tensor from training data, $\boldsymbol{\mathcal{Z}}_b = \boldsymbol{\mathcal{X}}_b \times_1 \bar{\boldsymbol{y}}_b^{tr}$, and set $\hat{\boldsymbol{\mathcal{Z}}}_b = \boldsymbol{\mathcal{Z}}_b$.
         (b) Using the ρ-PCA step for component k from Algorithm 1, compute the components of $\hat{\boldsymbol{\mathcal{Z}}}_b$,
         $$\left[\hat{\boldsymbol{v}}_{jbk}, \hat{\boldsymbol{w}}_{jbk}, \hat{\boldsymbol{t}}_{jbk}\right] = \rho\text{-PCA}(\hat{\boldsymbol{\mathcal{Z}}}_b, k, \lambda_j)$$
         (c) Reduce the dimensionality of the data,
         $$\tilde{\boldsymbol{X}}_{jbk}^{tr} = \boldsymbol{\mathcal{X}}_b^{tr} \times_2 \hat{\boldsymbol{v}}_{jbk} \times_3 \hat{\boldsymbol{w}}_{jbk} \times_4 \hat{\boldsymbol{t}}_{jbk}$$
         $$\tilde{\boldsymbol{X}}_{jbk}^{ts} = \boldsymbol{\mathcal{X}}_b^{ts} \times_2 \hat{\boldsymbol{v}}_{jbk} \times_3 \hat{\boldsymbol{w}}_{jbk} \times_4 \hat{\boldsymbol{t}}_{jbk}$$
         (d) Predict the class memberships of $\boldsymbol{y}_b^{ts}$ using choice SLM, $\hat{\boldsymbol{y}}_{jbk} = \text{SLM}\left(\tilde{\boldsymbol{X}}_{jbk}^{tr}, \tilde{\boldsymbol{X}}_{jbk}^{ts}, \boldsymbol{y}_b^{tr}\right)$.
         (e) Calculate the accuracy, denoted as ϕ_{jbk}, between $\boldsymbol{Y}_b^{ts}$ and $\hat{\boldsymbol{y}}_{jbk}$.
   (B) Find the optimal parameter set, $\lambda_k^* = \lambda_{j_k^*}$, for k using the average accuracy, $j_k^* = \arg\max_j \frac{1}{B} \sum_{b=1}^{B} \phi_{jbk}$
   (C) For each fold $b = 1, \ldots, B$,
      i. Cache the estimates $\hat{\boldsymbol{v}}_{j_k^* bk}$, $\hat{\boldsymbol{w}}_{j_k^* bk}$, and $\hat{\boldsymbol{t}}_{j_k^* bk}$ using the optimal parameter set λ_k^*.
      ii. Estimate $\hat{d}_{j_k^* bk}$ and deflate $\hat{\boldsymbol{\mathcal{Z}}}_b$ as described by Eq. (9), from Alg. 1.
      $$\hat{d}_{j_k^* bk} = \hat{\boldsymbol{\mathcal{Z}}}_b \times_1 \hat{\boldsymbol{v}}_{j_k^* bk} \times_2 \hat{\boldsymbol{w}}_{j_k^* bk} \times_3 \hat{\boldsymbol{t}}_{j_k^* bk}$$
      $$\hat{\boldsymbol{\mathcal{Z}}}_b = \hat{\boldsymbol{\mathcal{Z}}}_b - \hat{d}_{j_k^* bk} \cdot \hat{\boldsymbol{v}}_{j_k^* bk} \circ \hat{\boldsymbol{w}}_{j_k^* bk} \circ \hat{\boldsymbol{t}}_{j_k^* bk}$$
3) Return the optimal parameters for each component, $\{(\lambda_v^*, \lambda_w^*, \alpha_w^*, \alpha_t^*)_k\}_{k=1}^{K}$
4) Run Alg. 2 using the optimal parameters and full data set.

REFERENCES

[1] Allen, G. (2012). Sparse higher-order principal components analysis. In *Artificial Intelligence and Statistics*, pages 27–36.

[2] Allen, G. I. (2013). Multi-way functional principal components analysis. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 220–223. IEEE.

[3] Allen, G. I., Grosenick, L., and Taylor, J. (2014). A generalized least-square matrix decomposition. *Journal of the American Statistical Association*, 109(505):145–159.

[4] Allen, G. I. and Maletić-Savatić, M. (2011). Sparse non-negative generalized pca with applications to metabolomics. *Bioinformatics*, 27(21):3029–3035.

[5] Allen, G. I. and Weylandt, M. (2019). Sparse and functional principal components analysis. In *2019 IEEE Data Science Workshop (DSW)*, pages 11–16.

[6] Goutte, C. and Amini, M.-R. (2010). Probabilistic tensor factorization and model selection. *Tensors, Kernels, and Machine Learning (TKLM 2010)*, pages 1–4.

[7] Lee, M., Shen, H., Huang, J. Z., and Marron, J. (2010). Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095.

[8] Parikh, N., Boyd, S., et al. (2014). Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239.

[9] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

[10] Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494.

[11] Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.

[12] Yu, Y.-L. (2013). On decomposing the proximal map. In *Advances in Neural Information Processing Systems*, pages 91–99.