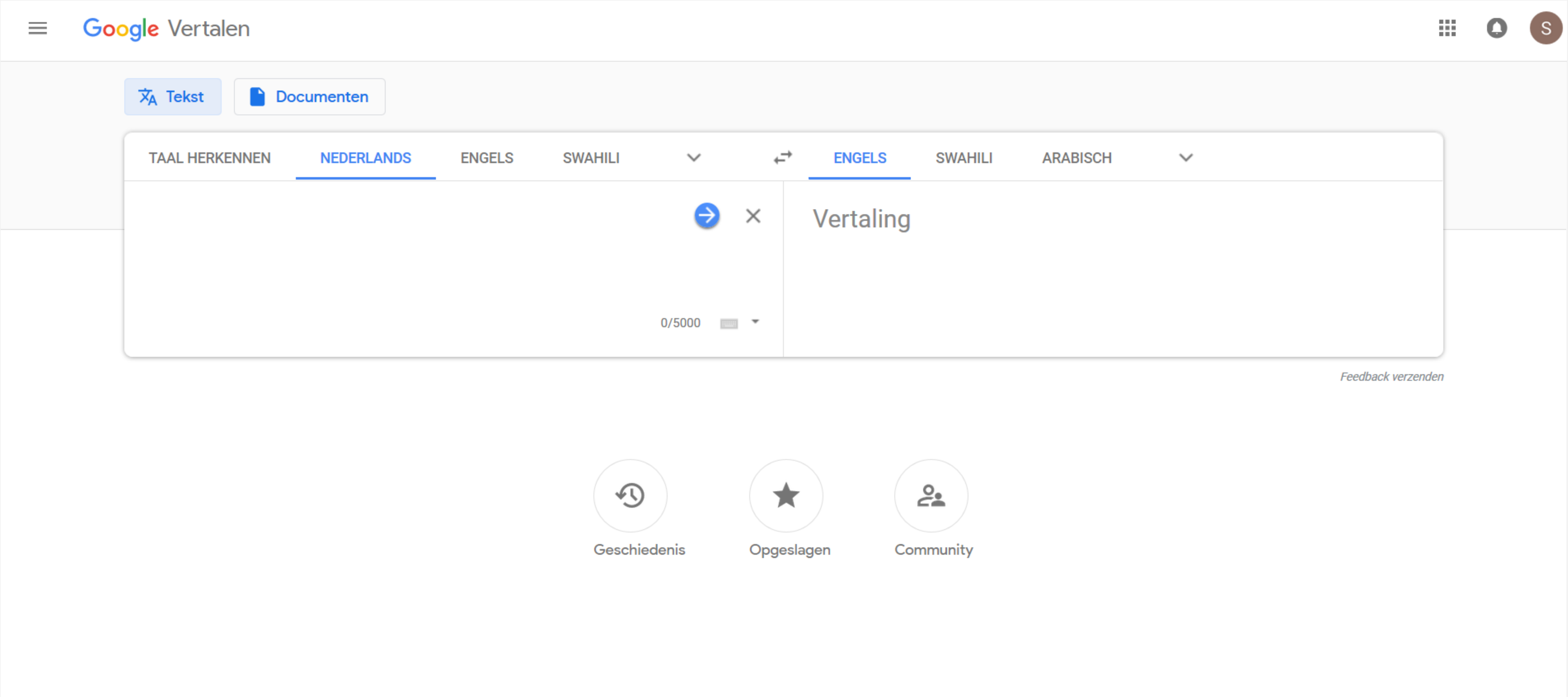
Identifying stable components of matrix/tensor factorizations via low-rank approximation of inter-factorization similarity

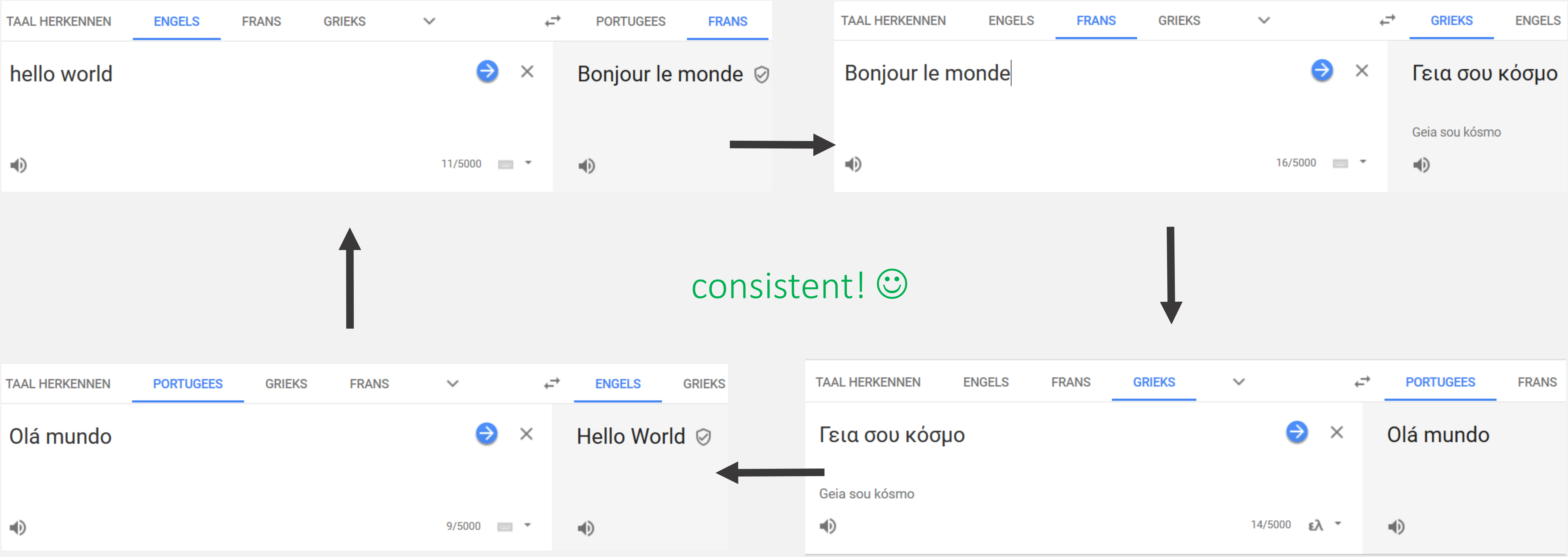
Simon Van Eyndhoven*, Nico Vervliet, Lieven De Lathauwer, Sabine Van Huffel

European Signal Processing Conference, A Coruña, Spain
3 September 2019

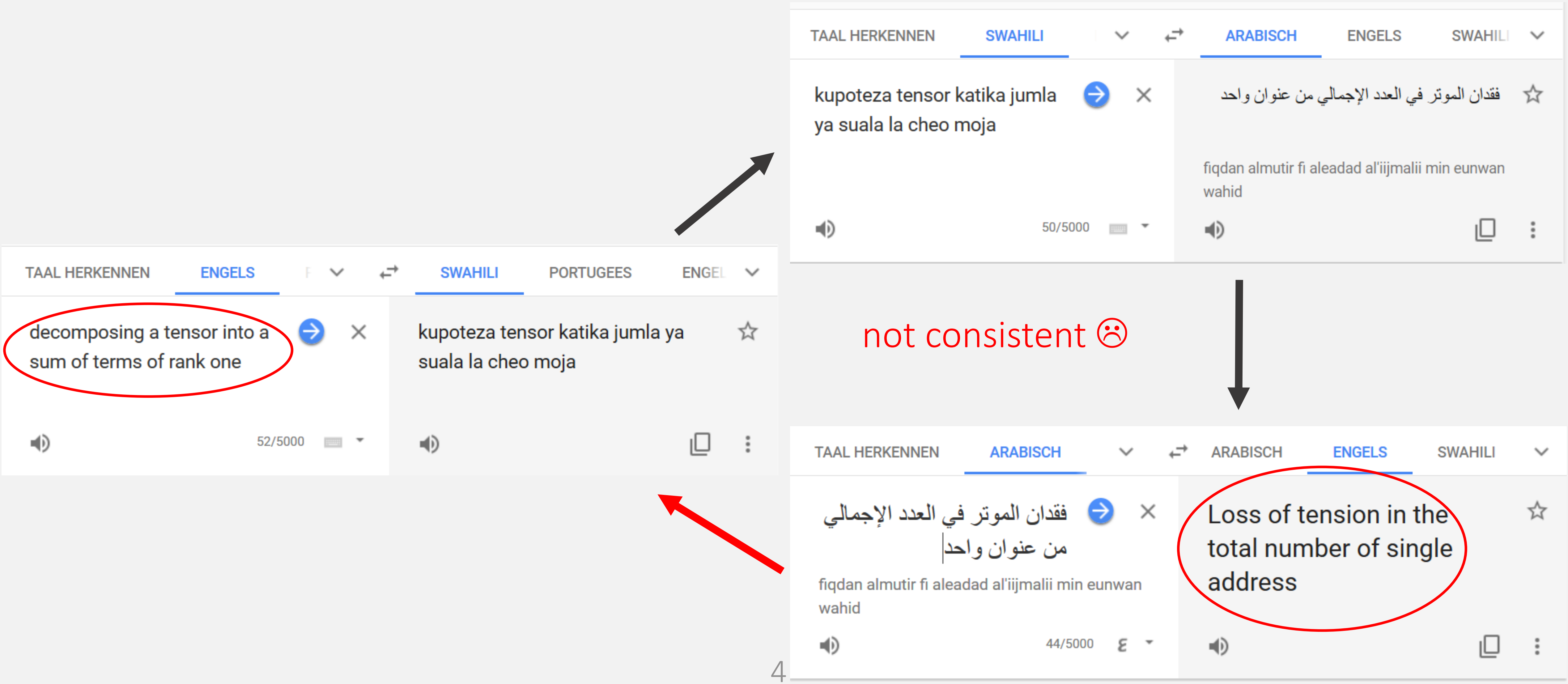
Example 1: machine translation



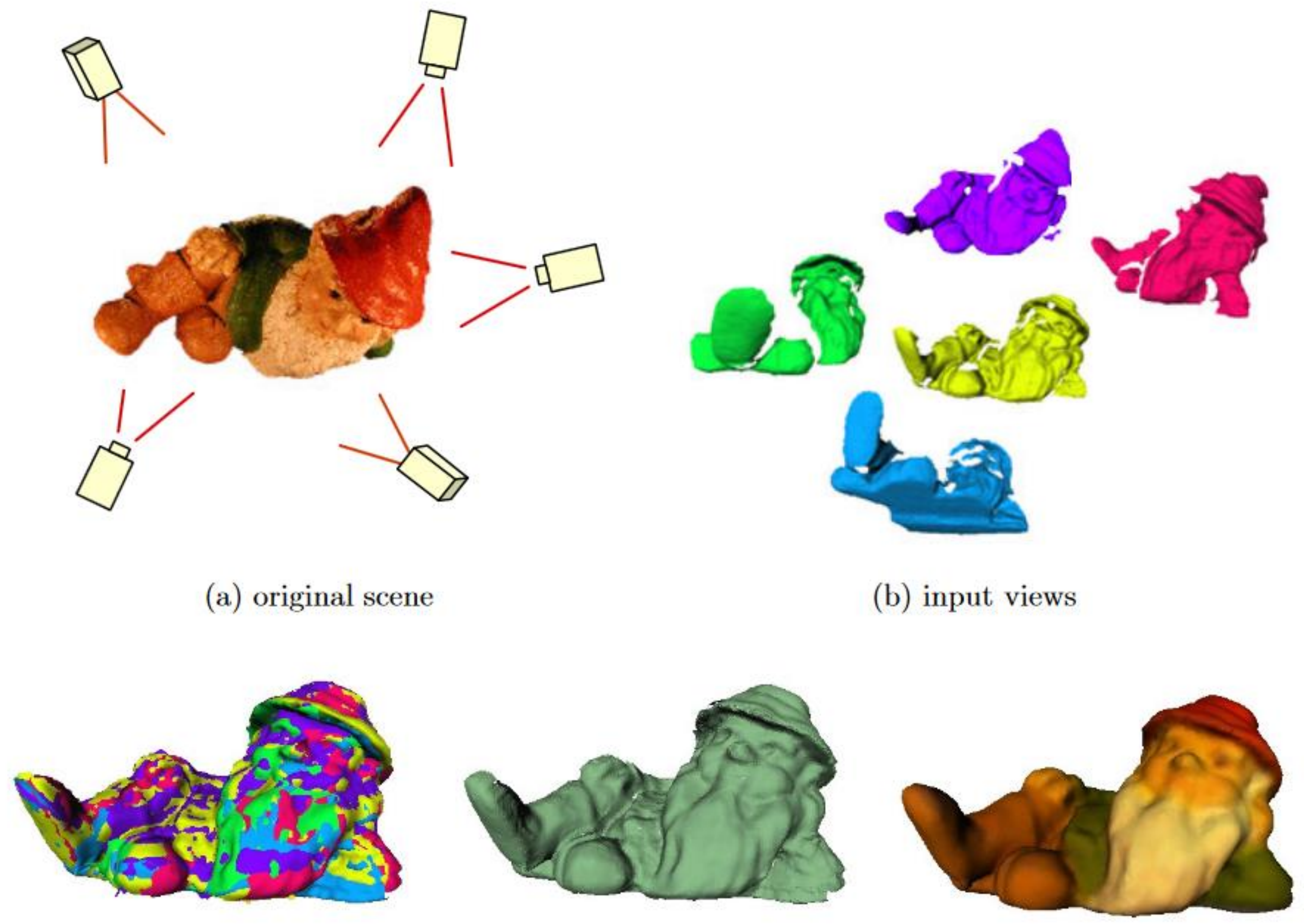
Example 1: machine translation



Example 1: machine translation



Example 2: Reconstructing 3D models



(a) original scene

(b) input views

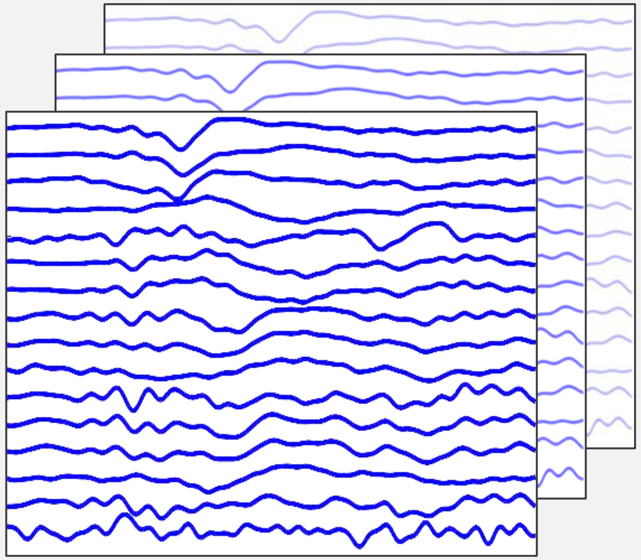
(c) registered views

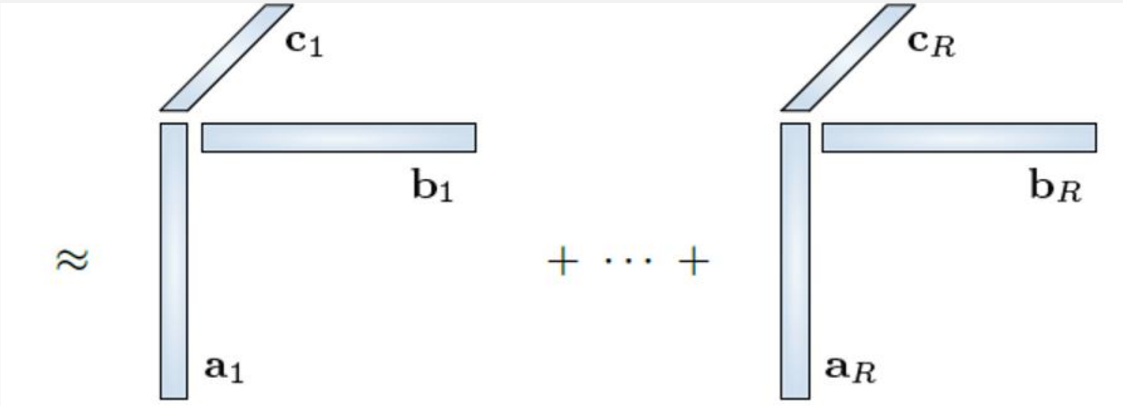
(d) integrated model

(e) texture mapped model

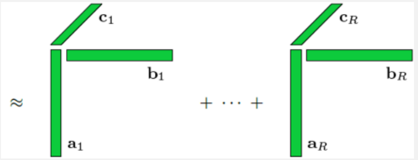
D. Huber, *“Automatic three-dimensional modeling from reality”,*

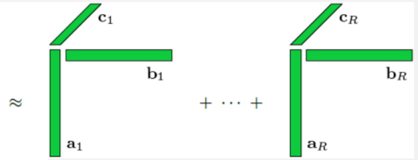
Example 3: non-convex blind source separation

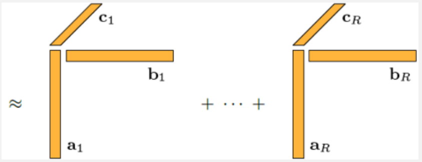


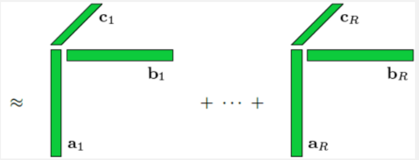


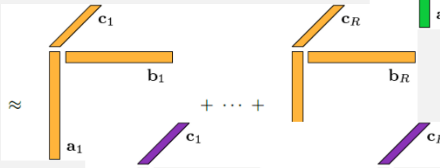
trust these estimates?

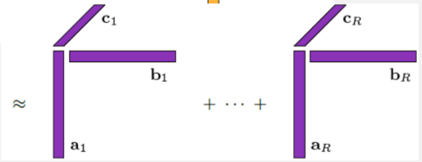




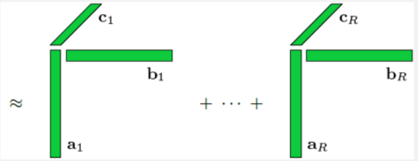


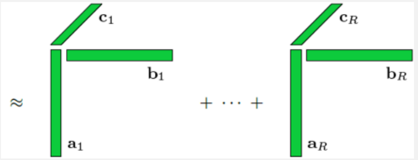


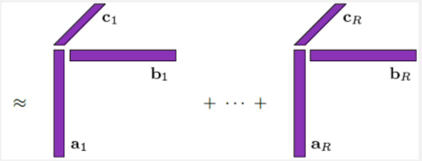


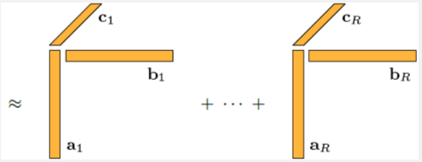


= which CPD components are reproducible?

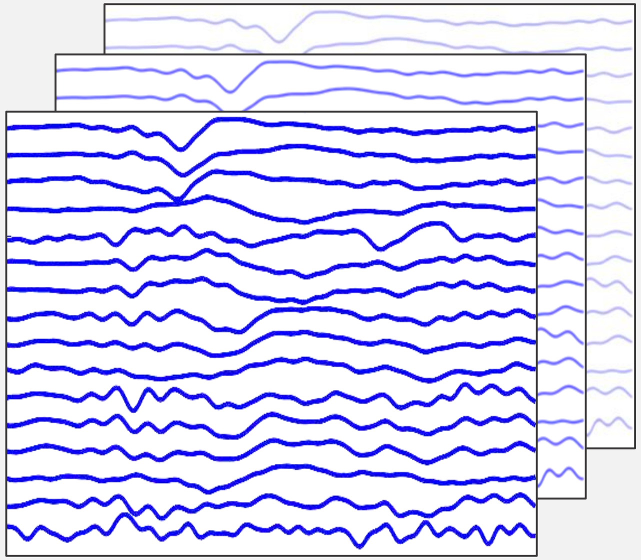


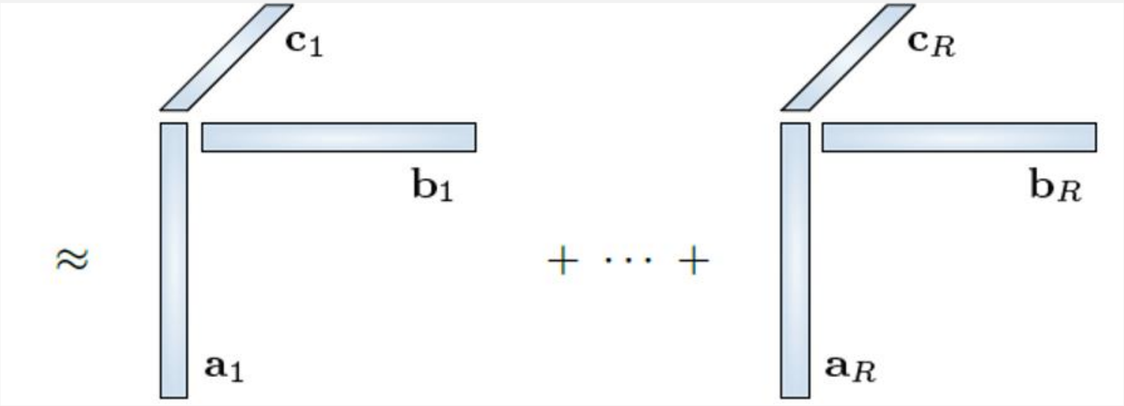






Problem statement: stability analysis in BSS





Identify clusters of local solutions found by optimization.

We can use the same notion of consistency as before!

Cycle consistency

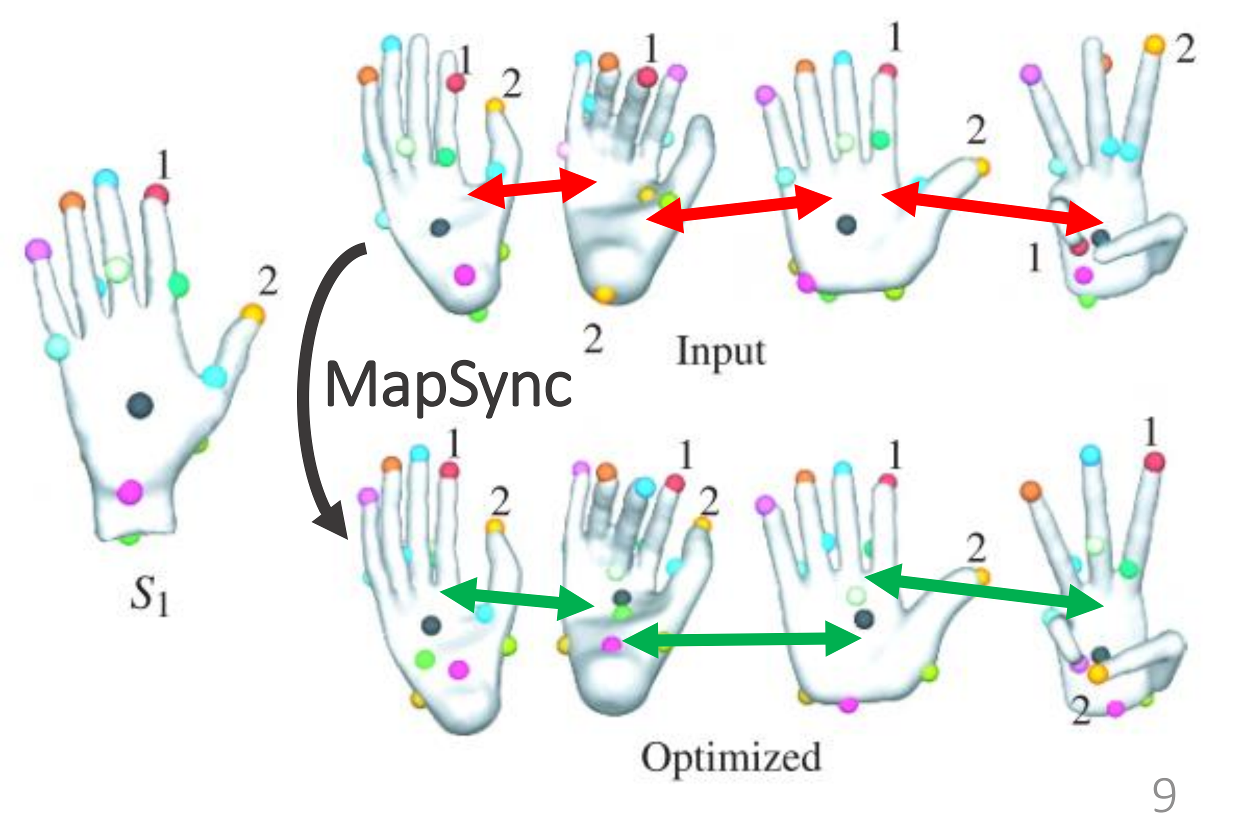
= an intuitive property of (some) graphs

Leveraging cycle consistency

from noisy, pairwise, local matching between objects…

reconstruct globally consistent correspondence maps

This is called **map synchronization**.



Huang, Qi-Xing, and Leonidas Guibas.
"Consistent shape maps via semidefinite programming."

Cycle consistency is a strong constraint

local (pairwise) map:

m[th] decomposition

n[th] decomposition

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{X}_{nm}$$

all pairwise maps:

$$\begin{bmatrix} \mathbf{I} & \mathbf{X}_{12} & \cdots & \mathbf{X}_{1N} \\ \mathbf{X}_{12}^{\mathrm{T}} & \mathbf{I} & \cdots & \mathbf{X}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{1N}^{\mathrm{T}} & \mathbf{X}_{2N}^{\mathrm{T}} & \cdots & \mathbf{I} \end{bmatrix}$$

$$\mathbf{X}_{mp}\mathbf{X}_{pr}\cdots\mathbf{X}_{kn}\mathbf{X}_{nm} \stackrel{?}{=} \mathbf{I} \quad \forall\,\text{cycle}$$

= inter-factorization adjacency / mapping / matching matrix

Different view of cycle consistency

1 To step from m^{th} decomposition to n^{th} decomposition: premultiply with map $\mathbf{X}_{nm}$

2 iff cycle consistent: all decompositions are (partial) instances of a latent "universe" that contains the R true components

3 Alternative map from m to n:
1. map from m to the universe: $\mathbf{X}_{um} = \mathbf{X}_{mu}^T$
2. map from the universe to n: $\mathbf{X}_{nu}$

$$\mathbf{X}_{nm} = \mathbf{X}_{nu}\mathbf{X}_{mu}^T$$

Cycle consistent graphs are low-rank

translates to a low-rank,
positive semi-definite structure!

$$\mathbf{X}_{nm} = \mathbf{X}_{nu}\mathbf{X}_{mu}^{T} \quad \forall\, n, m$$

$$\underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{X}_{12} & \cdots & \mathbf{X}_{1N} \\ \mathbf{X}_{12}^{\mathrm{T}} & \mathbf{I} & \cdots & \mathbf{X}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{1N}^{\mathrm{T}} & \mathbf{X}_{2N}^{\mathrm{T}} & \cdots & \mathbf{I} \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} \mathbf{X}_{1u} \\ \mathbf{X}_{2u} \\ \vdots \\ \mathbf{X}_{Nu} \end{bmatrix}}_{\mathbf{X}_u} \underbrace{\begin{bmatrix} \mathbf{X}_{1u}^{\mathrm{T}} & \mathbf{X}_{2u}^{\mathrm{T}} & \cdots & \mathbf{X}_{Nu}^{\mathrm{T}} \end{bmatrix}}_{\mathbf{X}_u^{T}}$$

$rank(\mathbf{X})$ = size of universe = # distinct BSS components

Identifying stable BSS components via MapSync

1 Build (thresholded) inter-factorization similarity matrix $\mathbf{X} \in \mathbb{R}^{NR \times NR}$

2 Determine Q, the size of the universe $Q = \arg\max_r \frac{\lambda_r - \lambda_{r+1}}{|\lambda_r| + |\lambda_{r+1}|}$

3 Compute a symmetric rank-Q factorization of X $\mathbf{X} \approx \mathbf{X}_u \mathbf{X}_u^T$

= enforce cycle consistency

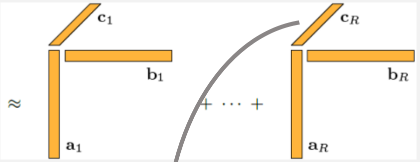
4 Project the obtained maps onto space of permutations $\tilde{\mathbf{X}} \in \{0, 1\}^{NR \times NR}$

= restore a denoised graph

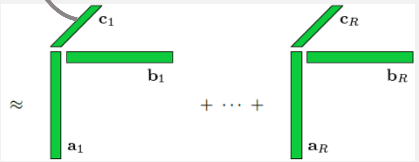
5 Identify the Q clusters of recurrent components in the approximated graph

Similarity metric

similarity between
rth component in run **i** --- **s**th component in run **j**:



$$\sigma(r_i, s_j) = \prod_{m \in \mathcal{M}} \sigma_m(r_i, s_j) = \left| \prod_{m \in \mathcal{M}} \langle \mathbf{a}_{r_i}^{(m)}, \mathbf{a}_{s_j}^{(m)} \rangle \right|^{\frac{1}{|\mathcal{M}_\sigma|}}$$



apply a threshold: e.g. 0.95

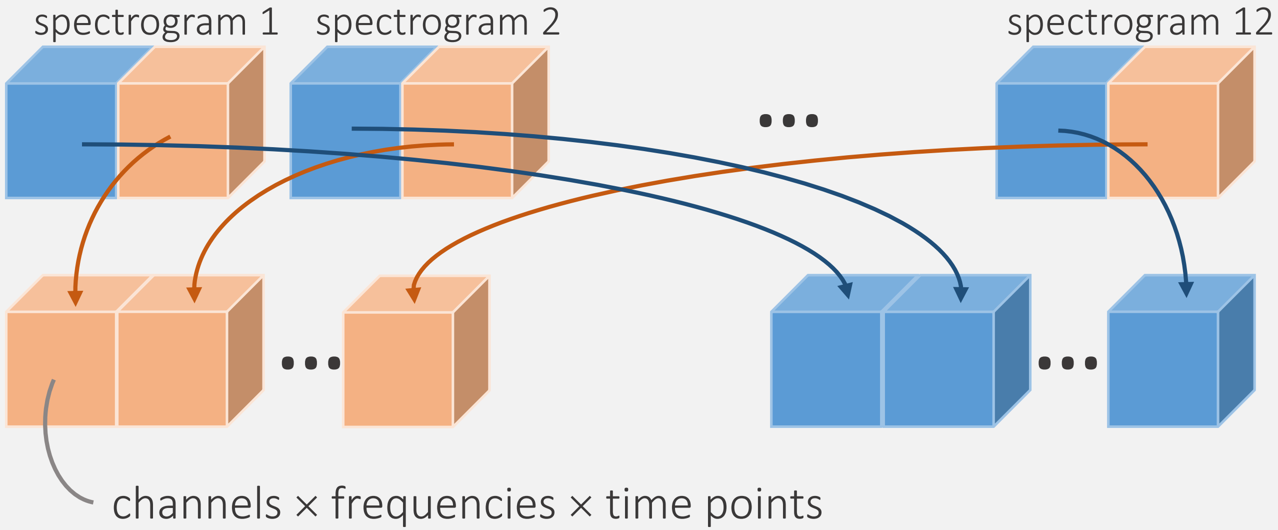
Note:
Metric can be tailored to the decomposition, e.g. KL divergence for non-negative factorizations.

MapSync4BSS: experiment 1

Canonical Polyadic Decomposition (CPD)

CPD of resting-state EEG data

We concatenated EEG spectrograms from 12 patients, and divided the data in 2 halves

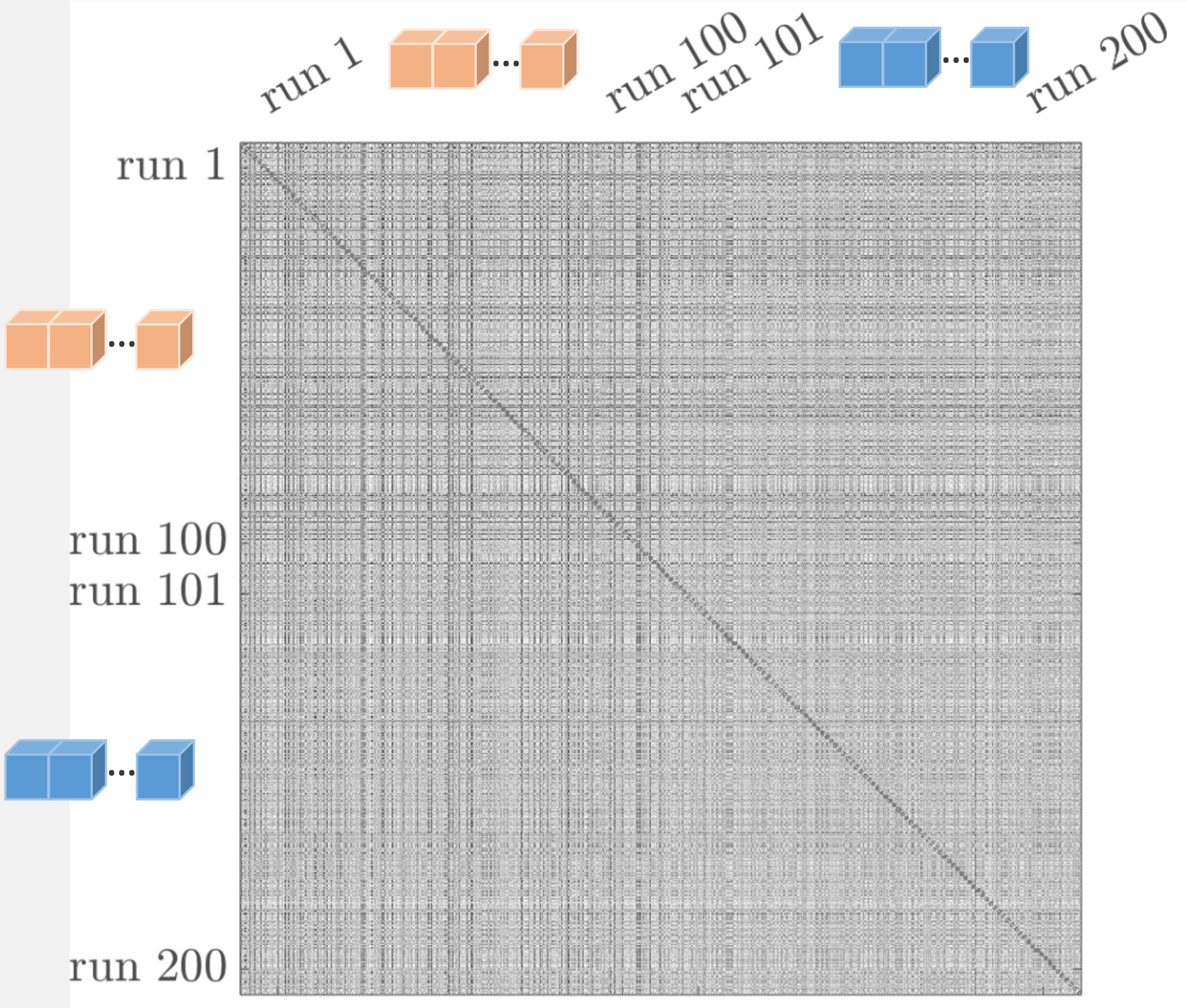


We compute 100× a rank-10 CPD of both halves

Q: Which spatial-spectral patterns are common to both halves, and which ones are specific?

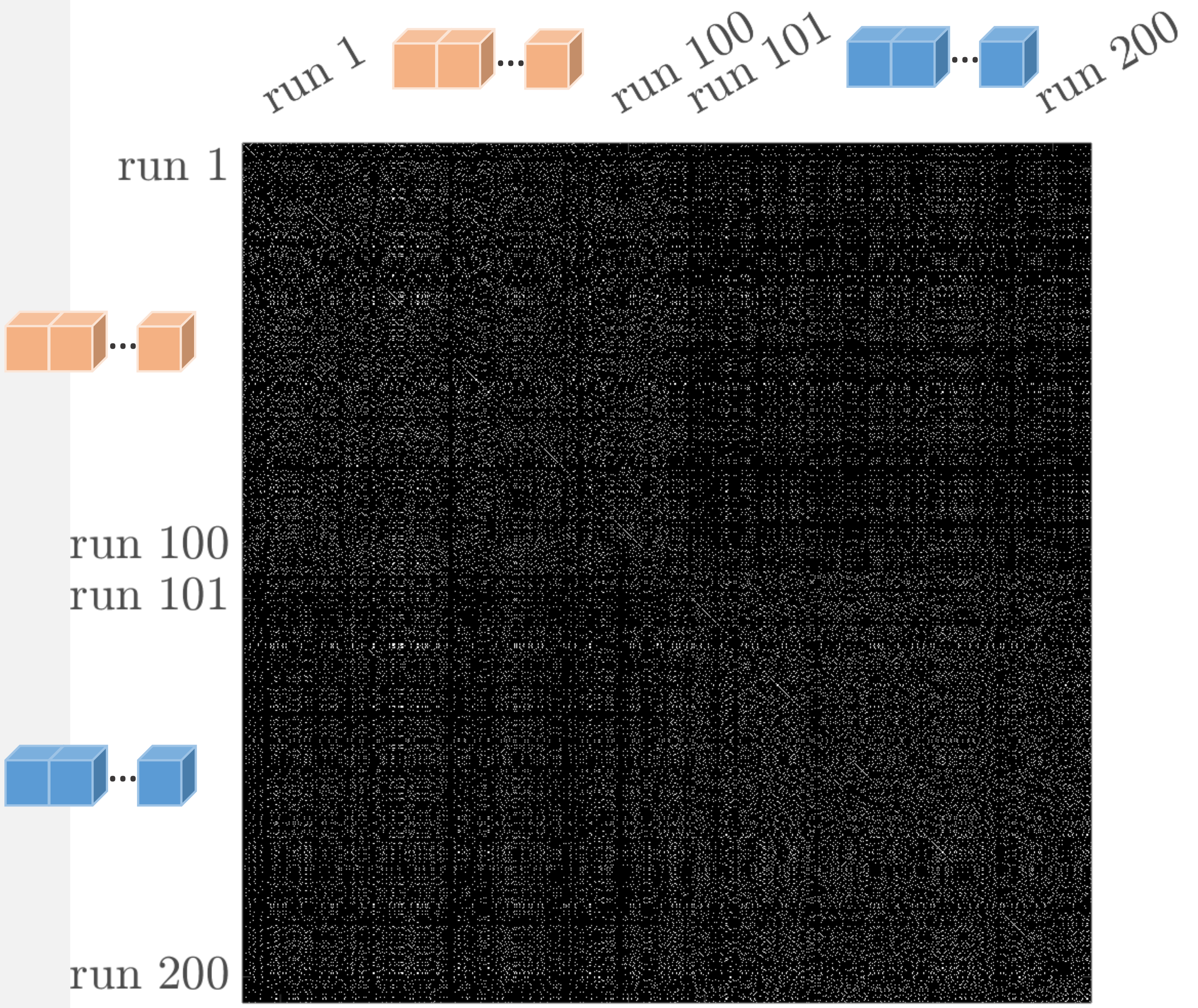
CPD of EEG data: similarity

1



CPD of EEG data: similarity

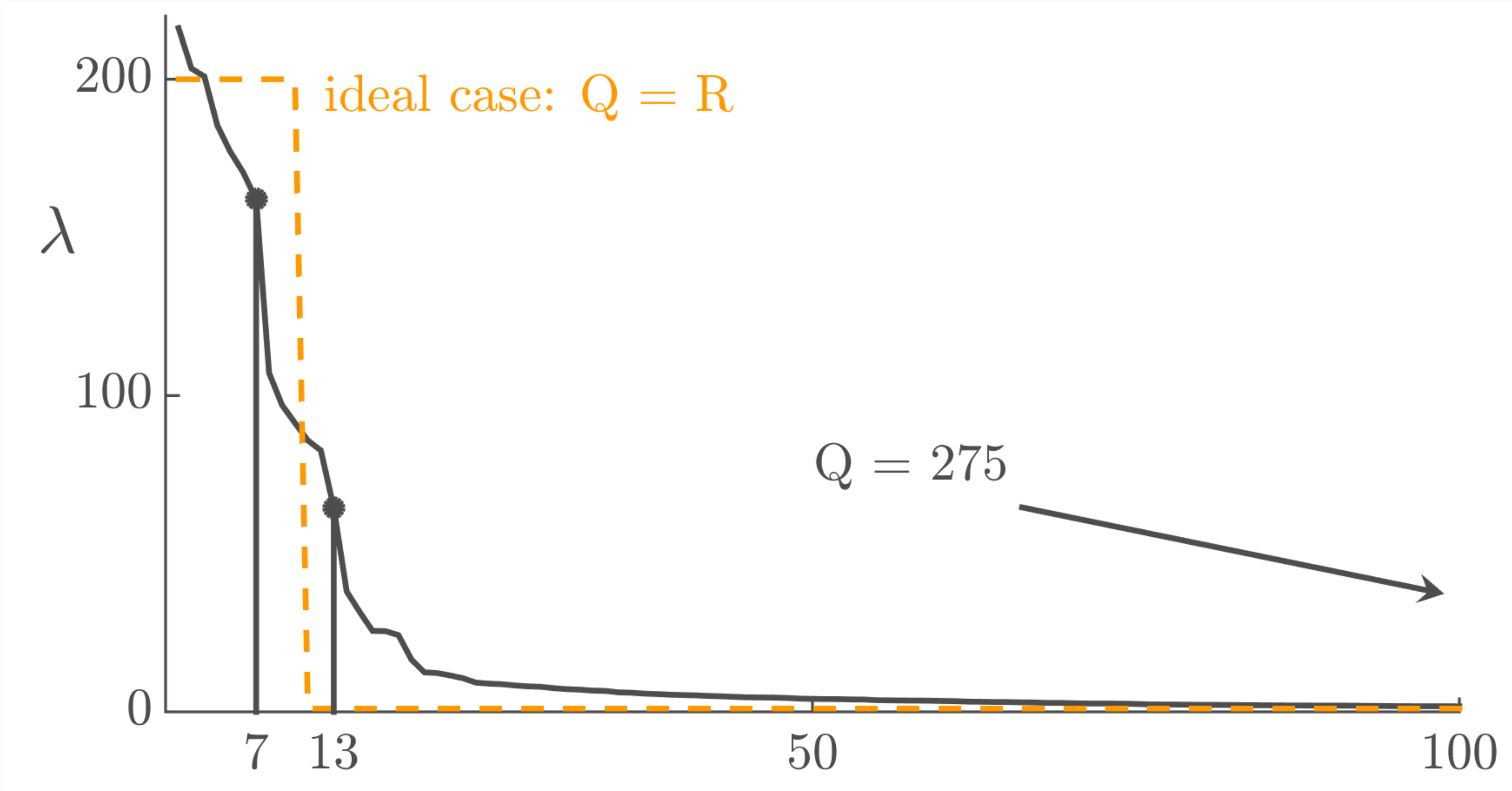
1



threshold at 0.95

CPD of EEG data: variability

2



CPD of EEG data: enforcing consistency

3

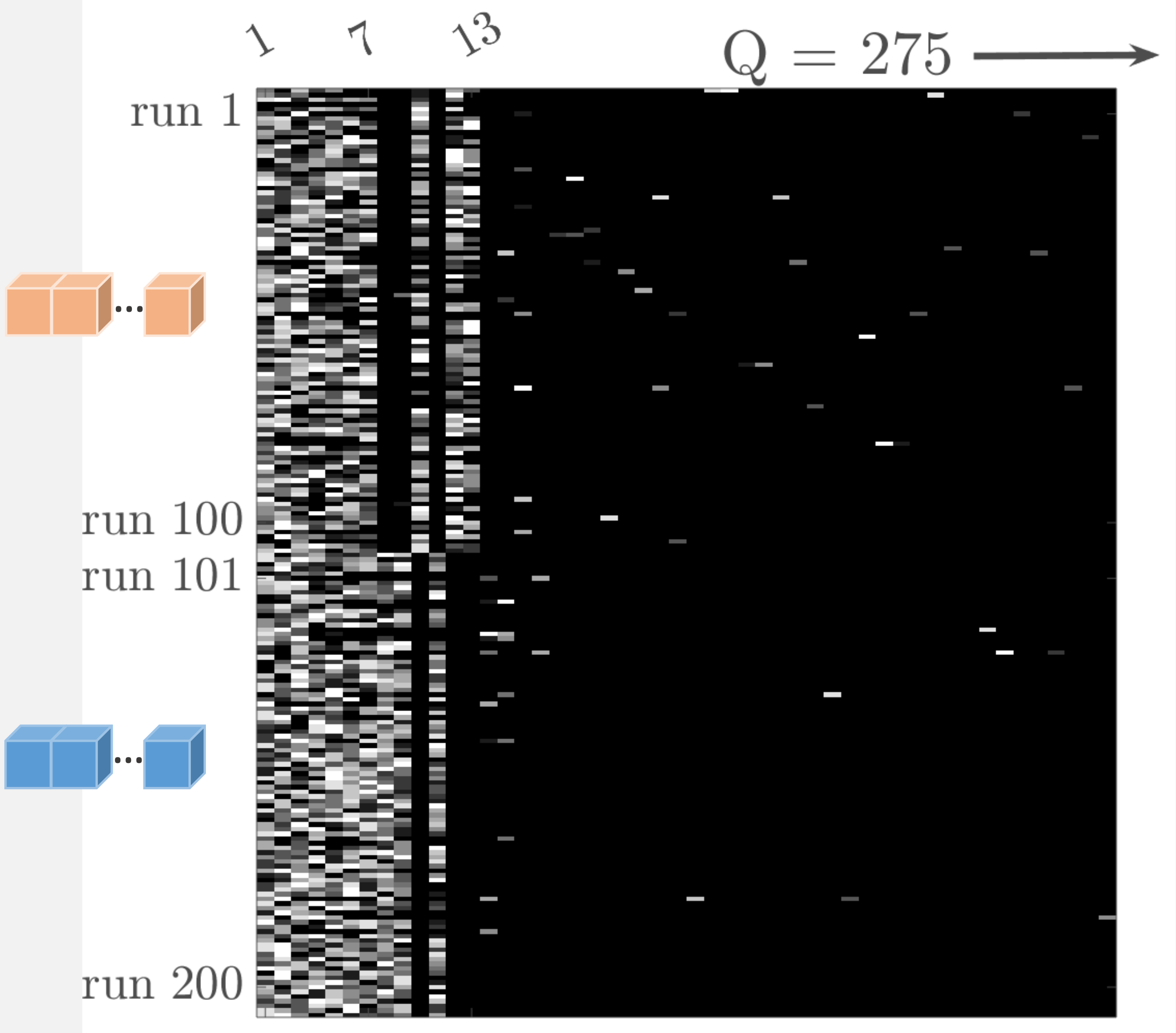
```
>> [ V , D ] = eig(X)
>> % sort eigenvalues and vectors
>> Xu = V(:,1:Q) * sqrt(D(1:Q,1:Q))
```

CPD of EEG data: denoise / cluster

4

+

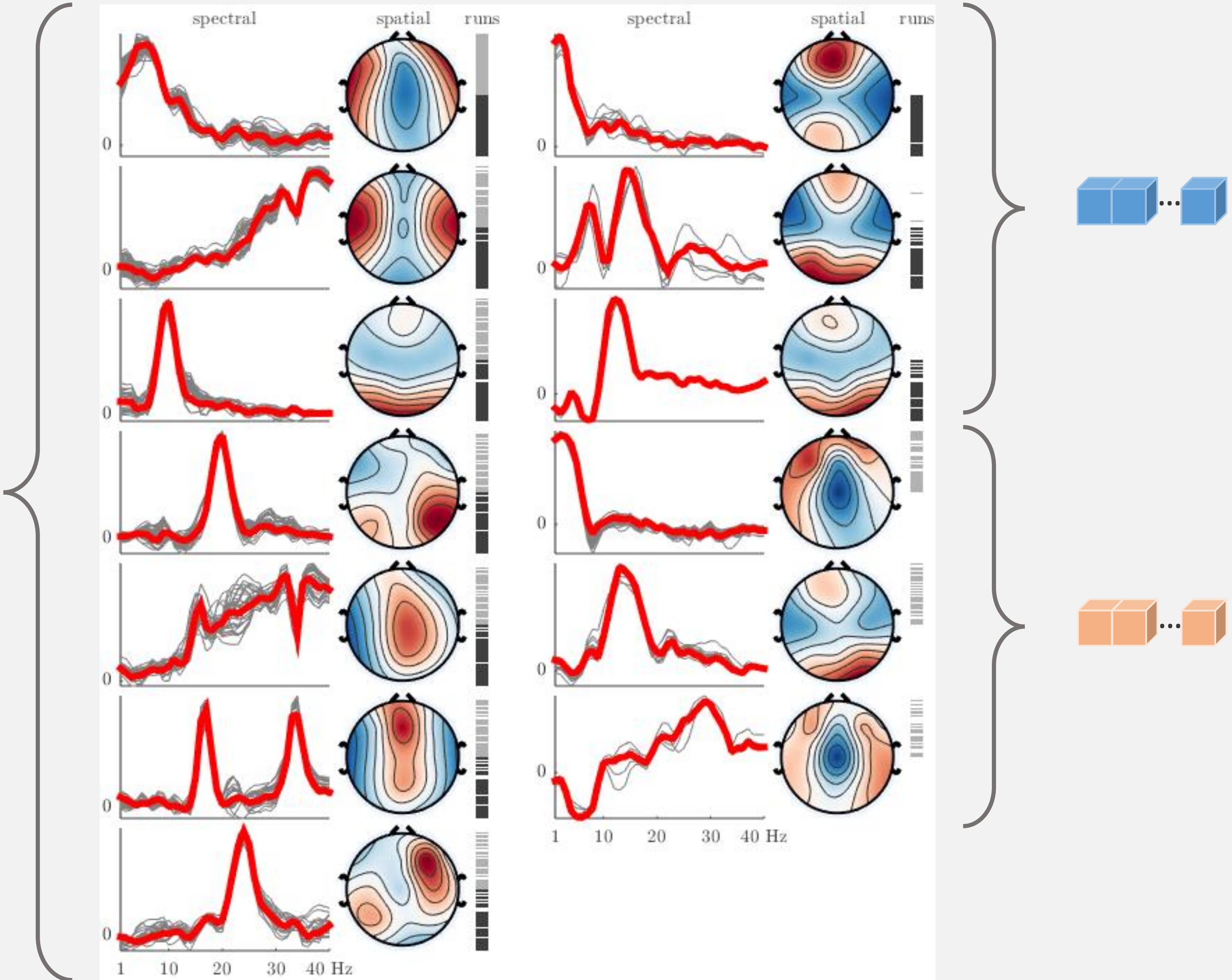
5



assignment matrix of components to clusters

5

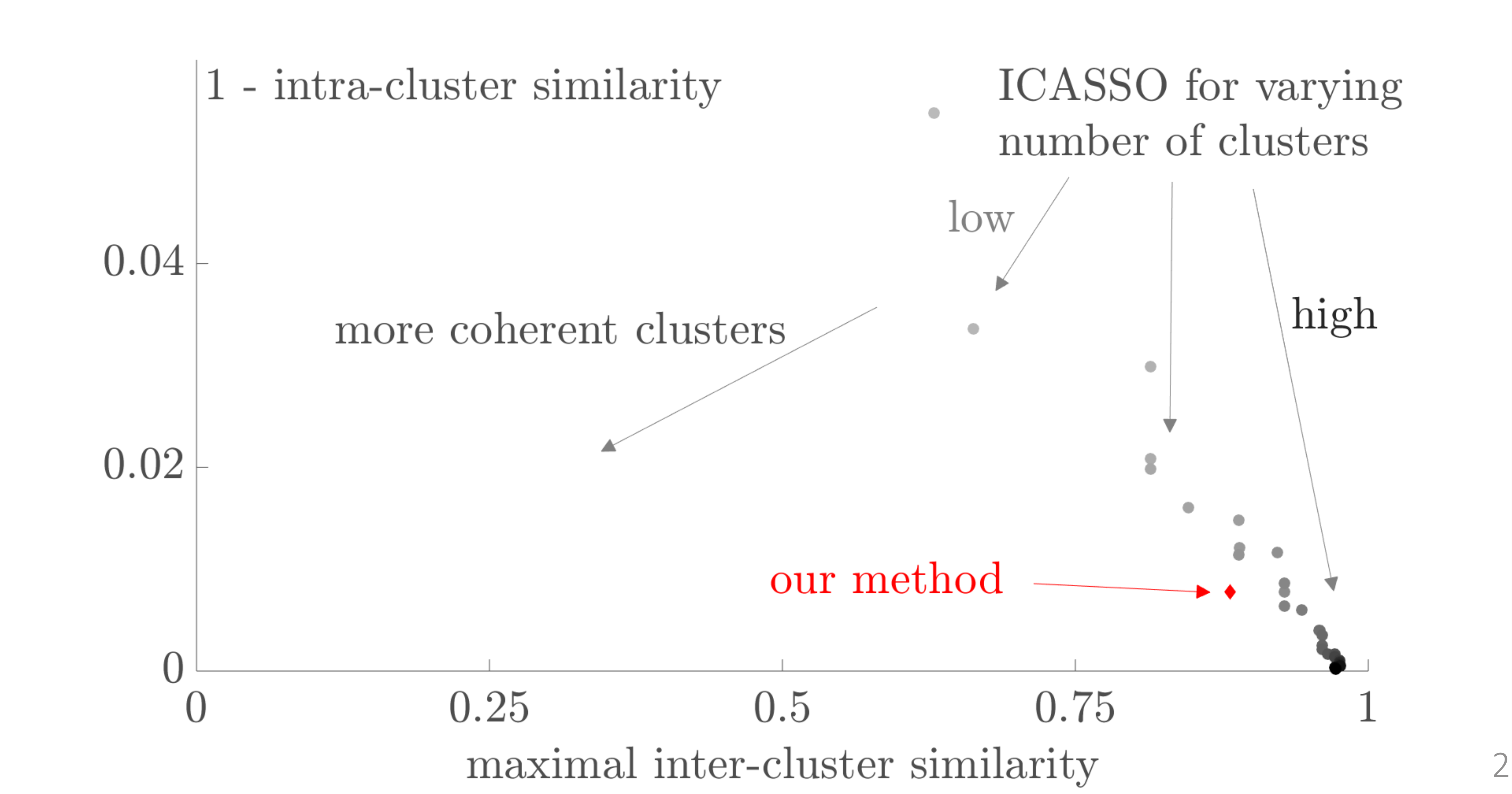
&



spectral
spatial
runs
spectral
spatial
runs
0
0
0
0
0
0
0
0
0
0
0
0
0
1
10
20
30
40 Hz
1
10
20
30
40 Hz

CPD of EEG data: cluster

5

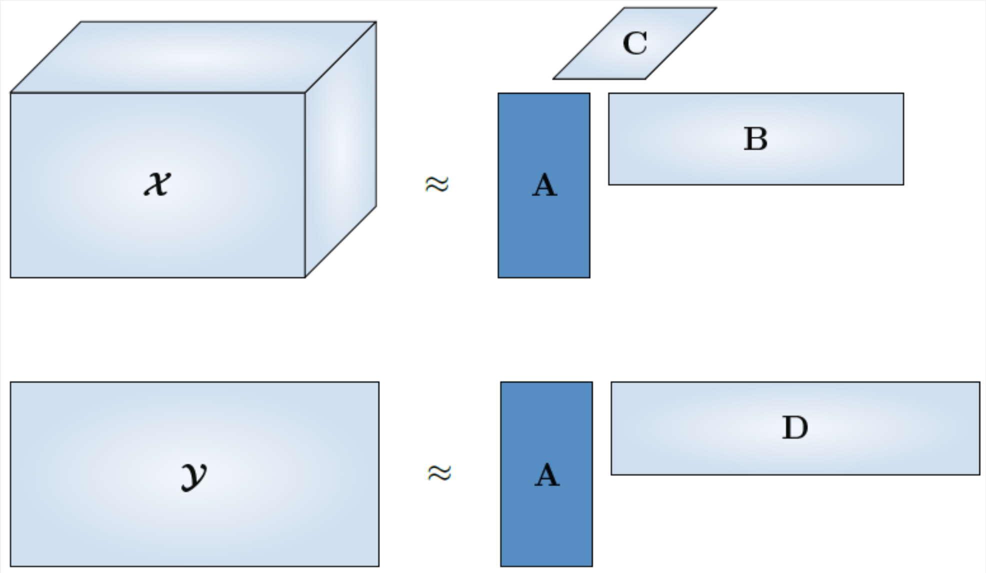


MapSync4BSS: experiment 2

Coupled matrix-tensor factorization (CMTF)

CMTF of synthetic data

We coupled a non-negative tensor with a matrix, both of rank 7.

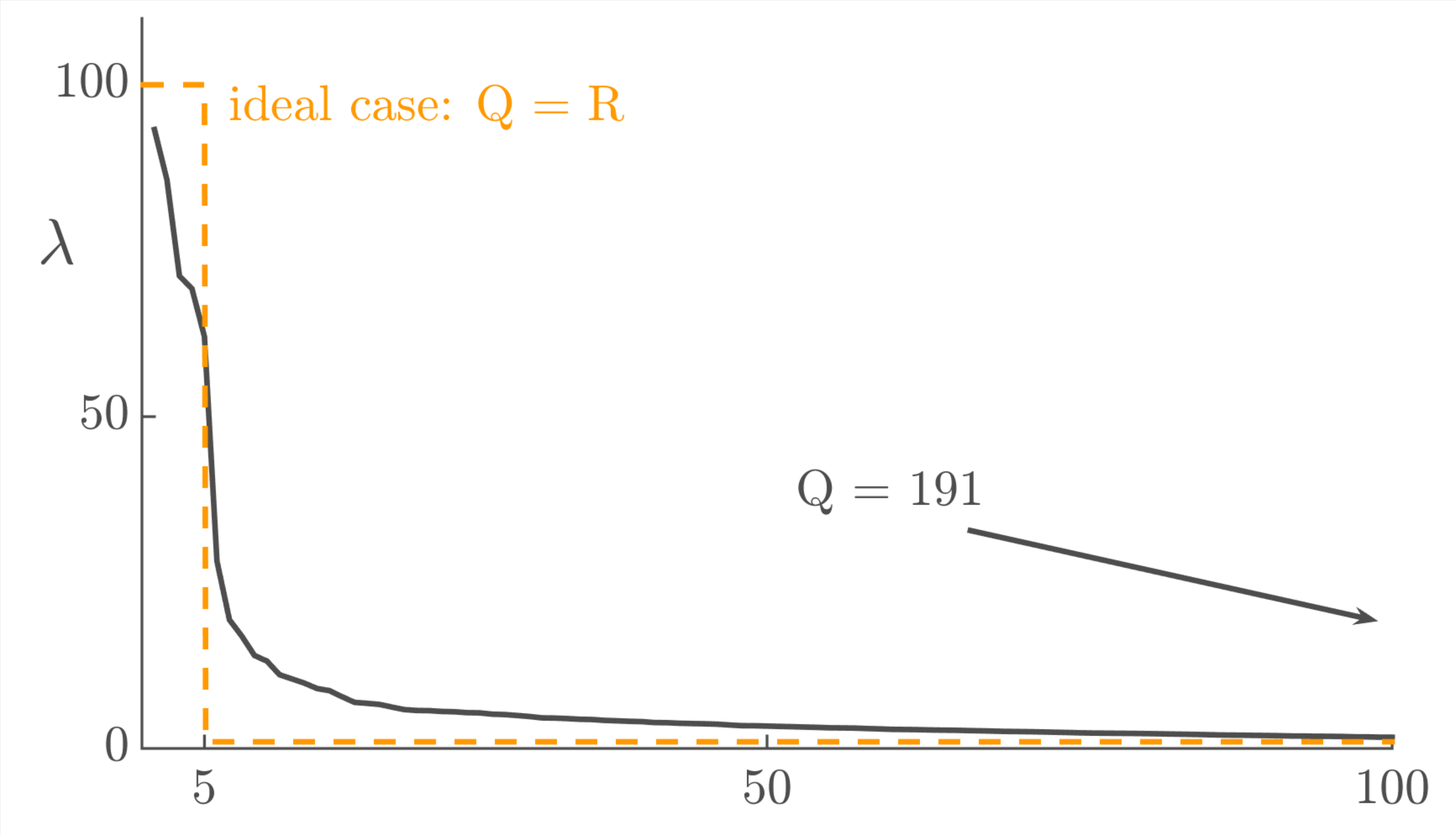


We compute 100× a CMTF with 5 components (underfitting the data),

Q: How stable is the coupled matrix-tensor factorization?

CMTF of synthetic data: variability

2

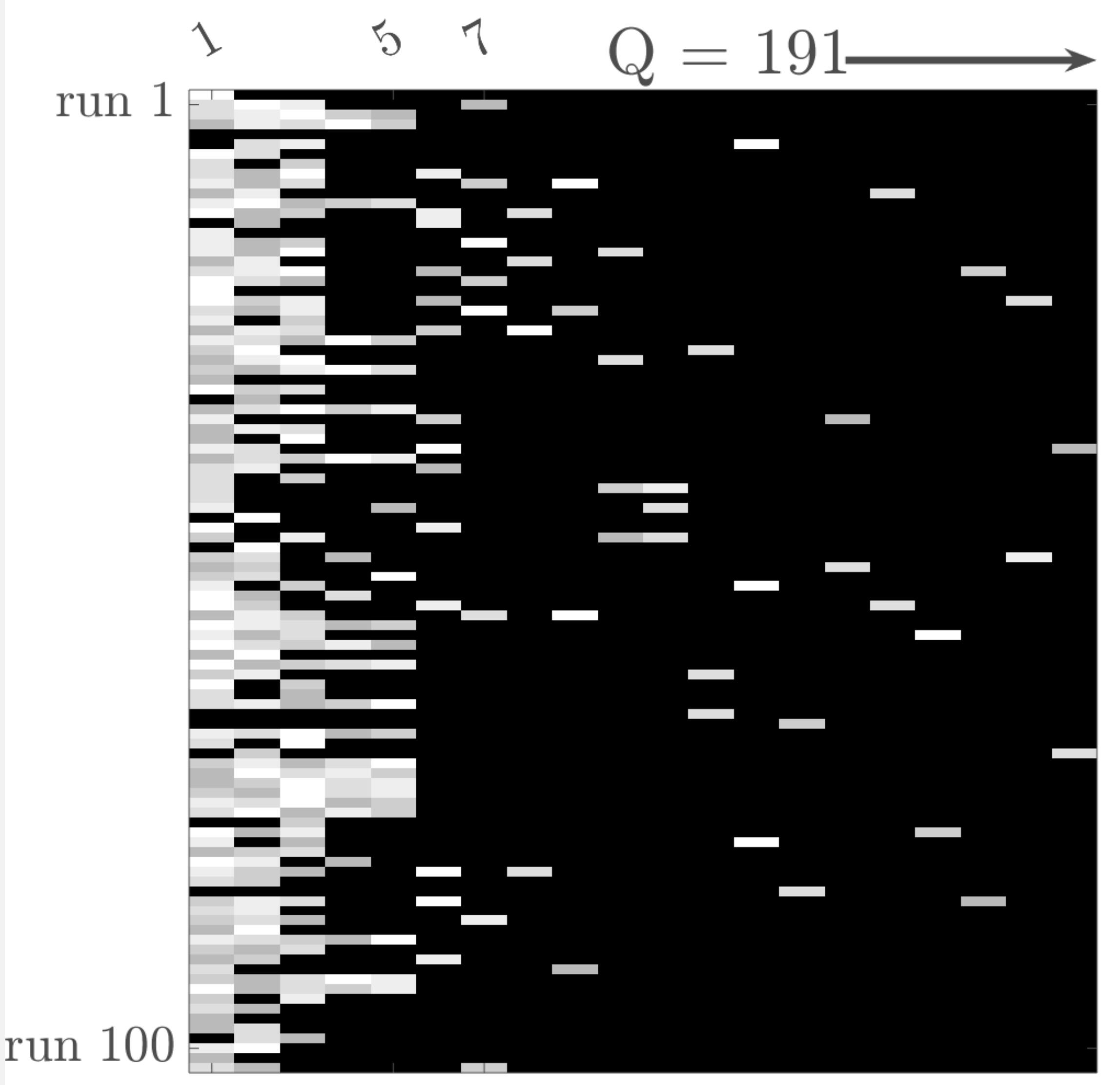


CMTF of synthetic data: denoise / cluster

4

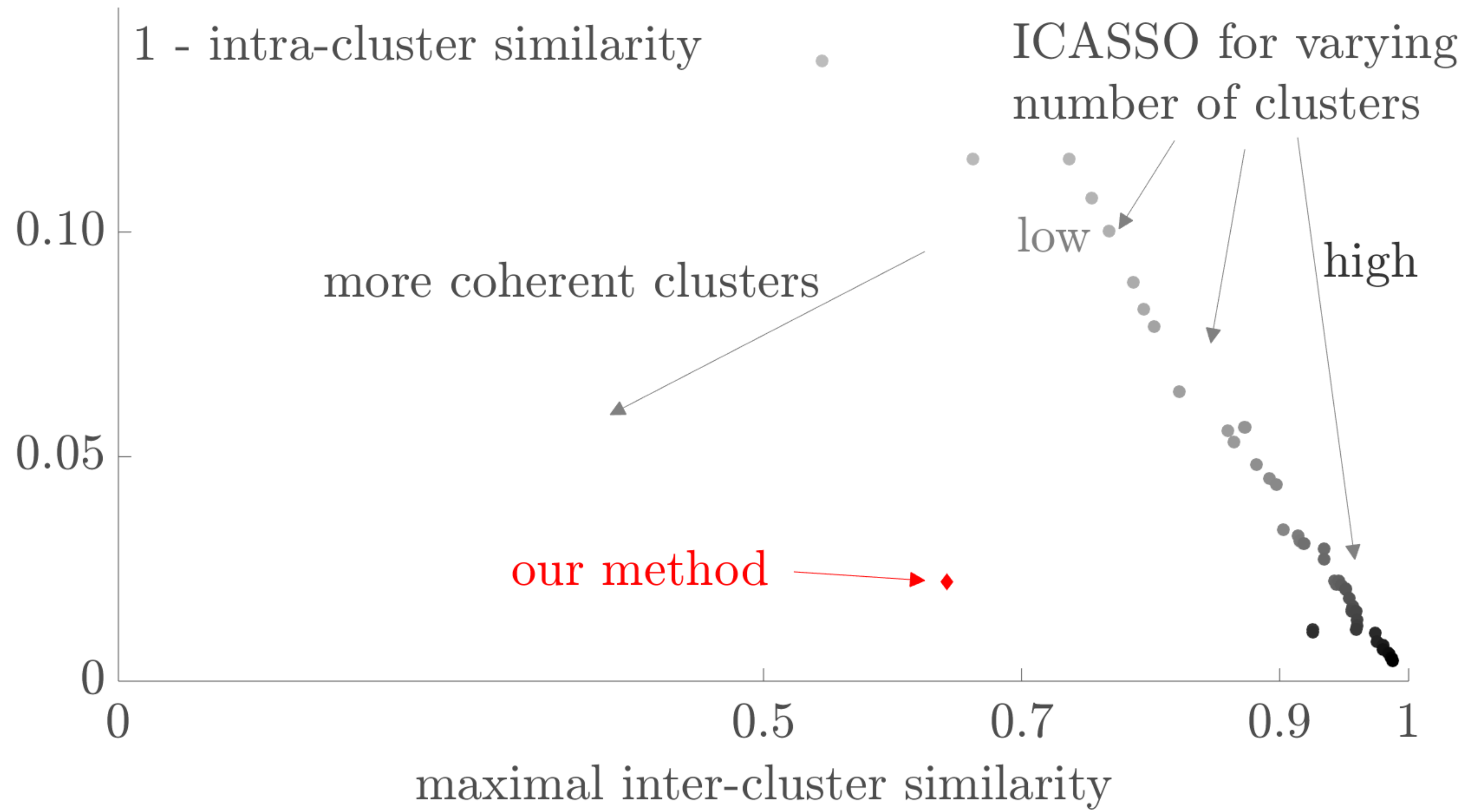
+

5



CMTF of synthetic data: cluster

5



MapSync4BSS is a diagnostic tool for practitioners

accurate

principled

intuitive

generic

easier than ICASSO

PS: Interested in the MATLAB implementation?

✉ simon.vaneyndhoven@kuleuven.be ☺

Thank you!

PS: Interested in the MATLAB implementation?

✉ simon.vaneyndhoven@kuleuven.be ☺

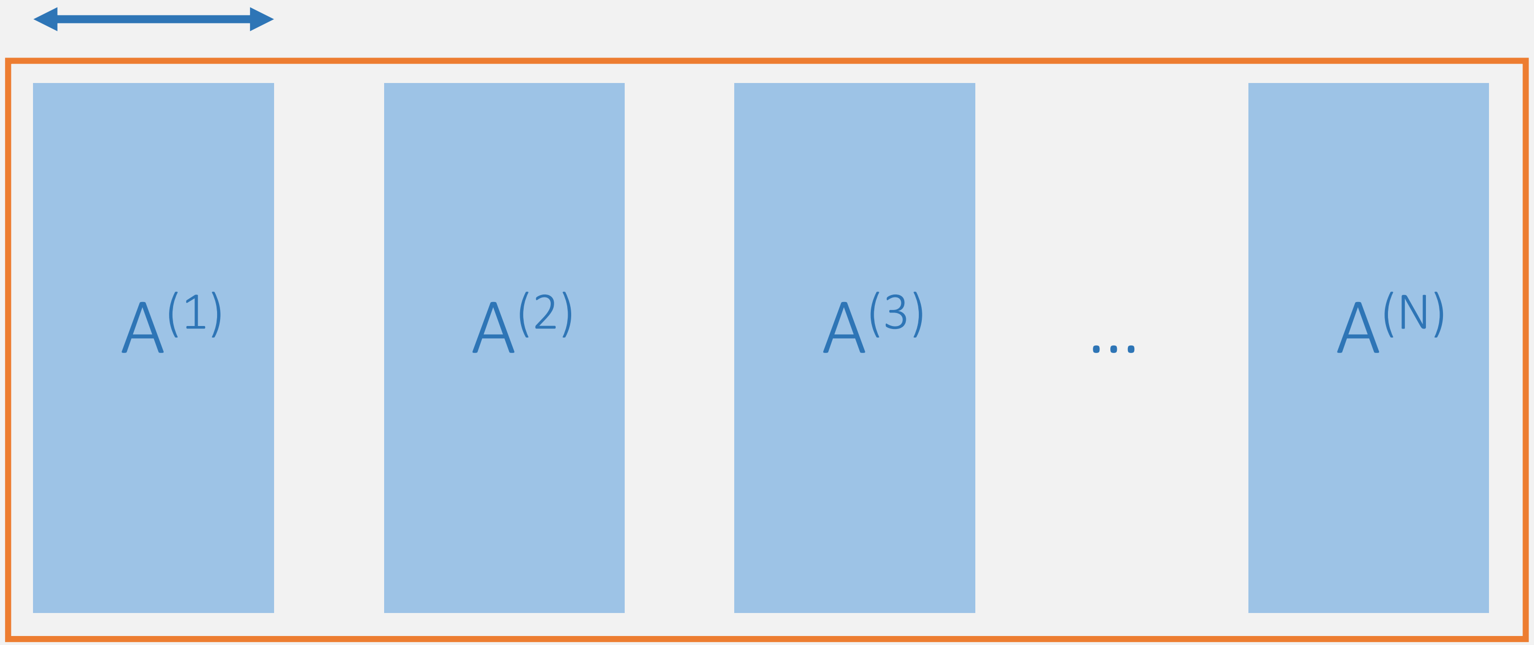
erc
European Research Council
Established by the European Commission

KU LEUVEN

Different view of cycle consistency

$\mathrm{rank}(A^{(i)}) = R$

recompute the CPD N times: $[A^{(i)},B^{(i)},C^{(i)}]$



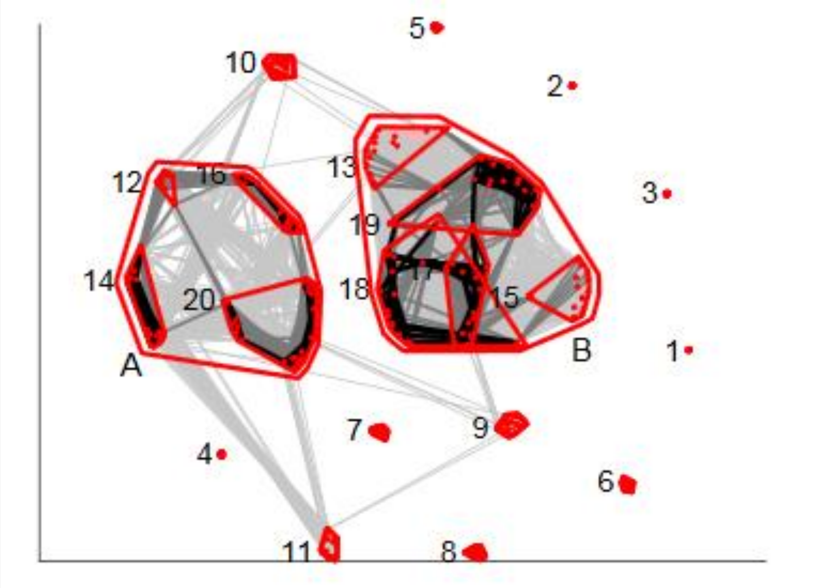
A

$\mathrm{rank}(A) = R$
iff factor $A^{(i)}$ is reproducible
(analogous for mode 2 and 3)

Existing work

ICASSO: SOFTWARE FOR INVESTIGATING THE RELIABILITY OF ICA ESTIMATES BY CLUSTERING AND VISUALIZATION

Johan Himberg[1] and Aapo Hyvärinen[1,2]
1) Neural Networks Research Centre
Helsinki Univ. of Technology, P.O. Box 5400, 02015 HUT, Finland
2) Helsinki Institute for Information Technology/BRU
Dept. of Computer Science, Univ. of Helsinki, Finland
johan.himberg@hut.fi, aapo.hyvarinen@helsinki.fi



N samples of k-dimensional vectors. The estimates of demixing matrices $\hat{\mathbf{W}}_i$ from each run $i = 1, 2, \ldots, M$ are collected into a single matrix $\hat{\mathbf{W}} = [\hat{\mathbf{W}}_1^T\ \hat{\mathbf{W}}_2^T \cdots \hat{\mathbf{W}}_M^T]^T$. If n_i independent components are estimated on each round, we get $K = \sum_i n_i$ estimates, and the size of $\hat{\mathbf{W}}$ will be $K \times k$.

We can resample independent component estimates by a) *Randomizing the initial condition*: FastICA is run M times for the same data $\mathbf{X}$, so that for each run the algorithm starts from a new random initial condition; b) *Bootstrapping*: FastICA is run M times. The initial condition is kept the same in every run, but the data is bootstrapped every time; and c) *Bootstrapping with randomized initial condition* as a combination of a) and b).

A natural measure of similarity between the estimated independent components is the absolute value of their mutual correlation coefficients r_{ij}, $i, j = 1, \ldots, K$. Straightforward calculations show that they can be obtained as elements of $\mathbf{R} = \hat{\mathbf{W}}\Sigma\hat{\mathbf{W}}^T$ where Σ is the covariance matrix for $\mathbf{X}$. The final similarity matrix has then elements

$$\sigma_{ij} = |r_{ij}|. \qquad (1)$$

similar algorithm found in ICA lore

extension to tensors not explicit

different parametrization

Selected references

1. Bajaj, Chandrajit, et al. "***SMAC: simultaneous mapping and clustering using spectral decompositions***." *International Conference on Machine Learning*. 2018.

2. Huang, Qi-Xing, and Leonidas Guibas. "***Consistent shape maps via semidefinite programming***." *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*. Eurographics Association, 2013.

3. Shen, Yanyao, et al. "***Normalized spectral map synchronization***." *Advances in Neural Information Processing Systems*. 2016.

4. Himberg, Johan, Aapo Hyvärinen, and Fabrizio Esposito. "***Validating the independent components of neuroimaging time series via clustering and visualization***." *Neuroimage* 22.3 (2004): 1214-1222.