Article:
Song, L., Zhou, S. orcid.org/0000-0002-8069-2814 and Lu, H. (2022) Direct ICA on data tensor via random matrix modeling. Signal Processing, 196. 108508. ISSN 0165-1684

https://doi.org/10.1016/j.sigpro.2022.108508

Direct ICA on Data Tensor via Random Matrix Modeling

Liyan Song[a,b], Shuo Zhou[c], Haiping Lu[c,*]

[a] *Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China.*
[b] *Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, Shenzhen 518055, China.*
[c] *Department of Computer Science, University of Sheffield, Sheffield S1 4DP, United Kingdom.*

Abstract

Independent component analysis (ICA) is a fundamental method for blind source separation (BSS). Classical ICA takes data matrix input formed by vector data. This paper focuses on ICA for BSS with third-order data tensor input formed by matrix data, such as 2D images. Two approaches exist for this problem. The first reshapes each matrix into a vector to apply classical ICA, with structural information lost. The second approach unfolds a data tensor into a data matrix along different modes to perform classical ICA mode-wise, which partially preserves structures but has strong or ill BSS assumptions. This paper proposes a third approach via RAndom Matrix ICA (RAMICA) modeling. RAMICA works on data tensor directly, without vectorization or unfolding, and preserves row or column structures under more general BSS assumptions. We develop the RAMICA model, algorithm, and related theories via defining new statistics for random matrices and new procedures for whitening and independent component estimation. We study the identifiability, higher-order extension, and relationships with existing methods. Experiments on both synthetic and real data show superior BSS performance of RAMICA over competing methods and

[*] Corresponding author. Address: The University of Sheffield, 211 Portobello, Sheffield, S1 4DP, United Kingdom. Tel: +44 (0) 114 222 1853. Email: h.lu@sheffield.ac.uk

Email addresses: `songly@sustech.edu.cn` (Liyan Song), `shuo.zhou@sheffield.ac.uk` (Shuo Zhou), `h.lu@sheffield.ac.uk` (Haiping Lu)

offer insights on the trade-offs between different factors.

Keywords: Blind source separation, independent component analysis, tensor.

1. Introduction

Blind source separation (BSS) [1] assumes that *observed data* are generated from unknown *latent sources*, and aims to recover these sources from the observations only. Independent component analysis (ICA) [2] is a fundamental method for BSS. Classical ICA methods treat P sources as random variables, and assume they are mutually independent and linearly mixed to produce M observations $\{\underline{x_m}\}$ as:

$$\underline{x_m} = a_{m1}\underline{s_1} + \cdots + a_{mP}\underline{s_P}, \tag{1}$$

where random variables are underlined, $\underline{x_m}$ is the mth observation, $\underline{s_1}, \cdots, \underline{s_P}$ are the latent sources named as *independent components* (ICs), and $a_{m1}, \cdots, a_{mP}$ are mixing coefficients. Stacking random variables into vectors, the *vector-matrix notation* of (1) is:

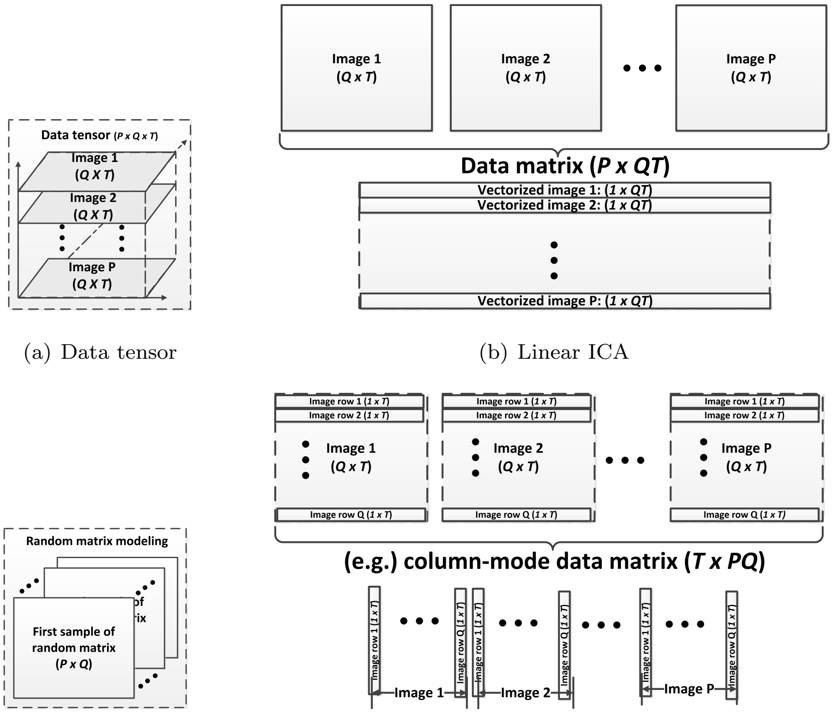
$$\underline{\mathbf{x}} = \mathbf{A}\underline{\mathbf{s}} \in \mathbb{R}^M, \tag{2}$$

where $\underline{\mathbf{x}} = [\underline{x_1}, \cdots, \underline{x_M}]^\top$ and $\underline{\mathbf{s}} = [\underline{s_1}, \cdots, \underline{s_P}]^\top$ are observation (mixture) and source random vectors, and $\mathbf{A} = \{a_{mp}\} \in \mathbb{R}^{M\times P}$ is the unknown constant matrix, namely *mixing matrix*. It is usually assumed that $M = P$ so that $\mathbf{A} \in \mathbb{R}^{P\times P}$ is square, which we follow hereafter. Considering T available samples of observations $\underline{\mathbf{x}}$ (sources $\underline{\mathbf{s}}$), align all samples in column to form the *data matrix* $\mathbf{X}$ (*source matrix* $\mathbf{S}$). We can then write the ICA model in $\mathbf{X}$ and $\mathbf{S}$ as:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \in \mathbb{R}^{P\times T}. \tag{3}$$

In other words, $\mathbf{X}$ contains T samples $\mathbf{x}(1), \cdots, \mathbf{x}(T)$ of *random vector* $\underline{\mathbf{x}}$ in column, and $\mathbf{S}$ contains T samples[1] of random vector $\underline{\mathbf{s}}$ in column. ICA aims

[1] In this paper, the term ‘sample’ refers to samples of random variables, and the number of observations (P) is the number of data samples/examples in typical machine learning terminology.



(a) Data tensor

(b) Linear ICA

(c) *Proposed* ICA

(d) Mode-wise ICA

Figure 1: Given a $P \times Q \times T$ data tensor in (a), classical linear ICA vectorizes each image mixture into a long row vector and stacks them into a $P \times QT$ data matrix as in (b) for source recovery. Mode-wise ICA models, such as DTICA [4, 5] and MMICA [6], unfold the data tensor along an image (column or row) mode, to obtain a $T \times PQ$ (or $Q \times PT$) data matrix as in (d), on which to apply classical ICA (twice). The proposed RAMICA model deals with the data tensor directly *without* vectorization or unfolding as in (c).

to estimate the source matrix $\mathbf{S}$ and the mixing matrix $\mathbf{A}$ simultaneously with data matrix $\mathbf{X}$ as the only input [3].

Real-world data are often matrices or even higher-order tensors, such as multichannel electroencephalography (EEG) signals, images, videos, or social networks [7]. In such cases, all observed data of a particular problem form a *data tensor* $\boldsymbol{\mathcal{X}}$ with their natural multidimensional structures. All observed matrix data form a third-order data tensor, while all observed Nth-order tensor form an $(N+1)$th-order data tensor. We can view such data tensor as *stacking* all data along a particular dimension (the first ‘mode’ by default). This paper focuses on *ICA for such data tensor input*. For convenience of discussion, we consider stacking P *2D images* of size $Q \times T$ into a third-order *data tensor* of size $P \times Q \times T$. Figure 1(a) shows the stacking of images into a data tensor.

There are two existing ICA approaches for data tensor input. The first *classical approach* is a linear one, which *vectorizes* (*reshapes*) images into vectors so that we can apply classical ICA methods such as FastICA [8], JADE [9], or Infomax [10]. Equivalently, we can view this *vectorization* process as *unfolding* the data tensor of size $P \times Q \times T$ into a data matrix of size $P \times QT$ along the first mode, as shown in Fig. 1(b). The sources can be recovered as vectors first and then folded (reshaped) back to images (matrices). However, the vectorization breaks the original structure and leads to high-dimensional vectors, imposing significant theoretical and computational challenges. There are some other ICA variations [11, 12, 13] where more complicated data inputs are considered (e.g., images with known forming factors), but images are still represented as vectors and they degenerate to classical ICA under basic (simplified) settings [6].

The second approach is to do *mode-wise (linear) ICA* to partially preserve structural information and explore computational benefits (as shown in Fig. 1(d)), where the data tensor is unfolded along each of the original image dimensions (image row or column) into data matrices of size $Q \times PT$ and $T \times PQ$ to apply classical ICA. This brings an additional issue of mixing modeling, with two ways summarized below.

The first way of modeling in mode-wise ICA is a *multilinear-mixing model* as in directional tensor ICA (DTICA) [4, 5], where an image mixture is generated by *one source matrix* and *two mode-wise mixing matrices* (one for each mode). DTICA forms row and column directional images by shifting the rows/columns and then estimates two mixing matrices by mode-wise FastICA. Similarly, Virta *et al.* [14, 15] generalize JADE and FOBI [16] to mode-wise versions for data tensor input, using *multiple mode-wise mixing matrices* to mix *one latent tensor*. Such multilinear-mixing models have an inherent limitation. They *cannot do BSS* to recover multiple matrix/tensor sources since they model a single source/tensor matrix only, which is hard to interpret in a BSS context.

The second way of modeling in mode-wise ICA is a *multilinear-source model* as in multilinear mode-wise ICA (MMICA) [6], where an image mixture is generated by two mode-wise source matrices via a multilinear mixing matrix. This model resembles the mixing model (1) more closely. However, it assumes the sources are rank one and constructed by mode-wise source matrices. Thus, although MMICA can do BSS, it can only recover rank-one sources due to its strong assumptions.

This paper proposes a new, third approach for ICA with data tensor input. Different from all existing works, we aim to recover general (not only rank-one) sources as the first approach can do while preserving multidimensional structure as the second approach. We do so by working with the original data tensor *directly* as shown in Fig. 1(c), *without vectorization or unfolding.* We develop our new method by considering *random matrix* in modeling so we name it RAndom Matrix ICA (RAMICA). In the RAMICA model, a random matrix consists of multiple random vectors. It assumes observed image data are generated by mixing source images with row-wise or column-wise structures. We make three major contributions in developing this RAMICA model and deriving the RAMICA algorithm:

1. With random matrix modeling, we propose RAMICA, a new ICA approach for data tensor input that deals with data tensor directly without vectorization or unfolding. Thus, RAMICA preserves multidimensional structure and can recover general source matrices in BSS.
2. We define new statistics of random matrix including covariance matrix, white matrix, independence, and higher-order cumulants, as the basis for developing RAMICA for data tensor input.
3. We formulate the RAMICA objective function by introducing a *whitening* step for a respective *whitened* random matrix and a new *cumulant operator* for random matrices. Then we derive a new RAMICA algorithm to recover source matrices with the Jacobi method.

The rest of the paper is organized as follows. Section 2 offers a brief overview of notation and technical preliminaries. Section 3 formulates the proposed RAMICA model and Section 4 derives the proposed RAMICA algorithms. Section 5 discusses identifiablity and uniqueness, higher-order extensions, relationship to other methods, computational and memory cost, and the significance. Section 6 reports our numerical studies on both synthetic and real data. Finally, Section 7 draws conclusions.

2. Preliminaries

2.1. Notation

Table 1 lists the symbols used for easy reference. Constants are in normal fonts, and random variables are underlined. Scalars, vectors, matrices, and tensors are denoted by lowercase, lowercase boldface, uppercase boldface, and bold calligraphic letters e.g., x, $\mathbf{x}$, $\mathbf{X}$, and $\boldsymbol{\mathcal{X}}$ for constant variables, and $\underline{x}$, $\underline{\mathbf{x}}$, $\underline{\mathbf{X}}$, and $\underline{\boldsymbol{\mathcal{X}}}$ for random variables.

2.2. Random vector and cumulants

A *random vector* is a vector of random variables $\underline{\mathbf{x}} = [\underline{x_1}, \cdots, \underline{x_Q}]^\top$. Its expectation is a vector $E(\underline{\mathbf{x}}) = [E(\underline{x_1}), \cdots, E(\underline{x_Q})]^\top$. Its covariance matrix is

$$\Sigma(\underline{\mathbf{x}}) = \begin{bmatrix} \sigma^2(\underline{x_1}) & \cdots & cov(\underline{x_1}, \underline{x_Q}) \\ \vdots & & \vdots \\ cov(\underline{x_Q}, \underline{x_1}) & \cdots & \sigma^2(\underline{x_Q}) \end{bmatrix}, \tag{4}$$

where $cov(\underline{x_i}, \underline{x_j})$ is the covariance of $\underline{x_i}$ and $\underline{x_j}$, and $\sigma^2(\underline{x_i})$ is the variance of $\underline{x_i}$. If $\underline{x_1}, \cdots, \underline{x_Q}$ are mutually independent, $\underline{\mathbf{x}}$ has independent components, and $\Sigma(\underline{\mathbf{x}})$ becomes diagonal. $E(\underline{\mathbf{x}})$ and $\Sigma(\underline{\mathbf{x}})$ are the first and second order *cumulants* of $\underline{\mathbf{x}}$. Higher-order cumulants with order $r \geq 3$ are denoted by $[\mathcal{Q}_r(\underline{\mathbf{x}})]_{i_1 \cdots i_r}$ or $cum(\underline{x_{i_1}}, \cdots, \underline{x_{i_r}})$, where $i_1, \cdots, i_r$ are the mode-wise indices.

Cumulants of a random vector $\underline{\mathbf{x}}$ have the following four properties:

- ***symmetry***: $[\mathcal{Q}_r(\underline{\mathbf{x}})]_{i_1 \cdots i_r} = [\mathcal{Q}_r(\underline{\mathbf{x}})]_{i_{\sigma(1)} \cdots i_{\sigma(r)}}$ for any permutation $\sigma(\cdot)$;

Table 1: Symbols with descriptions and types, sorted alphabetically.

Symbol	**Description**	**Type**
$\mathbf{A}$	Mixing matrix	constant matrix
$cum(\cdots)$	Conventional rth order cumulant tensor of random vector $\underline{\mathbf{x}}$	statistic operator
$\widetilde{cum}(\cdots)$	New rth order cumulant tensor of random matrix $\underline{\mathbf{X}}$	statistic operator
$\mathscr{F}_{\underline{\mathbf{X}}}(\cdot)$	New cumulant operator	statistic operator
$\mathcal{K}_r(\underline{\mathbf{x}_p})$	$\widetilde{cum}(\underline{\mathbf{x}_p}, \cdots, \underline{\mathbf{x}_p})$, a special $\widetilde{\mathcal{Q}_r}(\underline{\mathbf{X}})$	statistic operator
M	Number of observations	constant scalar
P	Number of sources	constant scalar
$\widetilde{\mathcal{Q}_r}(\underline{\mathbf{X}})$	$\widetilde{cum}(\underline{\mathbf{x}_{i_1}}, \cdots, \underline{\mathbf{x}_{i_r}})$	statistic operator
$\mathcal{Q}_r(\underline{\mathbf{x}})$	$cum(\underline{x_{i_1}}, \cdots, \underline{x_{i_r}})$	statistic operator
r	Order of higher-order cumulants	constant scalar
$\boldsymbol{\mathcal{S}}$	Source data tensor	constant tensor
$\mathbf{S}$	Source data matrix	constant matrix
$\underline{\mathbf{S}}$	Source random matrix	random matrix
$\mathbf{S}(t)$	tth sample of $\underline{\mathbf{S}}$	constant matrix
$\underline{\mathbf{s}}$	Source random vector	random vector
$\underline{\mathbf{s}_p}$	pth source vector	random vector
$\underline{s_p}$	pth source scalar	random scalar
$\mathbf{s}(t)$	tth sample of $\underline{\mathbf{s}}$	constant vector
$\widetilde{\Sigma}(\underline{\mathbf{X}})$	New covariance matrix of random matrix $\underline{\mathbf{X}}$	statistic operator
$\Sigma(\underline{\mathbf{x}})$	Conventional covariance matrix of random vector $\underline{\mathbf{x}}$	statistic operator
T	Number of random samples	constant scalar
$\mathbf{U}$	Whitened mixing matrix	constant matrix
$\mathbf{W}$	Whitening matrix	constant matrix
$\boldsymbol{\mathcal{X}}$	Observation data tensor	constant tensor
$\mathbf{X}$	Observation data matrix	constant matrix
$\underline{\mathbf{X}}$	Original observation random matrix	random matrix
$\widehat{\mathbf{X}}$	Whitened observation random matrix	random matrix
$\mathbf{X}(t)$	tth sample of $\underline{\mathbf{X}}$	constant matrix
$\underline{\mathbf{x}}$	Observation random vector	random vector
$\underline{\mathbf{x}_m}$	mth observation vector	random vector
$\underline{x_m}$	mth observation scalar	random scalar
$\mathbf{x}(t)$	tth sample of $\underline{\mathbf{x}}$	constant vector

- ***linearity***: $cum(\underline{x_1}, \cdots, \underline{x_i}+\underline{y}, \cdots, \underline{x_r}) = cum(\underline{x_1}, \cdots, \underline{x_i}, \cdots, \underline{x_r})+cum(\underline{x_1}, \cdots, \underline{y}, \cdots, \underline{x_r})$ and $cum(\underline{x_1}, \cdots, \alpha\, \underline{x_i}, \cdots, \underline{x_r}) = \alpha\, cum(\underline{x_1}, \cdots, \underline{x_i}, \cdots, \underline{x_r})$ for any random variable $\underline{y}$ and constant α;

- ***independence***: if $\exists p, q \in \{1, \cdots, r\}$ where $\underline{x_{i_p}}$ and $\underline{x_{i_q}}$ are independent, then $[\mathcal{Q}_r(\underline{\mathbf{x}})]_{i_1 \cdots i_r} = 0$;

- ***vanishing Gaussian***: if $\underline{\mathbf{x}}$ is Gaussian, $[\mathcal{Q}_r(\underline{\mathbf{x}})]_{i_1 \cdots i_r} = 0$ for any order $r \geq 3$.

*2.3. ICA steps and tensor mode-*1 *product*

ICA has three standard steps.

1. *Centering*: remove the first-order statistics from the data by shifting the sample mean to the origin.
2. *Whitening*: remove the second-order statistics from the data to obtain *whitened* variables. Here, the second-order statistics and the whitening process need to be redefined for the proposed model.
3. *IC Estimation*: use higher-order statistics of the data to estimate ICs. It is the core step of ICA, and different methods do it differently.

The *mode*-1 *product* of a third-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$, denoted by $\mathcal{A} \times_1 \mathbf{U}$, is a tensor with entries [17, 18]:

$$(\mathcal{A} \times_1 \mathbf{U})_{j_1 i_2 i_3} = \sum_{i_1} \mathcal{A}_{i_1 i_2 i_3} \cdot \mathbf{U}_{j_1 i_1}. \tag{5}$$

3. A Random Matrix Model for ICA

3.1. RAMICA model and assumptions

First, we view ICA from a random matrix perspective. A *random matrix* is a matrix of random variables $\underline{\mathbf{X}} = [\underline{x_{ij}}] \in \mathbb{R}^{P \times Q}$ and its expectation matrix is $E(\underline{\mathbf{X}}) = [E(\underline{x_{ij}})] \in \mathbb{R}^{P \times Q}$. Traditionally, its covariance matrix is defined as

$$\Sigma(\underline{\mathbf{X}}) := \Sigma(vec(\underline{\mathbf{X}})) \in \mathbb{R}^{(PQ) \times (PQ)}, \tag{6}$$

where $vec(\cdot)$ is the vectorization operator [19, 20, 21], and its higher-order cumulants are defined via vectorizing the random matrix to the random vector. Thus, the independence of random matrix involves the independence of all elements. Equivalently, these definitions treat random matrix as its vectorized version, without considering any structural information.

Here, we follow (1), (2), and (3) in classical ICA. Instead of (1), we have M mixtures $\underline{\mathbf{x}_1}, \cdots, \underline{\mathbf{x}_M} \in \mathbb{R}^Q$ of P random vector sources (ICs) as:

$$\underline{\mathbf{x}_m} = a_{m1}\underline{\mathbf{s}_1} + \cdots + a_{mP}\underline{\mathbf{s}_P}, \tag{7}$$

where $\underline{\mathbf{x}_m}$ is the mth mixture random vector, $\underline{\mathbf{s}_1}, \cdots, \underline{\mathbf{s}_P}$ are the independent source random vectors, and $a_{m1}, \cdots, a_{mP}$ are mixing coefficients. Again, we assume $M = P$. Stacking P random vectors $\{\underline{\mathbf{x}_m}\}$ and $\{\underline{\mathbf{s}_p}\}$ into random matrices along the first mode respectively, we obtain the *observation (mixture) matrix* $\underline{\mathbf{X}} = \left[\underline{\mathbf{x}_1}, \cdots, \underline{\mathbf{x}_P}\right]^\top \in \mathbb{R}^{P \times Q}$, and the source matrix $\underline{\mathbf{S}} = \left[\underline{\mathbf{s}_1}, \cdots, \underline{\mathbf{s}_P}\right]^\top \in \mathbb{R}^{P \times Q}$. We have the full matrix notation version of (7) instead of (2) below:

$$\underline{\mathbf{X}} = \mathbf{A}\underline{\mathbf{S}}, \tag{8}$$

where $\mathbf{A} \in \mathbb{R}^{P \times P}$ is the *mixing matrix* and assumed to be full-rank. With T samples of such *random matrices*, we form the data tensor $\boldsymbol{\mathcal{X}}$ and source tensor $\boldsymbol{\mathcal{S}}$ of size $P \times Q \times T$. Then we can write RAMICA model in data tensors $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{S}}$ instead of (3) as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{S}} \times_1 \mathbf{A}, \tag{9}$$

where $\times_1$ denotes the mode-1 multiplication of a tensor by a matrix as defined in (5). We can view (9) as a *partial* Tucker decomposition [22].

The RAMICA objective is to estimate the source tensor $\boldsymbol{\mathcal{S}}$ and mixing matrix $\mathbf{A}$ given the data tensor $\boldsymbol{\mathcal{X}}$ only. Figure 1(c) shows this RAMICA model for data tensor, which is viewed along the third mode (instead of the first mode of stacking) as T samples of the random matrix $\underline{\mathbf{X}}$, i.e. $\mathbf{X}(1), \cdots, \mathbf{X}(T)$.

Remark. Note that (7), (8), and (9) in RAMICA correspond to (1), (2), and (3) in classical ICA, respectively. When $Q = 1$, the RAMICA model degenerates to the classical ICA model. Thus, RAMICA is a natural second-order generalization of classical ICA, without vectorization or unfolding. A key difference between (7) and (1) is that while $\underline{\mathbf{s}_1}, \cdots, \underline{\mathbf{s}_P}$ are independent, *the components of each source random vector* $\underline{\boldsymbol{s}_p}$ *can be dependent and encode structural infor-*

mation. This allows structural information to be better preserved in RAMICA than in classical ICA.

To derive RAMICA algorithm, we make three assumptions analogous to classical ICA [23]:

1. The expectation of mixture or source matrix is zero, i.e. $E(\underline{\mathbf{S}}) = E(\underline{\mathbf{X}}) = \mathbf{0}_{P\times Q}$. A *centering* process will be performed if the input data are not centered.
2. The covariance matrix of the source random matrix is an identity matrix. Note the definition in (6) is equivalent to vectorizing $\underline{\mathbf{X}}$, leading to the first, vectorization-based ICA approach. Thus, we need *new definitions of related statistics for random matrix* to embody structural information.
3. At most one independent random vector from $\{\underline{\mathbf{s}_p}\}$ has multivariate Gaussian distribution.

3.2. New statistics of random matrix

We define new statistics for RAMICA via *tensor contraction.* On its basis, we subsequently define the *white* random matrix and the whitened RAMICA model.

Definition 1. *The **contracted covariance matrix** of a zero-mean random matrix* $\underline{\boldsymbol{X}} = \left[\underline{\boldsymbol{x}_1}, \cdots, \underline{\boldsymbol{x}_P}\right]^\top \in \mathbb{R}^{P\times Q}$ *is defined as:*

$$\widetilde{\Sigma}(\underline{\boldsymbol{X}}) = \frac{1}{Q} E[\underline{\boldsymbol{X}}\,\underline{\boldsymbol{X}}^\top] \in \mathbb{R}^{P\times P}. \tag{10}$$

Each element of $\widetilde{\Sigma}(\underline{\boldsymbol{X}})$ *is the covariance of two corresponding random vectors:*

$$[\widetilde{\Sigma}(\underline{\boldsymbol{X}})]_{ij} = \widetilde{cov}(\underline{\boldsymbol{x}_i}, \underline{\boldsymbol{x}_j}) = \frac{1}{Q}\sum_{q=1}^{Q} cov(\underline{x_{iq}}, \underline{x_{jq}}), \tag{11}$$

where $cov(\cdot,\cdot)$ *is the conventional covariance.*

Definition 2. *Given a zero-mean random matrix* $\underline{\boldsymbol{X}} = \left[\underline{\boldsymbol{x}_1}, \cdots, \underline{\boldsymbol{x}_P}\right]^\top \in \mathbb{R}^{P\times Q}$*, its **contracted cumulant** of order* $r \geq 3$ *denoted by* $[\widetilde{\mathcal{Q}_r}(\underline{\boldsymbol{X}})]_{i_1\cdots i_r}$ *or* $\widetilde{cum}(\underline{\boldsymbol{x}_{i_1}}, \cdots, \underline{\boldsymbol{x}_{i_r}})$ *is:*

$$[\widetilde{\mathcal{Q}_r}(\underline{\boldsymbol{X}})]_{i_1\cdots i_r} = \frac{1}{Q}\sum_{q=1}^{Q} cum(\underline{x_{i_1q}}, \cdots, \underline{x_{i_rq}}), \tag{12}$$

where $i_1, \cdots, i_r \in \{1, \cdots, P\}$, and $cum(\cdot)$ is the conventional cumulant of a random vector. In particular, we denote a special case

$$\mathcal{K}_r(\underline{\boldsymbol{x}_p}) = \widetilde{cum}(\underbrace{\underline{\boldsymbol{x}_p}, \cdots, \underline{\boldsymbol{x}_p}}_{(r \text{ times})}). \tag{13}$$

Lemma 1. *The newly defined cumulants for random matrix $\underline{\boldsymbol{X}} = \left[\underline{\boldsymbol{x}_1}, \cdots, \underline{\boldsymbol{x}_P}\right]^\top \in \mathbb{R}^{P\times Q}$ satisfies the properties of conventional cumulants for random vectors: (1) symmetry, (2) linearity, (3) independence, and (4) vanishing Gaussian.*

Proof. This lemma can be obtained by using the corresponding properties of cumulants for random vector in Sec. 2.2 multiple times and do a final average operation as denoted in (12). □

Definition 3. *A random matrix $\underline{\boldsymbol{X}} = \left[\underline{\boldsymbol{x}_1}, \cdots, \underline{\boldsymbol{x}_P}\right]^\top \in \mathbb{R}^{P\times Q}$ is* ***independent*** *if the conditional distribution of $\underline{\boldsymbol{x}_i}$ given $\underline{\boldsymbol{x}_j} = \boldsymbol{x}$ does not depend on $\underline{\boldsymbol{x}_j}$ (i.e. $\underline{\boldsymbol{x}_i}$ and $\underline{\boldsymbol{x}_j}$ are mutually independent):*

$$f_{\underline{\boldsymbol{x}_i}|\underline{\boldsymbol{x}_j}}(\underline{\boldsymbol{x}_i}|\underline{\boldsymbol{x}_j}) = f_{\underline{\boldsymbol{x}_i}}(\underline{\boldsymbol{x}_i}). \tag{14}$$

Using these new statistics defined for random matrix, we can derive our RAMICA algorithm.

3.3. Alternative forms of the RAMICA model

Given P images of size $R \times C$ (R rows and C columns), we can form a data tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{P\times Q\times T}$ in (9) in two ways:

- *Row-wise* RAMICA (**rRAMICA**): each image row is treated as a random vector in $\mathbb{R}^C$ and all the R rows are considered R samples of this random vector so $Q = C$ and $T = R$.

- *Column-wise* RAMICA (**cRAMICA**): each image column is treated as a random vector in $\mathbb{R}^R$ and all the C columns are considered C samples of this random vector so $Q = R$ and $T = C$.

Thus, rRAMICA and cRAMICA consider row and column structural information of image data, respectively. Real-world image data often have structures in both rows and columns so both rRAMICA and cRAMICA can be effective. For example, in the real image BSS experiments in Sec. 6.2, we know *both* row and column structures present in most real images and the results in Fig. 2(b) indeed show that *both* versions lead to improvement in the noiseless and low noise settings.

These two versions of RAMICA capture different aspects of data, which can offer alternative explanations of the data. If we use RAMICA as a feature extractor, we can combine both versions to provide complementary features.

If we have to choose one from the two, a *model selection* problem arises. We can consider this choice as a hyper-parameter and determine it based on prior knowledge of the data, cross validation, or other statistical model selection strategies, e.g., picking the one with a higher kurtosis.

For convenience of discussion, when we talk about RAMICA, we refer to **cRAMICA** unless specified explicitly.

4. The RAMICA Algorithm

Given zero-mean input, the RAMICA algorithm has two steps, i.e., whitening and IC estimation.

4.1. RAMICA whitening

Definition 4. *A random matrix* $\underline{\boldsymbol{X}} = \left[\underline{\boldsymbol{x}_1}, \cdots, \underline{\boldsymbol{x}_P}\right]^\top \in \mathbb{R}^{P\times Q}$ *is* ***contracted white*** *if its covariance matrix is an identity matrix:*

$$\widetilde{\Sigma}(\underline{\boldsymbol{X}}) = \boldsymbol{I}_{P\times P}. \tag{15}$$

Therefore, the ***contracted whitening*** step of the RAMICA model (8) is to find a matrix $\mathbf{W}$ such that $\widetilde{\Sigma}(\mathbf{W}\underline{\mathbf{X}}) = \mathbf{I}_{P\times P}$.

Theorem 1. *For the RAMICA model* (8)*, let* $\boldsymbol{W}$ *denote any inverse square root of* $\widetilde{\Sigma}(\underline{\boldsymbol{X}})$*, i.e.* $[\widetilde{\Sigma}(\underline{\boldsymbol{X}})]^{-1/2}$*. Then* $\boldsymbol{W}$ *is the whitening matrix and* $\boldsymbol{W}\underline{\boldsymbol{X}}$ *is white. Literally, any square-root of the covariance matrix is a whitening matrix.*

Proof. Perform singular value decomposition (SVD) on the mixing matrix $\mathbf{A} = \mathbf{EDV}$, where $\mathbf{E}$ and $\mathbf{V}$ are orthogonal, and $\mathbf{D}$ is diagonal. Compute the covariance matrix of $\underline{\mathbf{X}}$ using the above decomposition:

$$\widetilde{\Sigma}(\underline{\mathbf{X}}) = \frac{1}{Q}E[\mathbf{EDVSS}^\top\mathbf{V}^\top\mathbf{DE}^\top] = \mathbf{ED}^2\mathbf{E}^\top. \tag{16}$$

The second equality is due to *Assumption 2* of RAMICA and orthogonality of $\mathbf{V}$. Denote by $\mathbf{M} \in \mathbb{R}^{P\times P}$ any *orthogonal* matrix, then according to [24], $\mathbf{W}$ can be written as:

$$\mathbf{W} = \mathbf{MD}^{-1}\mathbf{E}^\top \in \mathbb{R}^{P\times P}. \tag{17}$$

Next, we calculate the covariance matrix of $\mathbf{W}\underline{\mathbf{X}}$ as

$$\begin{aligned}
\widetilde{\Sigma}(\mathbf{W}\underline{\mathbf{X}}) &= \frac{1}{Q}E[\mathbf{MD}^{-1}\mathbf{E}^T\underline{\mathbf{X}}\underline{\mathbf{X}}^\top\mathbf{ED}^{-1}\mathbf{M}^\top] \\
&= \mathbf{MD}^{-1}\mathbf{E}^\top[\Sigma(\underline{\mathbf{X}})]\mathbf{ED}^{-1}\mathbf{M}^\top \\
&= \mathbf{MD}^{-1}\mathbf{E}^\top[\mathbf{ED}^2\mathbf{E}^\top]\mathbf{ED}^{-1}\mathbf{M}^\top \\
&= \mathbf{I}_{P\times P}.
\end{aligned} \tag{18}$$

Therefore, we have proved that $\mathbf{W}\underline{\mathbf{X}}$ is white and $\mathbf{W}$ is the whitening matrix. □

Similar to classical ICA, we can conduct eigenvalue decomposition on the covariance matrix as

$$\widetilde{\Sigma}(\underline{\mathbf{X}}) = \mathbf{E\Lambda E}^\top, \tag{19}$$

where $\mathbf{E} = [\mathbf{e}_1, \cdots, \mathbf{e}_P]$ has the unit-norm eigenvectors as columns, and the diagonal matrix $\mathbf{\Lambda}$ consists of the eigenvalues. According to Theorem 1, the whitening matrix is

$$\mathbf{W} = \mathbf{\Lambda}^{-1/2}\mathbf{E}^\top. \tag{20}$$

After RAMICA whitening, we have $\mathbf{W}\underline{\mathbf{X}} = \mathbf{WA}\underline{\mathbf{S}} = \mathbf{U}\underline{\mathbf{S}}$. Since $\widetilde{\Sigma}(\mathbf{W}\underline{\mathbf{X}}) = \mathbf{I}_{P\times P} = \mathbf{U}\widetilde{\Sigma}(\underline{\mathbf{S}})\mathbf{U}^\top = \mathbf{UU}^\top$, $\mathbf{U}$ is orthogonal. The *whitened RAMICA* model can be rewritten as

$$\widehat{\mathbf{X}} = \mathbf{U}\underline{\mathbf{S}}, \tag{21}$$

where $\widehat{\mathbf{X}}$ is the *whitened random matrix*, $\mathbf{U}$ is the *whitened mixing matrix.*

Lemma 2. *Given the whitened RAMICA model* (21)*, the fourth-order cumulants of (whitened)* $\widehat{\boldsymbol{X}}$ *satisfy:*

$$[\widetilde{\mathcal{Q}_4}(\widehat{\boldsymbol{X}})]_{ijkl} = \sum_p u_{ip}u_{jp}u_{kp}u_{lp}\mathcal{K}_4(\underline{\boldsymbol{s}_p}),\ \forall i,j,k,l \in \{1,\cdots,P\}. \tag{22}$$

Proof. Calculate the fourth-order cumulants according to Lemma 1. We have:

$$\begin{aligned}[\widetilde{\mathcal{Q}_4}(\widehat{\mathbf{X}})]_{ijkl} &= cum(\sum_p u_{ip}\underline{\mathbf{s}_p}, \sum_{p'} u_{jp'}\underline{\mathbf{s}_{p'}}, \sum_q u_{kq}\underline{\mathbf{s}_q}, \sum_{q'} u_{lq'}\underline{\mathbf{s}_{q'}}) \\ &= \sum_{p,p',q,q'} u_{ip}u_{jp'}u_{kq}u_{lq'}\mathrm{cum}(\underline{\mathbf{s}_p}, \underline{\mathbf{s}_{p'}}, \underline{\mathbf{s}_q}, \underline{\mathbf{s}_{q'}}).\end{aligned} \tag{23}$$

Due to the independence of $\{\widehat{\mathbf{s}}_p\}$, only those products with $p = p' = q = q'$ are nonzero. Therefore, we have proved that

$$[\widetilde{\mathcal{Q}_4}(\widehat{\mathbf{X}})]_{ijkl} = \sum_p u_{ip}u_{jp}u_{kp}u_{lp}\mathcal{K}_4(\underline{\mathbf{s}_p}). \tag{24}$$

□

4.2. RAMICA IC estimation

It is difficult to recover sources directly from cumulants. Instead, we can convert the BSS problem to a matrix diagonalization problem as in JADE [9], via a cumulant-based mapping of the whitened mixing matrix $\mathbf{U}$. To do this, we firstly define a new cumulant operator.

Definition 5. *Given the whitened RAMICA model* (21)*, the* ***cumulant operator*** $\mathscr{F}_{\widehat{\mathbf{X}}}$ *is defined by the fourth-order cumulant tensor* $\widetilde{\mathcal{Q}_4}(\widehat{\boldsymbol{X}})$ *of random matrix* $\widehat{\boldsymbol{X}}$ *as:*

$$\mathscr{F}_{\widehat{\mathbf{X}}} : \boldsymbol{M} \in \mathbb{R}^{P\times P} \mapsto [\mathscr{F}_{\widehat{\mathbf{X}}}(\boldsymbol{M})]_{ij} = \sum_{k,l} m_{kl}[\widetilde{\mathcal{Q}_4}(\widehat{\boldsymbol{X}})]_{ijkl} \in \mathbb{R}^{P\times P}. \tag{25}$$

Lemma 3. *Given the whitened RAMICA model* (21)*, the cumulant operator* $\mathscr{F}_{\widehat{\mathbf{X}}}$ *satisfies:*

$$[\mathscr{F}_{\widehat{\mathbf{X}}}(\boldsymbol{M})]_{ij} = \sum_p u_{ip}u_{jp}\mathcal{K}_4(\underline{\boldsymbol{s}_p})\sum_{k,l} m_{kl}u_{kp}u_{lp},\ \forall i,j \in \{1,\cdots,P\}. \tag{26}$$

Proof. This lemma can be proved by substituting cumulants $[\mathcal{Q}_4(\widehat{\mathbf{X}})]_{ijkl}$ in (25) by the derived (22) of Lemma 2. □

Theorem 2. *Given the whitened model* (21)*, the matrix* $[\boldsymbol{U}^\top \mathscr{F}(\boldsymbol{M})\boldsymbol{U}]$ *is diagonal for* $\forall \boldsymbol{M} \in \mathbb{R}^{P\times P}$*.*

Proof. According to matrix multiplication rules and Lemma 3, we know for $\forall i, j \in \{1, \cdots, P\}$:

$$\begin{aligned}
[\mathbf{U}^\top \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M})\mathbf{U}]_{ij} &= \sum_{p,q} u_{pi}u_{qj}[\mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M})]_{pq} \\
&= \sum_{p,q} u_{pi}u_{qj} \sum_m u_{pm}u_{qm}\mathcal{K}_4(\underline{\mathbf{s}_m}) \sum_{k,l} m_{kl}u_{km}u_{lm} \\
&= \sum_m \mathcal{K}_4(\underline{\mathbf{s}_m}) \sum_{k,l} m_{kl}u_{km}u_{lm} \sum_p u_{pi}u_{pm} \sum_q u_{qj}u_{qm}.
\end{aligned} \tag{27}$$

Since $\mathbf{U}$ is orthogonal, we have

$$[\mathbf{U}^\top \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M})\mathbf{U}]_{ij} = \sum_m \mathcal{K}_4(\underline{\mathbf{s}_m}) \sum_{k,l} m_{kl}u_{km}u_{lm}\delta_{im}\delta_{jm}. \tag{28}$$

Only those products with $i = j = m$ are nonzero. Thus, we have

$$[\mathbf{U}^\top \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M})\mathbf{U}]_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ \mathcal{K}_4(\underline{\mathbf{s}_i}) \sum_{k,l} m_{kl}u_{ki}u_{li}, & \text{if } i = j. \end{cases} \tag{29}$$

Therefore, we have proved the diagonality of $\mathbf{U}^\top \mathscr{F}(\mathbf{M})\mathbf{U}$. □

4.3. RAMICA objective

Theorem 2 reveals the connection between the whitened RAMICA model (21) and $\mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M})$. We can take a set of matrices $\mathbf{M}_i$ and make the matrix set $\{\mathbf{U}^\top \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M}_i)\mathbf{U}\}$ as diagonal as possible for BSS. In practice, they cannot be made exactly diagonal because the model does not hold exactly and there are sampling errors. In fact, the diagonality of a symmetric matrix $\mathbf{Q} = \mathbf{U}^\top \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M})\mathbf{U}$ can be measured by the sum of the squares of off-diagonal entries: $\sum_{i\neq j} q_{ij}^2$ [25]. Since for a given matrix $\mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M})$, the square sum over all elements of the matrix is preserved under an orthogonal transformation, minimizing the sum of squares of off-diagonal elements is equivalent to maximizing the sum of squares of diagonal elements [26]. We formulate our objective function based on this property. For a set of basis matrix $\{\mathbf{M}_i \in \mathbb{R}^{P\times P}\}$, we maximize the following objective function with respect to orthogonal matrix $\mathbf{U}$:

$$\sum_i ||\mathrm{diag}(\mathbf{U}^\top \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M}_i)\mathbf{U})||^2, \tag{30}$$

where $||\mathrm{diag}(\cdot)||^2$ denotes the sum of squares of the diagonal elements. In this paper, we use the standard basis of $\mathbb{R}^{P\times P}$, i.e. $\{\mathbf{E}^{ij} = \mathbf{e}_i\mathbf{e}_j^\top\}_{i,j=1}^P$, which can reduce computational cost significantly.

Algorithm 1 Random Matrix Modeling ICA (RAMICA)

```
Input: a zero-mean data tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{P\times Q\times T}$. The last dimension contains the samples.
Contracted whitening: viewing $\boldsymbol{\mathcal{X}}$ as $T$ samples of $\underline{\mathbf{X}}$, compute the sample estimate of the whitening matrix $\mathbf{W}$ according to Theorem 1, and compute the whitened data tensor: $\boldsymbol{\mathcal{X}} \times_1 \mathbf{W}$.
Construct the contracted cumulant tensor by (12).
Construct cumulant matrices $\{\mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{E}^{ij})\}$ according to (25) for $\forall i, j \in \{1, \cdots P\}$.
Align the above matrices to form the matrix $\mathbf{M} = [\mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{E}^{11}), \cdots, \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{E}^{PP})]$.
Apply Jacobi method to $\mathbf{M}$ to get rotation matrices.
Get $\mathbf{U}^{-1}$ by multiplying the rotation matrices.
Compute $\mathbf{A}^{-1} = \mathbf{U}^{-1}\mathbf{W}$, and $\mathbf{A} = (\mathbf{A}^{-1})^{-1}$.
Compute $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{A}^{-1}$.
Output: mixing matrix $\mathbf{A}$ and source tensor $\boldsymbol{\mathcal{S}}$.
```

4.4. RAMICA optimization

Similar to JADE, we apply Jacobi method to optimize (30) to compute the whitened mixing matrix $\mathbf{U}$. Specifically, to use the Jacobi method, we first align the set of matrices $\{\mathscr{F}(\mathbf{E}^{ij})\}_{i,j=1}^P$ into an extended matrix $\mathbf{M} = [\mathscr{F}(\mathbf{E}^{11}), \cdots, \mathscr{F}(\mathbf{E}^{PP})]$. Then, we apply the Jacobi method on $\mathbf{M}$ to conduct a series of Jacobi rotations, each of which handles two rows and two columns at a time [27]. After this, we obtain $\mathbf{U}^{-1}$ by multiplying the rotation matrices. Subsequently, we get $\mathbf{A}$ from $\mathbf{A}^{-1} = \mathbf{U}^{-1}\mathbf{W}$. The RAMICA algorithm is summarized in Algorithm 1, where we view $\boldsymbol{\mathcal{X}}$ as T samples of $\underline{\mathbf{X}}$ to obtain sample-based estimates of various statistics. The theoretical properties of the newly defined cumulant can be extended to such empirical estimations following [23, §2.7].

5. Discussions

5.1. Identifiability and uniqueness

In our RAMICA formulation, the mixing matrix $\mathbf{A}$ in (8) plays exactly the same role as the mixing matrix $\mathbf{A}$ in (2) of classical ICA. The cumulant operator in Definition 5 maps the fourth-order cumulants of (whitened) mixture $\widehat{\mathbf{X}}$ to a $P \times P$ matrix in (5). This contraction operates on the tensor observation data to reduce the RAMICA problem to JADE. Thus, the RAMICA objective function (30) is similar to the objective function in JADE [9] and theoretical results on JADE can be similarly applied here.

Miettinen *et al.* [28] have proved that the joint diagonalization procedure achieves mixing matrix recovery for the identifiable case of at most one Gaussian source in Theorems 8 and 9 of their paper. These theorems help establish that the same joint diagonalization procedure in our proposed RAMICA can recover the mixing matrix when there is at most one Gaussian source and the contraction operation via $\widetilde{\mathcal{Q}_4}(\cdot)$ does not lead to an ICA instance with all Gaussian sources.

Note however that the mixing matrix can be identified only up to the order and signs (independent components are not ordered as in principal components). Therefore, if $\mathbf{V}^\top \mathscr{F}_{\widehat{\mathbf{X}}}(\mathbf{M}_i)\mathbf{V}$ is diagonal for all $\mathbf{M}$ under the RAMICA formulation, we will get a $\mathbf{V}$ that is equivalent to $\mathbf{U}$ only up to the order (permutation) and signs.

5.2. Higher-order extensions

RAMICA can be extended to higher-order tensors by extending *random matrix modeling* to *random tensor modeling*, and tensor contraction over *one* mode to over *multiple* modes. For instance, given a fourth order tensor $\widetilde{\boldsymbol{\mathcal{X}}}$ of size $P \times Q \times R \times T$ formed by stacking P tensors of size $Q \times R \times T$, the mixing model in (7) can be extended to $M = P$ mixtures $\underline{\mathbf{X}_1}, \cdots, \underline{\mathbf{X}_M} \in \mathbb{R}^{Q \times R}$ of P random matrix sources (ICs) as:

$$\underline{\mathbf{X}_m} = a_{m1}\underline{\mathbf{S}_1} + \cdots + a_{mP}\underline{\mathbf{S}_P}. \tag{31}$$

Respectively, (8) becomes

$$\underline{\boldsymbol{\mathcal{X}}} = \underline{\boldsymbol{\mathcal{S}}} \times_1 \mathbf{A}. \tag{32}$$

We can then view $\widetilde{\boldsymbol{\mathcal{X}}}$ as T samples of *random tensors* with size $P \times Q \times R$. We can subsequently define mode-wise covariance matrices and cumulant for mode 1, now with summation over Q and R instead of just Q. Given P third-order tensors, we will have three (instead of two) ways of forming a fourth-order data tensor: i.e., treating its mode-n slice ($n \in \{2, 3, 4\}$) as a random matrix. Further extensions (e.g., to fifth-order tensor of $P \times Q \times R \times S \times T$) can be similarly formulated.

5.3. Relationship to existing methods

5.3.1. Differences with existing tensor-based ICA

The covariance matrix and cumulant of RAMICA have the sizes of $P \times P$ and P^r respectively, while their linear (vectorization-based) counterparts (of JADE) have much larger sizes of $PQ \times PQ$ and $(PQ)^r$ (typically $r = 4$ in ICA). Thus, they have much smaller computational and memory footprints than its linear counterparts. In terms of source separation capability, RAMICA captures structural information better than classical ICA, and lifts the restriction of rank-one sources in MMICA, while being superior over DTICA's single source assumption that is hard to interpret and unable to do source separation.

5.3.2. Connection to ISA

Independent subspace analysis (ISA) [29, 30], a.k.a. group ICA or subspace ICA, is a generalization of ICA, which assumes some random (scalar) sources are mutually dependent, but the dependencies among the sources of different groups are minimized. In this sense, it shares similarity with RAMICA by viewing these dependent sources as random vector sources. However, the ways of

RAMICA and ISA in modeling and solving the problems are essentially different. ISA arises from the relaxation of ICA assumptions by restricting its mixing matrix, and is often solved by first preprocessing the mixtures by an classical ICA algorithm and then grouping the estimated components with highest dependence. In contrast, RAMICA is a generalization of classical ICA designed from a random matrix formulation with a natural interpretation and our mixing matrix is general, as in the classical ICA.

5.3.3. Connection to IVA

Independent vector analysis (IVA) [31, 32] is another generalization of ICA to multiple datasets. IVA takes K data matrices as input and assumes that the kth data matrix is obtained by linearly mixing rows of the kth source matrix through the kth mixing matrix. Thus, there are K related ICA problems. IVA and RAMICA both take multiple data matrices as the input and make similar independence assumptions. However, IVA considers each data matrix as a dataset but RAMICA considers it as a sample of a dataset. Therefore, IVA estimates K mixing matrices but RAMIC estimates *only one* mixing matrix. RAMICA can be extended to deal with multiple matrix/tensor datasets following IVA.

5.4. Computational and memory cost

The covariance matrix and cumulant of RAMICA have sizes of $P \times P$ and P^r respectively, which are *much smaller* than the sizes of their linear (vectorization-based) counterparts, $PQ \times PQ$ and $(PQ)^r$ respectively. Thus, the covariance matrix and cumulant of RAMICA have much smaller computational and memory cost than its linear counterparts. For example, in the BSS experiments on 256×256 gray-level images (Sec. 6.2), the average running time (in second) over 100 runs with noise level 0.01 was *0.05450* and *0.00132* for JADE and RAMICIA, respectively. This confirms the *superior computational efficiency* of RAMICA over its vectorization-based counterpart.

Moreover, although computing higher-order statistics like cumulants is computationally expensive in general, most of the computations in RAMICA can be *parallelized* to improve the scalability, benefiting from the tensor-based formulation. This is particularly advantageous with the wide availability of parallel computing.

5.5. Potential significance of RAMICA

Direct ICA on data tensor is a technically challenging problem without much success before this work. RAMICA provides theoretical/conceptual contributions in overcoming the technical challenges in a tensor-based ICA without unfolding, including a new random matrix modeling approach for ICA and new statistics for random matrix. This enables further development of robust variants and higher-order extensions, as sketched above. Its connection with ISA discussed above also sheds new light on ICA research. Thus, the significance of such contributions may go beyond ICA/BSS and impact not only ICA, but also more general tensor-based learning (without unfolding) and random-matrix-based machine learning. Although we showed only separation of real images below, RAMICA is promising in solving practical neuroimaging problems such as for EEG and fMRI, where ICA is very popular [33, 34].

6. Numerical Results

In this section, we perform evaluations on BSS with data tensor input for sources having general 2D structures, rather than special structures such as rank one. We compare two versions of RAMICA, i.e. rRAMICA and cRAMICA, against classical ICA methods FastICA,[2] JADE,[3] and Infomax[4] with default settings. The mode-wise ICA with a *multilinear-mixing model* (DTICA) cannot do BSS because it assumes only one source matrix, while the one with a

[2] https://research.ics.aalto.fi/ica/fastica/code/dlcode.shtml

[3] http://bsp.teithe.gr/members/downloads/Jade

[4] https://inc.ucsd.edu/ marni/code.html

multilinear-source model (MMICA) can only recover rank-one sources by design. Therefore, both DTICA and MMICA will fail on this more general setting of BSS with data tensor input.

BSS experiments were conducted on both synthetic and real data. For synthetic data, we generate *column-wise* source random vectors for mixing. With only column structural information synthesized, we expect cRAMICA to perform well whereas rRAMICA not. For real data, we linearly mix natural images and then recover them from their mixtures. Natural images are expected to have both row and column structural information. Hence, both rRAMICA and cRAMICA are expected to recover the sources to some extent. Which one does better could depend on whether row or column structure is stronger.

For BSS performance measurement, we use the popular Amari error [35] calculated over the demixing matrices (i.e., the inverse of the mixing matrices). For convenience, we report Amari error values multiplied by 100 throughout this paper. We report the average performance with standard deviations (std) over *100* repetitions for each experimental setting below.

6.1. Blind source separation on synthetic data

We first study how well RAMICA can recover sources from synthetic data tensor generated according to the column-wise RAMICA model (9).

6.1.1. Data generation

To simulate *column-wise* structural information in each random vector $\underline{\mathbf{s}} \in \mathbb{R}^Q$, only its first component $\underline{s_1}$ is randomly generated, while its other components have the following linear relationships with $\underline{s_1}$:

$$\underline{s_q} = \alpha \underline{s_1} + \beta(q-1), \tag{33}$$

where $\alpha \sim \mathcal{N}(1,1)$ (Gaussian distribution) and $\beta \sim U(0,1)$ (uniform distribution) are randomly generated, and $q \in \{2, \cdots, Q\}$. Thus, $\{\underline{s_1}, \cdots, \underline{s_Q}\}$ are dependent rather than independent and $\underline{\mathbf{s}}$ is a column random vector with column structures. We consider the following four distributions that generate the first component of each source random vector $\underline{\mathbf{s}_p} \in \mathbb{R}^Q$ for $p \in \{1, \cdots, P\}$:

- Psn: Pearson distribution with zero mean, unit variance, unit skewness, and unit kurtosis.
- Stu: Student-t distribution with 5 freedom degrees.
- Exp: Exponential distribution with $\lambda = 1$.
- Lap: Laplace distribution with $\mu = 0$ and $b = 1/\sqrt{2}$.

Following the above generation, T samples of the pth random vector $\underline{\mathbf{s}_p}$ form the pth source image $\mathbf{S}_p = [\mathbf{s}_p(1), \cdots, \mathbf{s}_p(T)] \in \mathbb{R}^{Q \times T}$. Finally, stacking P source images along the first mode, we have the source tensor $\boldsymbol{\mathcal{S}} = [\mathbf{S}_1; \cdots; \mathbf{S}_P] \in \mathbb{R}^{P \times Q \times T}$. Such sources are much more realistic/general than the restricted rank-one sources synthesized in MMICA [6]. Note although rank-one sources can be combined to produce low-rank (or high-rank) sources, there is great indeterminacy.

In generating the mixing matrix $\mathbf{A}$, we need to guarantee its invertibility. We generate $\mathbf{A}$ in three steps: (i) uniformly generate a $P \times P$ matrix with each entry between zero and one; (ii) normalize the generated matrix by column; (iii) add an identity matrix to the one in (ii). With $\boldsymbol{\mathcal{S}}$ and $\mathbf{A}$ generated, we further generate $\boldsymbol{\mathcal{X}}$ according to (9).

6.1.2. Design factors

In simulations, we have the following design factors investigated with several choices:

- $\mathcal{D} \in \{Psn, \boldsymbol{Stu}, Exp, Lap\}$: the distribution used to generate the (first components of) sources.
- $P \in \{2, \mathbf{4}, 8, 16\}$: the number of sources.
- $Q \in \{16, \mathbf{32}, 64, 128\}$: the dimension of the source random vectors.
- $T \in \{16, 32, \mathbf{64}, 128\}$: the number of random samples.

- $\sigma^2 \in \{\mathbf{0}, 0.01, 0.02, \cdots, 0.1, 0.15, 0.2\}$: the Gaussian noise level that is added to the observation as:

  $$\underline{\mathbf{X}} = \mathbf{A}\underline{\mathbf{S}} + \underline{\mathbf{E}}, \tag{34}$$

  where $\underline{\mathbf{E}}$ denotes standard Gaussian noise $vec(\underline{\mathbf{E}}) \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. The default setting is noise-free.

When studying one factor, we vary it with other factors fixed to their default settings in bold above.

6.1.3. Results on synthetic data

Tables 2, 3, 4, 5 report the performance with varying $\mathcal{D}$, P, Q and T, respectively. When varying one factor, default settings were applied for other factors. The effect of each factor are summarized below.

Effect of $\mathcal{D}$. From Table 2, almost all of the methods get the best results on *Exp* but the worst results on *Psn*, indicating *Psn* is more challenging than *Exp*. Among the five ICA methods, cRAMICA consistently achieves the best performance, though this is expected due to the *column-wise* data tensor generation. In particular, cRAMICA improves over JADE (the second best) by 31.4% on average. On the other hand, it is not surprising that rRAMICA gives poorer results. Nonetheless, real-world data often have both row-wise and column-wise structures so both rRAMICA and cRAMICA can be effective and reveal different aspects of data. This will be confirmed in the real data experiments.

Table 2: Effect of the underlying source distribution $\mathcal{D}$ for synthetic BSS. Other factors use default settings. Amari errors are reported and each entry is the mean±std of 100 repetitions. The best (second-best) Amari errors are highlighted in **bold** (underline).

$\mathcal{D}$	Infomax	FastICA	JADE	cRAMICA	rRAMICA
Psn	10.03±1.71	7.07±1.47	6.90±1.61	**5.12±2.11**	7.63±1.23
Stu	9.35±1.48	6.48±1.41	6.33±1.44	**4.45±2.09**	6.88±1.39
Exp	9.04±1.50	6.03±1.22	5.93±1.19	**3.72±1.73**	6.73±1.24
Lap	9.29±1.44	6.37±1.42	6.22±1.37	**4.12±1.82**	6.60±1.19

Table 3: Effect of the number of sources P for synthetic BSS as in Table 2.

P	Infomax	FastICA	JADE	cRAMICA	rRAMICA
2	1.97±0.65	1.51±0.53	1.47±0.54	**0.88±0.77**	1.42±0.33
4	9.35±1.48	6.48±1.41	6.33±1.44	**4.45±2.09**	6.88±1.39
8	36.6±4.29	25.4±5.07	24.5±4.53	**20.9±5.53**	26.7±3.92
16	134.4±10.1	103.2±13.0	99.4±12.4	**92.5±14.9**	106.0±11.2

Effect of P. From Table 3, cRAMICA consistently achieves the best performance and outperforms others by a large margin. For example, cRAMICA outperforms JADE by 29.70% for $P = 4$. rRAMICA is inferior to cRAMICA, but it outperforms Infomax.

Effect of Q. In Table 4, classical ICA methods perform differently with respect to Q. Infomax performs similarly as RAMICA but FastICA and JADE achieve slightly better performance with increasing Q. Such difference could be due to the trade-off between the benefits of having more samples QT and the detriments of more dependent samples. The detriments dominate for Infomax but the benefits dominate for JADE and FastICA.

Table 4: Effect of the dimension of the source random vectors Q for synthetic BSS as in Table 2.

Q	Infomax	FastICA	JADE	cRAMICA	rRAMICA
16	8.80±1.63	6.56±1.72	6.33±1.68	**4.29±2.20**	6.76±1.43
32	9.35±1.48	6.48±1.41	6.33±1.44	**4.45±2.09**	6.88±1.39
64	9.79±1.50	6.37±1.17	6.28±1.28	**4.58±2.09**	6.92±1.29
128	10.09±1.44	6.26±1.12	6.20±1.21	**4.72±2.09**	6.93±1.19

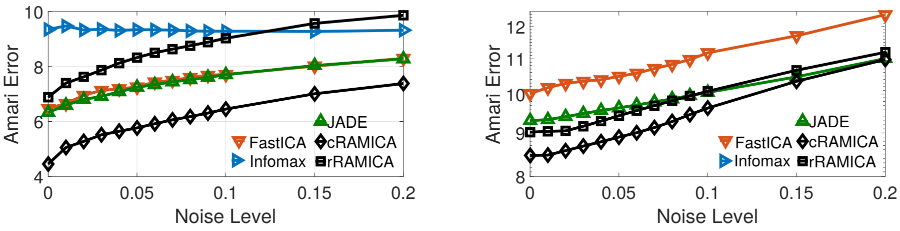
Effect of T. From Table 5, the performance of all of the methods become better with the increasing of T, where the improvement of cRAMICA is the most significant (as shown in the last row in Table 5). The improvements for classical ICA methods are less significant because they have to compensate the detriments of more dependent samples. Though the benefits of larger sample size dominate, the detriments of more dependent samples reduce their improvement

Table 5: Effect of the number of samples T for synthetic BSS as in Table 2. The last row reports the improvement rate when we increase T from 16 to 128.

T	Infomax	FastICA	JADE	cRAMICA	rRAMICA
16	10.81±1.69	7.59±1.66	7.42±1.58	**6.51±2.36**	8.01±1.24
32	9.91±1.39	7.55±1.39	7.13±1.25	**5.71±2.01**	7.59±1.33
64	9.35±1.48	6.48±1.41	6.33±1.44	**4.45±2.09**	6.88±1.39
128	9.08±1.58	5.87±1.13	5.78±1.14	**3.19±1.47**	6.47±0.98
↑	16.00%	22.66%	22.10%	51.00%	19.26%

rate. Comparing the results in Tables 4 and 5, we can also see that T has a larger effect on the performance of RAMICA than Q.

Effect of σ^2. In the last study on synthetic BSS, we examine the sensitivity of these ICA methods with respect to noise as shown in Fig. 2(a). We can see that cRAMICA and rRAMICA have similar sensitivity to noise with JADE and FastICA. It may be because they are all based on the fourth-order cumulants. Infomax is the least sensitive to noise but it performs the worst in most cases. In addition, cRAMICA largely outperforms the others in this experiment.



(a) Synthetic BSS.

(b) Real image BSS.

Figure 2: Effect of the noise level σ^2 for synthetic and real image BSS. Other factors for synthetic BSS use default settings. There are four 256×256 source images in the image BSS experiment. The average Amari errors over 100 repetitions are reported. The Infomax results in the image BSS are higher than the chosen upper limit so they are not visible.

6.2. Blind image separation

We further perform evaluations on real-world image data. Natural image data tend to have structures in both row and column. Thus, both rRAMICA and cRAMICA should work to some extent. Which one performs better will depend on whether row or column structure dominates. Next, we conduct real data experiments to verify this.

6.2.1. Data

Source images are taken from the Caltech256 repository [36]. We selected 4,424 images with strong higher-order statistics from the total 30,607 images for blind image separation experiments. All selected images are resized to a standard size of 256×256 with 256 gray levels so $Q = T = 256$.

6.2.2. Experimental settings

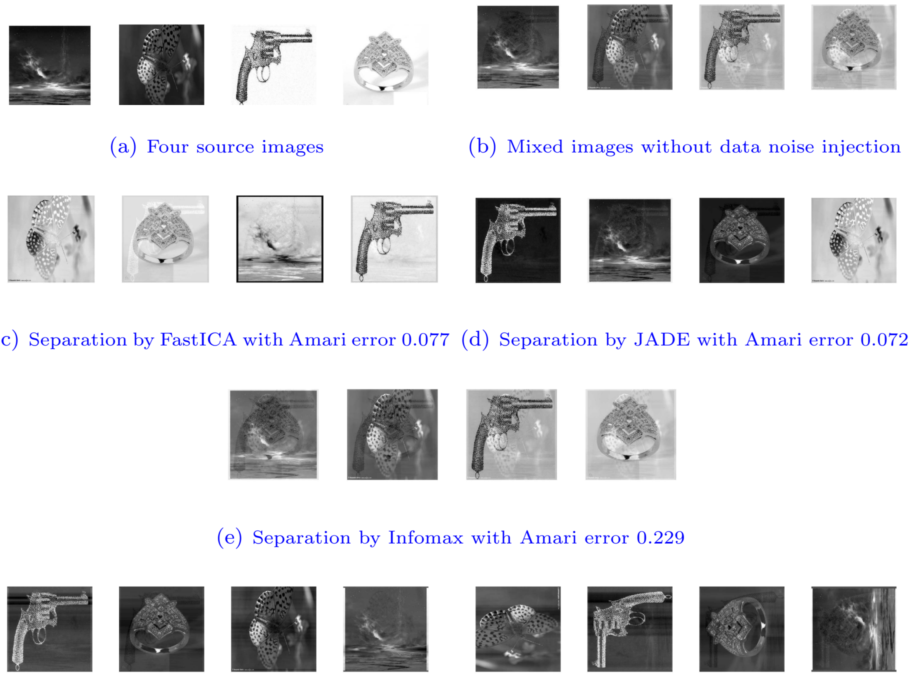
We repeat the following process 100 times:

- randomly select four source images ($P = 4$);
- mix them using a mixing matrix randomly generated in the same way as in the synthetic study to produce four mixture images according to (9) (equivalent to the classical ICA model (3));
- recover the sources from the four mixtures by ICA methods;
- compute the Amari errors accordingly.

In addition, we add noise to the mixing process to do sensitivity study. The average Amari errors are reported.

6.2.3. Image separation results

Figure 3 presents an example of blind image separation. For illustration, we show the four randomly selected source images in Figure 3(a) and their mixtures in Figure 3(b) for which the mixing matrix is generated randomly according to Eq. (9) and they do not suffer from data noise. We also show the performance of



(a) Four source images

(b) Mixed images without data noise injection

(c) Separation by FastICA with Amari error 0.077

(d) Separation by JADE with Amari error 0.072

(e) Separation by Infomax with Amari error 0.229

(f) By cRAMICA with Amari error 0.069

(g) By rRAMICA with Amari error 0.081

Figure 3: An example of the source images, mixed images and separated images by ICA methods for the blind image separation problem.

FastICA, JADE, Infomx, cRAMICA and rRAMICA in Figures 3(c), 3(d), 3(e), 3(f) and 3(g), respectively. We can see that cRAMICA can achieve the best blind image separation with the best (smallest) Amari error, and both FastICA and JADE can also perform good image separation.

Figure 2(b) shows the recovery performance across different noise levels. The results of Infomax are above the chosen upper limit in the figure (for clarity) so they are not visible there. We can see that both rRAMICA and cRAMICA can obtain better BSS performance than the other methods when the noise level is below 0.1. This confirms that real-world images contain both row and column structures and both rRAMICA and cRAMICA have their merits. Furthermore, cRAMICA outperforms the other methods by a *large margin* in the *noise-free* case. And this margin reduces as the noise level increases. The observation that cRAMICA outperforms rRAMICA indicates that the column structure is stronger than the row structure on the whole for the selected Caltech256 images.

7. Conclusion

This paper proposed a new ICA method RAMICA for BSS with data tensor input. It differs from the classical vectorization-based approach and more recent mode-wise approach by dealing with data tensor directly, without vectorization or unfolding. Thus, it can do more general BSS while preserving structural information. We build RAMICA based on random matrix modeling with two versions: row-wise and column-wise RAMICA. By defining new statistics of random matrix, we develop a two-step RAMICA algorithm with a new cumulant operator and the Jacobi method. Experimental results on both synthetic and real image BSS showed that RAMICA outperformed competing ICA methods greatly in BSS on data tensor, with its two versions having their respective merits. Future directions include extensions of the proposed approach to faster, more robust algorithms, and random-matrix-based machine learning and tensor analysis.

Acknowledgment

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/R014507/1, National Natural Science Foundation of China (Grant No. 62002148), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterprepreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531).

References

[1] P. Comon, C. Jutten, J. Herault, Blind separation of sources, part ii: Problems statement, Signal processing 24 (1) (1991) 11–20.

[2] P. Comon, Independent component analysis, a new concept?, Signal processing 36 (3) (1994) 287–314.

[3] A. Hyvärinen, E. Oja, Independent Component Analysis: Algorithms and Applications, IEEE Trans. Neural Netw. 13 (4-5) (2000) 411–430.

[4] L. Zhang, Q. Gao, D. Zhang, Directional Independent Component Analysis with Tensor Representation, in: CVPR, 2008, pp. 1–7.

[5] Q. Gao, L. Zhang, D. Zhang, H. Xu, Independent Components Extraction from Image Matrix, Pattern Recognition Letters 31 (3) (2010) 171–178.

[6] H. Lu, Learning Modewise Independent Components from Tensor Data Using Multilinear Mixing Model, in: ECMLPKDD, 2013, pp. 288–303.

[7] H. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data, CRC Press, 2013.

[8] A. Hyvärinen, E. Oja, A Fast Fixed-point Algorithm for Independent Component Analysis, Neural Computation 9 (1997) 1483–1492.

[9] J. F. Cardoso, A. Souloumiac, Blind Beamforming for Non-Gaussian Signals, IEE Proceedings F (Radar and Signal Processing) 140 (6) (1993) 362–370.

[10] A. J. Bell, T. J. Sejnowski, An Information-Maximization Approach to Blind Separation and Blind Deconvolution, Neural Computation 7 (1995) 1129–1159.

[11] M. A. O. Vasilescu, D. Terzopoulos, Multilinear Independent Components Analysis, in: CVPR, 2005, pp. 547–553.

[12] M. A. O. Vasilescu, D. Terzopoulos, Multilinear (Tensor) ICA and Dimensionality Reduction, in: Int. Conf. on Independent Component Analysis and Signal Separation, 2007, pp. 818–826.

[13] R. G. Raj, A. C. Bovik, MICA: A Multilinear ICA Decomposition for Natural Scene Modeling, IEEE Trans. Image Process. 17 (3) (2008) 259–271.

[14] J. Virta, B. Li, K. Nordhausen, H. Oja, JADE for tensor-valued observations, Journal of Computational and Graphical Statistics 27 (3) (2018) 628–637.

[15] J. Virta, B. Li, K. Nordhausen, H. Oja, Independent component analysis for tensor-valued data, Journal of Multivariate Analysis 162 (2017) 172–192.

[16] J. F. Cardoso, Source Separation Using Higher Order Moments, in: ICASSP, 1989, pp. 2109–2112.

[17] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. on Matrix Analysis and Applications 21 (4) (2000) 1253–1278.

[18] T. G. Kolda, B. W. Bader, Tensor Decompositions and Applications, SIAM Review 51 (3) (2009) 455–500.

[19] M. Bilodeau, D. Brenner, Theory of Multivariate Statistics, Springer-Verlag New York, 1999.

[20] M. S. Srivastava, T. von Rosen, D. von Rosen, Models with a Kronecker Product Covariance Structure: Estimation and Testing, Mathematical Methods of Statistics 17 (4) (2008) 357–370.

[21] M. John, Multivariate Statistics: Old School, CreateSpace Independent Publishing Platform, Department of Statistics, University of Illinois at Urbana-Champaign, 2015.

[22] L. R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.

[23] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley-Interscience, 2001.

[24] P. Ilmonen, H. Oja, R. Serfling, On Invariant Coordinate System (ICS) Functionals, Int. Statistical Review 80 (2012) 93–110.

[25] P. Comon, Tensor Diagonalization, A Useful Tool in Signal Processing, IFAC Symposium on System Identification 1 (1994) 77–82.

[26] G. Deco, D. Obradovic, An Information-Theoretic Approach to Neural Computing (1st Edition), Springer, 1996.

[27] B. D. Clarkson, A Least Squares Version of Algorithm as 211: The F-G Diagonalization Algorithm, Applied Statistics 37 (1988) 317–321.

[28] J. Miettinen, S. Taskinen, K. Nordhausen, H. Oja, Fourth moments and independent component analysis, Statistical Science 30 (3) (2015) 372–390.

[29] F. J. Theis, Blind Signal Separation into Groups of Dependent Signals Using Joint Block Diagonalization, in: ISCAS, 2005, pp. 5878–5881 Vol. 6.

[30] A. Hyvärinen, U. Köster, FastISA: A Fast Fixed-point Algorithm for Independent Subspace Analysis, in: ESANN, 2006.

[31] T. Kim, T. Eltoft, T.-W. Lee, Independent vector analysis: An extension of ica to multivariate components, in: Int. conf. on independent component analysis and signal separation, 2006, pp. 165–172.

[32] T. Adali, M. Anderson, G.-S. Fu, Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging, IEEE Signal Process. Mag. 31 (3) (2014) 18–33.

[33] A. Mahadevan, S. Acharya, D. B. Sheffer, D. H. Mugler, Ballistocardiogram artifact removal in EEG-fMRI signals using discrete hermite transforms, IEEE J. Sel. Topics Signal Process. 2 (6) (2008) 839–853.

[34] M. Pakravan, M. B. Shamsollahi, Joint, partially-joint, and individual independent component analysis in multi-subject fMRI data, IEEE Transactions on Biomedical Engineering 67 (7) (2019) 1969–1981.

[35] S. Amari, A. Cichocki, H. H. Yang, A New Learning Algorithm for Blind Signal Separation, in: NeurIPS, 1996, pp. 757–763.

[36] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, California Institute of Technology (2006).