Supporting Information for

Contrastive independent component analysis for salient patterns and dimensionality reduction

Kexin Wang, Aida Maraj, Anna Seigal

Corresponding Author Anna Seigal.
E-mail: aseigal@seas.havard.edu

This PDF file includes:

Supporting Information Text

1. Comparison of HTD with other tensor decomposition methods

1.1. Comparison of HTD with other hierarchical tensor decompositions. We compare HTD in Algorithm 1 to other hierarchical tensor decompositions. The goal of hierarchical tensor decomposition (1, Chapter 11) is to efficiently represent a tensor that lives in a high-dimensional space. Given a tensor of order d, a hierarchical decomposition is based on a hierarchy of vector spaces given by a dimension partition tree on indices $\{1,\dots,d\}$, such as those in Figure S1.

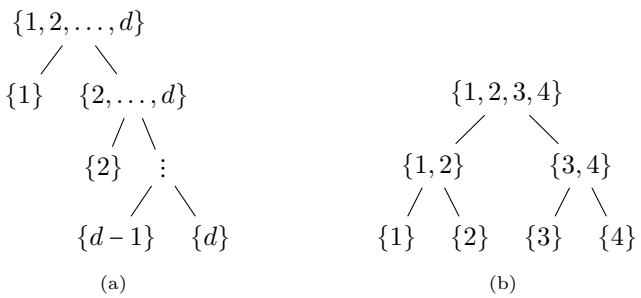

Fig. S1. The dimension partition trees used in (a) the PARATREE algorithm of (2) and (b) our HTD from Algorithm 1.

Hierarchical tensor representations in (1, Chapter 11) start at the leaves of the tree, which are labeled by single indices. One finds subspaces $U_i \subseteq \mathbb{R}^{n_i}$ such that the tensor is well-approximated by a tensor in the lower-dimensional space $U_1 \otimes \cdots \otimes U_d \subset \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_d}$. Proceeding from leaves to the root, when two indices $\{i\}$ and $\{j\}$ combine to form the subset $\{i,j\}$, the representation finds a subspace $U_{ij} \subset U_i \otimes U_j$ that well-approximates the tensor. This repeats until we have a low-dimensional subspace $U_{1\cdots d} \subseteq \mathbb{R}^{n_1} \otimes \cdots \otimes \mathbb{R}^{n_d}$ such that the tensor T lies in this subspace to reasonable accuracy. Fixing ranks in the representation fixes the allowable dimension of the subspaces U_I for the subsets $I \subseteq [d]$ in the tree. See (1, Figure 11.1).

The PARATREE model starts at the root of the tree. For example, if the root is the splitting of $\{1,2,3\}$ into $\{1\} \cup \{2,3\}$ (i.e. Figure S1 in the case $d = 3$) then one computes a decomposition of the flattened tensor in $\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2 n_3}$ to give a sum $\sum_{i=1}^{r_1} \mathbf{u}_i \otimes \mathbf{x}_i$, with $\mathbf{u}_i \in \mathbb{R}^{n_1}$ and $\mathbf{x}_i \in \mathbb{R}^{n_2 n_3}$. The second step is the splitting of indices $\{2,3\} = \{2\} \cup \{3\}$. This decomposes each vector $\mathbf{x}_i = \sum_{j=1}^{r_2} \mathbf{v}_{i,j} \otimes \mathbf{w}_{i,j}$, where $\mathbf{x}_i \in \mathbb{R}^{n_2 n_3}$ is viewed as a matrix of size $n_2 \times n_3$. This results in the decomposition

$$T = \sum_{i=1}^{r_1} \mathbf{u_i} \otimes \left(\sum_{j=1}^{r_2} \mathbf{v}_{i,j} \otimes \mathbf{w}_{i,j} \right). \qquad [1]$$

This pattern can be continued for larger d, see (2, Equation 9).

Our HTD takes a symmetric $p \times p \times p \times p$ tensor as input. We use the dimension partition tree in Figure S1(b). HTD can be viewed as a symmetric analog of the PARATREE model, but differs in that it uses a different dimension partition tree, and leverages the symmetry of the tensor and decomposition to produce a rank r decomposition, rather than the rank $r_1 r_2$ (or, more generally, rank $r_1 \cdots r_{d-1}$) decomposition obtained from [1]. Compared to the hierarchical tensor representations of (1, Chapter 11), it differs in that the tensor is symmetric and it uses the dimension partition tree from root to leaves rather than leaves to root.

1.2. Comparison of HTD with other linear algebra based tensor decompositions. We compare HTD in Algorithm 1 to other linear algebra based tensor decompositions.

Jennrich's Algorithm (3) decomposes an order 3 tensor $T = \sum_{i=1}^{r} \mathbf{u}_i \otimes \mathbf{v}_i \otimes \mathbf{w}_i$, requiring $\mathbf{u}_1,\dots,\mathbf{u}_r$ to be linearly independent and $\mathbf{v}_1,\dots,\mathbf{v}_r$ to be linearly independent. It computes two matrices $M_z = T(:,:,z), M_{z'} = T(:,:,z')$ for random unit norm vectors z, z' and then computes eigendecompositions of $M_z M_{z'}^{+}$ and $M_{z'} M_z^{+}$. The decomposition of T can then be recovered via pairing the eigenvalues of the two eigendecompositions. When applying Jennrich's algorithm to an order-4 symmetric tensor, we need to flatten the 3rd and 4th dimensions of the tensor to form an order-3 tensor first. It can decompose a symmetric $p \times p \times p \times p$ tensor of rank at most p due to the linear independence requirement and it takes $O(p^4)$ operations, where the most costly step is forming the matrices M_z and $M_{z'}$.

Orthogonal symmetric decomposition (4) decomposes a symmetric tensor $T = \sum_{i=1}^{r} \mathbf{u}_i^{\otimes d}$ where $\mathbf{u}_1,\dots,\mathbf{u}_r$ are orthogonal. It takes a random tensor S in $(\mathbb{R}^p)^{\otimes(d-2)}$ and computes the eigendecomposition of $T(S,:,:)$. The vectors $\mathbf{u}_1,\dots,\mathbf{u}_r$ are eigenvectors of the matrix $T(S,:,:)$. As in Jennrich's algorithm, it can also decompose a symmetric tensor in $(\mathbb{R}^p)^{\otimes 4}$ with rank at most p due to the orthogonal requirement and it takes $O(p^4)$ operations where the most costly step is forming the matrice $T(S,:,:)$.

In comparison, HTD can decompose a symmetric $p \times p \times p \times p$ tensor of rank up to p^2. The algorithm has a computational complexity of $O(p^4 r)$ for decomposing a rank r tensor, primarily due to the eigendecomposition of the tensor flattening. HTD recovers the orthogonal symmetric decomposition when the tensor is orthogonally decomposable.

2. Detailed proof of Theorem 2.4

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

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

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

$$|\nu_i - \mu_i| \le (2|\nu_i|L + K)\epsilon^2 + \left(\frac{|\nu_i|}{\nu} 2^{\frac{5}{2}} + 1\right)\delta + o(\epsilon^2) + o(\delta)$$

$$\min\{\|\mathbf{b}_i - \mathbf{c}_i\|, \|\mathbf{b}_i + \mathbf{c}_i\|\} \le 2^{3/2} L\epsilon^2 + \frac{8\delta}{\nu} + o(\epsilon^2) + o(\delta).$$

where

$$K = \sqrt{8} \sum_{i=1}^{\ell} |\nu_i|(i-1), \quad L = 2^{3/2}\frac{K}{\nu} + 2\ell - 2, \quad \nu = \min_{i \neq j}\{|\nu_i - \nu_j|, |\nu_i|\}.$$

We prove Theorem 2.4 via the following lemma.

Lemma 2.1. *Let $\mathbf{b}_1, \dots, \mathbf{b}_\ell$ be vectors in $\mathbb{R}^p$ such that $|\langle \mathbf{b}_i, \mathbf{b}_j \rangle| \le \epsilon$ for all $i \neq j$. Let $\mathbf{B}_i$ be the vectorization of $\mathbf{b}_i^{\otimes 2}$. Define $M = \sum_{i=1}^{\ell} \nu_i \mathbf{B}_i^{\otimes 2}$. Then there exists a matrix M' with eigendecomposition $M' = \sum_{i=1}^{\ell} \nu_i \mathbf{B}_i'^{\otimes 2}$ such that for all $i \in [\ell]$,*

$$\|\mathbf{B}_i - \mathbf{B}_i'\| \le 2(\ell - 1)\epsilon^2 + O(\epsilon^4) \quad \text{and} \quad \|M - M'\|_F \le \sqrt{8} \sum_{i=1}^{\ell} |\nu_i|(i-1)\epsilon^2 + O(\epsilon^4).$$

Proof. We generate orthogonal vectors via Gram-Schmidt:

$$\mathbf{B}_j'' = \mathbf{B}_j - \sum_{i=1}^{j-1} \langle \mathbf{B}_i', \mathbf{B}_j \rangle \mathbf{B}_i', \quad \mathbf{B}_j' = \frac{\mathbf{B}_j''}{\|\mathbf{B}_j''\|}.$$

The vectors $\mathbf{B}_i$ satisfy $\|\mathbf{B}_i\| = 1$ for all i and $\langle \mathbf{B}_i, \mathbf{B}_j \rangle \le \epsilon^2$ for $i \neq j$. We will prove by induction on j that

$$|\langle \mathbf{B}_j', \mathbf{B}_k \rangle| \le \epsilon^2 + O(\epsilon^4) \quad \text{for all } k > j.$$

When $j = 1$, $\mathbf{B}_1' = \mathbf{B}_1$, so the result follows immediately. Assume the result is true for $j - 1$. Then,

$$\begin{aligned}
|\langle \mathbf{B}_j'', \mathbf{B}_k \rangle| &= |\langle \mathbf{B}_j, \mathbf{B}_k \rangle - \sum_{i=1}^{j-1} \langle \mathbf{B}_i', \mathbf{B}_j \rangle \langle \mathbf{B}_i', \mathbf{B}_k \rangle| \\
&\le |\langle \mathbf{B}_j, \mathbf{B}_k \rangle| + \sum_{i=1}^{j-1} |\langle \mathbf{B}_i', \mathbf{B}_j \rangle||\langle \mathbf{B}_i', \mathbf{B}_k \rangle| \\
&\le \epsilon^2 + (j-1)(\epsilon^2 + O(\epsilon^4))^2 \\
&= \epsilon^2 + O(\epsilon^4).
\end{aligned}$$

The inner product with $\mathbf{B}_j'$ is obtained from that with $\mathbf{B}_j''$ via

$$|\langle \mathbf{B}_j', \mathbf{B}_k \rangle| = \frac{|\langle \mathbf{B}_j'', \mathbf{B}_k \rangle|}{\|\mathbf{B}_j''\|},$$

so we obtain

$$|\langle \mathbf{B}_j', \mathbf{B}_k \rangle| \le \frac{\epsilon^2 + O(\epsilon^4)}{\|\mathbf{B}_j\| - \|\mathbf{B}_j - \mathbf{B}_j''\|} \le \frac{\epsilon^2 + O(\epsilon^4)}{1 - (j-1)\epsilon^2 + O(\epsilon^4)} = \epsilon^2 + O(\epsilon^4),$$

which proves the inductive step. By Gram-Schmidt and the triangle inequality, we have

$$\|\mathbf{B}_j'' - \mathbf{B}_j\| = \|\sum_{i=1}^{j-1} \langle \mathbf{B}_i', \mathbf{B}_j \rangle \mathbf{B}_i'\| \le \sum_{i=1}^{j-1} |\langle \mathbf{B}_i', \mathbf{B}_j \rangle| \le (j-1)\epsilon^2 + O(\epsilon^4) \le (\ell - 1)\epsilon^2 + O(\epsilon^4).$$

Thus, we can bound the distance between $\mathbf{B}_j'$ and $\mathbf{B}_j$ using the triangle inequality and $\mathbf{B}_j' = \frac{\mathbf{B}_j''}{\|\mathbf{B}_j''\|}$ by

$$\begin{aligned}
\|\mathbf{B}_j' - \mathbf{B}_j\| &\le \|\mathbf{B}_j' - \mathbf{B}_j''\| + \|\mathbf{B}_j'' - \mathbf{B}_j\| \\
&= \left|\frac{1 - \|\mathbf{B}_j''\|}{\|\mathbf{B}_j''\|}\right| + \|\mathbf{B}_j'' - \mathbf{B}_j\| \\
&\le \frac{\|\mathbf{B}_j - \mathbf{B}_j''\|}{1 - \|\mathbf{B}_j - \mathbf{B}_j''\|} + \|\mathbf{B}_j - \mathbf{B}_j''\| \\
&\le 2(j-1)\epsilon^2 + O(\epsilon^4).
\end{aligned}$$

Finally, we can bound the Frobenius norm of the difference between the matrices M and M' by

$$\begin{aligned}\|M - M'\|_F &= \|\sum_{i=1}^{\ell} \nu_i \mathbf{B}_i^{\otimes 2} - \sum_{i=1}^{\ell} \nu_i \mathbf{B}_i'^{\otimes 2}\|_F \\ &\leq \sqrt{2} \sum_{i=1}^{\ell} |\nu_i| \|\mathbf{B}_i - \mathbf{B}_i'\| \\ &\leq \sqrt{8} \sum_{i=1}^{\ell} |\nu_i|(i-1)\epsilon^2 + O(\epsilon^4). \end{aligned}$$

□

Proof of Theorem 2.4. Fix $M = \sum_{i=1}^{r} \nu_i \mathbf{B}_i^{\otimes 2}$ and $M' = \sum_{i=1}^{r} \nu_i \mathbf{B}_i'^{\otimes 2}$ as in Lemma 2.1. Fix $\hat{M} = \mathrm{Mat}(\hat{T})$ and let

$$\hat{M} = \sum_{i=1}^{r} \hat{\nu}_i \hat{\mathbf{B}}_i^{\otimes 2}$$

be its eigendecomposition. By the triangle inequality and Lemma 2.1, we have

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

where $K = \sqrt{8} \sum_{i=1}^{\ell} |\nu_i|(i-1)$. By Weyl's theorem,

$$|\nu_i - \hat{\nu}_i| \leq \|\hat{M} - M'\|_{\mathrm{op}} \leq \|\hat{M} - M'\|_F.$$

By the variant of the Davis-Kahan theorem in (5),

$$\|\hat{\mathbf{B}}_i - \mathbf{B}_i'\| \leq \frac{2^{\frac{3}{2}}}{\nu} \|\hat{M} - M'\|_F \quad \text{where} \quad \nu = \min_{j \neq i} \{|v_i|, |v_i - v_j|\}.$$

Thus, we can bound the distance between $\mathbf{B}_i$ and $\hat{\mathbf{B}}_i$, using the triangle inequality, by

$$\begin{aligned}\|\mathbf{B}_i - \hat{\mathbf{B}}_i\| &\leq \|\mathbf{B}_i - \mathbf{B}_i'\| + \|\mathbf{B}_i' - \hat{\mathbf{B}}_i\| \\ &\leq 2(\ell - 1)\epsilon^2 + \frac{2^{\frac{3}{2}}}{\nu}\delta + \frac{2^{\frac{3}{2}}K}{\nu}\epsilon^2 + O(\epsilon^4) \\ &= L\epsilon^2 + 2^{\frac{3}{2}}\frac{\delta}{\nu} + O(\epsilon^4),\end{aligned}$$

where $L = 2^{3/2}\frac{K}{\nu} + 2\ell - 2$.

The top eigenvector of $\mathrm{Mat}(\hat{\mathbf{B}}_i)$ is $\mathbf{c}_i$, and we suppose its eigenvalue is α. The top eigenpair of $\mathrm{Mat}(\mathbf{B}_i)$ is $(\mathbf{b}_i, 1)$. Therefore, again by the Davis-Kahan theorem, we have

$$\min\{\|\mathbf{b}_i - \mathbf{c}_i\|, \|\mathbf{b}_i + \mathbf{c}_i\|\} \leq 2^{\frac{3}{2}} \|\mathbf{B}_i - \hat{\mathbf{B}}_i\| \leq 2^{\frac{3}{2}} L\epsilon^2 + 8\frac{\delta}{\nu} + O(\epsilon^2).$$

By Weyl's theorem,

$$|\alpha - 1| \leq \|\mathbf{B}_i - \hat{\mathbf{B}}_i\|_{op} \leq \|\mathbf{B}_i - \hat{\mathbf{B}}_i\|_F \leq L\epsilon^2 + 2^{\frac{3}{2}}\frac{\delta}{\nu} + O(\epsilon^4).$$

The algorithm of HTD implies

$$\mu_i = \hat{\nu}_i \alpha^2.$$

Hence, we obtain, by the triangle inequality,

$$\begin{aligned}|\mu_i - \nu_i| &\leq |\mu_i - \hat{\nu}_i| + |\hat{\nu}_i - \nu_i| \\ &\leq |\hat{\nu}_i||1 - \alpha^2| + |\hat{\nu}_i - \nu_i| \\ &\leq (|\hat{\nu}_i - \nu_i| + |\nu_i|)|1 - \alpha|(2 + |1 - \alpha|) + |\hat{\nu}_i - \nu_i| \\ &\leq 2|1 - \alpha||\nu_i| + |\hat{\nu}_i - \nu_i| + o(\epsilon^2) + o(\delta) \\ &\leq 2|\nu_i|L\epsilon^2 + 2^{\frac{5}{2}}|\nu_i|\frac{\delta}{\nu} + \delta + K\epsilon^2 + o(\epsilon^2) + o(\delta).\end{aligned}$$

□

3. Detailed proof of Theorem 3.7 and 3.8

3.1. Proof of Theorem 3.7. 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.

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

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

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

Proof. Let $\mathbf{a}_i'$ be the estimate of $\mathbf{a}_i$ obtained via Step 1 of Algorithm 2, and μ_i be the estimate of λ_i' via Step 2 of Algorithm 2. We can bound the difference between the true tensor T and the recovered tensor $\hat{T}$ as

$$\begin{aligned}
&\|\hat{T} - T\|_F \\
=&\|\hat{\kappa}_4(\mathbf{x}) - \sum_{i=1}^{r} \mu_i \mathbf{a}_i'^{\otimes 4} - \kappa_4(\mathbf{x}) + \sum_{i=1}^{r} \lambda_i' \mathbf{a}_i^{\otimes 4}\|_F \\
\leq&\|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \|\sum_{i=1}^{r} \mu_i (\mathbf{a}_i^{\otimes 4} - \mathbf{a}_i'^{\otimes 4})\|_F + \|\sum_{i=1}^{r} (\lambda_i' - \mu_i) \mathbf{a}_i^{\otimes 4}\|_F \\
\leq&\|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \sum_{i=1}^{r} |\mu_i| \|\mathbf{a}_i^{\otimes 4} - \mathbf{a}_i'^{\otimes 4}\| + \sum_{i=1}^{r} |\lambda_i' - \mu_i| \\
\leq&\|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \sum_{i=1}^{r} 2|\mu_i| \|\mathbf{a}_i - \mathbf{a}_i'\| + \sum_{i=1}^{r} |\lambda_i' - \mu_i|,
\end{aligned}$$

where the first two inequalities follow from the triangle inequality and the last inequality follows from

$$\begin{aligned}
&\|\mathbf{a}_i^{\otimes 4} - \mathbf{a}_i'^{\otimes 4}\|^2 = 2 - 2\langle \mathbf{a}_i, \mathbf{a}_i' \rangle^4 \\
=&2 - 2\left(1 - \frac{1}{2}\|\mathbf{a}_i - \mathbf{a}_i'\|^2\right)^4 \\
\leq&2 - 2 + 4\|\mathbf{a}_i - \mathbf{a}_i'\|^2 \quad \text{(using } (1-x)^4 \geq 1 - 4x \text{ for small } x) \\
=&4\|\mathbf{a}_i - \mathbf{a}_i'\|^2.
\end{aligned}$$

By (6, Lemma S.32), we have $\sigma_r(M_y) \geq \lambda \sigma_r(G_2)$, where $G_2 \in \mathbb{R}^{r \times r}$ is the matrix with (i,j) entry $\langle \mathbf{a}_i, \mathbf{a}_j \rangle^2$. By the proof of (6, Lemma 6), we have $\sigma_r(G_2) \geq 1 - \rho_2$ where $\rho_s = \sup_{\|x\|=1} \sum_{i=1}^{r} |\langle x, \mathbf{a}_i \rangle|^s - 1$ for $s > 0$ and $\rho_s \leq (r-1)\rho^{\lfloor s/2 \rfloor}$. Thus, we can lower bound $\sigma_r(M_y)$ by

$$\sigma_r(M_y) \geq \lambda - \lambda(r-1)\rho = \lambda' = \lambda + O(\rho).$$

Let $\tau = \frac{1}{6} - 4\rho_2 - 6\rho_4 = \frac{1}{6} + O(\rho)$. By (6, Theorem 7), if $\Delta_A < \frac{2\tau}{2+4\tau+12}$, we can bound the distance between the true component $\mathbf{a}_i$ and learned component $\mathbf{a}_i'$ by

$$\|\mathbf{a}_i - \mathbf{a}_i'\| \leq \sqrt{\frac{\Delta_A}{2}}.$$

The condition is satisfied when $\frac{\Delta_M}{\lambda - \Delta_M + O(\rho)} \leq \frac{1}{44} + O(\rho)$. This explains our second assumption $\Delta_M \leq \frac{\lambda}{45} + O(\rho)$.

By (6, Lemma S.31), the distance between the numbers $\frac{1}{\lambda_i'}$ and $\frac{1}{\mu_i}$ is bounded from above by

$$\begin{aligned}
|\frac{1}{\lambda_i'} - \frac{1}{\mu_i}| &\leq \frac{\sqrt{8}}{\sigma_r(M_y)} \|\mathbf{a}_i - \mathbf{a}_i'\| + \Delta_A (\frac{2}{\sigma_r(M_y)} + \frac{1}{\sigma_r(M_y) - \Delta_M}) \\
&\leq \frac{1}{\sigma_r(M_y)} (2\sqrt{\Delta_A} + 3\Delta_A) \\
&\leq \frac{1}{\lambda'} (2\sqrt{\Delta_A} + 3\Delta_A).
\end{aligned}$$

This implies that

$$|\lambda_i' - \mu_i| \le |\lambda_i'\mu_i|\frac{1}{\lambda'}(2\sqrt{\Delta_A} + 3\Delta_A)$$
$$\le (|\lambda_i'^2| + |\lambda_i'||\lambda_i' - \mu_i|)\frac{1}{\lambda'}(2\sqrt{\Delta_A} + 3\Delta_A).$$

Rearranging, we obtain

$$|\lambda_i' - \mu_i|(1 - |\lambda_i'|\frac{1}{\lambda'}(2\sqrt{\Delta_A} + 3\Delta_A)) \le |\lambda_i'^2|\frac{1}{\lambda'}(2\sqrt{\Delta_A} + 3\Delta_A).$$

To obtain an upper bound on $|\lambda_i' - \mu_i|$ from the above inequality, we need $|\lambda_i'|\frac{1}{\lambda'}(2\sqrt{\Delta_A} + 3\Delta_A) < 1$, which is our third assumption in the statement. Thus, the distance between the true coefficient λ_i' of the rank one component $\mathbf{a}_i^{\otimes 2}$ and the learned coefficient μ_i is bounded by

$$|\lambda_i' - \mu_i| \le |\lambda_i'^2|\frac{1}{\lambda'}(2\sqrt{\Delta_A} + 3\Delta_A)(1 + |\lambda_i'|\frac{1}{\lambda'}(2\sqrt{\Delta_A} + 3\Delta_A) + O(\Delta_A))$$
$$= |\lambda_i'^2|\frac{2}{\lambda'}\sqrt{\Delta_A} + O(\Delta_A).$$

Plugging the bounds on $\|\mathbf{a}_i' - \mathbf{a}_i\|$ and $|\lambda_i' - \mu_i|$ into the bound on $\|\hat{T} - T\|_F$, we obtain

$$\|\hat{T} - T\|_F \le \|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \sum_{i=1}^r 2|\mu_i|\|\mathbf{a}_i - \mathbf{a}_i'\| + \sum_{i=1}^r |\lambda_i' - \mu_i|$$
$$\le \|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \sum_{i=1}^r 2(|\lambda_i' - \mu_i| + |\lambda_i'|)\sqrt{\frac{\Delta_A}{2}} + \sum_{i=1}^r |\lambda_i'^2|\frac{2}{\lambda'}\sqrt{\Delta_A} + O(\Delta_A)$$
$$\le \|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \sum_{i=1}^r |\lambda_i'|\sqrt{2\Delta_A} + \sum_{i=1}^r |\lambda_i'^2|\frac{2}{\lambda'}\sqrt{\Delta_A} + O(\Delta_A).$$

Note that $\sigma_r(M_y) \le \lambda'$ so $\Delta_A = \frac{\Delta_M}{\sigma_r(M_y) - \Delta_M} \le \frac{\Delta_M}{\lambda'} + O(\Delta_M^2)$. Hence, replacing Δ_A by Δ_M, we obtain

$$\|\hat{T} - T\|_F \le \|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + (\sum_{i=1}^r |\lambda_i'|\sqrt{\frac{2}{\lambda'}} + |\lambda_i'^2|\frac{2}{\lambda'^{\frac{3}{2}}})\sqrt{\Delta_M} + O(\Delta_M).$$ □

3.2. Detailed proof of Theorem 3.8.

We restate the theorem for convenience.

Theorem. 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 $\mathbf{c}_i$ are the output patterns of the cICA algorithm, with corresponding recovered scalars μ_i, obtained from the tensor of foreground patterns $T = \sum_{i=1}^\ell \nu_i \mathbf{b}_i^{\otimes 4}$. Under the assumptions of Theorem 2.4 and Theorem 3.7, we have

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

where

$$|\langle \mathbf{b}_i, \mathbf{b}_j\rangle| \le \epsilon, \quad \text{for all } i \ne j$$
$$\delta = \widetilde{O}\left(\frac{p^{\frac{3}{2}}\ell'^2}{N_2} + \sqrt{\frac{\ell'^4}{N_2}} + \sqrt{\frac{pr'^2}{N_1} + \sqrt{\frac{r'^4}{pN_1}}}\right),$$

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

We prove the theorem via the following lemmas.

Lemma 3.1. *Let $A \in \mathbb{R}^{p\times r}$ be a matrix with columns $\mathbf{a}_1, \dots, \mathbf{a}_r$, where $\|\mathbf{a}_i\| = 1$ for all i, and $\max_{i\ne j} |\langle \mathbf{a}_i, \mathbf{a}_j\rangle| \le \rho$. Then*

$$\|A\|_2 = 1 + O(\rho).$$

Proof. Let $C = A^\mathsf{T} A$. For any $v \in \mathbb{R}^r$, we have

$$\|(C - I_r)v\| \le \rho \|v\|_1 \le \sqrt{r}\rho \|v\|,$$

thus

$$\|C - I_r\|_2 \le \sqrt{r}\rho.$$

Let σ be the top eigenvalue of C. Then $\sigma = \|A\|_2^2$. By Weyl's theorem, we have

$$|\sigma - 1| \le \|C - I_r\|_2 \le \sqrt{r}\rho,$$

so $\sigma = 1 + O(\rho)$, and hence

$$\|A\|_2 = \sqrt{1 + O(\rho)} = 1 + O(\rho). \qquad \square$$

Suppose T is a symmetric tensor in $(\mathbb{R}^p)^{\otimes 4}$. Its operator norm is

$$\|T\| = \sup_{\|v_1\|=\|v_2\|=\|v_3\|=\|v_4\|=1} |T(v_1, v_2, v_3, v_4)|$$

where

$$T(v_1, v_2, v_3, v_4) = \sum_{i=1}^p \sum_{j=1}^p \sum_{k=1}^p \sum_{\ell=1}^p T_{ijkl} (v_1)_i (v_2)_j (v_3)_k (v_4)_\ell.$$

Lemma 3.2. *Suppose T is a symmetric tensor in $(\mathbb{R}^p)^{\otimes 4}$. Then, we have*

$$\|\operatorname{Mat}(T)\|_2 \le p\|T\| \quad \textit{and} \quad \|T\|_F \le p^{\frac{3}{2}} \|T\|.$$

Proof. Let $B \in \mathbb{R}^{p^2}$ such that $\|B\| = 1$ and

$$B^\mathsf{T} \operatorname{Mat}(T) B = \|\operatorname{Mat}(T)\|_2.$$

The matrix $\operatorname{Mat}(B)$ is symmetric since it lies in the column span of $\operatorname{Mat}(T)$. Let $\operatorname{Mat}(B) = \sum_{i=1}^p \lambda_i \mathbf{b}_i^{\otimes 2}$ be its eigendecomposition. Then, we have

$$B^\mathsf{T} \operatorname{Mat}(T) B = \sum_{i=1}^p \sum_{j=1}^p \lambda_i \lambda_j T(\mathbf{b}_i, \mathbf{b}_i, \mathbf{b}_j, \mathbf{b}_j) \le \left(\sum_{i=1}^p \lambda_i \right)^2 \|T\|.$$

Note that $\|B\| = 1$, so $\sum_{i=1}^p \lambda_i^2 = 1$. By the AM–GM inequality, $|\sum_{i=1}^p \lambda_i| \le \sqrt{p}$. Thus

$$\|\operatorname{Mat}(T)\|_2 = B^\mathsf{T} \operatorname{Mat}(T) B \le p\|T\|.$$

The quantity $\min_{T \neq 0} \frac{\|T\|}{\|T\|_F}$ is the best rank-one approximation ratio, see (7, 8). For fourth-order tensors of size p, we have $\|T\|_F \le p^{3/2}\|T\|$ since T can be written as a sum of at most p^3 tensors whose vectorizations are orthogonal, see (7, Theorem 3.5) or (8, Theorem 1.1). $\square$

We will use the following sample complexity result of ICA from (9, Theorem 2).

Theorem 3.3. *Consider N samples $x^i = Ah^i$, $i \in [N]$, from the ICA model with mixing matrix $A \in \mathbb{R}^{d \times k}$. Suppose $\|A\| \le O(1 + \sqrt{k/d})$ and the entries of $h \in \mathbb{R}^k$ are independent subgaussian variables with $\mathbb{E}[h_j^2] = 1$ and constant nonzero 4th order cumulant. Define $m = \max(d, k)$. For the 4th order cumulant κ_4 in (8) and its empirical estimate $\hat{\kappa}_4$, if $n \ge d$, we have with high probability*

$$\|\hat{\kappa}_4 - \kappa_4\| \le \widetilde{O}\left(\frac{m^2}{N} + \sqrt{\frac{m^4}{d^3 N}} \right).$$

Proof of Theorem 3.8. We have $\|A\| = 1 + O(\rho)$ and $\|B\| = 1 + O(\epsilon^2)$ by Lemma 3.1. Using the triangle inequality, we obtain

$$\|(A, B)\| \le \|A\| + \|B\| \le 2 + O(\epsilon^2) + O(\rho).$$

Thus, we have $\|A\| = O(1)$ and $\|(A, B)\| = O(1)$.

We obtain that the following bounds on the operator norm of the difference between the sample cumulants and true cumulants hold with high probability:

$$\|\kappa_4(\mathbf{y}) - \hat{\kappa}_4(\mathbf{y})\| = \widetilde{O}\left(\frac{r'^2}{N_1} + \sqrt{\frac{r'^4}{p^3 N_1}} \right),$$

$$\|\kappa_4(\mathbf{x}) - \hat{\kappa}_4(\mathbf{x})\| = \widetilde{O}\left(\frac{\ell'^2}{N_2} + \sqrt{\frac{\ell'^4}{p^3 N_2}} \right),$$

by Theorem 3.3, under the assumptions on z_i, z_i', s_j in the statement, and using $\|A\| = O(1)$ and $\|(A,B)\| = O(1)$. Let $T = \sum_{i=1}^{\ell} \nu_i \mathbf{b}_i^{\otimes 4}$, and let $\hat{T}$ be the tensor obtained after Steps 1 and 2 of Algorithm 2. We can bound the distance between the true T and the recovered $\hat{T}$ by

$$\begin{aligned}\|\hat{T} - T\|_F &\le \|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\|_F + \beta\sqrt{\Delta_M} + O(\Delta_M)\\ &\le p^{\frac{3}{2}}\|\hat{\kappa}_4(\mathbf{x}) - \kappa_4(\mathbf{x})\| + \beta\sqrt{p\|\hat{\kappa}_4(\mathbf{y}) - \kappa_4(\mathbf{y})\|} + O(\Delta_M)\\ &= \widetilde{O}\left(\frac{p^{\frac{3}{2}}\ell'^2}{N_2} + \sqrt{\frac{\ell'^4}{N_2}} + \sqrt{p\left(\frac{r'^2}{N_1} + \sqrt{\frac{r'^4}{p^3 N_1}}\right)}\right)\\ &= \widetilde{O}\left(\frac{p^{\frac{3}{2}}\ell'^2}{N_2} + \sqrt{\frac{\ell'^4}{N_2}} + \sqrt{\frac{pr'^2}{N_1} + \sqrt{\frac{r'^4}{pN_1}}}\right),\end{aligned}$$

using Theorem 3.7. Hence, we obtain the final bounds via Theorem 2.4 that

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

and

$$\min\{\|\mathbf{b}_i - \mathbf{c}_i\|, \|\mathbf{b}_i + \mathbf{c}_i\|\} \le 2^{3/2}L\epsilon^2 + \widetilde{O}(\delta) = O(\epsilon^2) + \widetilde{O}(\delta),$$

where

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

4. Proportional cICA

In this section, we present a variant of cICA called proportional cICA.

Recall that the cICA model expresses the background $\mathbf{y}$ and foreground $\mathbf{x}$ as

$$\mathbf{y} = A\mathbf{z} \qquad \text{and} \qquad \mathbf{x} = A\mathbf{z}' + B\mathbf{s}. \tag{2}$$

Proportional cICA assumes assumes $\mathbf{z}' = \gamma\mathbf{z}$ for some scalar $\gamma > 0$. This assumption also appears in cPCA (10). There, the choice of the hyperparameter γ is not unique. However, in our setting—which involves the fourth-order cumulants $\kappa_4(\mathbf{y})$ and $\kappa_4(\mathbf{x})$, under the assumption that $r + \ell \le \binom{p+1}{2}$—the value of γ is uniquely determined, with a closed-form expression, see Theorem 4.1. The details of the ensuing algorithm for computing matrix B are as follows.

Algorithm 1 Recover B from the background and foreground cumulants when $\mathbf{z}' = \gamma\mathbf{z}$

Input: $\kappa_4(\mathbf{x}), \kappa_4(\mathbf{y})$ and ℓ as in [5].

1: **Compute** γ using the following theorem.

2: **Recover** B**:** Compute rank ℓ symmetric decomposition of $\kappa_4(\mathbf{x}) - \gamma^4\kappa_4(\mathbf{y})$, using Algorithm 1.

Output: Mixing matrix B.

Theorem 4.1. *Consider proportional cICA with $\mathbf{z}' = \gamma\mathbf{z}$, for $\gamma > 0$. For generic $\mathbf{a}_1, \dots, \mathbf{a}_r$ and $\mathbf{b}_1, \dots, \mathbf{b}_\ell$ with $r + \ell \le \binom{p+1}{2}$ and $r \ne 8$, the hyperparameter γ is the unique value $(\frac{1}{\lambda_i}(\mathbf{a}_i^\mathsf{T} V D^{-1} V^\mathsf{T} \mathbf{a}_i)^{-1})^{\frac{1}{4}}$, where i is any index between 1 and r, λ_i is the coefficient of $\mathbf{a}_i^{\otimes 4}$ in $\kappa_4(\mathbf{x})$ and VDV^T is the thin eigendecomposition of* $\mathrm{Mat}(\kappa_4(\mathbf{x}))$.

Proof. The flattenings of the cumulants $\kappa_4(\mathbf{y})$ and $\kappa_4(\mathbf{x})$ are, respectively,

$$M_{\mathbf{y}} := \sum_{i=1}^{r} \lambda_i \mathbf{A}_i^{\otimes 2}, \qquad M_{\mathbf{x}} := \gamma^4\left(\sum_{i=1}^{r} \lambda_i \mathbf{A}_i^{\otimes 2}\right) + \sum_{j=1}^{\ell} \nu_j \mathbf{B}_j^{\otimes 2},$$

where $\mathbf{A}_i, \mathbf{B}_j \in \mathbb{R}^{p^2}$ vectorize the matrices $\mathbf{a}_i^{\otimes 2}$ and $\mathbf{b}_i^{\otimes 2}$, respectively and we use that $\lambda_i' = \gamma^4\lambda_i$. We have $\operatorname{rank} M_{\mathbf{y}} = r$ and $\operatorname{rank} M_{\mathbf{x}} = r + \ell$, by the assumptions in the statement.

Let $A \in \mathbb{R}^{p^2 \times r}$ be the matrix with columns $\mathbf{A}_1, \dots, \mathbf{A}_r$ and define $D' = \gamma^4\mathrm{Diag}(\lambda_1, \dots, \lambda_r)$. We have $\operatorname{rank}(M_{\mathbf{x}} - AD'A^\mathsf{T}) = \operatorname{rank}(\sum_{j=1}^{\ell} \nu_j \mathbf{B}_j^{\otimes 2}) = \ell$. Suppose that VDV^T is the thin eigendecomposition of $M_{\mathbf{x}}$. We have

$$V^\mathsf{T}(M_{\mathbf{x}} - AD'A^\mathsf{T})V = D - (V^\mathsf{T}A)D'(V^\mathsf{T}A)^\mathsf{T}.$$

We have that $\operatorname{rank} D = r + \ell$, the upper bound $\operatorname{rank}(V^\mathsf{T}A)D'(V^\mathsf{T}A)^\mathsf{T} = \operatorname{rank} V^\mathsf{T}M_{\mathbf{y}}V \le r$, and finally that $\operatorname{rank}(D - (V^\mathsf{T}A)D'(V^\mathsf{T}A)^\mathsf{T}) = \operatorname{rank}(V^\mathsf{T}(M_{\mathbf{x}} - AD'A^\mathsf{T})V) \le \ell$. Hence

$$D' = (A^\mathsf{T}VD^{-1}V^\mathsf{T}A)^{-1},$$

by Lemma 3.6. Matrices $A, \mathrm{Diag}(\lambda_1, \ldots, \lambda_r), V, D$ can be recovered uniquely from tensor decomposition of $\kappa_4(\mathbf{y})$ and the eigendecomposition of $M_{\mathbf{x}}$. So D' can also be recovered uniquely, and hence γ is unique: it is $\gamma^4\lambda_i = (\mathbf{a}_i^\mathsf{T} V D^{-1} V^\mathsf{T} \mathbf{a}_i)^{-1}$ for any $i \in [r]$. □

One can test the proportionality assumption by seeing whether the values $\left(\frac{1}{\lambda_i}(\mathbf{a}_i^\mathsf{T} V D^{-1} V^\mathsf{T} \mathbf{a}_i)^{-1}\right)^{\frac{1}{4}}$ from the above Theorem are approximately equal as i varies. In practice, exact proportionality may not hold, and learning γ via the above Theorem could be challenging. An alternative is to use a sweep of γ values and choose γ according to visualization plots, a similar method to that used in cPCA (10).

We also implement the proportional cICA algorithm and report its performance on various datasets in Section 6 of the Appendix along with the numerical experiments details.

5. Practicalities and interpretation of cICA

In this section, we discuss the practicalities of cICA: preprocessing the input to speed up the algorithm and how to choose the ranks r and ℓ. We also discuss how to interpret coordinates when viewing cICA as a dimensionality reduction method.

5.1. Choosing the ranks. When computing the tensor decompositions in cICA, a key step is to determine the ranks r and ℓ. To choose the ranks, we can use the flattenings of the cumulants, the matrices $\mathrm{Mat}(\kappa_4(\mathbf{x})), \mathrm{Mat}(\kappa_4(\mathbf{y})) \in \mathbb{R}^{p^2 \times p^2}$. If the expressions for the cumulant tensors $\kappa_4(\mathbf{x})$ and $\kappa_4(\mathbf{y})$ in [5] hold exactly, and if $r + \ell \leq \binom{p+1}{2}$ and the vectors $\mathbf{a}_i, \mathbf{b}_j$ are generic, then

$$r = \mathrm{rank}(\mathrm{Mat}(\kappa_4(\mathbf{y}))) \quad \text{and} \quad r + \ell = \mathrm{rank}(\mathrm{Mat}(\kappa_4(\mathbf{x}))).$$

For non-exact cumulants, such as sample cumulants, we do not work with the exact ranks of the flattening matrices, but instead examine plots of the eigenvalues in descending magnitude (see Appendix) to choose an appropriate cut-off. We choose r such that the decrease of the eigenvalue plot of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ slows down, choose q such that the decrease of the eigenvalue plot of $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ slows down, and calculate $\ell = q - r$. The algorithm cICA has hyperparameters r and ℓ; proportional cICA has one hyperparameter ℓ.

We discuss how the results may be affected by an incorrect choice of r and ℓ and justify our proposed way to order the foreground patterns $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ by importance in [10].

Let the true ranks be r and ℓ and assume that we have used r' and ℓ' in the input to Algorithm 2.

- If $\ell' > \ell$, then $\ell' - \ell$ foreground patterns are noise.
- If $\ell' < \ell$, then $\ell - \ell'$ foreground patterns are not recovered.
- If $r' < r$, then background patterns are mixed with foreground patterns, as follows. Assuming without loss of generality that we have recovered $\mathbf{a}_1, \ldots, \mathbf{a}_{r'}$, the third step of Algorithm 2 decomposes the tensor $\sum_{i=r'+1}^{r} \lambda_i' \mathbf{a}_i^{\otimes 4} + \sum_{j=1}^{\ell} \nu_j \mathbf{b}_j^{\otimes 4}$ via HTD, as in Algorithm 1. If the orthogonality hypotheses of Proposition 2.3 hold, then the recovered foreground patterns are recovered together with some background patterns that are incorrectly interpreted as foreground patterns. If the approximate orthogonality hypotheses of Theorem 2.4 hold, then the foreground patterns are recovered approximately, together with background patterns that are classed as foreground patterns. Without an orthogonality condition, the recovered foreground patterns $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ will be polluted but still roughly collinear to the true foreground patterns for small $r - r'$ or when the dimension of the dataset is large, resulting in almost orthogonality between random vectors.
- If $r' > r$, then foreground patterns are mixed with background noise, as follows. Some background patterns from Algorithm 2 will be noise, say $\mathbf{a}_{r+1}', \ldots, \mathbf{a}_{r'}'$. Step 2 of Algorithm 2 computes the coefficients of the tensors $(\mathbf{a}_{r+1}')^{\otimes 4}, \ldots, (\mathbf{a}_{r'}')^{\otimes 4}$ in $\kappa_4(\mathbf{x})$, though they are not true rank one components of $\kappa_4(\mathbf{x})$. In Step 3, the tensor to be decomposed has the form $\sum_{i=1}^{r'-r} \mu_i (\mathbf{a}_{r+i}')^{\otimes 4} + \sum_{i=1}^{\ell} \nu_i \mathbf{b}_i^{\otimes 4}$ for some $\mu_1, \ldots, \mu_{r'-r} \in \mathbb{R}$. As in the case $r' < r$, the foreground patterns can still be exactly or approximately recovered, under the hypotheses of Proposition 2.3 and Theorem 2.4 respectively, albeit with some background noise recovered as foreground patterns.

The above discussion shows that when $r' \neq r$, the vectors $\mathbf{b}_1, \ldots, \mathbf{b}_\ell$ obtained from Algorithm 2 could represent foreground patterns, background patterns, or noise. We order the vectors according to [10]. The denominator of [10] is the variance of the linearly transformed background dataset $Y\mathbf{b}$. The numerator is that of the transformed dataset $X\mathbf{b}$. Their ratio enables us to select the most relevant foreground patterns, as follows.

- If $\mathbf{b}$ is a foreground pattern, we expect $\mathbf{b}^\mathsf{T}\kappa_2(\mathbf{y})\mathbf{b}$ to be small relative to $\mathbf{b}^\mathsf{T}\kappa_2(\mathbf{x})\mathbf{b}$, hence a large $k(\mathbf{b})$.
- If $\mathbf{b}$ is a background pattern, we expect $\mathbf{b}^\mathsf{T}\kappa_2(\mathbf{y})\mathbf{b} \approx \alpha \mathbf{b}^\mathsf{T}\kappa_2(\mathbf{x})\mathbf{b}$ for some constant α and hence $k(\mathbf{b}) \approx \alpha$.
- If $\mathbf{b}$ is foreground noise, we expect a small $\mathbf{b}^\mathsf{T}\kappa_2(\mathbf{x})\mathbf{b}$, hence small $k(\mathbf{b})$.
- If $\mathbf{b}$ is background noise, we expect a small $\mathbf{b}^\mathsf{T}\kappa_2(\mathbf{y})\mathbf{b}$, hence a large $k(\mathbf{b})$. To prevent the background noise from showing up in the recovered foreground pattern, we require $r' \leq r$.

In practice, we consider those patterns for which $k(\mathbf{b})$ exceeds a certain threshold or take the patterns with the two highest values of $k(\mathbf{b})$.

5.2. Visualization. We discuss how to interpret coordinates when using cICA for dimensionality reduction. The following proposition relates the projections $\mathbf{b}_i^T\mathbf{x}$ for $i \in [\ell]$ to the latent variables s_i.

Proposition 5.1. *Consider the cICA model in [2]. Suppose $\|\mathbf{b}_i\| = 1$ for $i \in [\ell]$. Assume that for some small $\epsilon > 0$ that $|\langle \mathbf{b}_i, \mathbf{b}_j\rangle| < \epsilon$ and $|\langle \mathbf{b}_i, \mathbf{a}_k\rangle| < \epsilon$ for $i \neq j \in [\ell]$, $k \in [r]$. Then, for each $i \in [\ell]$,*

$$|s_i - \mathbf{b}_i^T\mathbf{x}| = (rC_{\mathbf{z}'} + (\ell-1)C_{\mathbf{s}})O(\epsilon),$$

where $C_{\mathbf{z}'}$ and $C_{\mathbf{s}}$ are upper bounds on the magnitudes of random variables in $\mathbf{z}'$ and $\mathbf{s}$. In particular, $\mathbf{b}_i^T\mathbf{x}$ approximates the component s_i with an error linear in ϵ.

Proof. Recall from [2] that $\mathbf{x} = A\mathbf{z}' + B\mathbf{s}$. Hence

$$\begin{aligned}\mathbf{b}_i^T\mathbf{x} &= (\mathbf{b}_i^T A)\mathbf{z}' + (\mathbf{b}_i^T B)\mathbf{s} \\ &= \sum_{k=1}^{r}\langle \mathbf{b}_i, \mathbf{a}_k\rangle z_k' + \sum_{j=1, j\neq i}^{\ell}\langle \mathbf{b}_i, \mathbf{b}_j\rangle s_j + s_i.\end{aligned}$$

The almost orthogonality conditions of the proposition then imply that

$$\begin{aligned}|s_i - \mathbf{b}_i^T\mathbf{x}| &\leq \sum_{k=1}^{r}|\langle \mathbf{b}_i, \mathbf{a}_k\rangle||z_k'| + \sum_{j=1}^{\ell}|\langle \mathbf{b}_i, \mathbf{b}_j\rangle||s_j| \\ &\leq (rC_{\mathbf{z}'} + (\ell-1)C_{\mathbf{s}})\epsilon.\end{aligned}$$

□

The almost orthogonality conditions in Proposition 5.1 are strong requirements. However, they can be relaxed – if $|\langle \mathbf{b}_i, \mathbf{b}_j\rangle| < \epsilon$ for chosen $i, j \in [\ell]$ and sources s_i and s_j have wider variance than $(\mathbf{b}_i^\mathsf{T} A)\mathbf{z}'$ and $(\mathbf{b}_j^\mathsf{T} A)\mathbf{z}'$, then plotting $\mathbf{b}_i^\mathsf{T} X$ against $\mathbf{b}_j^\mathsf{T} X$ still approximates the plot of s_i against s_j.

If $(\mathbf{b}_i^\mathsf{T} A)\mathbf{z}'$ and $(\mathbf{b}_j^\mathsf{T} A)\mathbf{z}'$ are uncorrelated, we expect the plot of $X\mathbf{b}_i$ against $X\mathbf{b}_j$ to show axis-aligned clusters; otherwise, clusters may not be axis-aligned. We specify the condition for $(\mathbf{b}_i^\mathsf{T} A)\mathbf{z}'$ and $(\mathbf{b}_j^\mathsf{T} A)\mathbf{z}'$ to be uncorrelated, assuming that all variables in the tuple $\mathbf{z}'$ have the same variance.

Proposition 5.2. *Consider the cICA model in [2]. Suppose that the independent variables $\mathbf{z}'$ is a tuple of independent random variables with the same variance. Then $(\mathbf{b}_i^\mathsf{T} A)\mathbf{z}'$ and $(\mathbf{b}_j^\mathsf{T} A)\mathbf{z}'$ are uncorrelated if and only if $\langle \mathbf{b}_i^\mathsf{T} A, \mathbf{b}_j^\mathsf{T} A\rangle = 0$.*

Proof. Write $\mathbf{u} = \mathbf{b}_i^\mathsf{T} A$ and $\mathbf{v} = \mathbf{b}_j^\mathsf{T} A$. By the bilinearity of the covariance

$$\begin{aligned}\mathrm{Cov}(\mathbf{u}\mathbf{z}', \mathbf{v}\mathbf{z}') &= \sum_{1\leq i,j\leq r} u_i v_j \mathrm{Cov}(z_i', z_j') \\ &= \sum_{1\leq i\leq r} u_i v_i \mathrm{Var}(z_i') \\ &= \mathrm{Var}(z_1')\sum_{1\leq i\leq r} u_i v_i.\end{aligned}$$

The last expression is zero if and only if $\langle \mathbf{u}, \mathbf{v}\rangle = 0$. □

6. Details of numerical experiments

All experiments are run on an Apple M2 Pro with 16 GB memory. Each run of each algorithm takes at most 1 minute.

6.1. Choices of Methods in Algorithm 2. We describe the details of the synthetic data setup in Section 4.1. Our setup involves a background dataset of three independent uniform random variables and a foreground dataset with five sources: three uniform random variables and 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

$$\begin{pmatrix} 0.74280923 & 0.91366784 & 0.52707773 \\ -0.61857537 & 0.32868577 & 0.83815881 \\ 0.23109269 & -0.2120887 & -0.08650875 \\ -0.0153426 & 0.07115626 & -0.07315634 \\ 0.10936053 & 0.08445063 & 0.08272407 \end{pmatrix}.$$

We show in Figure 3 of the main text that projecting the foreground dataset using the matrix B reveals four distinct clusters and we illustrate the performance of our algorithm SPM-HTD and the variants SPM-SPM, HTD-HTD. Here, we report the performance of other combinations of tensor decompositions methods, ICA methods and HTD in Figure S2. Only the two methods JADE-HTD and FastICA-HTD find the four clusters in the foreground dataset.

To demonstrate the necessity of our proposed three-step decomposition (Algorithm 2) instead of separately decomposing the foreground and background tensors, we introduce an additional comparison method called SPM-SPM-Separate. Here, SPM

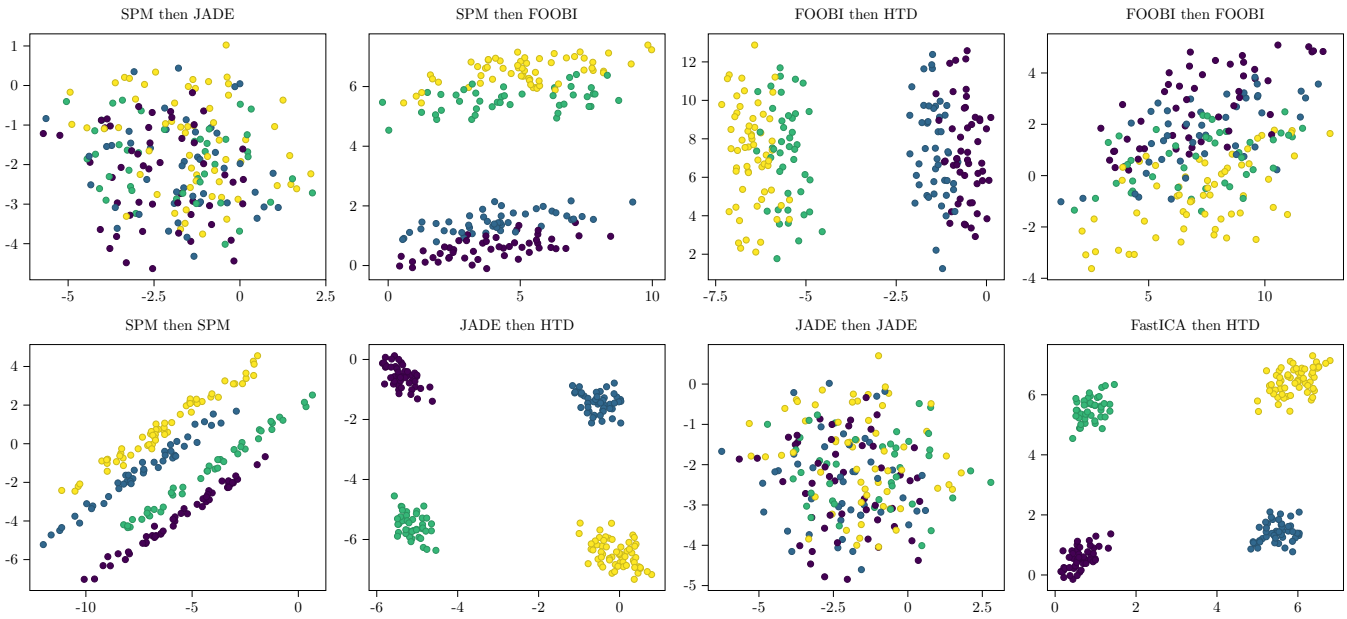

Fig. S2. The performance of SPM-JADE, SPM-FOOBI, FOOBI-HTD, FOOBI-FOOBIM, SPM-SPM, JADE-HTD, JADE-JADE and FastICA-HTD on synthetic data. Only JADE-HTD and FastICA-HTD find the four clusters in the foreground dataset.

is applied separately to the foreground and background cumulant tensors. The resulting patterns are matched using cosine similarity to identify the foreground patterns.

We vary the sample size of both datasets from 100 to 1000. For each sample size, we repeat the experiment 20 times by randomly drawing datasets, applying all eleven 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 that the estimated matrix B accurately recovers the four clusters. To mitigate randomness, we record the best silhouette score from 20 independent runs for each method and then average these best scores across experiments. Apart from the methods in Figure 3, we also report the performance of the method in Figure S3. The method, SPM-SPM-Separate yields the lowest scores. This confirms the need to use the three-step decomposition procedure described in Algorithm 2 over separate foreground and background tensor decompositions.

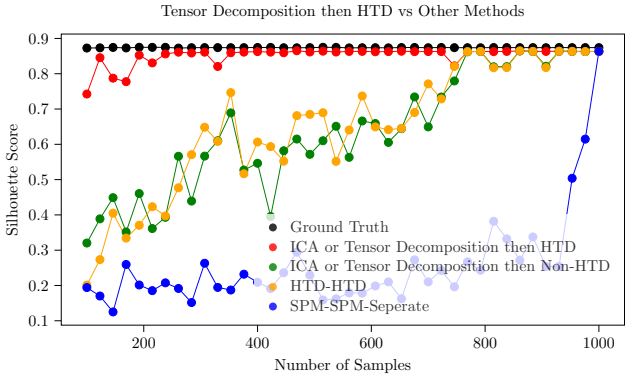

Fig. S3. We study the accuracy of different approaches to cICA as the number of samples varies. We compare methods using ICA or tensor decomposition followed by HTD against HTD-HTD, methods using ICA or tensor decomposition methods followed by non-HTD alternatives, and SPM-SPM-Separate, in which SPM is applied separately to the foreground and background datasets. Performance is evaluated using the silhouette score, which measures how effectively the estimated matrix B recovers the four clusters shown in the top-right plot of Figure 3. The SPM-SPM-Separate method performs worst among all methods, emphasizing the importance of employing the three-step decomposition procedure in Algorithm 2. Methods using ICA or tensor decomposition followed by HTD consistently outperform both ICA or tensor decomposition methods followed by non-HTD approaches, and the HTD-HTD combination. These results justify our decision to use SPM in Step 1 and HTD in Step 3 of our algorithm.

6.2. Salient patterns.

6.2.1. Synthetic data. We describe the details of the synthetic data setup in Section 4.2.1 that produced Figure 5. We consider $p \in [4, 12]$. Our samples come from the distributions [2], where matrices $A \in \mathbb{R}^{p\times p}$ and $B \in \mathbb{R}^{p\times(p-1)}$ are random with unit vector columns, and the columns of B are assumed to be orthogonal. We assume the orthogonality of the columns of B to facilitate comparison with the methods cPCA and PCPCA, which require this assumption.

For testing Algorithm 2 in Figure 5(a) and (b) in the main text, variables s_i are exponential distributions $\exp(\theta_i)$ where $\theta_i = 2$ when i is odd and $\theta_i = 1.5$ when i is even. Variables z_i and z_i' are exponential distributions $\exp(\nu_i), \exp(\nu_i')$ where $\nu_i = 2, \nu_i' = 1$ when i is odd and $\nu_i = 1, \nu_i' = 2$ when i is even. We generate 10^5 data points for both the foreground and background data and apply cICA to the sample cumulant tensors. cICA has randomness coming from the subspace power method. We apply our algorithm 100 times and get 100 recovered foreground mixing matrices $B \in \mathbb{R}^{p\times(p-1)}$.

We also test Algorithm 1 here. The result is shown in Figure S4.

We let z_i, z_i' be exponential distributions $\exp(\nu_i), \exp(\nu_i')$ where $\nu_i = \nu_i' = 1$. We learn the hyperparameter γ' via Theorem 4.1 of the Appendix. The true γ' is 1 and the recovered γ' are all in the range $[0.94, 1.08]$.

We describe the implementation of the two methods we compare to. For cPCA (10), we test 100 log-evenly spaced hyperparameters α between 0 and 1000 with $p-1$ components. Each run returns a matrix of size $p \times (p-1)$, whose columns are contrastive principal components with norm 1. For PCPCA, we test 100 evenly spaced hyperparameters γ between 0 and 0.9 and fix $p-1$ components. Each run returns a matrix of size $p \times (p-1)$. We normalize the columns to unit norm, to compare PCPCA with the other algorithms.

Since the columns of B that are recovered are only unique up to permutation and sign, we describe how to align the outputs. Let $B' \in \mathbb{R}^{p\times(p-1)}$ be a recovered matrix. Rather than searching over all ways to match the columns of B to those of B', we use a greedy algorithm to approximate the matching, as follows. We fix the first column of B, denoted $\mathbf{b}_1$. We choose one of the columns of B' whose cosine similarity with $\mathbf{b}_1$ has the largest absolute value. We set this to be the first column of B', changing its sign if the cosine similarity is negative. Then we select among the remaining columns, the one with the largest absolute cosine similarity with $\mathbf{b}_2$ and set this as the second column of B'(again, changing the sign if the cosine similarity is negative). We continue until we reach the last column. Then we compute the relative Frobenius error and mean cosine similarity which are, respectively,

$$\sqrt{\sum_{i=1}^{p}\sum_{j=1}^{p-1}(b_{ij} - b'_{ij})^2/(p-1)} \qquad \text{and} \qquad \frac{1}{p-1}\sum_{i=1}^{p-1}\langle \mathbf{b}_i, \mathbf{b}'_i\rangle.$$

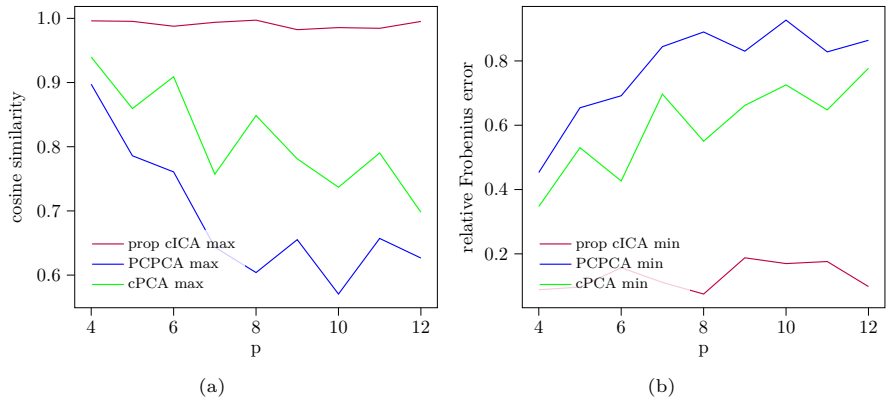

Fig. S4. The similarity of the recovered vs. true foreground patterns (i.e. the accuracy of recovering matrix B), measured via cosine similarity in (a) and relative Frobenius error in (b) . The x-axis is the number of variables p, which ranges from 4 to 12. For cPCA and PCPCA, we test 100 hyperparameter values and plot the one with the lowest error.

6.2.2. Corrupted MNIST dataset with continuous strength. For the hyperparameters of cICA, we choose the number of components to be 30, which explains 85% of the variance. We then choose r, ℓ for cICA and ℓ for proportional cICA. We order the eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ according to their absolute values and plot parts of the ordered eigenvalues in Figure S5. Based on these plots, we choose $r = 65$ and $r + \ell = 130$.

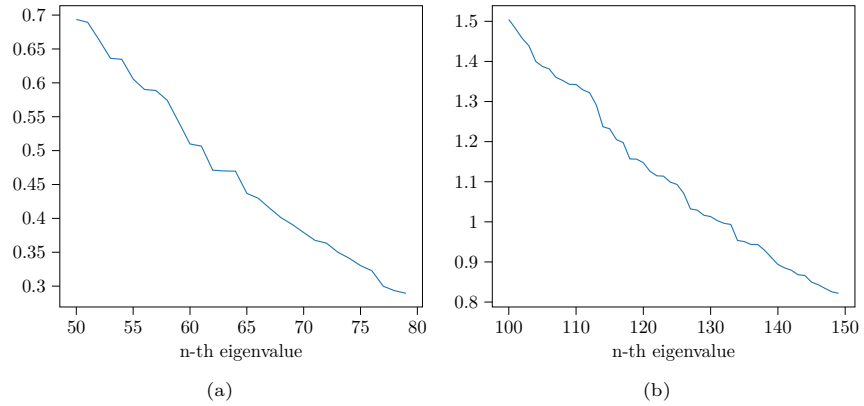

Fig. S5. Absolute values of eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ (left) and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ (right).

We fix the random seed to be 0 for cICA. We check that the absolute values of the foreground-to-background cumulant ratios for the background patterns $\mathbf{a}_1, \ldots, \mathbf{a}_r$ range from 7.2×10^{-3} to 91.

For cPCA, we run the experiment for $\alpha = 1$. We run PCPCA for $\gamma' = 0.9$.

6.2.3. Human and monkey gene expression data. We describe the patterns obtained from the comparison of human and monkey gene expression in Section 4.2.3. The selected 15 highest variance genes among the 139 selected genes in (11) are EIF3K, NDUFA13, SARNP, MYL10, TAF9, PRCD, BBS5, MRPS14, RING1, AGPAT5, FLOT1, BTBD7, MASTL, KANK1, BDP1. The 15 highest variance genes among the remaining 3244 = 3383 – 139 genes are LUC7L3, RBKS, RBM7, AP4S1, CLCN1, CLASP1, ADTRP, CNNM3, NDUFAF7, CNIH4, RPUSD2, NELFCD, RPP14, ROMO1, RNF181.

For cICA, we fix the random seed to be 0. We use the plots of the eigenvalues of the flattenings of $\kappa_4(\mathbf{y}), \kappa_4(\mathbf{x})$ to choose $r = 22$ and $\ell = 46 - 22 = 24$. We check that the absolute values of the foreground-to-background cumulant ratios for the background patterns $\mathbf{a}_1, \ldots, \mathbf{a}_r$ range from 4.6×10^{-2} to 55 illustrating that the shared gene patterns between human and monkey have different strength across the two datasets.

The top two foreground patterns are:

$$\begin{aligned}\mathbf{b}_1^T = [&-0.04, -0.041, -0.09, -0.051, -0.12, 0.075, 0.01, -0.004, 0.002, 0.007,\\ &-0.07, -0.061, 0.95, 0.192, -0.009, -0.007, -0.002, -0.001, -0.076, -0.042,\\ &-0.008, -0.04, 0.005, -0.058, 0.012, -0.012, -0.05, -0.006, -0.046, -0.005]\\ \mathbf{b}_2^T = [&0.615, -0.166, 0.185, 0.119, 0.113, -0.099, -0.118, 0.011, 0.045, -0.025,\\ &0.098, 0.141, -0.482, -0.339, 0.054, 0.028, -0.005, 0.03, 0.247, -0.017,\\ &-0.031, 0.043, 0.012, 0.043, 0.015, 0.04, 0.025, 0.002, 0.236, -0.016],\end{aligned}$$

where the coordinates are labeled by the 30 genes in the order listed above. The 15 genes with the largest absolute values of the top foreground pattern include 10 genes among the 139 selected in (11). The 15 genes with the largest absolute values of the second foreground pattern include 13 genes from (11). Therefore, the foreground patterns obtained via cICA demonstrate consistency with the finding in (11) that this subset of 139 genes captures human-specific information.

For ICA, we run HTD for $r = 46$ and rank the patterns according to [10].

We denote $(\mathbf{b}_1 < 15)$ (resp. $(\mathbf{b}_2 < 15)$) to be the number of genes among top 15 ones with the largest absolute value in $\mathbf{b}_1$ that are contained in the 139 evolutional relevant genes.

We run cPCA for 100 α between 0 to 1000 and choose α that achieves the highest value of $(\mathbf{b}_1 < 15) + (\mathbf{b}_2 < 15)$. The highest value is obtained at $\alpha = 0.17$. Note that our parameters for proportional cICA are square of the cPCA parameters, since if $\mathbf{z} = \lambda \mathbf{z}'$, then $\kappa_2(\mathbf{z}) = \lambda^2 \kappa_2(\mathbf{z}')$ and $\kappa_4(\mathbf{z}) = \lambda^4 \kappa_4(\mathbf{z}')$.

We run PCPCA for 100 evenly spaced γ' values between 0 and 0.9. The best score of $(\mathbf{b}_1 < 15) + (\mathbf{b}_2 < 15)$ is obtained for $\gamma' = 0$.

We also run the algorithm for 100 log-evenly spaced γ between 0 and 10^6 and choose γ that achieves the highest value of $(\mathbf{b}_1 < 15) + (\mathbf{b}_2 < 15)$. The highest score is achieved at $\gamma = 0.03$. 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 (11). The number of misclassified genes in this case is 6.

6.3. Dimensionality reduction.

6.3.1. Mouse protein data. There are 270 foreground samples. These are the protein expression in the cortex of mice subjected to shock therapy. Of these samples, 135 have Down syndrome and 135 do not. There are 135 background samples, protein expression measurements from mice without Down Syndrome who did not receive shock therapy. Each sample measures the expression of 77 proteins; that is, $p = 77$.

For cICA, we preprocess using PCA as described in Section 3.2. We take $k = 15$ components, which explain 90% of the variance. We then choose r and ℓ, as described in Appendix section 5.1. That is, we compute the eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$, ranking the eigenvalues by magnitude, see Figure S6. Based on these plots, we choose $r = 27$ and $\ell = 53 - 27 = 26$.

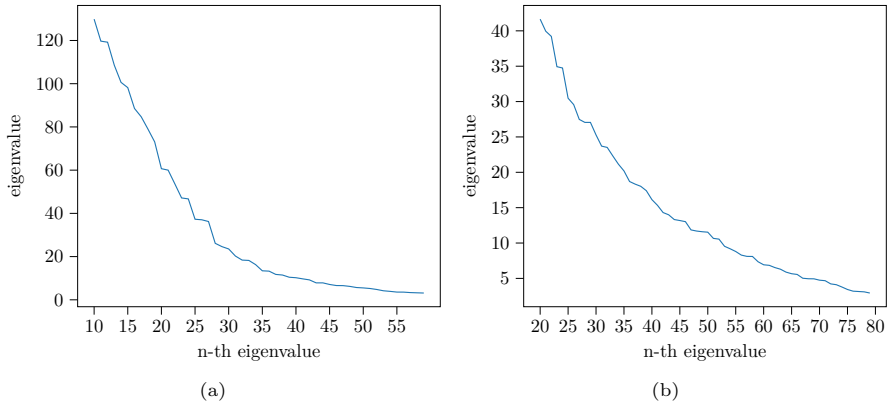

Fig. S6. Absolute values of eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ (left) and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ (right).

For cICA, we fix the random seed to be 0. For proportional cICA, we run the algorithm for 100 log-evenly spaced γ between 0 and 10^6. The highest silhouette score is obtained at $\gamma = 0$, equivalent to running ICA.

We run cPCA for 100 α between 0 to 1000. These are the default values of α in the code of (10). We plotted the choice with the highest silhouette score, which was achieved for $\alpha = 26.2$.

We run PCPCA for 100 evenly spaced γ' values between 0 and $0.9 \cdot \frac{270}{135}$. 270 and 135 are the number of samples in the foreground and background datasets, respectively. Such choices of γ' are in accordance with the setup in (12) and are sufficient to find the highest silhouette score. The best score was obtained when $\gamma' = 0.9 \cdot \frac{270}{135}$. In (12), the authors take a further step to scale the probabilistic contrastive principal components, before calculating the silhouette score. The silhouette score obtained after this additional step is 0.450.

6.3.2. Corrupted MNIST data with discrete strength. For the hyperparameters of cICA, we choose the number of components to be 30, which explains 85% of the variance. We then choose r, ℓ for cICA and ℓ for proportional cICA. We order the eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ according to their absolute values and plot parts of the ordered eigenvalues in Figure S7. Based on these plots, we choose $r = 51$ and $r + \ell = 192$. The absolute values of the foreground-to-background cumulant ratios for the background patterns $\mathbf{a}_1, \ldots, \mathbf{a}_r$ range from 6.7×10^{-3} to 16.

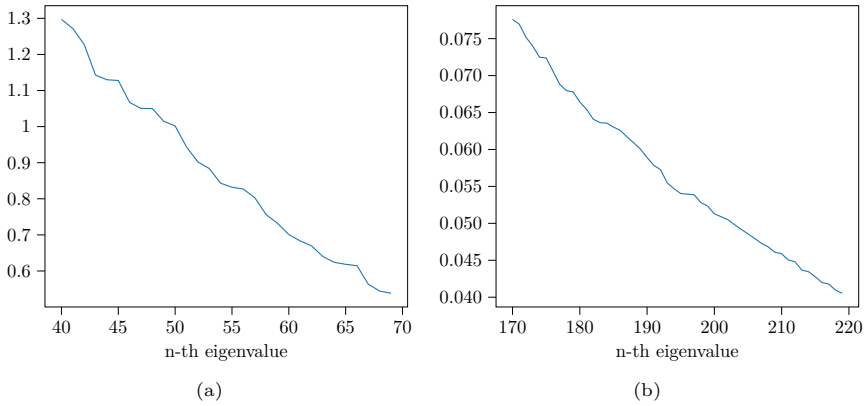

Fig. S7. Absolute values of eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ (left) and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ (right).

We fix the random seed to be 0 for cICA. For cPCA, we run experiments for 100 α values between 0 and 1000 and choose $\alpha = 6.6$ that achieves the highest silhouette score when plotting the mixed images of digits 0 and 1 using their inner product with the first two patterns. We run PCPCA for 100 evenly spaced γ' between 0 and 0.9 and choose the $\gamma' = 0.9$ with the highest silhouette score when plotting with the first two patterns.

We also include ICA with $r = 192$ to illustrate that cICA performs significantly better than ICA.

7. Additional numerical experiment

7.1. Single cell RNA data. We study the single-cell RNA sequencing data from (13). The foreground data points are gene expressions of bone marrow mononuclear cells from patients with acute myeloid leukemia before and after they received a stem-cell transplant; the background dataset contains gene expression measurements of healthy people. The foreground dataset includes 7525 pre-transplant patients and 4874 post-transplant patients, while the background dataset consists of 4457 healthy patients. Each sample contains gene expression measurements of bone marrow mononuclear cells. We preprocess the data by log-transforming and subsetting to the 500 most variable genes, in accordance with previous analyses on these data (12–14).

For cICA, the absolute values of the foreground-to-background cumulant ratios for the background patterns $\mathbf{a}_1, \ldots, \mathbf{a}_r$ range from 1.5×10^{-4} to 564. The projection plots of cICA, proportional cICA, cPCA, and PCPCA are shown in Figure S8. The method cPCA has the highest silhouette score (0.451), followed by proportional cICA (0.402), then cICA (0.344), then PCPCA (0.164). We also run ICA to the foreground dataset and it has silhouette score 0.202 for comparison with cICA.

For the hyperparameters of cICA and proportional cICA, we choose the number of components to be 30 which explains 54.5% of the variance. We then choose r, ℓ for cICA and ℓ for proportional cICA. We order the eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ according to their absolute values and plot out parts of the ranked eigenvalues in Figure S9. We choose $r = 53$ and $r + \ell = 116$.

We fix the random seed to be 0 for cICA and ICA. For ICA, we run the HTD algorithm for $r = 116$.

For proportional cICA, we run the algorithm for 100 log-evenly spaces γ between 0 and 10^6. The highest silhouette score is 0.402, obtained when $\gamma = 0.50$.

For cPCA, we plot the first two cPCA components. As above, we run cPCA using 100 α between 0 to 1000, the default values from (10). The highest silhouette score is 0.457, obtained when $\alpha = 3.5$.

We run PCPCA for 100 evenly spaced γ' between 0 and $0.9 \cdot \frac{12399}{4457}$, in accordance with (12). The numbers 12399 and 4457 are the sample sizes of the foreground and background datasets, respectively. In accordance with the experiment in (10), we run PCPCA with 4 components. The best silhouette score over any γ' and any pair of probabilistic contrastive principal components is 0.164, obtained when $\gamma' = 0.41$ using the third and fourth components. If we normalize the probabilistic contrastive principal components and then calculate the silhouette score, the score is 0.184.

There are three reasons why the silhouette score for cICA methods is suboptimal compared to that of cPCA.

1. Due to the computational cost of forming large tensors, cICA methods is applied to the PCA transformed dataset using the top 30 principal components, which explains only 54.5% of the variance. Consequently, the clustering quality is expected to be lower than when cPCA is applied to the complete dataset.

2. Our cICA methods return patterns that only exist in the foreground while cPCA learns patterns that are more prominent in the foreground than in the background.

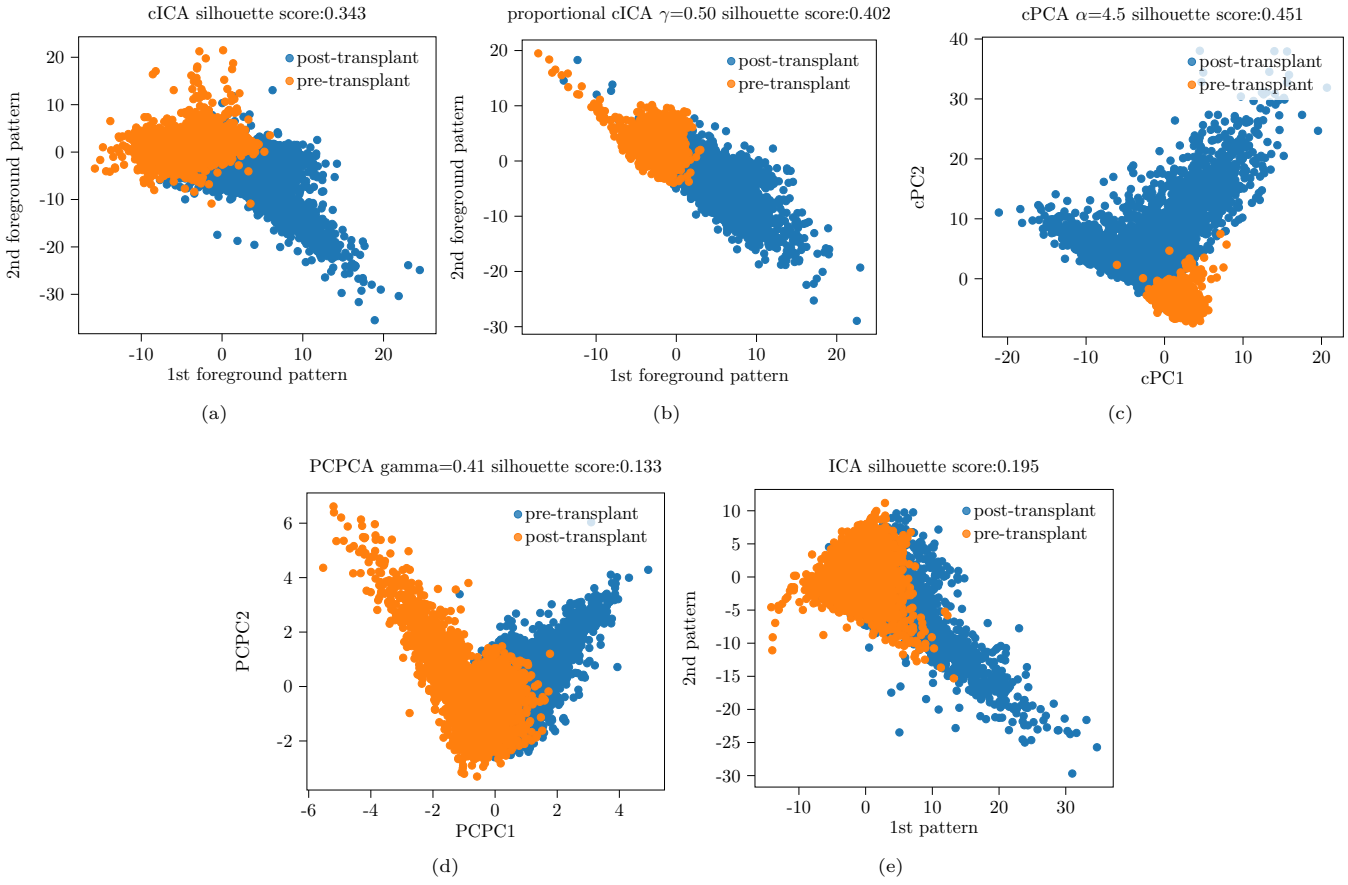

Fig. S8. Dimensionality reduction of the single-cell RNA sequencing data from (13) via (a) cICA (b) proportional cICA (c) cPCA (d) PCPCA (e) ICA.

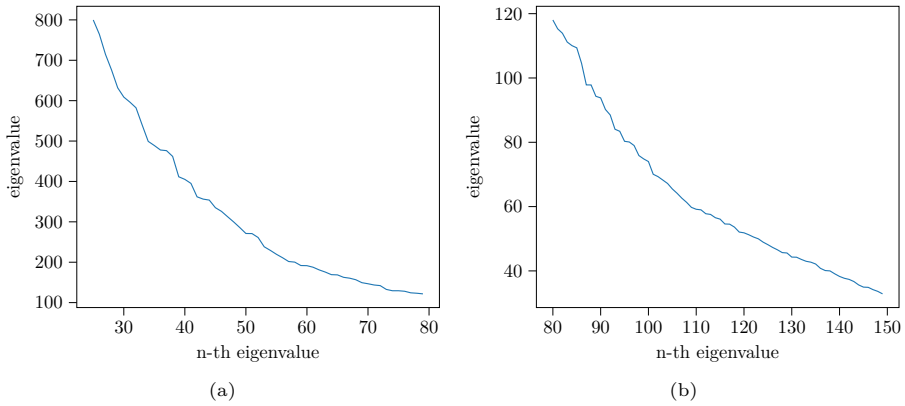

Fig. S9. Absolute values of eigenvalues of $\mathrm{Mat}(\kappa_4(\mathbf{y}))$ (left) and $\mathrm{Mat}(\kappa_4(\mathbf{x}))$ (right).

3. The patterns learned by cICA do not have any relation while cPCA returns perfectly orthogonal patterns. The patterns from cICA may enjoy better intrepretability but produce suboptimal plots than cPCA.

To illustrate these arguments, we generate plots using cPCA and cICA as follows. We apply cPCA to the PCA transformed dataset using the top 30 principal components. The plot obtained using the top two cPCA components is shown in Figure S10(a). The silhouette score achieved is 0.434. For cICA, we apply proportional cICA to the PCA transformed dataset using the same hyperparameters as above. We select the top foreground pattern $\mathbf{b}$ and the top background pattern $\mathbf{a}$ ranked according to [10]. We then use $\mathbf{b}$, $\frac{\mathbf{a}-\langle\mathbf{a},\mathbf{b}\rangle\mathbf{b}}{\|\mathbf{a}-\langle\mathbf{a},\mathbf{b}\rangle\mathbf{b}\|}$ as directions to plot the data. The plot is shown in S10(b). The silhouette score obtained is 0.428, almost the same as that of cPCA.

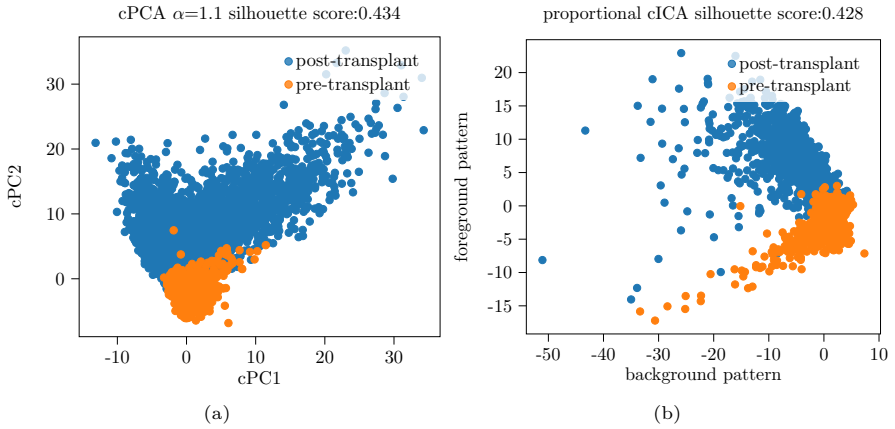

Fig. S10. (a) cPCA for the dataset obtained from the top 30 PCA components (b) Proportional cICA plot projected to the top foreground and the top background pattern.

References

1. W Hackbusch, *Tensor spaces and numerical tensor calculus.* (Springer) Vol. 42, (2012).
2. J Salmi, A Richter, V Koivunen, Sequential unfolding SVD for tensors with applications in array signal processing. *IEEE Transactions on Signal Process.* **57**, 4719–4733 (2009).
3. RA Harshman, Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Work. Pap. Phonetics* **16**, 84 (1970).
4. TG Kolda, Symmetric orthogonal tensor decomposition is trivial. *arXiv preprint arXiv:1503.01375* (2015).
5. Y Yu, T Wang, RJ Samworth, A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika* **102**, 315–323 (2015).
6. 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).
7. Z Li, Y Nakatsukasa, T Soma, A Uschmajew, On orthogonal tensors and best rank-one approximation ratio. *SIAM J. on Matrix Analysis Appl.* **39**, 400–425 (2018).
8. K Kozhasov, J Tonelli-Cueto, Probabilistic bounds on best rank-1 approximation ratio. *Linear Multilinear Algebr.* **72**, 3000–3028 (2024).

9. A Anandkumar, R Ge, M Janzamin, Sample complexity analysis for learning overcomplete latent variable models through tensor methods. *arXiv preprint arXiv:1408.0553* (2014).
10. A Abid, MJ Zhang, VK Bagaria, J Zou, Contrastive principal component analysis. *arXiv preprint arXiv:1709.06716* (2017).
11. H Suresh, et al., Comparative single-cell transcriptomic analysis of primate brains highlights human-specific regulatory evolution. *Nat. Ecol. & Evol.* **7**, 1930–1943 (2023).
12. D Li, A Jones, B Engelhardt, Probabilistic contrastive principal component analysis. *arXiv preprint arXiv:2012.07977* (2020).
13. GX Zheng, et al., Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
14. A Abid, MJ Zhang, VK Bagaria, J Zou, Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nat. Commun.* **9**, 2134 (2018).