PNAS RESEARCH ARTICLE | APPLIED MATHEMATICS

# Contrastive independent component analysis for salient patterns and dimensionality reduction

Kexin Wang[a], Aida Maraj[b], and Anna Seigal[a,1]

Edited by Peter Bickel, University of California, Berkeley, CA; received December 1, 2024; accepted October 29, 2025

**In recent years, there has been growing interest in jointly analyzing a foreground dataset, representing an experimental group, and a background dataset, representing a control group. The goal of such contrastive investigations is to identify salient features in the experimental group relative to the control. Independent component analysis (ICA) is a powerful tool for learning independent patterns in a dataset. We generalize it to contrastive ICA (cICA). For this purpose, we devise a linear algebra–based tensor decomposition algorithm, which is more expressive but just as efficient and identifiable as other linear algebra–based algorithms. We establish the identifiability of cICA and demonstrate its performance in finding patterns and visualizing data, using synthetic, semisynthetic, and real-world datasets, comparing the approach to existing methods.**

independent component analysis | tensor decomposition | contrastive methods

## Significance

Visualizing data and finding patterns in data are ubiquitous problems in the sciences. Increasingly, applications seek signal and structure in a contrastive setting: a foreground dataset relative to a background dataset. The goal is to learn patterns and visualize the foreground after "subtracting off" the effect of the background. For this purpose, we propose contrastive independent component analysis (cICA). We investigate cICA theoretically and computationally. We find that, relative to other approaches, cICA is more expressive; that is, able to model a broader range of settings, while simultaneously being identifiable, able to recover patterns uniquely.

Finding and understanding patterns in data is fundamental in various scientific fields. Often, data have been collected under two different settings, such as a group of patients receiving treatment and a control group, or a group of patients with a certain disease and a group without the disease. The goal is to understand the effect of the treatment or to understand the genetic changes that describe the disease. While standard data analysis methods can be used, which restrict attention to one of the datasets or combine them together, an alternate view is offered by contrastive methods. Contrastive methods view the two settings as a foreground and a background. They seek to learn patterns in the foreground after accounting for (or, "subtracting off") the background. The hope is that such patterns encode useful structures and offer a good basis for dimensionality reduction and visualization of the data, to identify fine-grained structures and clusters particular to the foreground.

Back in the 1980s, Flury initiated the idea of comparing covariance matrices and finding principal components across multiple datasets (1–3). The contrastive viewpoint was then addressed and formalized in ref. 4, where the authors discussed contrastive topic modeling and contrastive hidden Markov models. Principal component analysis (PCA) was generalized to contrastive PCA (cPCA) in refs. 5 and 6. A latent variable model perspective is taken in refs. 7 and 8. The present work extends such methods, specifically cPCA, to a more expressive and identifiable setting. Specifically, it removes simplifying assumptions that the amount of each background signal present in the foreground is the same (4–8), that the latent variables are Gaussians (5–8), and that the salient patterns in the foreground data are orthogonal (5, 6). The greater expressivity and identifiability are achieved using the higher-order cumulant tensors of the foreground and background data, which encode more fine-grained structure than the covariance matrices.

We call the method contrastive independent component analysis (cICA). Independent component analysis (ICA) is a blind source separation method, which seeks to recover latent sources and unknown mixing from observations of mixtures of signals (9). ICA assumes that latent sources are independent. In extending ICA to the contrastive setting, the idea is that background data is generated by mixing of independent sources while foreground data are generated by the background mixing together with a foreground mixing of independent sources.

We show using connections to classical algebraic geometry that cICA has strong identifiability properties. This enables the contribution of each background pattern to the foreground to be found uniquely, avoiding the need for a sweep of hyperparameters to find the best multiple of the background to subtract from the foreground and even avoids the assumption that the background contribution to the foreground is via a single scalar multiple, both of which are required in refs. 4–7.

To implement cICA, we devise a hierarchical tensor decomposition based on recursive eigendecompositions. The decomposition encourages (rather than imposes) orthogonality between the rank-one summands. We show that it recovers accurate

Author affiliations: [a]School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138; and [b]Max Planck Institute of Molecular Cell Biology and Genetics and Center for Systems Biology, Dresden 01307, Germany

Author contributions: K.W., A.M., and A.S. designed research; K.W., A.M., and A.S. performed research; K.W. and A.S. analyzed data; and K.W. and A.S. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.



[1]To whom correspondence may be addressed. Email: aseigal@seas.harvard.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2425119122/-/DCSupplemental.

Published December 10, 2025.

patterns for synthetic data. We turn cICA into a dimensionality reduction tool and investigate its performance on real-world data, comparing the plots to those obtained with other contrastive methods to see its competitiveness.

The paper is organized as follows. We define cICA in Section 1. We introduce the hierarchical tensor decomposition in Section 2. We study identifiability and algorithms for cICA in Section 3. Numerical results are in Section 4.

## 1. From ICA to cICA

Blind source separation seeks to recover latent sources and unknown mixing from observations of mixtures of signals (9). A special case is ICA, which assumes that the latent sources are independent. ICA was introduced in 1985 (10) and popularized by Comon in his paper (11).

ICA can be viewed as a generalization of PCA, where instead of finding uncorrelated components, it goes a step further by aiming to make the components statistically independent and instead of decomposing second-order information (covariance matrices), it decomposes higher-order statistics (via the cumulant tensors).

ICA studies observations that are a linear mixture of independent source variables. Applications include recovering speech and brain signals (12, 13), causal discovery (14), and image decomposition (15). We write the ICA model as

$$\mathbf{y} = A\mathbf{z}, \qquad [1]$$

where $\mathbf{z}$ is a vector of $r$ independent latent random variables, the mixing matrix is $A \in \mathbb{R}^{p\times r}$, and $\mathbf{y}$ is a vector of $p$ observed variables. The $i$-th column of $A$ records a pattern in the data: the contribution of variable $z_i$ to each of the $p$ observed variables. The identifiability of ICA refers to the uniqueness of the mixing matrix $A$ and sometimes also of the variables $\mathbf{z}$; see refs. 11, 16, and 17.

Many algorithms for ICA proceed via tensor decomposition; see, e.g., refs. 9 and 18–20. The cumulants of a distribution are symmetric tensors that encode it. The $d$-th cumulant $\kappa_d(\mathbf{y})$ of $\mathbf{y}$ is a symmetric order $d$ tensor of format $p \times \cdots \times p$ whose entry at position $(j_1, \ldots, j_d)$ is

$$\sum_{i=1}^{r} \lambda_i (\mathbf{a}_i)_{j_1} \cdots (\mathbf{a}_i)_{j_d}, \qquad [2]$$

where the scalar $\lambda_i$ is the $d$-th cumulant of $z_i$ and the vector $\mathbf{a}_i \in \mathbb{R}^p$ is the $i$-th column of $A$. We denote this by

$$\kappa_d(\mathbf{y}) = \sum_{i=1}^{r} \lambda_i \mathbf{a}_i^{\otimes d}. \qquad [3]$$

This decomposition Eq. **3** follows from the multilinear properties of cumulants and the fact that cumulant tensors of independent variables are diagonal, see (21, Chapter 2). The matrix $A$ can be recovered using tensor decomposition of the cumulant tensor Eq. **2**. If the tensor decomposition is identifiable, then the columns $\mathbf{a}_i$ with $\lambda_i \neq 0$ can be recovered uniquely up to permutation and scaling of columns. Thus tensor decomposition of higher-order cumulant tensors gives an algorithm for ICA, provided no source variable is Gaussian (this is required for nonzero higher-order cumulants).

In this paper, we extend ICA, and tensor decomposition for ICA, to the comparison of two distributions. We call this contrastive ICA (cICA), by analogy with cPCA (6). We have two observed distributions, a foreground, and a background. Both are assumed to be linear mixtures of independent source variables. Our cICA model expresses the background $\mathbf{y}$ and foreground $\mathbf{x}$ as

$$\mathbf{y} = A\mathbf{z} \qquad \text{and} \qquad \mathbf{x} = A\mathbf{z}' + B\mathbf{s}. \qquad [4]$$

The background distribution $\mathbf{y}$ is a linear mixture of a random vector $\mathbf{z}$ of $r$ independent random variables, as in Eq. **1**. The foreground $\mathbf{x}$ is a mixture of $r + \ell$ independent variables $\mathbf{z}' = (z'_1, \ldots, z'_r)$ and $\mathbf{s} = (s_1, \ldots, s_\ell)$. The columns of $A$ are the patterns in the background: Column $\mathbf{a}_i \in \mathbb{R}^p$ records how source variable $z_i$ appears among the $p$ background variables as well as how source variable $z'_i$ appears among the $p$ foreground variables. The columns of $B$ are patterns that appear only in the foreground. They correspond to the variables $s_i$, referred to as the salient variables in ref. 22.

We propose a tensor decomposition algorithm to recover mixing matrices $A$ and $B$ from Eq. **4**. These matrices record the patterns that encode our background and foreground distributions. We apply the algorithm to empirical cumulant tensors of $\mathbf{x}$ and $\mathbf{y}$ obtained from sample data. We order the columns of matrix $B$ to obtain a dimensionality reduction tool. We work under the assumption that $\mathbf{z}, \mathbf{z}', \mathbf{s}$ are non-Gaussian, an assumption that also appears for usual ICA. This can likely be relaxed to that at most one source is Gaussian, cf. (11, 17).

Under the model Eq. **4**, the $d$-th cumulants of the background and foreground data are, respectively,

$$\kappa_d(\mathbf{y}) = \sum_{i=1}^{r} \lambda_i \mathbf{a}_i^{\otimes d}, \qquad \kappa_d(\mathbf{x}) = \sum_{i=1}^{r} \lambda'_i \mathbf{a}_i^{\otimes d} + \sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes d}, \qquad [5]$$

where $\lambda_i$ is the $d$-th cumulant of $z_i$, $\lambda'_i$ is the $d$-th cumulant of $z'_i$, and $\nu_j$ is the $d$-th cumulant of $s_j$. This follows from the multilinearity of cumulants and that cumulant tensors of independent sources are diagonal, as for usual ICA. See Fig. 1 for an illustration of $\kappa_3(\mathbf{x}) = \sum_{i=1}^{r} \lambda'_i \mathbf{a}_i^{\otimes d} + \sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes d}$ when $d = 3$.

We have the following optimization problem to recover $A$ and $B$: find a joint decomposition of cumulant tensors $\kappa_d(\mathbf{y})$ and $\kappa_d(\mathbf{x})$ of the form in Eq. **5**. Our approach is

1. Compute a symmetric tensor decomposition of $\kappa_d(\mathbf{y})$ to learn $A$.
2. Find the coefficients $\lambda'_i$ of each $\mathbf{a}_i^{\otimes d}$ in $\kappa_d(\mathbf{x})$ to obtain $\sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes d}$.

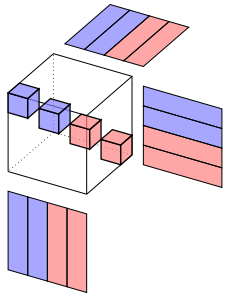

**Fig. 1.** Tensor decomposition for $\kappa_3(\mathbf{x}) = \sum_{i=1}^{r} \lambda'_i \mathbf{a}_i^{\otimes d} + \sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes d}$ when $d = 3$ and $r = \ell = 2$. The central $4 \times 4 \times 4$ diagonal tensor is multiplied along each index by a matrix with four columns, whose first two columns (blue) are the background patterns and last two columns (red) are the foreground patterns.

3. Compute a symmetric tensor decomposition of $\sum_{j=1}^{\ell} v_j \mathbf{b}_j^{\otimes d}$ to learn $B$.

We work with the fourth-order cumulants $d = 4$, since the tensor decomposition we use works better for an even order symmetric tensor. For the third step of our approach, we require a tensor decomposition method that is efficient and promotes orthogonality among the rank-1 components, which aids interpretability and improves visualizations. To address this, we propose a hierarchical eigendecomposition based algorithm, which we describe in more detail in the next section. The algorithm uses linear algebra and can handle tensors of rank up to $p^2$ [unlike $p$ in other linear algebra-based methods (23, 24)].

**1.1. Related Work.** We relate cICA to other contrastive models. In cPCA, the contrastive patterns are principal components of the foreground covariance matrix minus a scalar multiple of the background covariance matrix (5, 6). We can specialize cICA to cPCA by setting $\mathbf{z}' = \gamma \mathbf{z}$ and studying observed distributions $\mathbf{x}$ and $\mathbf{y}$ via their covariance matrices ($d = 2$). Probabilistic contrastive PCA (PCPCA) is introduced in ref. (7), where foreground patterns are inferred by maximizing a likelihood ratio of linear Gaussian mixtures. Contrastive ICA also relates to PCPCA (7) but we do not impose distributional assumptions, beyond independence and non-Gaussianity, on the variables $\mathbf{z}$ and $(\mathbf{z}', \mathbf{s})$. The paper (8) studies a linear contrastive latent variable model. The contrastive ICA model aligns with the framework of the contrastive latent variable model proposed in ref. 8, but it does not assume any relationship between $\mathbf{z}$ and $\mathbf{z}'$ while the contrastive latent variable model assumes $\mathbf{z} = \mathbf{z}'$.

The setting of cICA relates to usual ICA, with block structure on the mixing matrix:

$$\text{if } \mathbf{z}', \mathbf{z}, \mathbf{s} \text{ are independent, } \quad \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} 0 & A & B \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ \mathbf{z}' \\ \mathbf{s} \end{pmatrix};$$

$$\text{if } \mathbf{z}' = \gamma \mathbf{z}, \quad \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \gamma A & B \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ \mathbf{s} \end{pmatrix}.$$

Identifiability can be characterized using (11), or using (16, 17) if the model is overcomplete (i.e., the number of sources exceeds the number of observations, which occurs for $2r + \ell > 2p$). However, learning parameters via usual ICA requires access to the joint distribution of $(\mathbf{x}, \mathbf{y})$, which is generally unavailable because the data from the two datasets are unpaired. For example, single-cell RNA data for patients with a disease (foreground) and a control group (background) have each person assigned to either the foreground set or the background.

In ref. 25, the authors study multimodal linear ICA. They recover the mixing matrices from each mode via usual linear ICA and use a hypothesis test to decide which latent variables are shared across modes. Our method differs from this as we seek patterns unique to the foreground rather than shared patterns.

Nonlinear contrastive methods have been explored in the literature. Nonlinear ICA is studied using contrastive learning (26–28). Here contrastive is used in a different context: It describes a method to train a network to distinguish two datasets. A nonlinear contrastive method called a contrastive variational autoencoder (cVAE) is introduced in refs. 8 and 22. The paper (29) presents a method for cVAE using maximum mean discrepancy to prevent leakage of information between the two sets of latent variables. Identifiability of cVAE is studied using connections to nonlinear ICA in ref. 30. These works produce a nonlinear latent encoding of data, whereas our focus is on pattern vectors to describe observed variables.

## 2. Hierarchical Tensor Decomposition

ICA has seen limited application in data visualization, one notable exception being (31). Existing algorithms to compute a symmetric tensor decomposition usually have randomness due to initialization and the details of the optimization process, such as the step size in gradient-based optimization. Another challenge is that the resulting vectors may be nearly parallel (32), which yields a suboptimal basis for projecting the data and hinders its interpretability. We overcome these difficulties with our proposed hierarchical tensor decomposition (HTD). Its output is deterministic and the components learned are almost orthogonal.

HTD decomposes an order four tensor via recursive eigendecompositions. The idea is to find a low-rank approximation of a tensor, whose rank-one summands offer an interpretable basis on which to project data. Later, we use the decomposition for cICA. In this section, we define the decomposition and study its properties. HTD for a tensor in $(\mathbb{R}^p)^{\otimes 4}$ uses linear structure in the space $(\mathbb{R}^p)^{\otimes 2}$ rather than $\mathbb{R}^p$, so it handles tensors of rank up to $p^2$ [unlike $p$ in other linear algebra-based methods (23, 24)]. The detailed comparison with other tensor decomposition methods is in *SI Appendix*, section 1.

**2.1. The HTD Algorithm.** Consider a symmetric tensor $T$ of format $p \times p \times p \times p$. We compute a rank $r$ approximation,

$$T \approx \sum_{i=1}^{r} v_i \mathbf{b}_i^{\otimes 4}, \qquad [6]$$

as follows. Let $\mathrm{Mat}(T)$ be the flattening of $T$ that rearranges its $p^4$ entries into a matrix of size $p^2 \times p^2$. The entries of $\mathrm{Mat}(T)$ are indexed $((i_1, i_2), (j_1, j_2))$, where $i_1, i_2, j_1, j_2 \in [p] := \{1, \ldots, p\}$. We compute the approximation Eq. **6** by first computing the eigendecomposition of $\mathrm{Mat}(T)$, whose eigenvectors lie in $\mathbb{R}^{p^2}$, and then by reshaping these eigenvectors into $p \times p$ matrices and computing their top eigenvalue and corresponding eigenvector. By top eigenvalue we mean those of highest magnitude. This decomposition has not to our knowledge been studied before but has connections to the hierarchical tensor representations of (33, Chapter 11) and the PARATREE model in ref. 34; see *SI Appendix*, section 1. See Fig. 2 for an illustration of the steps of HTD on a $2 \times 2 \times 2 \times 2$ tensor. Here is the HTD algorithm.

***Algorithm 1:*** Compute unit vectors $\mathbf{b}_1, \ldots, \mathbf{b}_r$ such that $T \approx \sum_{i=1}^{r} v_i \mathbf{b}_i^{\otimes 4}$

**Input:** Symmetric tensor $T$ of format $p \times p \times p \times p$ and rank $r$.

1: Compute the eigendecomposition of the $p^2 \times p^2$ flattening $\mathrm{Mat}(T)$. Take the top $r$ eigenvalues $\mu_1, \ldots, \mu_r$, with corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r \in \mathbb{R}^{p^2}$ of unit length.

2: For each $i \in [r]$, reshape $\mathbf{v}_i \in \mathbb{R}^{p^2}$ to $M_i \in \mathbb{R}^{p \times p}$.

3: For each $M_i$, find the top eigenvalue $\beta_i$ and a corresponding unit length eigenvector $\mathbf{b}_i \in \mathbb{R}^p$.

**Output:** Rank $r$ decomposition $\sum_{i=1}^{r} (\mu_i \beta_i^2) \mathbf{b}_i^{\otimes 4}$.

We record some observations pertaining to Algorithm 1. The matrix $\mathrm{Mat}(T) \in \mathbb{R}^{p^2 \times p^2}$ is symmetric since $T$ is symmetric. The matrices $M_1, \ldots, M_r \in \mathbb{R}^{p \times p}$ are also symmetric, because

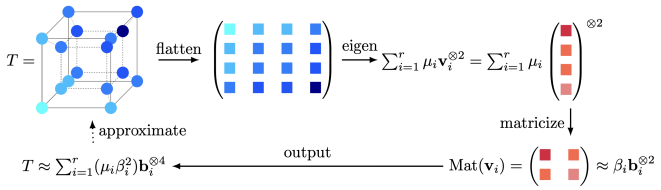


**Fig. 2.** Steps in the HTD algorithm: input tensor $T$, matrix flattening $\mathrm{Mat}(T)$, best rank $r$ approximation $\mathrm{Mat}(T) \approx \sum_{i=1}^{r} \mu_i \mathbf{v}_i^{\otimes 2}$, best rank-one approximation of each $\mathrm{Mat}(\mathbf{v}_i)$ and the output rank $r$ approximation for $T$.

the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ are in the column space of $\mathrm{Mat}(T)$, whose $(i_1, i_2)$-th row coincides with its $(i_2, i_1)$-th row. Although the output vectors $\mathbf{b}_i$ are in general not orthogonal, as each is an eigenvector of a distinct matrix, they can be nearly orthogonal in practice, see Section 2.2. This is because they are the leading eigenvectors of matrices that have been reshaped from orthogonal vectors $\mathbf{v}_i$.

***Example 2.1.*** [$2 \times 2 \times 2 \times 2$ example] Let $r = 2$. Fix

$$T = 2\begin{bmatrix}1\\0\end{bmatrix}^{\otimes 4} + \begin{bmatrix}0.0998\\0.995\end{bmatrix}^{\otimes 4}.$$

Then

$$\mathrm{Mat}(T) = \begin{bmatrix} 2.0001 & 0.0010 & 0.0010 & 0.0099 \\ 0.0010 & 0.0099 & 0.0099 & 0.0983 \\ 0.0010 & 0.0099 & 0.0099 & 0.0983 \\ 0.0099 & 0.0983 & 0.0983 & 0.9801 \end{bmatrix}$$

with eigenvalues $\mu_1 = 2.00019$, $\mu_2 = 0.99977$ and associated eigenvectors

$$\mathbf{v}_1^{\mathsf{T}} \approx [0.99995 \quad 0.00098 \quad 0.00098 \quad 0.00985],$$
$$\mathbf{v}_2^{\mathsf{T}} \approx [-0.00995 \quad 0.0993 \quad 0.0993 \quad 0.99003].$$

Their corresponding matrices $M_1, M_2 \in \mathbb{R}^{2\times 2}$ are symmetric with top eigenvalues $\beta_1 = 0.99995$ and $\beta_2 = 0.9998$, respectively, with associated eigenvectors $\mathbf{b}_1^{\mathsf{T}} = [0.99999 \quad 0.00099]$ and $\mathbf{b}_2^{\mathsf{T}} = [0.09787 \quad 0.99519]$. The HTD algorithm with input $T$ and $r = 2$ thus outputs

$$\sum_{i=1}^{2} (\mu_i \beta_i^2)\mathbf{b}_i^{\otimes 4} = 1.99999\begin{bmatrix}0.99999\\0.00099\end{bmatrix}^{\otimes 4} + 0.99937\begin{bmatrix}0.09787\\0.99519\end{bmatrix}^{\otimes 4}.$$

We note the similarity to the input tensor $T$.

**2.2. Properties of the Decomposition.** The HTD algorithm outputs a rank $r$ approximation of a tensor. In certain cases, the output closely approximates the input tensor, as in Example 2.1. We bound the distance between the HTD approximation and the input tensor. We give a bound that applies to all tensors in Proposition 2.2. We show that the input and output coincide for orthogonally decomposable tensors in Proposition 2.3. Our main result is Theorem 2.4, which bounds the distance between an input and output tensor for a tensor decomposition involving vectors that are close to orthogonal.

The norm $\|\cdot\|_F$ refers to the Frobenius norm for matrices and tensors and the 2-norm for vectors; i.e., the square root of the sum of the squares of the entries. The 2-norm of a matrix is denoted by $\|\cdot\|_2$.

**Proposition 2.2.** *Let $T$ be a symmetric tensor of format $p\times p\times p\times p$. Let $T' = \sum_{i=1}^{r} (\mu_i\beta_i^2)\boldsymbol{b}_i^{\otimes 4}$ be the rank $r$ HTD approximation of $T$. Then*

$$\|T' - T\|_F \le \left(\sum_{i=r+1}^{q} \mu_i^2\right)^{\frac{1}{2}} + \sum_{i=1}^{r} |\mu_i|(1+|\beta_i|)\left(\sum_{j=2}^{r_i} (\beta_i^{(j)})^2\right)^{\frac{1}{2}},$$

*where $q$ is the rank of $\mathrm{Mat}(T)$, $r_i$ is the rank of $M_i$, the numbers $\mu_1, \ldots, \mu_r$ are the eigenvalues of $\mathrm{Mat}(T)$ in descending order of magnitude, and $\beta_i := \beta_i^{(1)}$ is the highest magnitude eigenvalue of $M_i$ with $\beta_i^{(2)}, \ldots, \beta_i^{(r_i)}$ the other eigenvalues.*

***Proof:*** We use the notation from Algorithm 1. We have

$$\|\mathrm{Mat}(T) - \sum_{i=1}^{r} \mu_i \mathbf{v}_i^{\otimes 2}\|_F^2 = \sum_{i=r+1}^{q} \mu_i^2,\ \|M_i - \beta_i \mathbf{b}_i^{\otimes 2}\|_F^2 = \sum_{j=2}^{r_i} (\beta_i^{(j)})^2,$$

from the properties of the eigendecomposition of a symmetric matrix and the Frobenius norm. Let $T''$ be the $p\times p\times p\times p$ tensor obtained from reshaping the truncated eigendecomposition $\sum_{i=1}^{r} \mu_i \mathbf{v}_i^{\otimes 2}$ of $\mathrm{Mat}(T)$. Then $\|T - T''\|_F^2 = \sum_{i=r+1}^{q} \mu_i^2$. Let $\mathbf{B}_i \in \mathbb{R}^{p^2}$ be the vectorization of $\mathbf{b}_i^{\otimes 2} \in \mathbb{R}^{p\times p}$. Then

$$\begin{aligned}
\|T'' - T'\|_F &= \|\sum_{i=1}^{r} \mu_i(\mathbf{v}_i^{\otimes 2} - \beta_i^2 \mathbf{B}_i^{\otimes 2})\|_F \\
&\le \sum_{i=1}^{r} |\mu_i| \|\mathbf{v}_i^{\otimes 2} - \beta_i^2 \mathbf{B}_i^{\otimes 2}\|_F \\
&\le \sum_{i=1}^{r} |\mu_i| (\|\mathbf{v}_i^{\otimes 2} - \beta_i \mathbf{B}_i \otimes \mathbf{v}_i\|_F + \|\beta_i^2 \mathbf{B}_i^{\otimes 2} - \beta_i \mathbf{B}_i \otimes \mathbf{v}_i\|_F) \\
&= \sum_{i=1}^{r} |\mu_i| (\|\mathbf{v}_i\| + |\beta_i| \|\mathbf{B}_i\|) \|\mathbf{v}_i - \beta_i \mathbf{B}_i\| \\
&= \sum_{i=1}^{r} |\mu_i| (1 + |\beta_i|) \left(\sum_{j=2}^{r_i} (\beta_i^{(j)})^2\right)^{\frac{1}{2}},
\end{aligned}$$

where the penultimate equality follows from $\|\mathbf{x}\otimes\mathbf{y}\| = \|\mathbf{x}\| \cdot \|\mathbf{y}\|$ and the last equality uses $\|\mathbf{v}_i\| = \|\mathbf{B}_i\| = 1$. We conclude with the triangle inequality $\|T - T'\|_F \le \|T - T''\|_F + \|T'' - T'\|_F$. □

The quantity in Proposition 2.2 is small if $\mathrm{Mat}(T)$ is well approximated by a matrix of rank $r$, and each $M_i$ is well approximated by a matrix of rank one. Orthogonally decomposable tensors are those with a decomposition into orthogonal rank-one terms; that is, a decomposition $T = \sum_{i=1}^{r} \nu_i \mathbf{b}_i^{\otimes 4}$, where $\mathbf{b}_1, \ldots, \mathbf{b}_r$ are orthonormal (35). For orthogonally decomposable tensors, HTD recovers the exact decomposition.

**Proposition 2.3.** *Let $T = \sum_{i=1}^{r} \nu_i \boldsymbol{b}_i^{\otimes 4}$, where the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r$ are orthonormal and the coefficients $\nu_1, \ldots, \nu_r$ are distinct. Then the rank $r$ HTD approximation is the tensor $T$.*

***Proof:*** The flattening $\mathrm{Mat}(T)$ has decomposition $\sum_{i=1}^{r} \nu_i \mathbf{B}_i^{\otimes 2}$, where $\mathbf{B}_i \in \mathbb{R}^{p^2}$ is the vectorization of $\mathbf{b}_i^{\otimes 2} \in \mathbb{R}^{p\times p}$. We have the orthogonality $\langle \mathbf{B}_i, \mathbf{B}_j \rangle = \langle \mathbf{b}_i, \mathbf{b}_j \rangle^2 = 0$ for all $i \ne j$, since the vectors $\mathbf{b}_i, \mathbf{b}_j$ are orthogonal. Hence this expression for $\mathrm{Mat}(T)$ is a sum of outer products of orthogonal vectors, so

it is the eigendecomposition of $\mathrm{Mat}(T)$. The matrix reshaped from the eigenvector $\mathbf{B}_i$ is $M_i = \mathbf{b}_i^{\otimes 2}$. It has top eigenvalue 1 with corresponding eigenvector $\mathbf{b}_i$. Hence, the output of HTD is $\sum_{i=1}^r \nu_i \mathbf{b}_i^{\otimes 4}$. □

We extend Proposition 2.3 to decompositions where the vectors $\mathbf{b}_i$ are close to orthogonal and the input tensor is noisy. The condition that the matrices $\mathbf{b}_1^{\otimes 2}, \ldots, \mathbf{b}_r^{\otimes 2}$ are linearly independent ensures that $\mathrm{Mat}(T)$ has rank $r$. This condition holds for generic vectors $\mathbf{b}_i$, provided $r \leq \binom{p+1}{2}$. The quantity $\min\{\|\mathbf{b}_i - \mathbf{b}_i'\|, \|\mathbf{b}_i + \mathbf{b}_i'\|\}$ arises because of the sign indeterminacy in the vectors in the decompositions, due to the equality $(-\mathbf{b}_i)^{\otimes d} = \mathbf{b}_i^{\otimes d}$ for $d$ even.

We provide a sketch proof of Theorem 2.4. The full proof can be found in *SI Appendix*, section 2.

**Theorem 2.4.** *Fix vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell \in \mathbb{R}^p$ with $|\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle| \leq \epsilon$ for all $i \neq j$. Let*

$$T = \sum_{i=1}^{\ell} \nu_i \boldsymbol{b}_i^{\otimes 4},$$

*where $\nu_1 > \cdots > \nu_\ell$, $\ell \leq p$, and $\boldsymbol{b}_1^{\otimes 2}, \ldots, \boldsymbol{b}_\ell^{\otimes 2}$ are linearly independent. Fix $\hat{T}$ with $\|\hat{T} - T\|_F \leq \delta$. Let $\boldsymbol{c}_i$ be the output patterns of the HTD algorithm with input tensor $\hat{T}$ and $\mu_i$ the corresponding recovered scalars ordered so that $\mu_1 > \cdots > \mu_\ell$. Then for any $i \in [\ell]$,*

$$|\nu_i - \mu_i| = O(\epsilon^2) + O(\delta), \quad \textit{and}$$

$$\min\{\|\boldsymbol{b}_i - \boldsymbol{c}_i\|, \|\boldsymbol{b}_i + \boldsymbol{c}_i\|\} = O(\epsilon^2) + O(\delta).$$

***Sketch Proof:*** Fix $M = \mathrm{Mat}(T)$. Then $M = \sum_{i=1}^r \nu_i \mathbf{B}_i^{\otimes 2}$, where $\mathbf{B}_i = \mathrm{Vect}(\mathbf{b}_i^{\otimes 2})$. Using Gram–Schmidt orthogonalization, we can construct a matrix $M'$ in $\mathbb{R}^{p^2 \times p^2}$ with eigendecomposition $\sum_{i=1}^\ell \nu_i (\mathbf{B}_i')^{\otimes 2}$ such that

$$\|\mathbf{B}_i' - \mathbf{B}_i\| \leq 2(\ell - 1)\epsilon^2 + O(\epsilon^4), \qquad [7]$$

$$\|M - M'\|_F \leq K\epsilon^2 + O(\epsilon^4),$$

where $K = \sqrt{8} \sum_{i=1}^\ell |\nu_i|(i-1)$. Suppose $\hat{M} = \mathrm{Mat}(\hat{T})$ has eigendecomposition $\hat{M} = \sum_{i=1}^\ell \hat{\nu}_i \hat{\mathbf{B}}_i^{\otimes 2}$. The difference between $\hat{M}$ and $M'$ is bounded by

$$\|\hat{M} - M'\|_F \leq \|\hat{M} - M\|_F + \|M - M'\|_F \leq K\epsilon^2 + \delta + O(\epsilon^4)$$

using the triangle inequality. We thus obtain

$$|\nu_i - \hat{\nu}_i| \leq \delta + K\epsilon^2 + O(\epsilon^4), \qquad [8]$$

$$\|\hat{\mathbf{B}}_i - \mathbf{B}_i'\| \leq \frac{2^{\frac{3}{2}}}{\nu}(\delta + K\epsilon^2 + O(\epsilon^4)), \qquad [9]$$

by Weyl's Theorem and the variant of Davis-Kahan Theorem in ref. (36), where $\nu = \min_{i \neq j}\{|\nu_i - \nu_j|, |\nu_i|\}$. We bound the difference between $\mathbf{B}_i$ and $\hat{\mathbf{B}}_i$ using Eqs. **7** and **9**:

$$\|\mathbf{B}_i - \hat{\mathbf{B}}_i\| \leq \|\mathbf{B}_i' - \mathbf{B}_i\| + \|\mathbf{B}_i' - \hat{\mathbf{B}}_i\| \leq L\epsilon^2 + \frac{2^{\frac{3}{2}}}{\nu}\delta + O(\epsilon^4), \qquad [10]$$

where $L = 2^{3/2}\frac{K}{\nu} + 2\ell - 2$. Then, by Weyl's theorem,

$$|\alpha - 1| \leq \|\mathbf{B}_i - \hat{\mathbf{B}}_i\|, \qquad [11]$$

where $\alpha$ is the top eigenvalue of $\mathrm{Mat}(\hat{\mathbf{B}}_i)$. HTD implies $\mu_i = \alpha^2 \hat{\nu}_i$. The bound on $|\mu_i - \nu_i|$ then follows from Eq. **8** and Eq. **11**. The bound of $\min\{\|\mathbf{b}_i - \mathbf{c}_i\|, \|\mathbf{b}_i + \mathbf{c}_i\|\}$ follows from Eq. **10** and (36), since $\mathbf{c}_i$ is the top eigenvector of $\hat{\mathbf{B}}_i$. □

## 3. Tensor Decompositions for cICA

Our cICA model assumes $\mathbf{y} = A\mathbf{z}$ and $\mathbf{x} = A\mathbf{z}' + B\mathbf{s}$, for $A \in \mathbb{R}^{p \times r}$ and $B \in \mathbb{R}^{p \times \ell}$, see Eq. **4**. This leads to the cICA tensor decompositions Eq. **5**. One does not assume a relationship between $\mathbf{z}$ and $\mathbf{z}'$. We discuss the algorithm and identifiability of cICA in Section 3.1. We explain how to use cICA for dimensionality reduction in Section 3.2. This projects data onto a subspace given by certain columns of the foreground mixing $B$. We bound the end-to-end error of our algorithm in Section 3.3. When $\mathbf{z}' = \gamma\mathbf{z}$ for some scalar $\gamma$, we discuss an alternative algorithm in *SI Appendix*, section 4 and its performance for various datasets in *SI Appendix*, section 6.

### 3.1. cICA Algorithm and Identifiability.

We present Algorithm 2 for cICA. Steps 1 and 3 both decompose a symmetric order four tensor. We use the subspace power method (37) in Step 1 to prioritize the accuracy of the tensor decomposition. We use Algorithm 1 in Step 3 to prioritize interpretability and efficiency. We provide numerical experiments to justify these choices of algorithm in Section 4.1.

***Algorithm 2:*** Recover $A$ and $B$ from the cumulants of the background and foreground

**Input:** $\kappa_4(\mathbf{x})$, $\kappa_4(\mathbf{y})$ and $r, \ell$ as in Eq. **5**.

1: **Recover $A$:** Compute the symmetric tensor decomposition of $\kappa_4(\mathbf{y})$ via the subspace power method (37). This recovers $A$ up to permutation and scaling of columns.
2: **Subtract background from $\kappa_4(\mathbf{x})$:** Learn the coefficients $\lambda_i'$ of $\mathbf{a}_1^{\otimes 4}, \ldots, \mathbf{a}_r^{\otimes 4}$ in $\kappa_4(\mathbf{x})$ using the deflation step of the subspace power method.
3: **Recover $B$:** Compute the symmetric tensor decomposition of $\sum_{i=1}^\ell \nu_i \mathbf{b}^{\otimes 4} = \kappa_4(\mathbf{x}) - \sum_{i=1}^r \lambda_i' \mathbf{a}_i^{\otimes 4}$, using Algorithm 1.

**Output:** Mixing matrices $A$ and $B$.

We study the identifiability of the algorithm, that is, the uniqueness of the vectors and scalars it outputs, assuming genericity. Our genericity assumption holds almost surely in the space of parameters.

We use the following lemma.

**Lemma 3.1.** *Let vectors $\boldsymbol{a}_i \in \mathbb{R}^p$ and scalars $\lambda_i \in \mathbb{R}$ be generic. Then the decomposition $T = \sum_{i=1}^q \lambda_i \boldsymbol{a}_i^{\otimes d}$ of a symmetric $p \times p \times p \times p$ tensor $T$ is unique for*

$$q \leq \begin{cases} \lceil \frac{1}{p}\binom{p+3}{4} - 1 \rceil & \text{for } p \notin \{3, 4, 5\}, \\ \lceil \frac{1}{p}\binom{p+3}{4} \rceil & \text{for } p \in \{3, 5\}, \\ 9 & \text{for } p = 4, \text{ provided } q \neq 8. \end{cases}$$

***Proof:*** The rank of a generic $p \times p \times p \times p$ symmetric tensor is $\lceil \frac{1}{p}\binom{p+3}{4} \rceil$ for $p \notin \{3, 4, 5\}$ and $\lceil \frac{1}{p}\binom{p+3}{4} \rceil + 1$ for $p \in \{3, 4, 5\}$, by

the Alexander-Hirschowitz theorem (38). Generic rank $q$ tensors in this space, with $q$ strictly below the generic rank, have unique symmetric tensor decomposition for $(p, q) \neq (4, 8)$ and two tensor decompositions for $p = 4, q = 8$ by (39, Theorem 1.1). □

**Proposition 3.2.** *[Identifiability of the cICA tensor decomposition] The joint decomposition*

$$\kappa_4(\boldsymbol{y}) = \sum_{i=1}^{r} \lambda_i \boldsymbol{a}_i^{\otimes 4}, \qquad \kappa_4(\boldsymbol{x}) = \sum_{i=1}^{r} \lambda_i' \boldsymbol{a}_i^{\otimes 4} + \sum_{j=1}^{\ell} \nu_j \boldsymbol{b}_j^{\otimes 4},$$

*is unique for generic $\boldsymbol{a}_i, \boldsymbol{b}_j, \lambda_i, \lambda_i', \nu_j$, where $i \in [r]$ and $j \in [\ell]$, when $r + \ell < \lceil \frac{1}{p}\binom{p+3}{4} \rceil$ for $p \neq 3, 4, 5$, $r + \ell \leq \lceil \frac{1}{p}\binom{p+3}{4} \rceil$ for $p = 3, 5$, and when $r + \ell \leq 9$, $r + \ell \neq 8$ for $p = 4$.*

***Proof:*** The tensor decomposition for cICA in the statement is identifiable when the symmetric tensor decomposition of $\kappa_4(\mathbf{x})$ is unique, as follows. The tensor decomposition of $\kappa_4(\mathbf{x})$, gives vectors $\mathbf{a}_i, \mathbf{b}_j$ up to permutation and scaling. Then we can solve a linear system to find the decomposition $\kappa_4(\mathbf{y}) = \sum_{i=1}^{r} \lambda_i \mathbf{a}_i^{\otimes 4}$. It therefore remains to study the identifiability of the decomposition of $\kappa_4(\mathbf{x})$. It is a symmetric $p \times p \times p \times p$ tensor of rank $r+\ell$. Hence, the uniqueness follows from Lemma 3.1, setting $q = r + \ell$. □

When $(\lambda_1, \ldots, \lambda_r)$ and $(\lambda_1', \ldots, \lambda_r')$ are proportional, as vectors in $\mathbb{R}^r$, we have a stronger identifiability result than the one for two separate tensor decompositions in Proposition 3.2.

**Proposition 3.3.** *Consider the joint decomposition of*

$$\kappa_4(\boldsymbol{y}) = \sum_{i=1}^{r} \lambda_i \boldsymbol{a}_i^{\otimes 4}, \quad \kappa_4(\boldsymbol{x}) = \sum_{i=1}^{r} \lambda_i' \boldsymbol{a}_i^{\otimes 4} + \sum_{j=1}^{\ell} \nu_j \boldsymbol{b}_j^{\otimes 4}.$$

*Suppose that $(\lambda_1', \ldots, \lambda_r') = \mu(\lambda_1, \ldots, \lambda_r)$ for some $\mu \in \mathbb{R} \setminus \{0\}$. Suppose further that $\boldsymbol{a}_i, \boldsymbol{b}_j, \lambda_i, \mu, \nu_j$ for $i \in [r]$ and $j \in [\ell]$ are generic. Then, the joint decomposition of $\kappa_4(\boldsymbol{y})$ and $\kappa_4(\boldsymbol{x})$ is unique provided*

$$\max\left\{ \frac{1}{p}\left\lceil \frac{r}{\ell} \right\rceil + \ell, r \right\} < \frac{1}{p}\binom{p+3}{4}.$$

***Proof:*** We can assume that $r$ is a multiple of $\ell$: If the joint decomposition is unique with $r$ replaced by the possibly larger number $\lceil r/\ell \rceil \ell$, then the original joint decomposition with $r$ terms is also unique.

Let $k = \frac{r}{\ell}$ and define the tensors $T_1, \ldots, T_k$ by taking a subset of $\ell$ consecutive terms from $\kappa_4(\mathbf{y})$: $T_j = \sum_{i=(j-1)\ell+1}^{j\ell} \nu_i \mathbf{b}_i^{\otimes 4}$. Define

$$W = \operatorname{Span}\left\{ \kappa_4(\mathbf{x}), T_1, \ldots, T_k \right\}.$$

Then $\sum_{i=1}^{\ell} \nu_i \mathbf{b}_i^{\otimes 4} \in W$, since the difference between it and $\kappa_4(\mathbf{x})$ is $\mu(T_1 + \cdots + T_k)$.

Let $X \in \mathbb{P}^N$ be the variety of symmetric border rank at most $\ell$ tensors in $(\mathbb{R}^p)^{\otimes 4}$, where $N = \binom{p+3}{4} - 1$. The tensors

$$\sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes 4}, T_1, \ldots, T_k \qquad [12]$$

are generic points on $X$, since $\mathbf{a}_i, \mathbf{b}_j, \lambda_i, \nu_j$ are generic for $i \in [r], j \in [\ell]$. We have projective dimensions $\dim X \leq \ell p - 1$ and $\dim W = k$. When $k + \ell p < \binom{p+3}{4}$, we have the inequality

$$\dim X + \dim W < N.$$

Thus the intersection $W \cap X$ contains only the points in Eq. **12**, by the Generalized Trisecant Lemma (40, Proposition 2.6), The rank $r$ satisfies the condition in Lemma 3.1, since $rp < \binom{p+3}{4}$, so we can uniquely recover $T_1, \ldots, T_k$. We can thus recover the linear space $W$ and therefore we can recover $\sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes 4}$ from $W \cap X$. The decomposition of $\sum_{i=1}^{\ell} \nu_i b_i^{\otimes 4}$ is unique, since $p\ell < \binom{p+3}{4}$, and $\nu_j, \mathbf{b}_j$ are generic for $j \in [\ell]$. Hence, the overall joint decomposition is unique. □

***Remark 3.4.*** An alternative approach to study the identifiability of the joint decomposition is to stack $\kappa_4(\mathbf{x})$ and $\kappa_4(\mathbf{y})$ to form a partially symmetric tensor of size $2 \times p \times p \times p \times p$. This connects to the study of Segre-Veronese varieties (41). However, existing results do not apply to our setting, because the pair $(\kappa_4(\mathbf{x}), \kappa_4(\mathbf{y}))$ has additional structure: Proposition 3.3 is a first step toward identifiability for partially symmetric tensors with rank-one components that appear in a subset of slices.

We say that Algorithm 2 is identifiable if, for generic $\mathbf{a}_i, \mathbf{b}_j, \lambda_i, \lambda_i', \nu_j$, where $i \in [r], j \in [\ell]$, we can uniquely recover the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_r$, the coefficients $\lambda_1', \ldots, \lambda_r'$, and the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$.

**Proposition 3.5.** *Algorithm 2 is identifiable when $r + \ell \leq \binom{p+1}{2}$ for $p \neq 4$ and $r + \ell \leq 9$, $r, \ell \neq 8$ for $p = 4$.*

To prove Proposition 3.5, we use the following linear algebra result. See (37, Lemma B.1) for a proof.

**Lemma 3.6.** *Let $M \in \mathbb{R}^{n \times n}$, $U \in \mathbb{R}^{n \times k}$ and $V \in \mathbb{R}^{n \times k}$ be full-rank matrices with $k \leq n$. Let $C^* = (V^{\mathsf{T}} M^{-1} U)^{\dagger}$, where $\dagger$ denotes the pseudoinverse, and $d = \operatorname{rank}(C^*)$. Then*

$$\operatorname{rank}(M - UCV^{\mathsf{T}}) \geq n - d,$$

*with equality if and only if $C = C^*$.*

***Proof of Proposition 3.5:*** Tensors $\sum_{i=1}^{r} \lambda_i \mathbf{a}_i^{\otimes 4}$ and $\sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes 4}$ are generic rank $r$ and rank $\ell$ tensors, respectively. So, the identifiability of Steps 1 and 3 of Algorithm 2 hold if $r, \ell < \lceil \frac{1}{p}\binom{p+3}{4} \rceil$ for $p \notin \{3, 4, 5\}$ or $r, \ell \leq \lceil \frac{1}{p}\binom{p+3}{4} \rceil$ for $p \in \{3, 5\}$ or $r, \ell \leq 9$, $r, \ell \neq 8$ for $p = 4$, setting $q = r$ and $q = \ell$ in Lemma 3.1.

It remains to consider Step 2, learning the coefficients $\lambda_i'$ of $\mathbf{a}_i^{\otimes 4}$ in $\kappa_4(\mathbf{x})$. The flattening of $\kappa_4(\mathbf{x})$ has the form $M = \sum_{i=1}^{r} \lambda_i' \mathbf{A}_i^{\otimes 2} + \sum_{j=1}^{\ell} \nu_j \mathbf{B}_j^{\otimes 2} \in \mathbb{R}^{p^2 \times p^2}$, where $\mathbf{A}_i, \mathbf{B}_j \in \mathbb{R}^{p^2}$ vectorize $\mathbf{a}_i^{\otimes 2}$ and $\mathbf{b}_j^{\otimes 2}$, respectively. The scalar $\lambda_i'$ is unique if $\operatorname{rank}(M - \lambda_i' \mathbf{A}_i \otimes \mathbf{A}_i) = \operatorname{rank}(M) - 1$, by Lemma 3.6. It is $((\mathbf{A}_i^{\mathsf{T}} V) D^{-1} (\mathbf{A}_i^{\mathsf{T}} V)^{\mathsf{T}})^{-1}$, where $VDV^{\mathsf{T}}$ is the thin eigendecomposition of $M$. In particular, the coefficient $\lambda_i'$ is unique when

$$\mathbf{a}_i^{\otimes 2} \notin \operatorname{Span}(\{\mathbf{a}_1^{\otimes 2}, \ldots, \mathbf{a}_{i-1}^{\otimes 2}, \mathbf{a}_{i+1}^{\otimes 2}, \mathbf{a}_r^{\otimes 2}, \mathbf{b}_1^{\otimes 2}, \ldots, \mathbf{b}_\ell^{\otimes 2}\}).$$

For generic $\mathbf{a}_i$ and $\mathbf{b}_j$, this holds provided $r + \ell$ is at most $\binom{p+1}{2}$, the dimension of the space of $p \times p$ symmetric

matrices. Inequalities $\binom{p+1}{2} \le \lceil \frac{1}{p}\binom{p+3}{4} \rceil$ for $p \notin \{3,4,5\}$ and $\binom{p+1}{2} \le \lceil \frac{1}{p}\binom{p+3}{4} \rceil + 1$ for $p \in \{3,4,5\}$ hold. Combining the above conditions, Algorithm 2 is identifiable when $r+\ell \le \binom{p+1}{2}$ for $p \neq 4$ and $r+\ell \le 9$, $r, \ell \neq 8$ for $p = 4$. □

In some settings, we assume that the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ are orthogonal. In particular, $\ell \le p$. This assumption is natural for visualization purposes since the projection onto foreground patterns is orthogonal. In this case, HTD gives an exact decomposition, by Proposition 2.3. The identifiability requirements are the same as in Propositions 3.2 and 3.5, as follows. The identifiability conditions in the two propositions are unchanged under a change of basis by an invertible $p \times p$ matrix. When $\ell \le p$, we can apply a change of basis to $\kappa_4(\mathbf{x})$ so that the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ become orthogonal. We apply the same change of basis to $\kappa_4(\mathbf{y})$.

**3.2. cICA for Dimensionality Reduction.** Usual ICA has been used as a tool to project data, see refs. 31, 42, and 43. We extend this to cICA. In practice, the input to cICA consists of samples from the foreground $\mathbf{x}$ and background $\mathbf{y}$. These samples comprise the foreground data $X \in \mathbb{R}^{n\times p}$ and the background data $Y \in \mathbb{R}^{m\times p}$, where $n$ and $m$ are the numbers of samples in the foreground and background datasets respectively. We then construct the sample cumulants $\kappa_4(\mathbf{x})$ and $\kappa_4(\mathbf{y})$ as follows.

A dataset of $n$ samples in $\mathbb{R}^p$ gives a data matrix $X \in \mathbb{R}^{n\times p}$. Its fourth cumulant is computed as follows. Let $\bar{X} \in \mathbb{R}^p$ denote the mean vector over all observations. The $p \times p$ sample covariance matrix $\Sigma$ for $X$ has entries $\sigma_{ij} = \frac{1}{n}\sum_{t=1}^{n}(X_{ti} - \bar{X}_i)(X_{tj} - \bar{X}_j)$. The fourth-order central sample moment is a $p\times p\times p\times p$ tensor with entries $M_{ijkl} = \frac{1}{n}\sum_{t=1}^{n}(X_{ti}-\bar{X}_i)(X_{tj}-\bar{X}_j)(X_{tk}-\bar{X}_k)(X_{tl}-\bar{X}_l)$. Entry $(i,j,k,l)$ of the fourth-order sample cumulant is $M_{ijkl} - \sigma_{ij}\sigma_{kl} - \sigma_{ik}\sigma_{jl} - \sigma_{il}\sigma_{jk}$. If the data $X$ are samples from a distribution $\mathbf{x}$, this sample cumulant approximates $\kappa_4(\mathbf{x})$. The computation for $\kappa_4(\mathbf{y})$ is similar.

When $p$ is large, forming the fourth cumulants may be prohibitively expensive. To get around this, one can reduce the dimension before forming the cumulants, as follows.

We combine the foreground and background datasets to form a single dataset, a matrix of size $(m+n)\times p$. Let $U \in \mathbb{R}^{p\times k}$ have as its columns the top $k$ principal components of this combined data. The background and foreground transformed variables are then

$$U^\mathsf{T}A\mathbf{z} \qquad \text{and} \qquad U^\mathsf{T}A\mathbf{z}' + U^\mathsf{T}B\mathbf{s},$$

respectively, where $U^\mathsf{T}A \in \mathbb{R}^{k\times r}$ and $U^\mathsf{T}B \in \mathbb{R}^{k\times \ell}$. The recovered foreground patterns from cICA are the columns of $U^\mathsf{T}B$. The columns of $UU^\mathsf{T}B \in \mathbb{R}^{p\times\ell}$ convert these projected foreground patterns back into the original space.

In practice, for our data visualization in Section 4.3, we choose the number $k$ of PCA components to be 30 or the number of components that explains at least 90% variance, whichever comes first.

We compute the mixing matrix $B \in \mathbb{R}^{p\times\ell}$ with columns $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ using Algorithm 2. When employing cICA for dimensionality reduction, we project the foreground data $X$ onto $XB$. For a two-dimensional plot, we plot the projections $(X\mathbf{b}_i, X\mathbf{b}_j)$ for a pair $i,j$. To select the most relevant vectors out of our $\ell$ recovered vectors $\mathbf{b}_i \in \mathbb{R}^\ell$, we order them by the ratio

$$k(\mathbf{b}) := \frac{\mathbf{b}^\mathsf{T}\kappa_2(\mathbf{x})\mathbf{b}}{\mathbf{b}^\mathsf{T}\kappa_2(\mathbf{y})\mathbf{b}}. \qquad [13]$$

We justify this ranking and interpret the axes of a cICA dimensionality reduction plot in *SI Appendix*, section 5.

**3.3. Error Analysis for cICA.** Suppose we are in the setting of cICA, where the foreground and background datasets are described by ICA models

$$\mathbf{y} = A\mathbf{z}, \qquad \mathbf{x} = A\mathbf{z}' + B\mathbf{s}$$

and the population cumulant tensors are

$$\kappa_4(\mathbf{y}) = \sum_{i=1}^{r}\lambda_i\mathbf{a}_i^{\otimes 4}, \quad \kappa_4(\mathbf{x}) = \sum_{i=1}^{r}\lambda_i'\mathbf{a}_i^{\otimes 4} + \sum_{i=1}^{\ell}\nu_i\mathbf{b}_i^{\otimes 4}.$$

Let $\hat{\kappa}_4(\mathbf{y}), \hat{\kappa}_4(\mathbf{x})$ be the sample cumulant tensors for the two datasets. We prove the following upper bound on the error of estimating $\sum_{i=1}^{\ell}\nu_i\mathbf{b}_i^{\otimes 4}$.

**Theorem 3.7.** *Let $T = \sum_{i=1}^{\ell}\nu_i\boldsymbol{b}_i^{\otimes 4}$ and let $\hat{T}$ be the tensor obtained after Steps 1 and 2 of Algorithm 2 with input sample cumulant tensors $\hat{\kappa}_4(\boldsymbol{x}), \hat{\kappa}_4(\boldsymbol{y})$. Let $\rho = \max_{i\neq j}|\langle \boldsymbol{a}_i, \boldsymbol{a}_j\rangle|$, $M_y = \mathrm{Mat}(\kappa_4(\boldsymbol{y}))$ and $\Delta_M = \|M_y - \mathrm{Mat}(\hat{\kappa}_4(\boldsymbol{y}))\|_2$. Let $\sigma_r(M_y)$ denote the r-th largest singular value of $M_y$. Define*

$$\Delta_A = \frac{\Delta_M}{\sigma_r(M_y) - \Delta_M}, \quad \lambda = \min_i|\lambda_i|, \quad \lambda' = \lambda(1-(r-1)\rho).$$

*Under the assumptions that $(r-1)\rho = o(1)$, that $\Delta_M < \frac{\lambda}{45} + O(\rho)$, and moreover that $\max_i|\lambda_i'|\frac{2\sqrt{\Delta_A}+3\Delta_A}{\lambda'} = o(1)$, we have*

$$\|\hat{T} - T\|_F \le \|\hat{\kappa}_4(\boldsymbol{x}) - \kappa_4(\boldsymbol{x})\|_F + \beta\sqrt{\Delta_M} + O(\Delta_M),$$

*where $\beta = \sum_{i=1}^{r}(|\lambda_i'|\sqrt{\frac{2}{\lambda'}} + |\lambda_i'^2|2\lambda'^{-\frac{3}{2}})$.*

***Sketch Proof:*** Let $\mathbf{a}_i'$ be the estimate of $\mathbf{a}_i$ obtained via Step 1 of Algorithm 2, and let $\mu_i$ be the estimate of $\lambda_i'$ via Step 2 of Algorithm 2. Then $\|\hat{T} - T\|_F$ is at most

$$\|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \sum_{i=1}^{r}2|\mu_i|\|\mathbf{a}_i - \mathbf{a}_i'\| + \sum_{i=1}^{r}|\lambda_i' - \mu_i|, \qquad [14]$$

as can be shown using the triangle inequality and by comparing $\|\mathbf{a}_i^{\otimes 4} - \mathbf{a}_i'^{\otimes 4}\|$ and $\|\mathbf{a}_i - \mathbf{a}_i'\|$ for vectors $\mathbf{a}_i, \mathbf{a}_i'$. We will obtain bounds on the second and third terms in the sum Eq. **14**.

The distances between the numbers $\frac{1}{\lambda_i'}, \frac{1}{\mu_i}$ and between the vectors $\|\mathbf{a}_i - \mathbf{a}_i'\|$ can be bounded as

$$\|\mathbf{a}_i - \mathbf{a}_i'\| \le \sqrt{\frac{\Delta_A}{2}}, \quad \left|\frac{1}{\lambda_i'} - \frac{1}{\mu_i}\right| \le \frac{2}{\lambda'}\sqrt{\Delta_A} + O(\Delta_A), \qquad [15]$$

by applying results from the study of the optimization landscape of tensor decomposition (44). One can also show that

$$\sigma_r(M) \ge \lambda' = \lambda + O(\rho), \quad \Delta_A = \frac{\Delta_M}{\lambda'} + O(\Delta_M^2), \qquad [16]$$

by relating $\sigma_r(M_y)$ to $\sigma_r(G_2)$, where $G_2 \in \mathbb{R}^{r\times r}$ is the matrix with $(i,j)$ entry $\langle \mathbf{a}_i, \mathbf{a}_j\rangle^2$ and relating $\sigma_r(G_2)$ to $\rho$. Substituting Eqs. **15** and **16** into Eq. **14**, we obtain the result. □

We obtain the following end-to-end error bound for recovery of the foreground patterns and its coefficients via cICA, by combining Theorem 2.4, Theorem 3.7, and sample complexity results for cumulant tensors (45).

**Theorem 3.8.** *Suppose we have $N_1$ samples for the background dataset and $N_2$ samples for the foreground dataset. We can shift and scale our latent variables $z_i, z_i', s_j$ for $i, i' \in [r], j \in [\ell]$, so we assume without loss of generality that*

- $\mathbb{E}[z_i] = \mathbb{E}[z_i'] = \mathbb{E}[s_j] = 0$,
- $\mathbb{E}[z_i^2] = \mathbb{E}[z_i'^2] = \mathbb{E}[s_j^2] = 1$.

*Assume moreover that the fourth cumulants of $z_i, z_i', s_j$ are nonzero, and that the variables $z_i, z_i', s_j$ are sub-Gaussian. Suppose $\boldsymbol{c}_i$ are the output patterns of the cICA algorithm, with corresponding recovered scalars $\boldsymbol{\mu}_i$, obtained from the tensor of foreground patterns $T = \sum_{i=1}^{\ell} \nu_i \boldsymbol{b}_i^{\otimes 4}$. Under the assumptions of Theorem 2.4 and Theorem 3.7, we have*

$$|\nu_i - \mu_i| \le O(\epsilon^2) + \widetilde{O}(\delta),$$

$$\min\{\|\boldsymbol{b}_i - \boldsymbol{c}_i\|, \|\boldsymbol{b}_i + \boldsymbol{c}_i\|\} \le O(\epsilon^2) + \widetilde{O}(\delta)$$

*where*

$$|\langle \boldsymbol{b}_i, \boldsymbol{b}_j \rangle| \le \epsilon \quad \text{for all } i \ne j,$$

$$\delta = \widetilde{O}\left( \frac{p^{\frac{3}{2}} \ell'^2}{N_2} + \sqrt{\frac{\ell'^4}{N_2}} + \sqrt{\frac{pr'^2}{N_1} + \sqrt{\frac{r'^4}{pN_1}}} \right),$$

*$r' = \max\{r, p\}$, $\ell' = \max\{\ell, p\}$, and $\widetilde{O}$ absorbs polylog terms.*

***Remark 3.9.*** The $O(\epsilon^2)$ term in Theorem 3.8 captures model mismatch from the nonorthogonality of the true components. The $\widetilde{O}(\delta)$ term is error due to finite sample estimation of foreground patterns. Assuming $r$ and $\ell$ are $O(p)$, the $\widetilde{O}(\delta)$ term scales as

$$\widetilde{O}\left( \frac{p^{\frac{7}{2}}}{N_2} + \sqrt{\frac{p^4}{N_2}} + \sqrt{\frac{p^3}{N_1} + \sqrt{\frac{p^3}{N_1}}} \right).$$

We thus obtain a constant accuracy guarantee for recovering the foreground patterns and their coefficients if the background and foreground sample sizes satisfy

$$N_1 = \widetilde{O}(p^3), \qquad N_2 = \widetilde{O}(p^4).$$

These sample size requirements are beyond the optimal $O(p^2)$ sample complexity achievable by polynomial-time methods in ref. (46). The gap is due to two steps in our analysis that introduce dimension-dependent factors: i) bounding the spectral norm of $\hat{\kappa}_4(\mathbf{y}) - \kappa_4(\mathbf{y})$ by that of its flattening, and ii) converting between spectral and Frobenius norms for $\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})$.

An interesting direction for future work is to improve the sample efficiency, for instance using the structure of the tensors $\hat{\kappa}_4(\mathbf{y}) - \kappa_4(\mathbf{y})$ and $\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})$, by prewhitening the data, or by decomposing the stacked foreground and background cumulant tensors as a single tensor of size $p \times p \times p \times p \times 2$ to avoid the three-step procedure.

## 4. Numerical Experiments

We compare Algorithm 2 with other tensor decompositions and ICA methods to illustrate the necessity of HTD (Section 4.1). We investigate the performance of cICA for finding patterns in data (Section 4.2) and for data visualization (Section 4.3). Our code is available on GitHub at https://github.com/QWE123665/cICA.

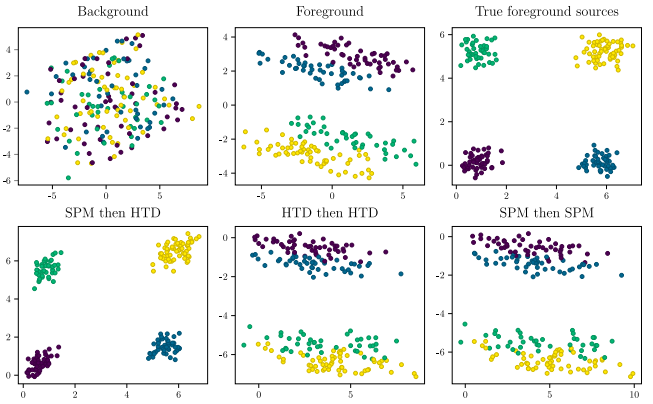


**Fig. 3.** We compare our algorithm SPM-HTD against other ICA and tensor decomposition methods in a synthetic setting to justify our algorithmic choices. The top-left and top-middle subplots illustrate the background and foreground datasets, each consisting of 200 samples in $\mathbb{R}^5$, projected onto their two leading principal components. The top-right subplot shows the foreground dataset projected onto the true foreground mixing matrix $B \in \mathbb{R}^{5\times 2}$, revealing four clusters. In the bottom row, we compare our algorithm (SPM-HTD) with applying HTD in both Steps 1 and 3, and applying SPM in both steps. Only our method (SPM-HTD, bottom-left) recovers the four clusters.

**4.1. Choices of Methods in Algorithm 2.** We evaluate our approach in Algorithm 2. We compare our method (SPM-HTD) against several alternative combinations involving SPM (37), HTD (Algorithm 1), FastICA (15), FOOBI (20), and JADE (18). The evaluated combinations include SPM-HTD, HTD-HTD, SPM-SPM, SPM-JADE, JADE-HTD, JADE-JADE, FOOBI-HTD, SPM-FOOBI, FOOBI-FOOBI, and FastICA-HTD.

Our setup has three background patterns and two foreground patterns. The background patterns are three independent uniform random variables. The foreground patterns are two mixtures of beta distributions $0.5B(2,5) + 0.5B(5,4)$. The foreground mixing matrix $B \in \mathbb{R}^{5\times 2}$ consists of the last two columns of the identity matrix $I_5$. The background mixing matrix $A \in \mathbb{R}^{5\times 3}$ is randomly generated and adjusted to ensure small inner products with columns of $B$.

We generate foreground and background datasets, each with 200 samples. Their projections to the leading two principal components are the first two subplots of Fig. 3. Projecting the foreground dataset via matrix $B$ reveals four distinct clusters, see the top-right subplot of Fig. 3.

We illustrate the performance of our algorithm SPM-HTD and the variants SPM-SPM, HTD-HTD in the second row of Fig. 3. SPM-HTD is the only method of the three to recover the four clusters. The performance of the other competing methods is in *SI Appendix*, section 6.1. All methods that find the four clusters use an ICA or tensor decomposition method in Step 1 and HTD in Step 3.

We vary the sample size of both datasets from 100 to 1,000. For each sample size, we repeat the experiment 20 times by randomly drawing datasets, applying all 11 methods to estimate the matrix $B$, and computing the silhouette score on the foreground data projected via the estimated $B$. A higher silhouette score indicates better recovery of the four clusters. To mitigate randomness, we record the best silhouette score from 20 independent runs for each method and then average these across experiments.

Fig. 4 compares silhouette scores for methods that apply an ICA or tensor decomposition approach in the first step followed by HTD (JADE-HTD, SPM-HTD, FOOBI-HTD, FastICA-

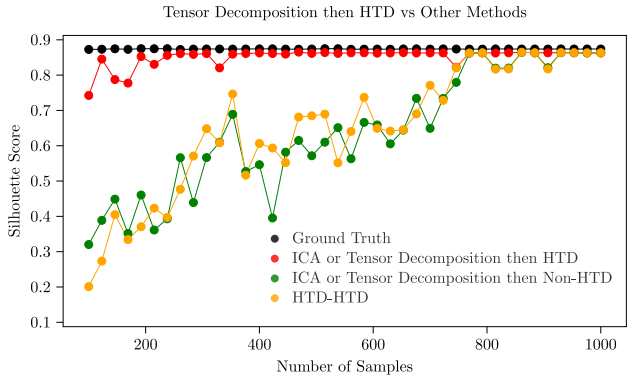


**Fig. 4.** We study the accuracy of different approaches to cICA as the number of samples varies. We compare methods using an ICA or tensor decomposition method followed by HTD (red), ICA, or tensor decomposition methods followed by non-HTD alternatives (green) and HTD-HTD (yellow). Performance is evaluated using the silhouette score, which measures how effectively the estimated matrix recovers the four clusters shown in the top-right plot of Fig. 3. Methods using ICA or tensor decomposition method followed by HTD outperform both non-HTD approaches and the HTD-HTD combination. This justifies our decision to use SPM in Step 1 and HTD in Step 3 of our cICA algorithm.

HTD) to methods that do not use HTD in the third step. It shows that methods using tensor decomposition or an ICA approach followed by HTD achieve superior silhouette scores, highlighting the importance of HTD in Step 3. The HTD-HTD method underperforms approaches combining another tensor decomposition method with HTD, revealing the necessity of an accurate decomposition in Step 1. The best choice in Step 1 cycles between FOOBI, FastICA, and SPM. We choose SPM for compatibility with Step 2. FastICA does not directly process cumulant tensors, making it unsuitable for Step 2.

**4.2. Salient Patterns.** The cICA patterns are the foreground vectors $\mathbf{b}_i$. We investigate the interpretability of the cICA patterns on synthetic, semisynthetic, and real-world datasets. We demonstrate that cICA recovers foreground patterns accurately for synthetic data, with comparisons to cPCA (5) and PCPCA (7). Our semisynthetic setup has background dataset consisting of images of grass and clouds from ref. 47. The foreground dataset consists of digits 0 and 1 superimposed, with varying intensity, onto images of grass and clouds. We find that, unlike other methods, cICA is able to recover as top two foreground patterns the digits 0 and 1. Additionally, we apply cICA to gene expression data from ref. 48, using monkey gene expression as the background and human gene expression as the foreground. We compare the cICA foreground patterns to results to identify genes responsible for human evolution.

***4.2.1. Synthetic data.*** We use synthetic data to assess the accuracy of the patterns recovered by cICA. We compare against cPCA and PCPCA, illustrating that cICA algorithms recover the foreground patterns more accurately when generated under a model Eq. **4** that assumes independence of latent variables, see Fig. 5. The details of the simulations are in *SI Appendix*, section 6.2.1.

We see from Fig. 5 that cICA outperforms cPCA and PCPCA in recovering the foreground patterns. Fig. 5 (*Top*) shows that the interquartile range for cICA in Algorithm 2 is above the maximum cosine similarity results for cPCA and PCPCA. The best performing cICA has cosine similarity above 0.9 for all tested $p$. Fig. 5 (*Bottom*) shows analogous results with accuracy measured via relative Frobenius norm. The variability as $p$ changes is due to randomness in the matrix $A$. The method

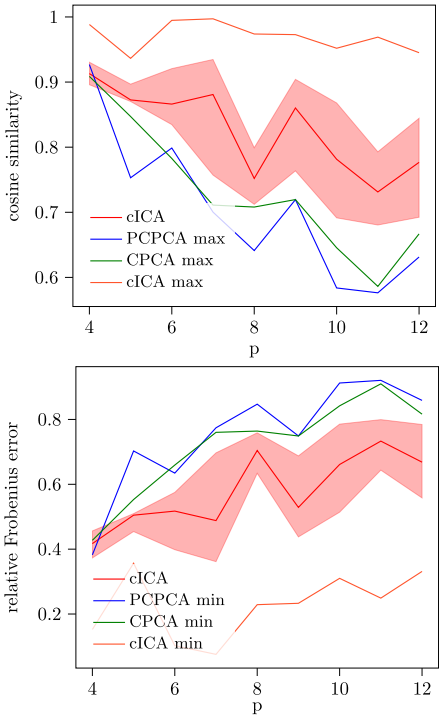


**Fig. 5.** The similarity of the recovered vs. true foreground patterns (i.e., the accuracy of recovering matrix $B$), measured via cosine similarity (*Top*) and relative Frobenius error (*Bottom*), via cICA in Algorithm 2. The interquartile range over 100 runs is shaded in red, with the best run shown as the red line. For cPCA and PCPCA, we test 100 hyperparameter values and plot the one with the lowest error.

outperforms cPCA and PCPCA, with the added benefit that no selection of hyperparameters is necessary.

***4.2.2. Corrupted MNIST dataset with continuous strength.*** We superimpose hand-written digits 0 and 1 from MNIST (49) onto grass and cloud images from ref. 47. The background dataset consists of 5,000 cloud images and 5,000 grass images. For the foreground dataset, we sample 8,000 grass and 2,000 cloud images. Next, we sample 10,000 pairs of images of digits 0 and 1 and superimpose them on the foreground grass and cloud images with independent strength following Uniform$[0, 1]$. Digits 0 and 1 images are expected to be the foreground patterns. The background patterns come from decomposing grass and cloud images and the ratios of grass and cloud images in the background vs. foreground reflects that the coefficient of the background signals may not be proportional, which often happens in reality. That is, the foreground-to-background ratio $\lambda_i'/\lambda_i$ from equation Eq. **5** would be $0.4 = 2{,}000/5{,}000$ for a pattern in the clouds and $1.6 = 8{,}000/5{,}000$ for a pattern in the grass. Samples of the foreground and background images are shown in Fig. 6.

To interpret the cICA patterns, we plot the vectors as grayscale images. We expect the images from the top two cICA patterns to look like 0 and 1. We also plot the top two images for cPCA and PCPCA for comparison. See Fig. 7. The cICA images most closely resemble the images obtained from averaging the sampled digits 0 and 1 images weighted by uniform strength. In the other methods, one component learned is a combination of the two digits 0 and 1. For details, see *SI Appendix*, section 6.2.2.

***4.2.3. Human and monkey gene expression data.*** We apply cICA to a dataset of human and monkey gene expression from ref. 48, in which the authors analyze human, chimp, gorilla, macaque, and marmoset datasets to identify genes that are

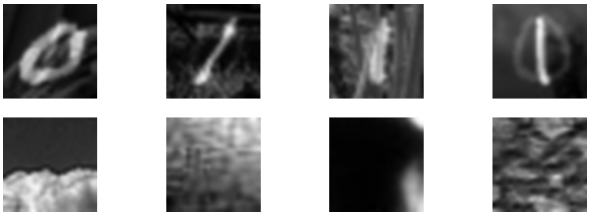

**Fig. 6.** Foreground (*Top*) and background images (*Bottom*) for the corrupted MNIST dataset.

responsible for evolutionary change. Out of 14,131 genes, they identify 3,383 genes with extensive differences between human and nonhuman primates, of which they identify a subset of 139 with deeply conserved coexpression across all nonhuman animals, and strongly divergent coexpression relationships in humans.

The idea is that the foreground patterns should be gene modules (considered as linear combinations of genes) that contribute to the human dataset but not the monkey dataset. By analogy to the MNIST dataset in the previous subsection, the foreground gene modules correspond to the digits 0 and 1. We evaluate the quality of the foreground patterns by testing its consistency with (48).

We select the 15 most variable genes among the 139 selected genes and the 15 most variable genes among the other $3{,}244 = 3{,}383-139$ genes. We combine 10,000 chimp and 10,000 gorilla data points to form the background dataset $Y \in \mathbb{R}^{20{,}000\times 30}$ and 10,000 human gene expression data points for the foreground dataset $X \in \mathbb{R}^{10{,}000\times 30}$. Then we apply cICA as in Algorithm 2 and use Eq. **13** to order the $\mathbf{b}_i$ and extract the first two vectors $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{30}$. We observe that the 15 genes with the highest absolute values in $\mathbf{b}_1$ (resp. $\mathbf{b}_2$) have 10 (resp. 13) genes among the 15 selected genes that come from the subset of 139 in ref. 48. This demonstrates consistency with the results from ref. 48: The vectors $\mathbf{b}_i$ assign higher weights to the genes from the subset of 139. In comparison, cPCA identifies 9 and 10 genes in its first two patterns and PCPCA identifies 10 and 11 genes.

We also report the number of genes misclassified by the methods, the size of the intersection of the $3{,}244 = 3{,}383 - 139$ evolution-irrelevant genes with the two sets of 15 genes in the foreground patterns (those with largest absolute values for $\mathbf{b}_1, \mathbf{b}_2$). The result can be found in Table 1. We see that cICA outperforms the other methods, with more recovered genes and fewer misclassified genes. The details of the experiments are in *SI Appendix*, section 6.2.3.

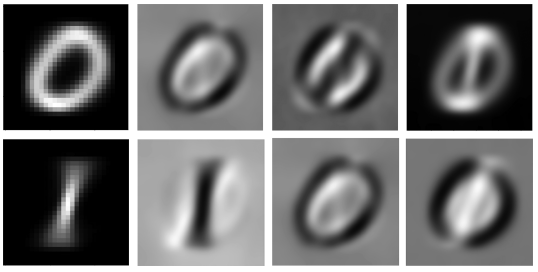

**Fig. 7.** Average images for digits 0 and 1 (first column). Patterns recovered plotted as images for cICA (second column), cPCA (third column), and PCPCA (fourth column).

**Table 1. Number of genes misclassified for the human-monkey gene expression data**

| Method | # Misclassified genes |
|---|---|
| cICA | 5 |
| ICA | 7 |
| PCPCA | 7 |
| cPCA | 9 |
| PCA | 9 |

**4.3. Dimensionality Reduction.** We use cICA for dimensionality reduction and data visualization, as described in Section 3.2. We investigate the performance of cICA on two datasets: mouse protein expression and corrupted MNIST images with discrete strength. Additional numerical experiments on transplant gene expression data are in *SI Appendix*, section 7.1. We quantify the performance of the methods using the silhouette score (50) of the projected data; higher values indicate better clustering of points.

***4.3.1. Mouse protein data.*** We study the mouse protein dataset from ref. 51. The foreground data measure protein expression in the cortex of mice subjected to shock therapy, some of whom have Down syndrome. The background dataset consists of protein expression measurements from mice without Down Syndrome who did not receive shock therapy. We compare cICA, ICA, as well as cPCA and PCPCA. All four algorithms can separate the two clusters in the foreground data, corresponding to mice with Down syndrome and those without, though the projections differ: cICA has the highest Silhouette score (0.606), followed by ICA (0.604), then cPCA (0.421), and then PCPCA (0.220); see Fig. 8. We consider the absolute values of the foreground-to-background cumulant ratios $|\lambda_i'/\lambda_i|$, for $\lambda_i, \lambda_i'$ defined in equation Eq. **5**. For $\mathbf{a}_1, \ldots, \mathbf{a}_r$, these range from $1.3 \times 10^{-4}$ to 0.12. Moreover, the foreground cumulants for $\mathbf{a}_1, \ldots, \mathbf{a}_r$ are in the range [0.1,30] while the foreground cumulants for $\mathbf{b}_1, \ldots, \mathbf{b}_5$ are much larger (in the range [200, 10,000]). This implies that the

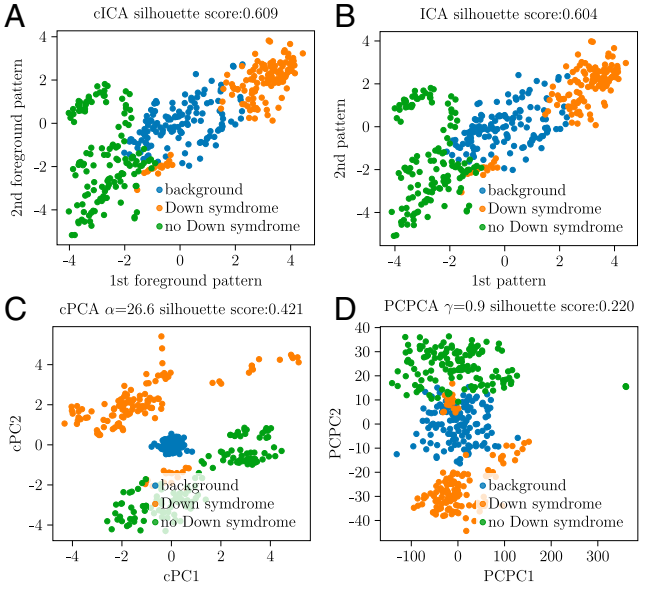


**Fig. 8.** Dimensionality reduction of the mouse protein data (51) via (*A*) cICA, (*B*) ICA, (*C*) cPCA, and (*D*) PCPCA. For (*A*), we fix a random seed. For (*B–D*), we plot the projection with the best silhouette score over 100 hyperparameter values.

**Fig. 9.** Foreground (*Top*) and background images (*Bottom*) for the mixed corrupted MNIST dataset .

background patterns are not obvious in the foreground dataset $X$ and explains the small difference between the experimental results for cICA and ICA. See *SI Appendix*, section 6.3.1 for details.

***4.3.2. Corrupted MNIST data with discrete strength.*** We superimpose hand-written digits 0 and 1 from MNIST (49) onto grass and cloud images from ref. 47. The background dataset consists of 5,000 cloud images and 5,000 grass images. For the foreground dataset, we sample 8,000 grass and 2,000 cloud images to create different foreground-to-background cumulant ratios for $\lambda_i'/\lambda_i$ in equation Eq. **5**. Similar to the corrupted MNIST data with continuous strength, we expect a ratio of 0.4, while for grass images, we expect a ratio of 1.6. Next, we sample 2,500 digit 0, 2,500 digit 1 images and form 2,500 images consisting of both digit 0 and digit 1. We then superimpose 2,500 digit 0, 2,500 digit 1, and 2,500 combined digit 0 and 1 images onto a randomly chosen subset of the background, as shown in the top row of Fig. 9. The inclusion of digits 0, 1, both, and none is to make the images of 0 and 1 independent patterns. Each image is of size $28 \times 28$.

We plot the 5,000 images of digits 0 or 1 superimposed on grass or cloud images using their inner product with the patterns learned in cICA, ICA, cPCA, and PCPCA. The plots are shown in Fig. 10. The algorithm cICA has the highest silhouette score (0.61), followed by cPCA (0.52), then PCPCA (0.44), then ICA (0.30). We also report the performance of each of the patterns for

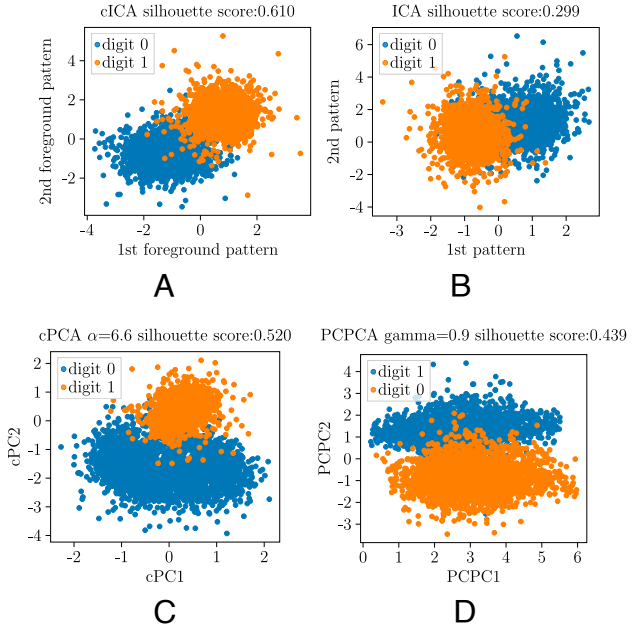


**Fig. 10.** Dimensionality reduction plots of the mixed corrupted MNIST data via (*A*) cICA, (*B*) ICA, (*C*) cPCA, and (*D*) PCPCA.

**Table 2. Classification accuracies for identifying digits 0 or 1 from corrupted images from each of the top two foreground patterns**

| Method | First pattern (%) | Second pattern (%) |
|---|---|---|
| cICA | 94 | 93 |
| cPCA | 71 | 94 |
| PCPCA | 50 | 94 |

classifying the digits 0 or 1 from the corrupted images using the sign of their inner product with the pattern. The classification accuracies for cICA, cPCA, and PCPCA are in Table 2. Both foreground cICA patterns can separate the digits 0 and 1 images with more than 0.9 accuracy, while cPCA and PCPCA only have one pattern that achieves this. See *SI Appendix*, section 6.3.2 for details.

## 5. Conclusion

We have presented cICA, a tool to explore patterns and visualize data in one setting relative to another. Unlike existing contrastive methods, cICA can model background patterns that each contribute to the foreground in different relative amounts $\lambda_i'/\lambda_i$. We designed an algorithm for cICA based on a hierarchical tensor decomposition (HTD). The algorithm uses linear algebra to decompose symmetric $p \times p \times p \times p$ tensors of rank at most $p^2$, encouraging orthogonality between rank-1 components. We use cICA to find salient patterns that describe a foreground dataset relative to a background, testing the results on synthetic, semisynthetic, and real-world datasets. We saw that it can extract foreground patterns of interest and is competitive with other methods.

We investigated the identifiability of cICA, via the uniqueness of its associated coupled tensor decomposition, seeing improvements relative to cPCA and PCPCA. This echoes the improved identifiability of ICA over PCA: a general linear mixing can be recovered uniquely via ICA, whereas PCA requires an orthogonal mixing.

We conclude with two directions for further study. This cICA model describes observations as a linear mixing of independent latent variables. Dropping the linearity assumption, we may seek patterns that have nonlinear signatures across the observed variables. This would combine the nonlinear contrastive methods of refs. 8, 22, 29, and 30 with approaches to find interpretable patterns, generalizing the vectors $\mathbf{b}_i$. Finally, dropping the independence assumption on the latent variables would connect cICA to other latent variable models such as those arising in causal disentanglement (52, 53).

## Materials and Methods

In our algorithm, we choose the rank by inspecting the singular values of the flattenings of the cumulant tensors. Details of the rank selection are provided in *SI Appendix*. Numerical experiments were conducted on both synthetic and real datasets. The synthetic datasets were generated in Python to model mixtures of statistically independent sources. Experiments on real data were conducted using publicly available image and gene expression datasets from refs. 48, 49, and 51.

**Data, Materials, and Software Availability.** The code and preprocessed data in all experiments are available on GitHub (https://github.com/QWE123665/cICA (54). Imagenet: A large-scale hierarchical image database. The mnist database of handwritten digit images for machine learning research. Comparative single-cell transcriptomic analysis of primate brains highlights human-

specific regulatory evolution. Self-organizing feature maps identify proteins critical to learning in a mouse model of Down syndrome. Massively parallel digital transcriptional profiling of single cells.

**ACKNOWLEDGMENTS.** We thank Salil Bhate for helpful discussions. A.M. and A.S. were partially supported by the NSF (DMS-2306672 and DMR-2011754).

1. B. Flury, Some relations between the comparison of covariance matrices and principal component analysis. *Comput. Stat. Data Anal.* **1**, 97–109 (1983).
2. B. N. Flury, Common principal components in k groups. *J. Am. Stat. Assoc.* **79**, 892–898 (1984).
3. B. K. Flury, Two generalizations of the common principal component model. *Biometrika* **74**, 59–69 (1987).
4. J. Y. Zou, D. J. Hsu, D. C. Parkes, R. P. Adams, Contrastive learning using spectral methods. *Adv. Neural Inf. Process. Syst.* **26** (2013).
5. A. Abid, M. J. Zhang, V. K. Bagaria, J. Zou, Contrastive principal component analysis. arXiv [Preprint] (2017). https://arxiv.org/abs/1709.06716 (Accessed 1 July 2024).
6. A. Abid, M. J. Zhang, V. K. Bagaria, J. Zou, Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nat. Commun.* **9**, 2134 (2018).
7. D. Li, A. Jones, B. Engelhardt, Probabilistic contrastive dimension reduction for case-control study data." *Ann. Appl. Stat.* **18.3**, 2207–2229 (2024).
8. K. A. Severson, S. Ghosh, K. Ng, "Unsupervised learning with contrastive latent variable models" in *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. **33**, pp. 4862–4869.
9. P. Comon, C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications* (Academic Press, 2010).
10. B. Ans, J. Hérault, C. Jutten, Architectures neuromimétiques adaptatives: Détection de primitives. *Proc. Cogn.* **85**, 593–597 (1985).
11. P. Comon, Independent component analysis, a new concept? *Signal Process.* **36**, 287–314 (1994).
12. M. S. Bartlett, J. R. Movellan, T. J. Sejnowski, Face recognition by independent component analysis. *IEEE Trans. Neural Networks* **13**, 1450–1464 (2002).
13. T. P. Jung *et al.*, Imaging brain dynamics using independent component analysis. *Proc. IEEE* **89**, 1107–1122 (2001).
14. S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, M. Jordan, A linear non-Gaussian acyclic model for causal discovery. *J. Mach. Learn. Res.* **7**, 2003–2030 (2006).
15. A. Hyvarinen, R. Cristescu, E. Oja, "A fast algorithm for estimating overcomplete ICA bases for image windows" in *Proceedings of the International Joint Conference on Neural Networks, IJCNN'99 (Cat. No. 99CH36339)* (IEEE, 1999), vol. **2**, pp. 894–899.
16. J. Eriksson, V. Koivunen, Identifiability, separability, and uniqueness of linear ICA models. *IEEE Signal Process. Lett.* **11**, 601–604 (2004).
17. K. Wang, A. Seigal, Identifiability of overcomplete independent component analysis. arXiv [Preprint] (2024). https://arxiv.org/abs/2401.14709 (Accessed 1 July 2024).
18. J. F. Cardoso, A. Souloumiac, "Blind beamforming for non-Gaussian signals" in *IEE Proceedings F (Radar and Signal Processing)* (IET, 1993), vol. **140**, pp. 362–370.
19. L. De Lathauwer, B. De Moor, J. Vandewalle, Independent component analysis and (simultaneous) third-order tensor diagonalization. *IEEE Trans. Signal Process.* **49**, 2262–2271 (2001).
20. L. De Lathauwer, J. Castaing, J. F. Cardoso, Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Trans. Signal Process.* **55**, 2965–2973 (2007).
21. P. McCullagh, *Tensor Methods in Statistics: Monographs on Statistics and Applied Probability* (Chapman and Hall/CRC, 2018).
22. A. Abid, J. Zou, Contrastive variational autoencoder enhances salient features. arXiv [Preprint] (2019). https://arxiv.org/abs/1902.04601 (Accessed 1 July 2024).
23. R. A. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Work. Pap. Phonetics* **16**, 84 (1970).
24. T. G. Kolda, Symmetric orthogonal tensor decomposition is trivial. arXiv [Preprint] (2015). https://arxiv.org/abs/1503.01375 (Accessed 1 July 2024).
25. N. Sturma, C. Squires, M. Drton, C. Uhler, Unpaired multi-domain causal representation learning. *Adv. Neural Inf. Process. Syst.* **36** (2024).
26. A. Hyvarinen, H. Morioka, Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. *Adv. Neural Inf. Process. Syst.* **29** (2016).
27. A. Hyvarinen, H. Sasaki, R. Turner, "Nonlinear ICA using auxiliary variables and generalized contrastive learning" in *The 22nd International Conference on Artificial Intelligence and Statistics* (PMLR, 2019), pp. 859–868.
28. Q. Lyu, X. Fu, "On finite-sample identifiability of contrastive learning-based nonlinear independent component analysis" in *International Conference on Machine Learning* (PMLR, 2022), pp. 14582–14600.
29. E. Weinberger, N. Beebe-Wang, S.-I. Lee, "Moment matching deep contrastive latent variable models" in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics* (PMLR, 2022), vol. 151, pp. 2354–2371.
30. R. Lopez, J. C. Huetter, E. Hajiramezanali, J. K. Pritchard, A. Regev, "Toward the identifiability of comparative deep generative models" in *Causal Learning and Reasoning*, F. Locatello, V. Didelez, Eds. (PMLR, 2024), pp. 868–912.
31. L. H. Lim, J. Morton, "Cumulant component analysis: A simultaneous generalization of PCA and ICA" in *CASTA2008* (2008), vol. **18**.
32. J. M. Landsberg, *Tensors: Geometry and Applications* (American Mathematical Society, 2011), vol. 128.
33. W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus* (Springer, 2012), vol. 42.
34. J. Salmi, A. Richter, V. Koivunen, Sequential unfolding SVD for tensors with applications in array signal processing. *IEEE Trans. Signal Process.* **57**, 4719–4733 (2009).
35. E. Robeva, Orthogonal decomposition of symmetric tensors. *SIAM J. Matrix Anal. Appl.* **37**, 86–102 (2016).
36. Y. Yu, T. Wang, R. J. Samworth, A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika* **102**, 315–323 (2015).
37. J. Kileel, J. M. Pereira, Subspace power method for symmetric tensor decomposition. *Numer. Algorithms*, 1–38 (2025).
38. Ah. J. Alexander, Polynomial interpolation in several variables. *J. Algebraic Geom.* **4**, 201–222 (1995).
39. L. Chiantini, G. Ottaviani, N. Vannieuwenhoven, On generic identifiability of symmetric tensors of subgeneric rank. *Trans. Am. Math. Soc.* **369**, 4021–4042 (2017).
40. L. Chiantini, C. Ciliberto, Weakly defective varieties. *Trans. Am. Math. Soc.* **354**, 151–178 (2002).
41. H. Abo, M. C. Brambilla, F. Galuppi, A. Oneto, Non-defectivity of Segre–Veronese varieties. *Proc. Am. Math. Soc. Ser. B* **11**, 589–602 (2024).
42. K. Domino, The use of fourth order cumulant tensors to detect outlier features modelled by a t-student copula. arXiv [Preprint] (2018). https://arxiv.org/abs/1804.00541 (Accessed 1 July 2024).
43. X. Geng, L. Wang, NPSA: Nonorthogonal principal skewness analysis. *IEEE Trans. Image Process.* **29**, 6396–6408 (2020).
44. J. Kileel, T. Klock, J. M. Pereira, Landscape analysis of an improved power method for tensor decomposition. *Adv. Neural Inf. Process. Syst.* **34**, 6253–6265 (2021).
45. A. Anandkumar, R. Ge, M. Janzamin, Sample complexity analysis for learning overcomplete latent variable models through tensor methods. arXiv [Preprint] (2014). https://arxiv.org/abs/1408.0553 (Accessed 1 July 2024).
46. A. Auddy, M. Yuan, Large-dimensional independent component analysis: Statistical optimality and computational tractability. *Ann. Stat.* **53**, 477–505 (2025).
47. J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database" in *2009 IEEE Conference on Computer Vision Pattern Recognition* (2009), pp. 248–255.
48. H. Suresh *et al.*, Comparative single-cell transcriptomic analysis of primate brains highlights human-specific regulatory evolution. *Nat. Ecol. Evol.* **7**, 1930–1943 (2023).
49. L. Deng, The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **29**, 141–142 (2012).
50. P. J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
51. C. Higuera, K. J. Gardiner, K. J. Cios, Self-organizing feature maps identify proteins critical to learning in a mouse model of Down syndrome. *PLoS ONE* **10**, e0129126 (2015).
52. M. Yang *et al.*, "CausalVAE: Disentangled representation learning via neural structural causal models" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 9593–9602.
53. C. Squires, A. Seigal, S. S. Bhate, C. Uhler, "Linear causal disentanglement via interventions" in *International Conference on Machine Learning* (PMLR, 2023), pp. 32540–32560.
54. K. Wang, A. Seigal, cICA: Contrastive Independent Component Analysis data and code repository. GitHub. https://github.com/QWE123665/cICA. Accessed 24 November 2025.