Brain Network Identification in Asynchronous Task fMRI Data Using Robust and Scalable Tensor Decomposition

Jian Li[a,*], Jessica L. Wisnowski[b,c], Anand A. Joshi[a], Richard M. Leahy[a]

[a]Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089
[b]Radiology and Pediatrics, Division of Neonatology, Children's Hospital Los Angeles, Los Angeles, CA 90027
[c]Keck School of Medicine, University of Southern California, Los Angeles, CA 90089

ABSTRACT

The goal of this work is to robustly identify common brain networks and their corresponding temporal dynamics across subjects in asynchronous task functional MRI (tfMRI) signals. We approached this problem using a robust and scalable tensor decomposition method combined with the BrainSync algorithm. We first used BrainSync algorithm to temporally align asynchronous tfMRI data, allowing us to study common brain networks across subjects. We mapped the synchronized tfMRI data into a 3D tensor (vertices × time × session) and performed a greedy canonical polyadic (CP) decomposition, reducing the rank to 20 in order to improve the signal-to-noise ratio (SNR). We incorporated the Nesterov-accelerated adaptive moment estimation into our previously developed scalable and robust sequential CP decomposition (SRSCPD) framework and applied this improved version of SRSCPD to the rank-reduced tensor to identify dynamic brain networks. We successfully identified 9 brain networks with their corresponding temporal dynamics from 40 subjects using Human Connectome Project tfMRI data without using any prior information with regard to the task designs. Three of these show the subjects' responses to cues at the beginning of each task block (fronto-parietal attentional control network, visual network and executive control network); one corresponds to the default mode network that exhibits deactivation during the tasks; four show motors networks (left hand, right hand, tongue, and both feet) where the temporal dynamics are strongly correlated to the task designs, and the remaining component reflects physiological noise (respiration).

Keywords: tensor decomposition, optimization, brain network identification, fMRI

1. INTRODUCTION

Functional magnetic resonance imaging (fMRI) is a powerful tool for imaging in-vivo human brain activity. In particular, in recent years, fMRI has been used to study spatio-temporal organization through studies of functional connectivity (FC)[1] and the manner in which brain activity changes over time during spontaneous and task related activity.

Independent component analysis (ICA) has been widely used for dynamic brain network identification. Spatial ICA (sICA) finds spatially disjoint brain regions that have coherent time-series, while temporal ICA (tICA) finds independent temporal dynamics with possible overlapping spatial maps. Several approaches applying ICA to multi-subject fMRI data have been proposed. Esposito *et al.*[2] performed ICA on each individual subject then combined the independent components together. Other methods differ in how data is organized/concatenated before applying ICA. Temporal concatenation[3,4] allows unique time-series for each subject but shared spatial maps, while spatial concatenation[5,6] assumes common time-series but unique spatial maps. Although meaningful components have been extracted using these ICA-based approaches[7], they require either spatial or temporal independence, which may not be realistic as brain networks can overlap and be correlated in both space and time[8]. Further, through concatenating these data into 2D matrices prior to applying ICA, low rank structure that may be inherent in the natural third-order tensor representation (space × time × subject) may be lost.

Higher-order tensor decompositions are a generalization of matrix decompositions and capable of representing multi-dimensional data using a low rank model. Application of tensor decomposition, especially the canonical polyadic (CP) model[9,10], to fMRI data for brain network identification have previously been explored. Generally, the decomposition is performed using the alternating least square (ALS) algorithm on a group-level fMRI study, to find common networks among subjects. For example, Andersen *et al.*[11] applied a third-order CP decomposition to finger-tapping task fMRI (tfMRI) data using the ALS algorithm. Beckmann and Smith[12] extended ICA to higher-order tensors by imposing an

---

[*] E-mail: jli981@usc.edu; Phone: +1(213)740-6430;

independence constraint in the spatial dimension. Instead of adding an independence constraint, Sen and Parhi[13] imposed an orthogonality constraint in the spatial dimension as with PCA. However, CP decomposition on fMRI data is not as popular as other methods because there are several issues that limit its applicability to fMRI studies:

Multi-subject group analysis on asynchronous fMRI data: Temporal synchrony across multiple subjects is a strict requirement for CP decomposition to work well with low rank models. However, this assumption may not be satisfied even when an identical task design is used across all subjects, because individual responses to tasks may differ (sometimes significantly for higher level cognitive tasks) in their latencies. Low rank CP decompositions will certainly fail when using task designs that vary across subjects. Such is the case with the tfMRI data we use in the experiments below where each subject has two sessions of recordings with different orders of the tasks. Finally, any brain processes independent of the task (such as spontaneous activity in the default mode network (DMN)) will not be identified using a CP decomposition since subject to subject DMN activity will not be synchronized.

Robustness against local minima and scalability to large dataset: The ALS algorithm is not guaranteed to converge to a global minimum or a stationary point for a CP model, even with multi-start[9,10]. Although adding additional constraints, such as independence[12] or orthogonality[13], may help avoid local minima, those constraints may not be physiologically reasonable for the brain network identification problem. Indeed, specific concerns[14,15] have been raised against imposing those constraints when performing CP decompositions on fMRI data. Moreover, the naïve ALS algorithm does not scale well to large datasets. As we have shown previously[16,17], the computational complexity is approximately quadratically proportional to the largest dimension of the tensor. In fact, most of the studies cited above heavily down-sampled the data in the spatial domain in order to have a tractable CP decomposition. In addition, multi-start strategies are required for robustness, particularly when dealing with fMRI data where the signal-to-noise ratio (SNR) is poor[16,17].

In this paper, we describe a method to robustly find common brain networks (i.e. identify *both spatial maps and temporal dynamics* simultaneously) across multiple subjects from *asynchronous* task fMRI data *without imposing unrealistic constraint* on the components such as independence or orthogonality. We approach this problem by incorporating the Nesterov-accelerated adaptive moment estimation (Nadam) method[18] into our recently developed scalable and robust sequential canonical polyadic decomposition (SRSCPD) framework[16,17], which was originally designed for brain network identification in EEG data, and combine it with the BrainSync[19] algorithm, which uses a temporal orthogonal transform to align time-series across subjects. We refer to our proposed tensor decomposition algorithm as Nadam-accelerated scalable and robust CP decomposition (NSRCPD).

2. PRELIMINARIES

We first define our mathematical notation, mostly following the conventions used in Kolda and Bader[9]. We then review the ALS algorithm, the SRSCPD framework, the adaptive moment estimation (Adam) first-order solver and its Nesterov-accelerated version, and the BrainSync algorithm, all of which will be used in part of our NSRCPD algorithm and brain network identification pipeline.

2.1 Notations

We use a lowercase letter to represent a scalar, e.g., x; a bold lowercase letter for a vector, e.g., $\boldsymbol{x}$; a bold uppercase letter for a matrix, e.g., $\boldsymbol{X}$ and a bold script letter for a tensor, e.g., $\boldsymbol{\mathcal{X}}$. The number of dimensions is called the *order* and each dimension is referred to as a *mode*. A third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ will be used in the following sections as an example with $x_{i,j,k}$ denoting its individual element. However, notations and algorithms can be extended naturally to higher-order tensors. A norm of a tensor $\boldsymbol{\mathcal{X}}$, analogous to the Frobenius norm for a matrix, is defined as:

$$\|\boldsymbol{\mathcal{X}}\| = \sqrt{\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} x_{i,j,k}^2} \tag{1}$$

A tensor can be matricized or unfolded into a matrix along the nth dimension, denoted by $\boldsymbol{X}_{(n)}$. Therefore, for a third-order tensor, $\boldsymbol{X}_{(1)} \in \mathbb{R}^{I \times JK}$ or $\boldsymbol{X}_{(2)} \in \mathbb{R}^{J \times IK}$ or $\boldsymbol{X}_{(3)} \in \mathbb{R}^{K \times IJ}$. The Kronecker product between matrix $\boldsymbol{X}$ and $\boldsymbol{Y}$ is

$$\boldsymbol{X} \otimes \boldsymbol{Y} = \begin{bmatrix} x_{11}\boldsymbol{Y} & x_{12}\boldsymbol{Y} & \cdots & x_{1J}\boldsymbol{Y} \\ x_{21}\boldsymbol{Y} & x_{22}\boldsymbol{Y} & \cdots & x_{2J}\boldsymbol{Y} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I1}\boldsymbol{Y} & x_{I2}\boldsymbol{Y} & \cdots & x_{IJ}\boldsymbol{Y} \end{bmatrix} \tag{2}$$

The Khatri-Rao product between matrix $\boldsymbol{X}$ and $\boldsymbol{Y}$ is

$$\boldsymbol{X} \odot \boldsymbol{Y} = [\boldsymbol{x}_1 \otimes \boldsymbol{y}_1 \quad \boldsymbol{x}_2 \otimes \boldsymbol{y}_2 \quad \cdots \quad \boldsymbol{x}_K \otimes \boldsymbol{y}_K] \tag{3}$$

where $\boldsymbol{x}_i$ is the ith column of $\boldsymbol{X}$. The Hadamard product between matrix $\boldsymbol{X}$ and $\boldsymbol{Y}$ is

$$\boldsymbol{X} * \boldsymbol{Y} = \begin{bmatrix} x_{11}y_{11} & x_{12}y_{12} & \cdots & x_{1J}y_{1J} \\ x_{21}y_{21} & x_{22}y_{22} & \cdots & x_{2J}y_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I1}y_{I1} & x_{I2}y_{I2} & \cdots & x_{IJ}y_{IJ} \end{bmatrix} \tag{4}$$

A useful property of the Khatri-Rao product is:

$$(\boldsymbol{X} \odot \boldsymbol{Y})^{\dagger} = (\boldsymbol{X}^T\boldsymbol{X} * \boldsymbol{Y}^T\boldsymbol{Y})^{\dagger}(\boldsymbol{X} \odot \boldsymbol{Y})^T \tag{5}$$

where $\boldsymbol{X}^{\dagger}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{X}$.

2.2 CP Decomposition and the ALS Algorithm

The CP decomposition approximates a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ as a sum of rank-1 tensors with the following objective function:

$$f = \min_{\lambda_r, \boldsymbol{a}_r, \boldsymbol{b}_r, \boldsymbol{c}_r} \frac{1}{2} \left\| \boldsymbol{\mathcal{X}} - \sum\nolimits_{r=1}^{R} \lambda_r\, \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \right\|^2 \tag{6}$$

where λ_r represents the scale of the rth component, $\boldsymbol{a}_r \in \mathbb{R}^I$, $\boldsymbol{b}_r \in \mathbb{R}^J$, $\boldsymbol{c}_r \in \mathbb{R}^K$ have unit norm, "$\circ$" represents the vector outer product and R is the *rank*. If we concatenate the $\boldsymbol{a}_r$ together, forming a matrix $\boldsymbol{A} = [\boldsymbol{a}_1\ \boldsymbol{a}_2 \cdots \boldsymbol{a}_R] \in \mathbb{R}^{I\times R}$ and similarly for $\boldsymbol{B} \in \mathbb{R}^{J\times R}$ and $\boldsymbol{C} \in \mathbb{R}^{K\times R}$, then we can re-write (6) as one of the following three equations[9]:

$$f = \min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \frac{1}{2} \left\|\boldsymbol{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^T\right\|_F^2 = \min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \frac{1}{2} \left\|\boldsymbol{X}_{(2)} - \boldsymbol{B}(\boldsymbol{C} \odot \boldsymbol{A})^T\right\|_F^2 = \min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \frac{1}{2} \left\|\boldsymbol{X}_{(3)} - \boldsymbol{C}(\boldsymbol{B} \odot \boldsymbol{A})^T\right\|_F^2 \tag{7}$$

A regularized version of the ALS algorithm solves this problem iteratively: we first solve for $\boldsymbol{A}$ with $\boldsymbol{B}$ and $\boldsymbol{C}$ fixed, then solve for $\boldsymbol{B}$ with $\boldsymbol{A}$ and $\boldsymbol{C}$ fixed, and so on until convergence. Specifically, suppose we fix $\boldsymbol{B}$ and $\boldsymbol{C}$ and solve for $\boldsymbol{A}$:

$$\widehat{\boldsymbol{A}} = \underset{\boldsymbol{A}}{\operatorname{argmin}} \frac{1}{2} \left\|\boldsymbol{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^T\right\|_F^2 + \frac{\mu_1}{2} g_1(\boldsymbol{A}) \tag{8}$$

where $g_1(\boldsymbol{A})$ is a regularizing function on the loading matrix $\boldsymbol{A}$ and μ_1 is the corresponding regularization parameter. The solution of (8) reduces to a least square solution if $\mu_1 = 0$ and has the following expression using (5):

$$\widehat{\boldsymbol{A}} = \boldsymbol{X}_{(1)}[(\boldsymbol{C} \odot \boldsymbol{B})^T]^{\dagger} = \boldsymbol{X}_{(1)}(\boldsymbol{C} \odot \boldsymbol{B})(\boldsymbol{C}^T\boldsymbol{C} * \boldsymbol{B}^T\boldsymbol{B})^{\dagger} \tag{9}$$

In cases where $\mu_1 \neq 0$, the solution of (8) will have a close form expression if $g_1(\boldsymbol{A})$ is a quadratic term but may require an iterative solution in other cases. The full ALS algorithm is shown in Algorithm I.

ALGORITHM I: CP-ALS

```
Algorithm CP-ALS ($\boldsymbol{\mathcal{X}}, R, \{\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{C}^*, \boldsymbol{\lambda}^*\}$)
    [1]Initialize $\boldsymbol{A} \in \mathbb{R}^{I\times R}, \boldsymbol{B} \in \mathbb{R}^{J\times R}, \boldsymbol{C} \in \mathbb{R}^{K\times R}, \boldsymbol{\lambda} \in \mathbb{R}^R$
    While not converged[2]
        $\boldsymbol{A} \leftarrow \underset{\boldsymbol{A}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^T\|_F^2 + \frac{\mu_1}{2} g_1(\boldsymbol{A})$
        $\boldsymbol{B} \leftarrow \underset{\boldsymbol{B}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{X}_{(2)} - \boldsymbol{B}(\boldsymbol{C} \odot \boldsymbol{A})^T\|_F^2 + \frac{\mu_2}{2} g_2(\boldsymbol{B})$
        $\boldsymbol{C} \leftarrow \underset{\boldsymbol{C}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{X}_{(3)} - \boldsymbol{C}(\boldsymbol{B} \odot \boldsymbol{A})^T\|_F^2 + \frac{\mu_3}{2} g_3(\boldsymbol{C})$
    End While
    Normalize $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ such that each column has unit norm and set $\lambda_r$ equal to the normalization factor for the $r^{\text{th}}$ component, $r = 1, \dots, R$.
    Return $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{\lambda}$
End Algorithm
```

[1] The initialization is performed using random matrices if not specifically supplied or using the provided values $\{\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{C}^*, \boldsymbol{\lambda}^*\}$.

[2] Effective convergence is defined as the point when the mean of the absolute difference of the loading matrices between two adjacent iterations over all modes is less than some small constant, e.g. 10^{-5}.

2.3 The SRSCPD Framework

We recently developed a SRSCPD framework for robust and scalable tensor decompositions of EEG data[16,17]. The robustness and scalability were achieved by using the results from the rank $r-1$ decomposition as a warm initialization for the rank r decomposition. The generic SRSCPD framework is shown in Algorithm II. For a third-order tensor example, we take a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ and a desired maximum rank R as the inputs. At each iteration r, a rank-r fit is computed using any CP decomposition algorithm with initialization $\{\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{C}^*, \boldsymbol{\lambda}^*\}$, which is formed by concatenating the result $\{\boldsymbol{A}^{r-1}, \boldsymbol{B}^{r-1}, \boldsymbol{C}^{r-1}, \boldsymbol{\lambda}^{r-1}\}$ from the previous iteration $r-1$ with the rank-1 approximation $\{\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}', \lambda'\}$ of the residue tensor $\boldsymbol{\mathcal{X}}_{res}$, where $\boldsymbol{\mathcal{X}}_{res}$ is obtained by subtracting the reconstructed tensor using $\{\boldsymbol{A}^{r-1}, \boldsymbol{B}^{r-1}, \boldsymbol{C}^{r-1}, \boldsymbol{\lambda}^{r-1}\}$ from the original data tensor $\boldsymbol{\mathcal{X}}$.

ALGORITHM II: SRSCPD

```
Algorithm SRSCPD ($\boldsymbol{\mathcal{X}}, R$)
    $\boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{c}^1, \lambda^1 \leftarrow$ CP-Algorithm ($\boldsymbol{\mathcal{X}}$, 1)
    $\boldsymbol{\mathcal{X}}_{res} \leftarrow \boldsymbol{\mathcal{X}} -$ Tensor_Recon ($\boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{c}^1, \lambda^1$)
    $\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}', \lambda' \leftarrow$ CP-Algorithm ($\boldsymbol{\mathcal{X}}_{res}$, 1)
    $\boldsymbol{A}^* \leftarrow [\boldsymbol{a}^1\ \boldsymbol{a}']$; $\boldsymbol{B}^* \leftarrow [\boldsymbol{b}^1\ \boldsymbol{b}']$; $\boldsymbol{C}^* \leftarrow [\boldsymbol{c}^1\ \boldsymbol{c}']$; $\boldsymbol{\lambda}^* \leftarrow \begin{bmatrix}\lambda^1 \\ \lambda'\end{bmatrix}$
    For $r = 2, 3, \dots, R$
        $\boldsymbol{A}^r, \boldsymbol{B}^r, \boldsymbol{C}^r, \boldsymbol{\lambda}^r \leftarrow$ CP-Algorithm ($\boldsymbol{\mathcal{X}}, r, \{\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{C}^*, \boldsymbol{\lambda}^*\}$)
        $\boldsymbol{\mathcal{X}}_{res} \leftarrow \boldsymbol{\mathcal{X}} -$ Tensor_Recon ($\boldsymbol{A}^r, \boldsymbol{B}^r, \boldsymbol{C}^r, \boldsymbol{\lambda}^r$)
        $\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}', \lambda' \leftarrow$ CP-Algorithm ($\boldsymbol{\mathcal{X}}_{res}$, 1)
        $\boldsymbol{A}^* \leftarrow [\boldsymbol{A}^r\ \boldsymbol{a}']$; $\boldsymbol{B}^* \leftarrow [\boldsymbol{B}^r\ \boldsymbol{b}']$; $\boldsymbol{C}^* \leftarrow [\boldsymbol{C}^r\ \boldsymbol{c}']$; $\boldsymbol{\lambda}^* \leftarrow \begin{bmatrix}\boldsymbol{\lambda}^r \\ \lambda'\end{bmatrix}$
    End For
    Return a set of solutions
    $\{\boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{c}^1, \lambda^1\}, \{\boldsymbol{A}^2, \boldsymbol{B}^2, \boldsymbol{C}^2, \boldsymbol{\lambda}^2\}, \dots, \{\boldsymbol{A}^R, \boldsymbol{B}^R, \boldsymbol{C}^R, \boldsymbol{\lambda}^R\}$
End Algorithm
```

2.4 Gradient of the CP Model, Adam and Nadam

If we treat the variables in a CP model as a high-dimensional vector lying in the space $\boldsymbol{x} \in \mathbb{R}^N$, where $N = I \times R + J \times R + K \times R$, i.e., $\boldsymbol{x}$ represents the vectorized loading matrices stacked together, then the objective function $f(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ in (6) and (7) can be thought of as a scalar-valued cost function $f(\boldsymbol{x}): \mathbb{R}^N \rightarrow \mathbb{R}$. Therefore, the solutions can be obtained using any optimization-based approach, which typically requires computation of the gradients. The partial gradient of the objective function f with respect to the loading matrix $\boldsymbol{A}$, without any regularization, is:

$$\nabla_{\boldsymbol{A}} f = -\boldsymbol{X}_{(1)}(\boldsymbol{C} \odot \boldsymbol{B}) + \boldsymbol{A}\,(\boldsymbol{C}^T \boldsymbol{C} * \boldsymbol{B}^T \boldsymbol{B}) \qquad (10)$$

and like-wise for $\boldsymbol{B}$ and $\boldsymbol{C}$. Proof and detailed derivations are given in [20]. A gradient-based search on the unregularized cost function will not produce a unique solution because all solutions in the form of $\{\eta_1 \boldsymbol{A}, \eta_2 \boldsymbol{B}, \eta_3 \boldsymbol{C}\}$ with $\eta_1 \eta_2 \eta_3 = 1$ are equivalent[9]. Therefore, a regularization term needs to be added. This is achieved here using the Tikhonov regularizer:

$$\widehat{\boldsymbol{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \frac{1}{2} \left\| \boldsymbol{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^T \right\|_F^2 + \frac{\mu_1}{2} \|\boldsymbol{A}\|_F^2 \qquad (11)$$

and similarly for $\boldsymbol{B}$ and $\boldsymbol{C}$, which encourages the norms of the factors to be equally spread. In this regularized case, the corresponding gradient becomes[20]:

$$\nabla_{\boldsymbol{A}} f = -\boldsymbol{X}_{(1)}(\boldsymbol{C} \odot \boldsymbol{B}) + \boldsymbol{A}\,(\boldsymbol{C}^T \boldsymbol{C} * \boldsymbol{B}^T \boldsymbol{B}) + \mu_1 \boldsymbol{A} \qquad (12)$$

and like-wise for $\boldsymbol{B}$ and $\boldsymbol{C}$.

ALGORITHM III: NADAM

```
Algorithm Nadam ($f(\boldsymbol{x})$, $\boldsymbol{x}_0$, $\alpha$, $\beta_1$, $\beta_2$, $\epsilon$)
    $m_0 \leftarrow 0, v_0 \leftarrow 0, t \leftarrow 0$
    Initialize $\alpha$, $\beta_1$, $\beta_2$, $\epsilon$ using default values if not provided
    While not converged
        $t \leftarrow t + 1$
        $g_t \leftarrow \nabla_{\mathbf{x}} f(\boldsymbol{x}_{t-1})$
        $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$
        $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t^2$
        $\widehat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t/(1 - \beta_1^t)$
        $\widehat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t/(1 - \beta_2^t)$
        $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_{t-1} - \frac{\alpha}{\sqrt{\widehat{v}_t} + \epsilon}\left(\beta_1 \widehat{\boldsymbol{m}}_t + \frac{(1-\beta_1)\boldsymbol{g}_t}{1-\beta_1^t}\right)$
    End While
    Return $\boldsymbol{x}_t$
End Algorithm
```

Adam[21] is a popular first-order solvers used in the deep learning community[22]. The superior performance of Adam was achieved by using momentum-based acceleration together with an adaptive learning rate. Recently, Dozat[18] described a modified algorithm, Nadam, in which Nesterov acceleration is incorporated into Adam. The update rules for Nadam are shown in Algorithm III, where $f(\boldsymbol{x})$ is a scalar-valued objective function of $\boldsymbol{x}$, $\boldsymbol{x}_0$ is the initial starting point, α is the global learning rate, β_1 and β_2 are the decay rates for the first moment $\boldsymbol{m}_t$ and second moment $\boldsymbol{v}_t$, respectively, and ϵ is a small constant avoiding division by zero. The default values for these parameters are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ per [21]. Detailed derivations are shown in [18,22].

2.5 BrainSync

Functional MRI time series from two different subjects, or even two different sessions, are often not directly comparable. This is clearly the case for resting fMRI studies in which spontaneous activity varies over subjects and time. Even in event-related studies, brain activity can vary due to differing latencies in response and variations in underlying brain states at the time a stimulus is given. But to perform group analysis or decomposition based directly on temporal data, temporal synchronization across subjects is necessary. We recently developed a synchronization technique for fMRI data called BrainSync that addresses this problem[19]. We summarize the methodology below.

BrainSync assumes that the fMRI data between any two subjects or sessions have been mapped onto a tessellated representation of the mid-cortical layer of the cortex and non-rigidly aligned and resampled onto a common mesh. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be the matrices representing the cortically mapped fMRI data for two subjects, each of size $T \times V$, where T is the number of time points and V is the number of vertices with $V \gg T$, which is typically true in fMRI data. BrainSync finds an orthogonal transform $\boldsymbol{O}^S$ that minimize the overall squared error:

$$\boldsymbol{O}^S = \underset{\boldsymbol{O} \in \boldsymbol{O}(T)}{\operatorname{argmin}} \|\boldsymbol{X} - \boldsymbol{O}\boldsymbol{Y}\|^2 \qquad (13)$$

where $\boldsymbol{O}(T)$ represents the group of $T \times T$ orthogonal matrices. The problem is well-posed given the fact that $V \gg T$ and can be solved using the Kabsch algorithm[23]. The BrainSync algorithm is shown in Algorithm IV, where $\operatorname{svd}(\boldsymbol{X})$ represents the singular value decomposition of a matrix $\boldsymbol{X}$. A group version of BrainSync is described in an accompanying paper in this proceeding[24].

ALGORITHM IV: BRAINSYNC

```
Algorithm BrainSync ($\boldsymbol{X}$, $\boldsymbol{Y}$)
    $\boldsymbol{U\Sigma V} = \operatorname{svd}(\boldsymbol{X}\boldsymbol{Y}^T)$
    $\boldsymbol{O}^S = \boldsymbol{U}\boldsymbol{V}^T$
    Return $\boldsymbol{O}^S$
End Algorithm
```

3. MATERIALS AND METHODS

3.1 NSRCPD

We have developed a Nesterov accelerated scalable and robust sequential CP decomposition (NSRCPD) algorithm which incorporates Nadam into the SRSCPD framework[16,17]. As with SRSCPD, for each rank, NSRCPD uses warm initializations from lower rank solutions to improve its scalability and robustness. However, unlike SRSCPD, where for each rank the solutions were obtained using the ALS algorithm, NSRCPD uses Nadam to update all modes simultaneously. It has been shown that the optimization-based CP algorithms are more robust than ALS but at the expense of a higher computational cost[9,10], making them expensive to scale to large datasets. Incorporating Nadam into the SRSCPD framework allows us to further improve the robustness relative to ALS without sacrificing scalability.

The full NSRCPD algorithm is shown in Algorithm V. It is built on the SRSCPD framework (Algorithm II). We use CP-ALS (Algorithm I) to solve any rank-1 decomposition, including the first rank-1 approximation to the data tensor and all of the following rank-1 fits to the current residue. We use Nadam (Algorithm III) to solve the main decomposition problem at each rank from 2 to the desired maximum rank R with warm initializations $\{\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{C}^*\}$, where f in line 7 is the Tikhonov regularized objective with gradient with respect to $\boldsymbol{A}$ shown in (12), with similar forms for $\boldsymbol{B}$ and $\boldsymbol{C}$. Note that the components returned from the ALS algorithm have unit norms and the normalization factors are stored in $\boldsymbol{\lambda}$ while Nadam optimizes all (non-normalized) components simultaneously. Therefore, in order to have Nadam start from the correct warm initialization point, we normalize and then re-scale the components before and after the Nadam procedure as shown in lines 8 and 6 respectively.

3.2 Simulation

We simulated third-order tensors $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{20\times10\times8}$ of ranks $R = 1, \dots 10$ from the outer product of factors randomly sampled from a standard normal distribution. We then added Gaussian white noise to the simulated tensor $\boldsymbol{\mathcal{X}}$ with a SNR of 2. We performed a CP decomposition with desired rank R on $\boldsymbol{\mathcal{X}}$ using the ALS algorithm, the original SRSPCD algorithm (with the ALS solver internally), and the NSRCPD algorithm. For a fair comparison, we generated and used the same random initializations for all three algorithms. We assessed the quality of the solutions using the averaged congruence product (ACP)[25] (we have previously shown that this metric is superior to the Frobenius norm error[17] in quantifying the ability of tensor decompositions to correctly identify multiple components in a low rank tensor decomposition). ACP is a measure of correlation between components defined as

$$\mathrm{ACP} = \max_{\mathbf{P}} \mathrm{tr}((\boldsymbol{A}^T\widehat{\boldsymbol{A}}) * (\boldsymbol{B}^T\widehat{\boldsymbol{B}}) * (\boldsymbol{C}^T\widehat{\boldsymbol{C}})\ \boldsymbol{P}) \tag{14}$$

where $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are the column-wise normalized ground truth loading matrices, $\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{B}}, \widehat{\boldsymbol{C}}$ their estimated counterparts, $\boldsymbol{P}$ is a permutation matrix accounting for the ambiguity of the ordering of the solutions[26] and $\mathrm{tr}(\boldsymbol{X})$ indicates the trace of $\boldsymbol{X}$. We evaluated the ACP of the solutions for each of the three methods as a function of the rank R. For each rank R we ran 100 Monte Carlo trials and the corresponding boxplots were generated.

ALGORITHM V: NSRCPD

```
s   Algorithm NSRCPD ($\boldsymbol{\mathcal{X}}$, $R$)
1       $\boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{c}^1, \lambda^1 \leftarrow$ CP-ALS ($\boldsymbol{\mathcal{X}}$, 1)
2       $\boldsymbol{\mathcal{X}}_{res} \leftarrow \boldsymbol{\mathcal{X}} -$ Tensor_Recon ($\boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{c}^1, \lambda^1$)
3       $\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}', \lambda' \leftarrow$ CP-ALS ($\boldsymbol{\mathcal{X}}_{res}$, 1)
4       $\boldsymbol{A}^* \leftarrow [\boldsymbol{a}^1\ \boldsymbol{a}']$; $\boldsymbol{B}^* \leftarrow [\boldsymbol{b}^1\ \boldsymbol{b}']$; $\boldsymbol{C}^* \leftarrow [\boldsymbol{c}^1\ \boldsymbol{c}']$; $\boldsymbol{\lambda}^* \leftarrow \begin{bmatrix}\lambda^1 \\ \lambda'\end{bmatrix}$
5       For $r = 2, 3, \dots, R$
6           Scale the $i$th components of $\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{C}^*$ by $\sqrt[3]{\boldsymbol{\lambda}_i^*}$
7           $\boldsymbol{A}^r, \boldsymbol{B}^r, \boldsymbol{C}^r \leftarrow$ Nadam ($f$, $\{\boldsymbol{A}^*, \boldsymbol{B}^*, \boldsymbol{C}^*\}$)
8           Normalize the $i$th components of $\boldsymbol{A}^r, \boldsymbol{B}^r, \boldsymbol{C}^r$ and store the norm product into $\boldsymbol{\lambda}_i^r$
9           $\boldsymbol{\mathcal{X}}_{res} \leftarrow \boldsymbol{\mathcal{X}} -$ Tensor_Recon ($\boldsymbol{A}^r, \boldsymbol{B}^r, \boldsymbol{C}^r, \boldsymbol{\lambda}^r$)
10          $\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}', \lambda' \leftarrow$ CP-ALS ($\boldsymbol{\mathcal{X}}_{res}$, 1)
11          $\boldsymbol{A}^* \leftarrow [\boldsymbol{A}^r\ \boldsymbol{a}']$; $\boldsymbol{B}^* \leftarrow [\boldsymbol{B}^r\ \boldsymbol{b}']$; $\boldsymbol{C}^* \leftarrow [\boldsymbol{C}^r\ \boldsymbol{c}']$; $\boldsymbol{\lambda}^* \leftarrow \begin{bmatrix}\boldsymbol{\lambda}^r \\ \lambda'\end{bmatrix}$
12      End For
13      Return a set of solutions $\{\boldsymbol{a}^1, \boldsymbol{b}^1, \boldsymbol{c}^1, \lambda^1\}, \{\boldsymbol{A}^2, \boldsymbol{B}^2, \boldsymbol{C}^2, \boldsymbol{\lambda}^2\}, \dots, \{\boldsymbol{A}^R, \boldsymbol{B}^R, \boldsymbol{C}^R, \boldsymbol{\lambda}^R\}$
e   End Algorithm
```

3.3 In-vivo motor tfMRI data

40 subjects (2 sessions for each subject) of minimally processed motor tfMRI data[27], represented in the grayordinate format[28], from the Human Connectome Project (HCP)[29] were used. The cortical data on each hemisphere were further downsampled, by a factor of 3, onto an approximately $11K$ tessellated surface for computational tractability. Each session was represented as a $V \times T$ matrix, where $V \approx 22K$ is the number of vertices on both hemispheres and $T = 284$ is the number of time points. Note that the order of presentation of the cues (hand/foot/tongue) varied across subjects and sessions. We therefore applied the BrainSync algorithm to align all sessions of tfMRI datasets to the first session of the first subject (this reference subject is 100307 given by HCP by default). The temporally aligned tfMRI data were then combined in a third-order data tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{V\times T\times S}$, where $S = 80$ is the total number of subjects (40) by sessions (2).

Analogous to PCA in matrix cases, we performed a greedy CP decomposition[20] to the tensor $\boldsymbol{\mathcal{X}}$ to reduce its rank to 20. Specifically, we recursively fit a rank-1 component to the data tensor and then subtracted this from the residual data tensor until we had found 20 components in total. Let $\boldsymbol{a}_i \in \mathbb{R}^V, \boldsymbol{b}_i \in \mathbb{R}^T, \boldsymbol{c}_i \in \mathbb{R}^S$ be the ith normalized spatial, temporal and session component found by the greedy CP decomposition and λ_i be the corresponding norm. Then we reconstructed the rank-reduced tensor as $\boldsymbol{\mathcal{Y}} = \sum_{i=1}^{20} \lambda_i\, \boldsymbol{a}_i \circ \boldsymbol{b}_i \circ \boldsymbol{c}_i$. This dimensionality reduction step was necessary to improve the SNR of the data. Next, we applied the NSRCPD algorithm to the rank-reduced tensor $\boldsymbol{\mathcal{Y}}$ to extract brain networks with a desired rank of 20 and a non-negativity constraint on the session mode since we assumed each subject could either participate or not participate in a network but could not negatively contribute to the network. Specifically, we used (11) for the spatial and temporal mode and the following objective function for the session mode within the Nadam update

$$\widehat{\boldsymbol{C}} = \underset{\mathbf{C}}{\operatorname{argmin}} \frac{1}{2}\left\|\boldsymbol{X}_{(3)} - \boldsymbol{C}(\boldsymbol{B}\odot\boldsymbol{A})^T\right\|_F^2 + \frac{\mu}{2}\|\boldsymbol{C}\|_F^2, \qquad \text{s.t. } \boldsymbol{C} \succcurlyeq 0 \tag{15}$$

where $\succcurlyeq$ represents the element-wise inequality, the regularization parameter μ was chosen to be 0.001 empirically based on the data for all three modes. Finally, the identified components were manually inspected and combined if they were recognized as sub-networks of a known network.

4. RESULTS

4.1 Simulation

Fig. 1 shows the boxplots of the ACP over 100 trials as a function of the rank R for each of the three methods: ALS, SRSCPD with ALS solver, and NSRCPD. When R is small, they perform almost equally well. However, as R increases, NSRCPD outperforms the SRSCPD with ALS solver as well as the original ALS algorithm by a margin that increases with rank, indicating robustness of the NSRCPD algorithm.

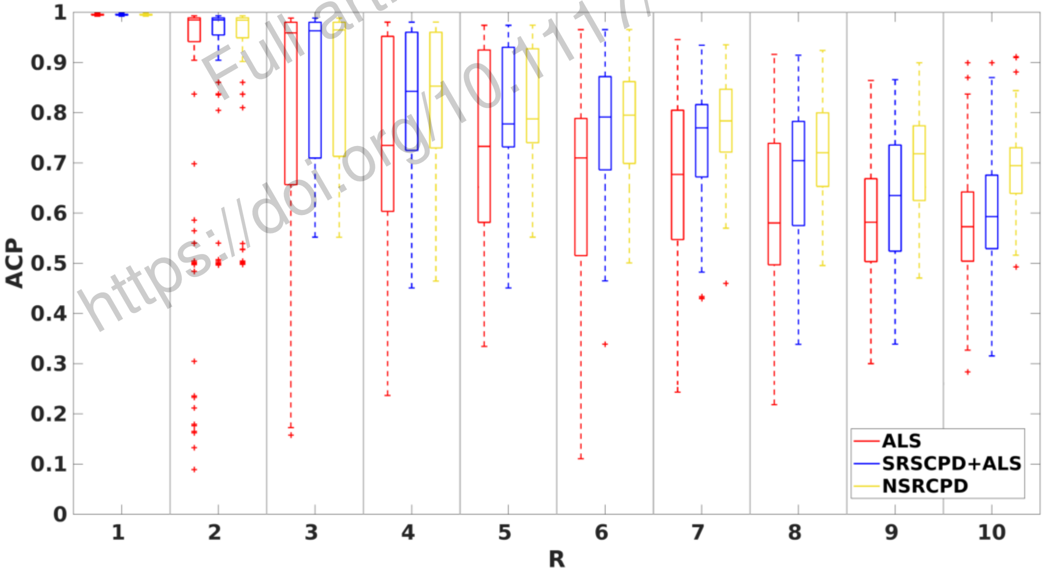

Figure 1: Simulation results. Boxplots of ACP over 100 Monte Carlo trials are shown as a function of rank R.

4.2 In-vivo motor task fMRI data

Fig. 2 shows 9 brain networks identified by the NSRCPD method. For each network, the left column shows the spatial map and the right column shows the dynamic temporal variations overlaid with the color-coded task design blocks (see caption and legend) for the reference subject.

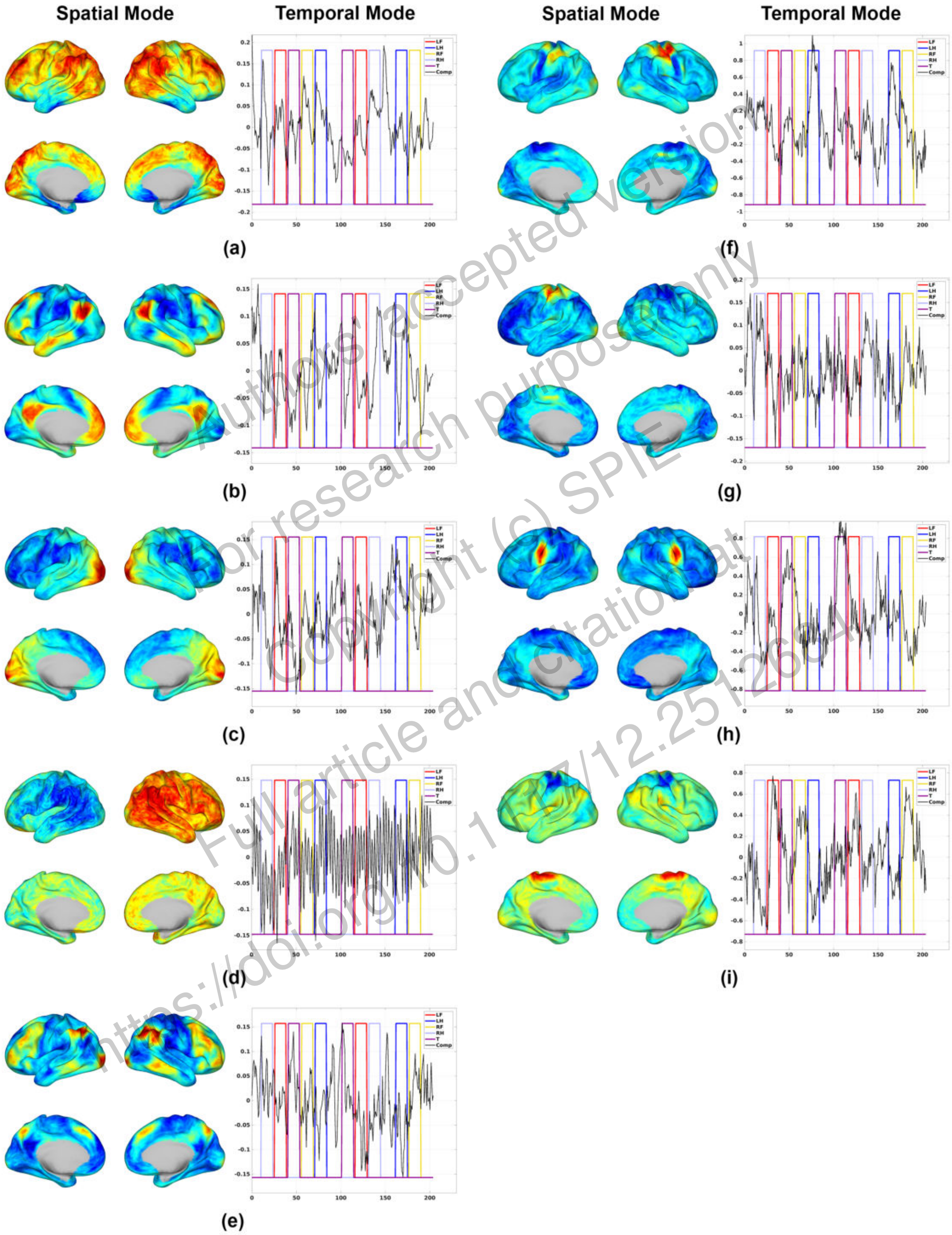

Figure 2: NSRCPD results on motor tfMRI data. For each component, the spatial map is shown on the left with the temporal dynamics on the right. (a) Frontal-parietal attentional control network (FPACN); (b) Default mode network (DMN); (c) Visual network (VN); (d) Respiration activity (Resp); (e) Executive control network (ECN); (f) Left hand (LH); (g) Right hand (RH); (h) Tongue (T); (i) Both feet (F).

The temporal dynamics of (a), (c) and (e) show clear peaks at the beginning of each task block, which visually correlate well with the cues for each task, with a short delay consistent with the effects of the hemodynamic response function. The spatial pattern of these networks indicates a fronto-parietal attentional control network (FPACN)[30] for (a); a visual network (VN) for (c) and an executive control network (ECN)[31] for (e), all reflecting the subjects' responses to the task cues.

The spatial map for (b) shows a very conventional and typical DMN. The DMN was first known as a task-negative network[32] and in fact a strong negative correlation between the temporal mode of (b) and the task blocks can be clearly observed (dips within task blocks and peaks in between task blocks). Note that it is the synchronization of spontaneous (non-task-related) activity across subjects using BrainSync that allows us to identify these components from the third-order tensor in addition to the task-related networks.

Fig. 2 (d) shows a spatially global, temporally relatively fast (~0.3Hz) and non-task-related activity, suggesting that it may represent a residual respiration (Resp) effect common (again after BrainSync synchronization) across subjects. The last 4 networks in Fig. 2 (f) – (i) correspond to clear motor activity for left hand (LH), right hand (RH), tongue (T) and both feet (F), that can be verified by inspection of their spatial modes. However, individual networks for left foot and right foot were not separated.

5. CONCLUSION

Using NSRCPD with BrainSync, we identified 9 spatially overlapped and temporally correlated common networks across multiple subjects: seven task-related (related to visual cues, attentional and executive control, and motor activity) networks, the DMN, and one corresponding to physiological noise (respiration) in tfMRI data. Although we did not use any prior information regarding the task designs, our results not only replicated the task timing, but also showed expected differences in the temporal dynamics of FPACN, DMN, ECN, visual and motor networks. Further, the use of BrainSync synchronization allows us to extract the DMN, and the effects of it on the motor task, jointly across subjects.

ACKNOWLEDGMENT

This work was supported in part by the National Institutes of Health under grants R01-NS074980, R01-EB026299 and R01-NS089212.

REFERENCES

[1] Friston, K. J., "Functional and effective connectivity: A review," Brain Connect. **1**(1), 13–36 (2011).

[2] Esposito, F., Scarabino, T., Hyvarinen, A., Himberg, J., Formisano, E., Comani, S., Tedeschi, G., Goebel, R., Seifritz, E. and Di Salle, F., "Independent component analysis of fMRI group studies by self-organizing clustering," Neuroimage **25**(1), 193–205 (2005).

[3] Calhoun, V. D., Adali, T., Pearlson, G. D. and Pekar, J. J., "A method for making group inferences from functional MRI data using independent component analysis," Hum. Brain Mapp. **14**(3), 140–151 (2001).

[4] Guo, Y. and Pagnoni, G., "A unified framework for group independent component analysis for multi-subject fMRI data," Neuroimage **42**(3), 1078–1093 (2008).

[5] Schmithorst, V. J. and Holland, S. K., "Comparison of Three Methods for Generating Group Statistical Inferences from Independent Component Analysis of Functional Magnetic Resonance Imaging Data," J. Magn. Reson. Imaging **19**(3), 365–368 (2004).

[6] Svensén, M., Kruggel, F. and Benali, H., "ICA of fMRI group study data," Neuroimage **16**(3), 551–563 (2002).

[7] Calhoun, V. D., Liu, J. and Adali, T., "A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data.," Neuroimage **45**(1), S163–S172 (2009).

[8] Karahanoğlu, F. I. and Van De Ville, D., "Transient brain activity disentangles fMRI resting-state dynamics in terms of spatially and temporally overlapping networks," Nat. Commun. **6**, 7751 (2015).

[9] Kolda, T. G. and Bader, B. W., "Tensor Decompositions and Applications," SIAM Rev. **51**(3), 455–500 (2009).

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

[11] Andersen, A. H. and Rayens, W. S., "Structure-seeking multilinear methods for the analysis of fMRI data," Neuroimage **22**(2), 728–739 (2004).

[12] Beckmann, C. F. and Smith, S. M., "Tensorial extensions of independent component analysis for multisubject FMRI

analysis," Neuroimage **25**(1), 294–311 (2005).

[13] Sen, B. and Parhi, K. K., "Extraction of common task signals and spatial maps from group fMRI using a PARAFAC-based tensor decomposition technique," IEEE Int. Conf. Acoust. Speech Signal Process., 1113–1117 (2017).

[14] Stegeman, A., "Comparing Independent Component Analysis and the Parafac model for artificial multi-subject fMRI data" (2007).

[15] Helwig, N. E. and Hong, S., "A critique of Tensor Probabilistic Independent Component Analysis: Implications and recommendations for multi-subject fMRI data analysis," J. Neurosci. Methods **213**(2), 263–273 (2013).

[16] Li, J., Mosher, J. C., Nair, D. R., Gonzalez-Martinez, J. and Leahy, R. M., "Robust tensor decomposition of resting brain networks in stereotactic EEG," Signals, Syst. Comput. 2017 51th Asilomar Conf., IEEE, Pacific Grove (2017).

[17] Li, J., Haldar, J., Mosher, J. C., Nair, D., Gonzalez-Martinez, J. and Leahy, R. M., "Scalable and Robust Tensor Decomposition of Spontaneous Stereotactic EEG Data," IEEE Trans. Biomed. Eng. (2018).

[18] Dozat, T., "Incorporating Nesterov Momentum into Adam" (2016).

[19] Joshi, A. A., Chong, M., Li, J., Choi, S. and Leahy, R. M., "Are you thinking what I'm thinking? Synchronization of resting fMRI time-series across subjects," Neuroimage **172**, 740–752 (2018).

[20] Acar, E., Dunlavy, D. M. and Kolda, T. G., "A scalable optimization approach for fitting canonical tensor decompositions," J. Chemom. **25**(2), 67–86 (2011).

[21] Kingma, D. P. and Ba, J., "Adam: A method for stochastic optimization," arXiv Prepr. arXiv1412.6980 (2014).

[22] Ruder, S., "An overview of gradient descent optimization algorithms," arXiv Prepr. arXiv1609.04747 (2016).

[23] Kabsch, W., "A solution for the best rotation to relate two sets of vectors," Acta Crystallogr. Sect. A **32**(5), 922–923 (1976).

[24] Akrami, H., Joshi, A. A., Li, J. and Leahy, R. M., "Group-wise alignment of resting fMRI in space and time," Med. Imaging 2019 Image Process. **10949**, 10949103 (2019).

[25] Tomasi, G. and Bro, R., "PARAFAC and missing values," Chemom. Intell. Lab. Syst. **75**(2), 163–180 (2005).

[26] Harshman, R. A., "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multimodal factor analysis," UCLA Work. Pap. Phonetics **16**(10), 1–84 (1970).

[27] Barch, D. M., Burgess, G. C., Harms, M. P., Petersen, S. E., Schlaggar, B. L., Corbetta, M., Glasser, M. F., Curtiss, S., Dixit, S., Feldt, C., Nolan, D., Bryant, E., Hartley, T., Footer, O., Bjork, J. M., Poldrack, R., Smith, S., Johansen-Berg, H., Snyder, A. Z., et al., "Function in the human connectome: Task-fMRI and individual differences in behavior," Neuroimage **80**, 169–189 (2013).

[28] Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., Van Essen, D. C. and Jenkinson, M., "The minimal preprocessing pipelines for the Human Connectome Project," Neuroimage **80**, 105–124 (2013).

[29] Van Essen, D. C., Smith, S. M., Barch, D. M., Behrens, T. E. J., Yacoub, E. and Ugurbil, K., "The WU-Minn Human Connectome Project: An overview," Neuroimage **80**, 62–79 (2013).

[30] Hopfinger, J. B., Buonocore, M. H. and Mangun, G. R., "The neural mechanisms of top-down attentional control," Nat. Neurosci. **3**(3), 284–291 (2000).

[31] Seeley, W. W., Menon, V., Schatzberg, A. F., Keller, J., Glover, G. H., Kenna, H., Reiss, A. L. and Greicius, M. D., "Dissociable Intrinsic Connectivity Networks for Salience Processing and Executive Control," J. Neurosci. **27**(9), 2349–2356 (2007).

[32] Raichle, M. E., MacLeod, A. M., Snyder, A. Z., Powers, W. J., Gusnard, D. A. and Shulman, G. L., "A default mode of brain function," Proc. Natl. Acad. Sci. **98**(2), 676–682 (2001).