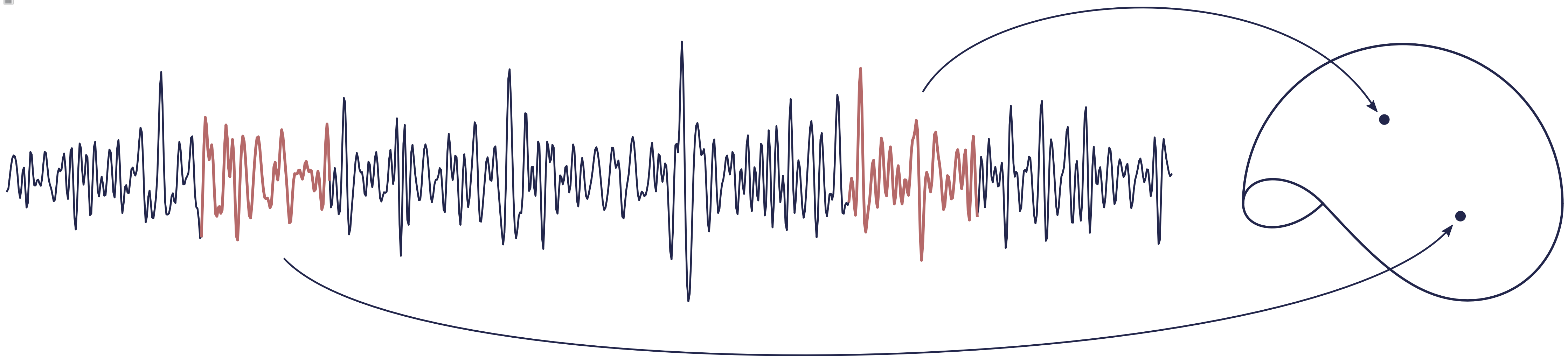
Riemannian Procrustres Analysis

Transfer Learning for Brain Computer Interfaces

Pedro L. C. Rodrigues   Marco Congedo   Christian Jutten



Presentation for the Machine Learning group at GIPSA-lab

22th May 2018, Grenoble, France

Outline

Introduction

Procrustes Analysis

Results

Concluding remarks

Outline

Introduction

Procrustes Analysis

Results

Concluding remarks

Classification of BCI signals always relies on an appropriate choice of **discriminant features**

N sensors
T samples
K trials

$$\{X_1, X_2, \ldots, X_K\}$$

$$X_i \in \mathbb{R}^{N \times T}$$

$$\{Y_1, Y_2, \ldots, Y_K\}$$

$$Y_i \in \mathbb{R}^d$$

d features

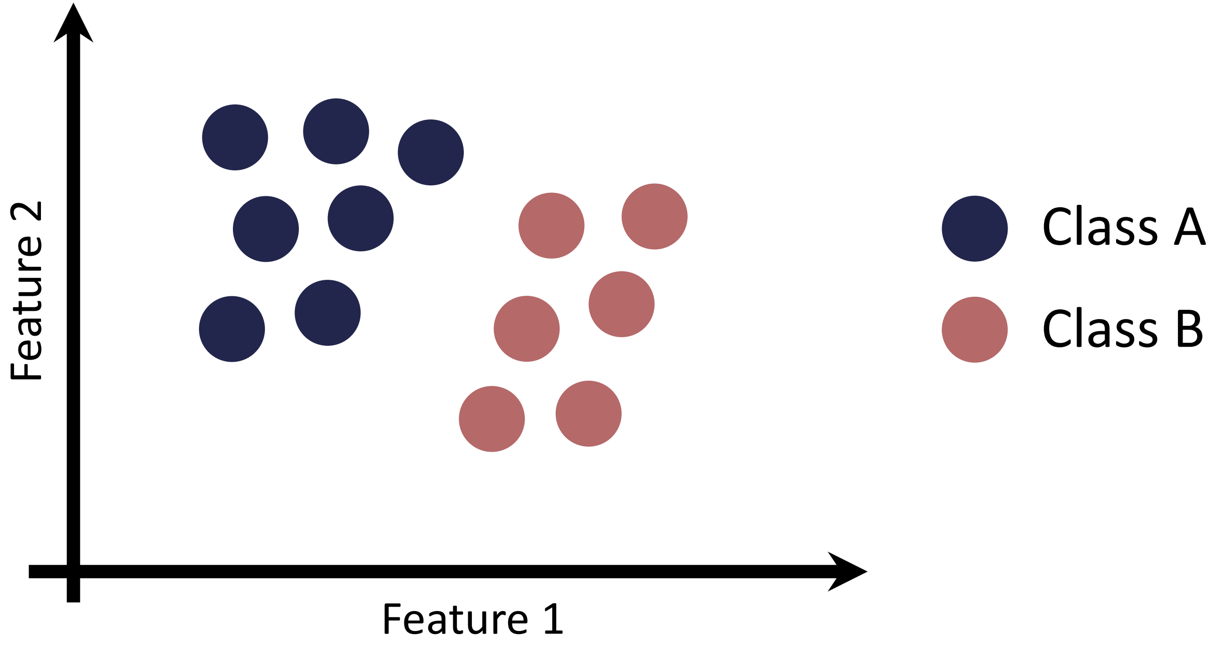
Classification of BCI signals always relies on an appropriate choice of **discriminant features**

N sensors
T samples
K trials

$$\{X_1, X_2, \dots, X_K\}$$
$$X_i \in \mathbb{R}^{N \times T}$$

$$\{Y_1, Y_2, \dots, Y_K\}$$
$$Y_i \in \mathbb{R}^d$$

d features



gipsa-lab

Classification of BCI signals always relies on an appropriate choice of **discriminant features**

N sensors
T samples
K trials

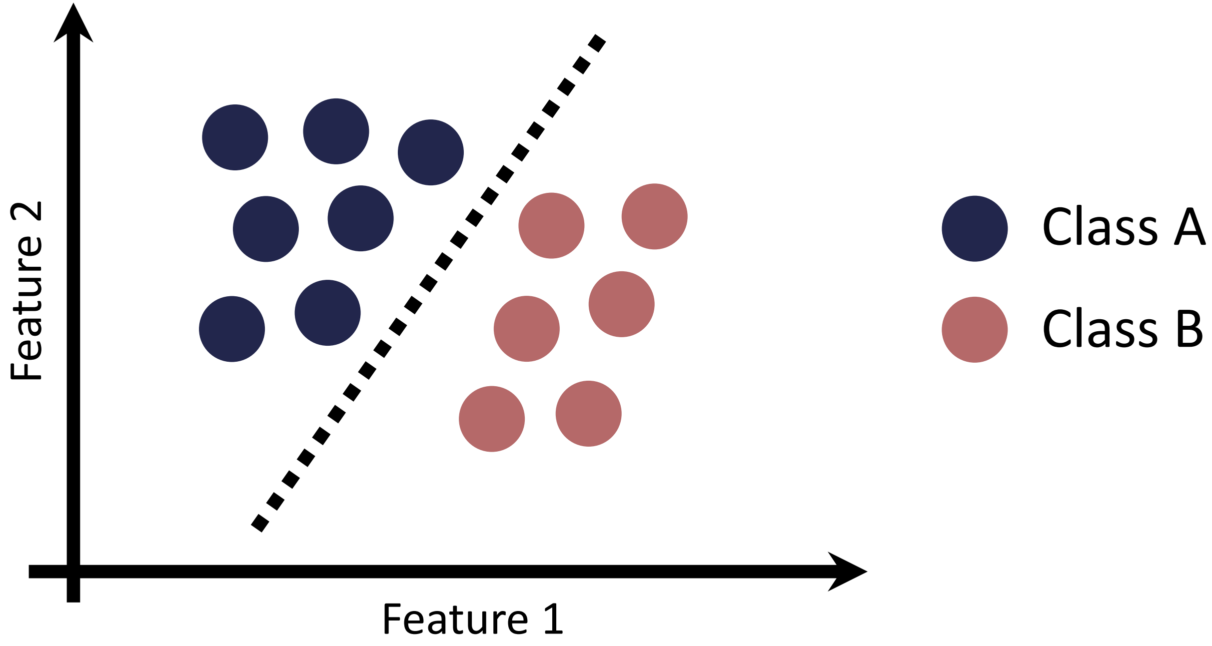
$$\{X_1, X_2, \ldots, X_K\}$$

$$X_i \in \mathbb{R}^{N \times T}$$

$$\{Y_1, Y_2, \ldots, Y_K\}$$

$$Y_i \in \mathbb{R}^{d}$$

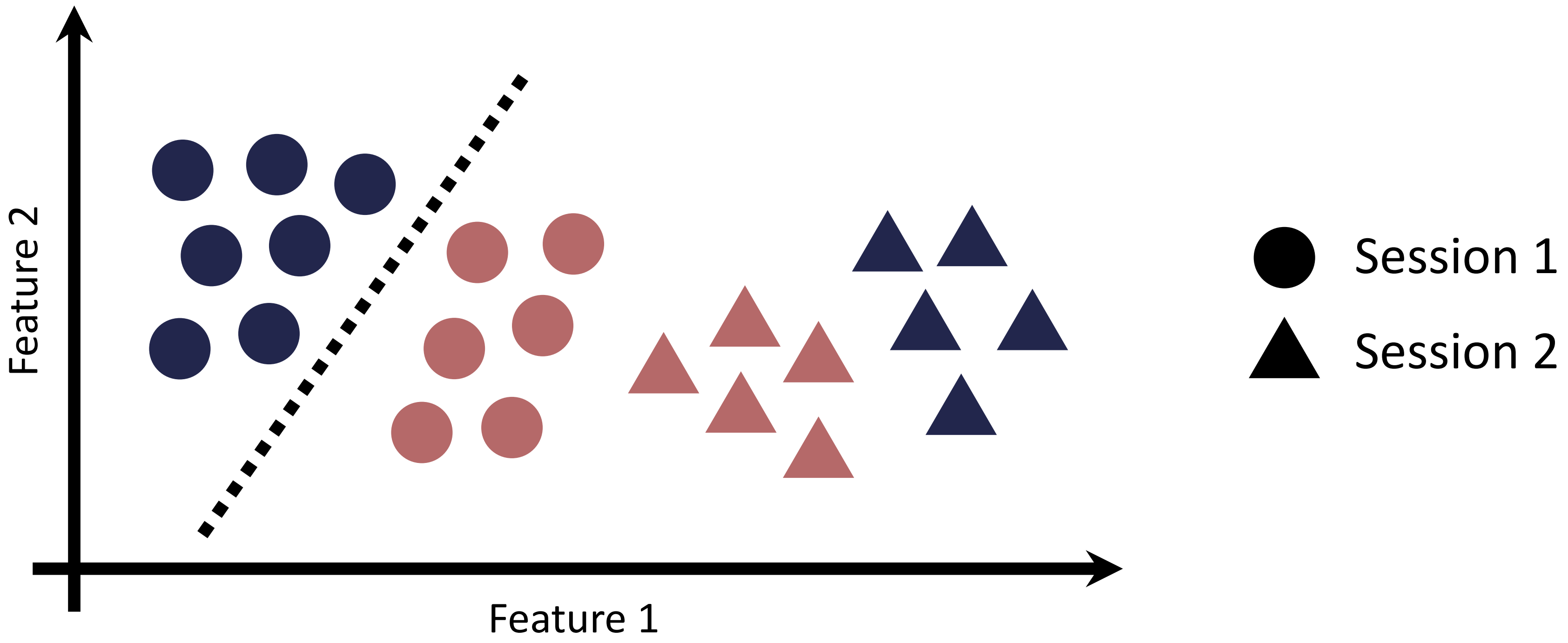
d features



gipsa-lab

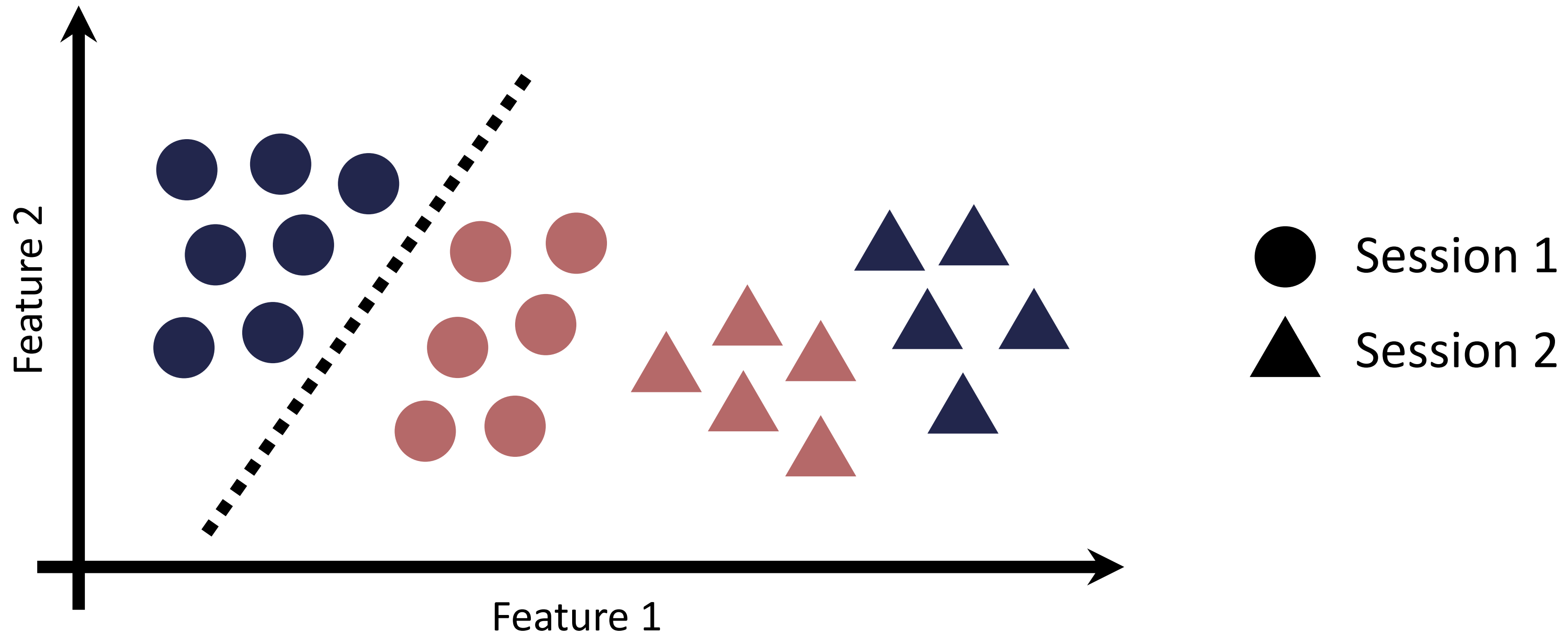
The **statistical distribution** of features may not be the same on different sessions

Covariate Shift



The **statistical distribution** of features may not be the same on different sessions

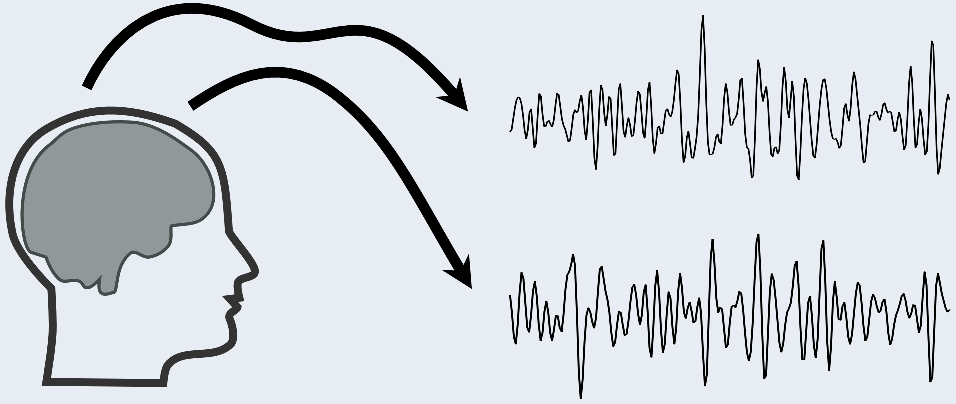
Covariate Shift



Transfer Learning aims at using the data from Session 1 to improve the classification of Session 2

gipsa-lab

The BCI paradigm – classification of EEG epochs



Cognitive task
(*K* trials)

EEG recording

$\{X_1, \ldots, X_K\}$

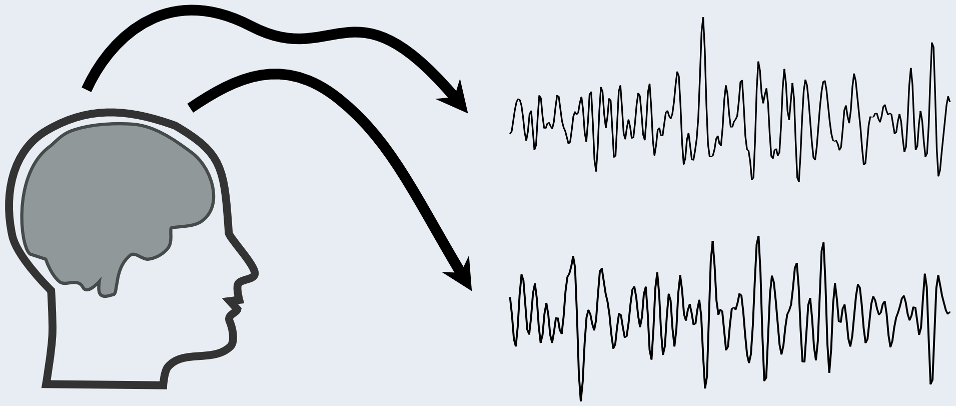
n – electrodes $X_k \in \mathbb{R}^{n \times T}$
T – samples

FEATURE ENGINEERING

Many features can be used to **describe** and **discriminate** the signals of each class

- Waveforms
- Spectra
- Neural connectivity
- Etc.

The BCI paradigm – classification of EEG epochs



Describes the instantaneous **correlations** between pairs of recorded signals

Cognitive task
(*K* trials)

EEG recording

$\{X_1, \dots, X_K\}$

n – electrodes $X_k \in \mathbb{R}^{n \times T}$
T – samples

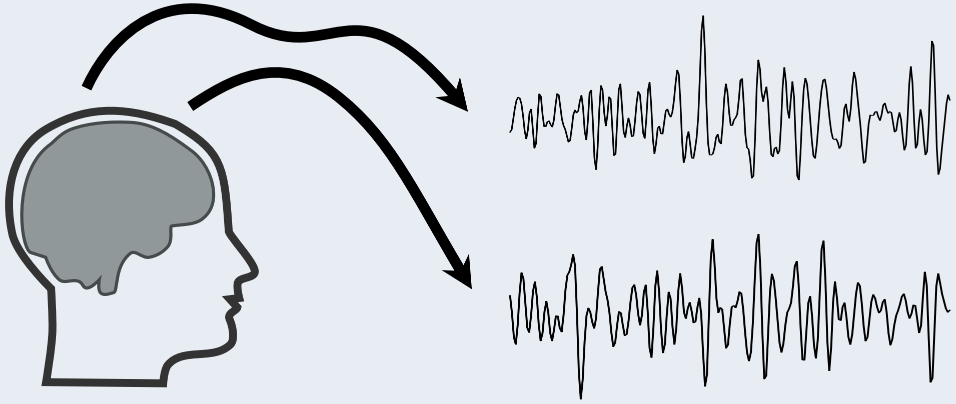
Covariance matrix

$\{C_1, \dots, C_K\}$

$C_k \in \mathbb{R}^{n \times n}$

$C_k = \frac{1}{T} X_k X_k^T$

The BCI paradigm – classification of EEG epochs



Cognitive task
(*K* trials)

EEG recording

$\{X_1, \ldots, X_K\}$

n – electrodes $X_k \in \mathbb{R}^{n \times T}$
T – samples

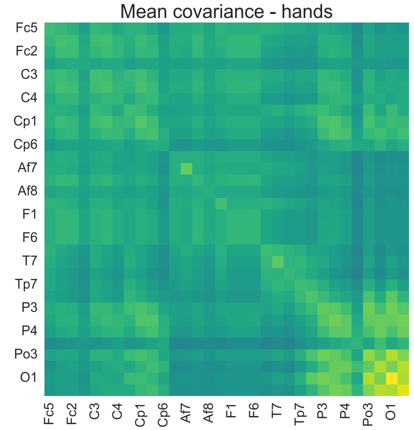
Covariance matrix

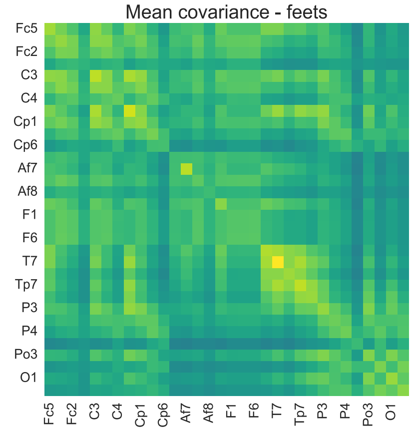
$\{C_1, \ldots, C_K\}$

$C_k \in \mathbb{R}^{n \times n}$

$C_k = \frac{1}{T} X_k X_k^T$

Classification





The Symmetric Positive Definite (SPD) manifold

Covariance matrix is **positive definite** → **Manifold** of positive definite matrices → Riemannian distance
Geometric mean
Tangent space

M is **positive definite** if, and only if,

$$\lambda \in \mathbb{R},\ \ \lambda_i > 0$$

(**positive** eigenvalues)

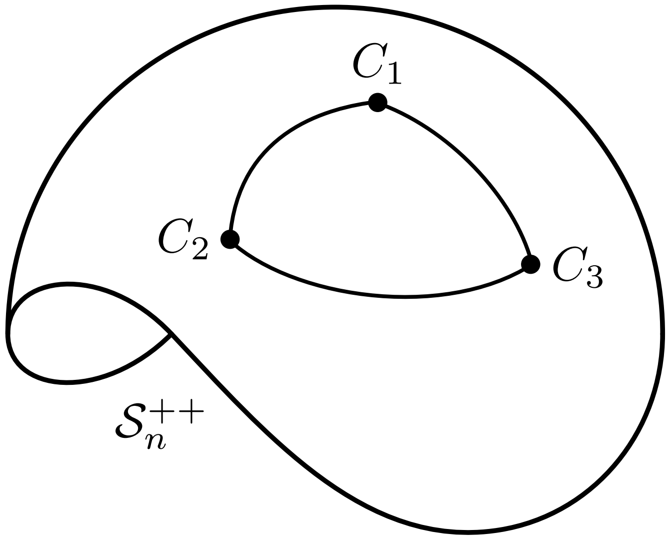
$$q_i^T q_j = \delta_{ij}$$

(**orthogonal** eigenvectors)

gipsa-lab

The Symmetric Positive Definite (SPD) manifold

Covariance matrix is **positive definite** → **Manifold** of positive definite matrices → Riemannian distance, Geometric mean, Tangent space



SPD manifold

The Symmetric Positive Definite (SPD) manifold

Covariance matrix
is **positive definite**

Manifold of positive
definite matrices

Riemannian distance
Geometric mean
Tangent space

gipsa-lab

The Symmetric Positive Definite (SPD) manifold

Covariance matrix
is **positive definite**

Manifold of positive
definite matrices

Riemannian distance
Geometric mean
Tangent space

Geodesic distances between
points on the globe

gipsa-lab

The Symmetric Positive Definite (SPD) manifold

Riemannian Geometry

Covariance matrix is **positive definite** → **Manifold** of positive definite matrices → Riemannian distance, Geometric mean, Tangent space

- Natural distance between SPD matrices

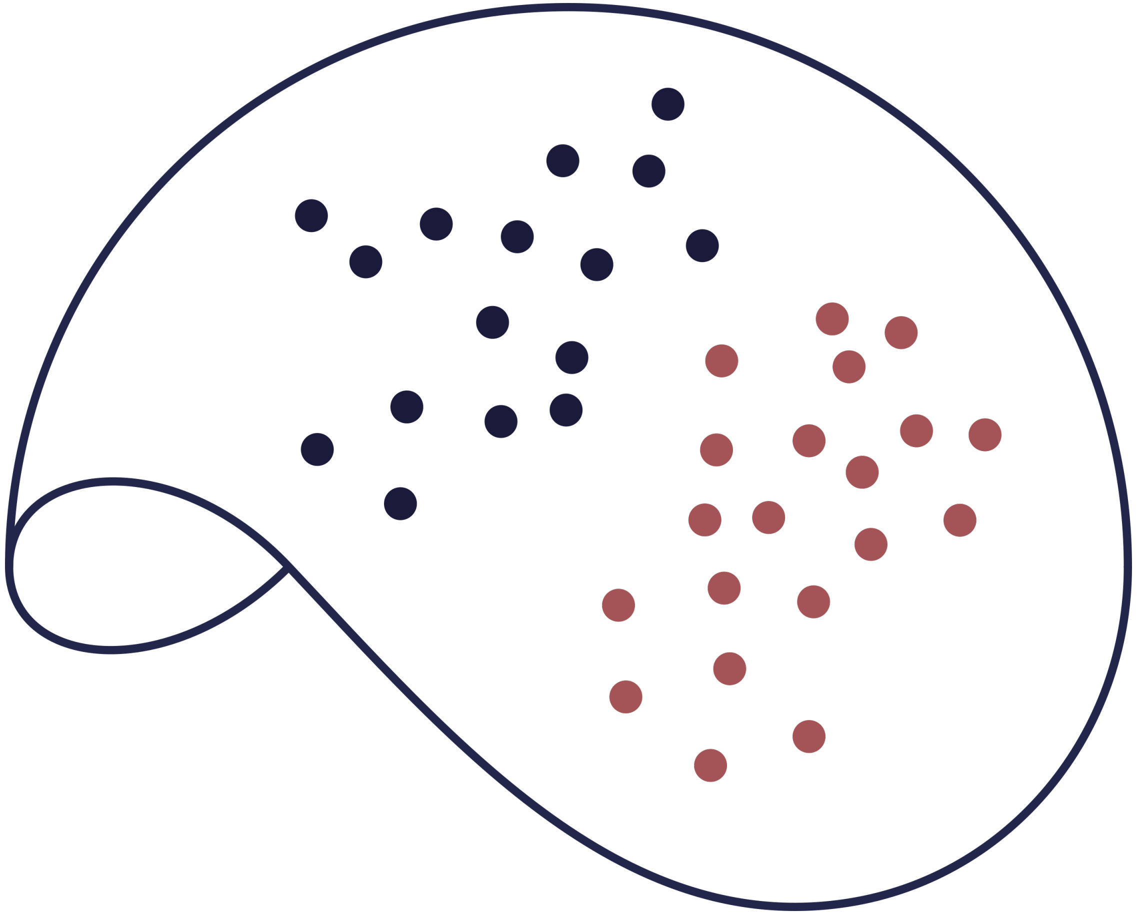
$$\delta_R^2(C_i, C_j) = \left\| \mathrm{Log}\left(C_i^{-1/2} C_j C_i^{-1/2}\right) \right\|_F^2 = \sum_{k=1}^{n} \log^2(\lambda_k)$$

- Center of mass of *K* SPD matrices

$$M = \underset{M \in \mathcal{S}_n^{++}}{\operatorname{argmin}} \sum_{k=1}^{K} \delta_R^2(M, C_k)$$

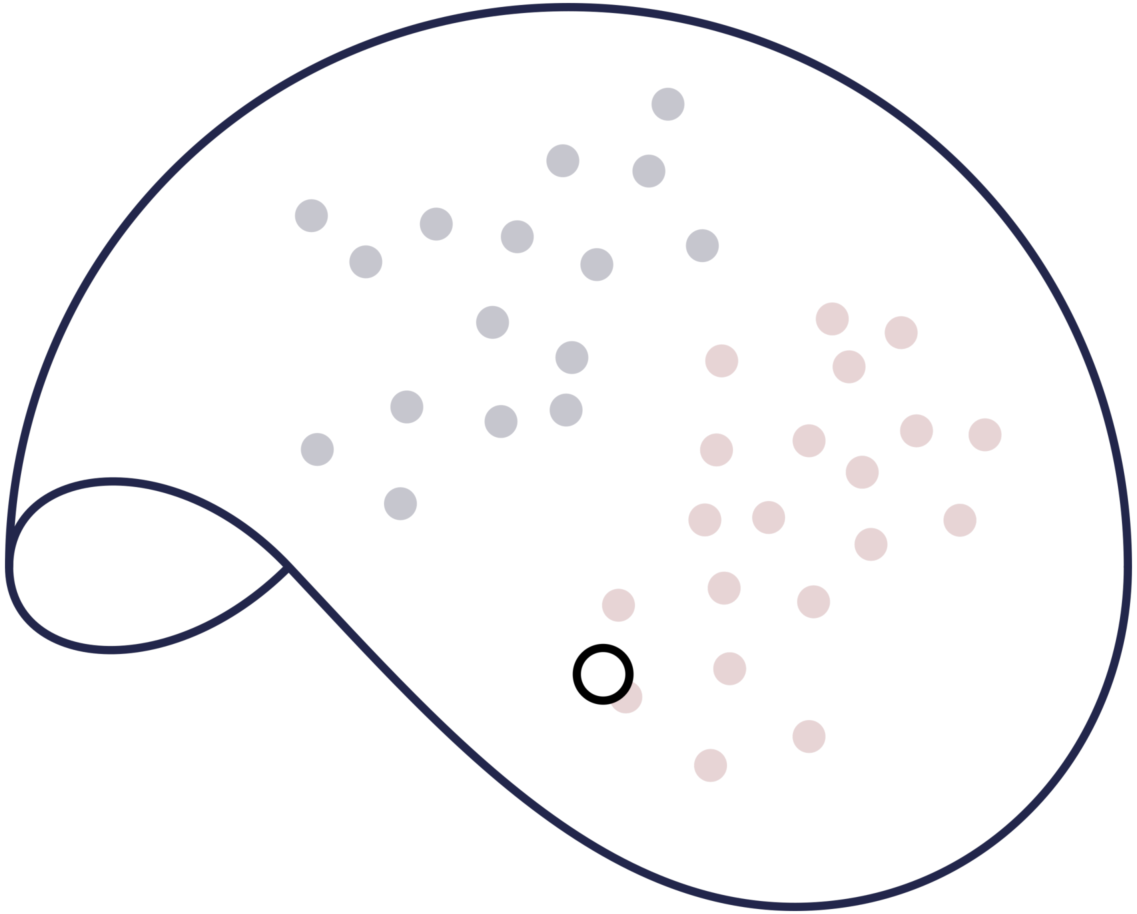
A very simple **classifier** using the geometry of the SPD manifold

$\{C_1^{(a)}, \dots, C_{K_a}^{(a)}\} \quad \{C_1^{(b)}, \dots, C_{K_b}^{(b)}\}$



A very simple **classifier** using the geometry of the SPD manifold

$\{C_1^{(a)}, \dots, C_{K_a}^{(a)}\} \quad \{C_1^{(b)}, \dots, C_{K_b}^{(b)}\}$

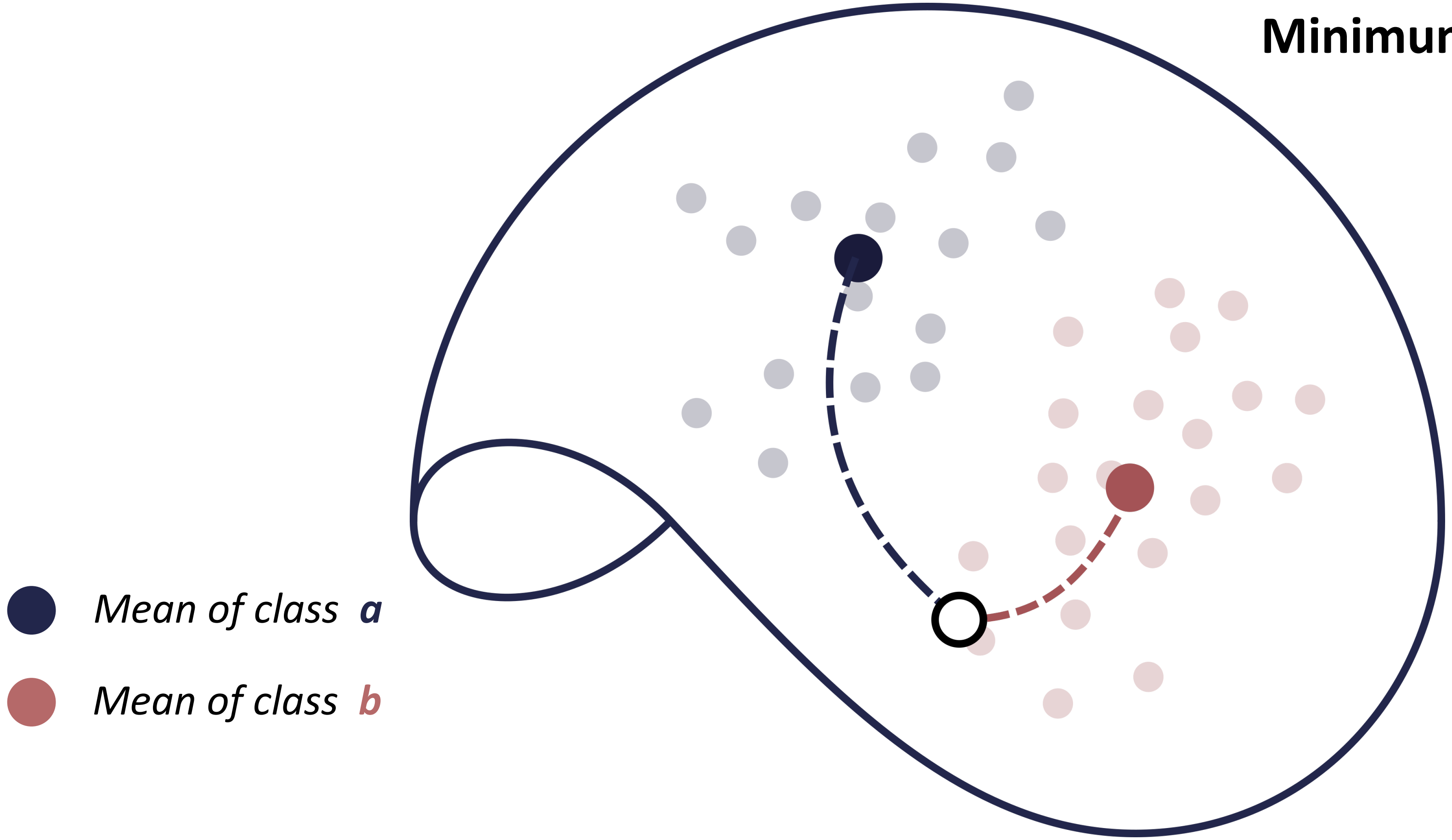


gipsa-lab

A very simple **classifier** using the geometry of the SPD manifold

$\{C_1^{(a)}, \ldots, C_{K_a}^{(a)}\}$ $\{C_1^{(b)}, \ldots, C_{K_b}^{(b)}\}$

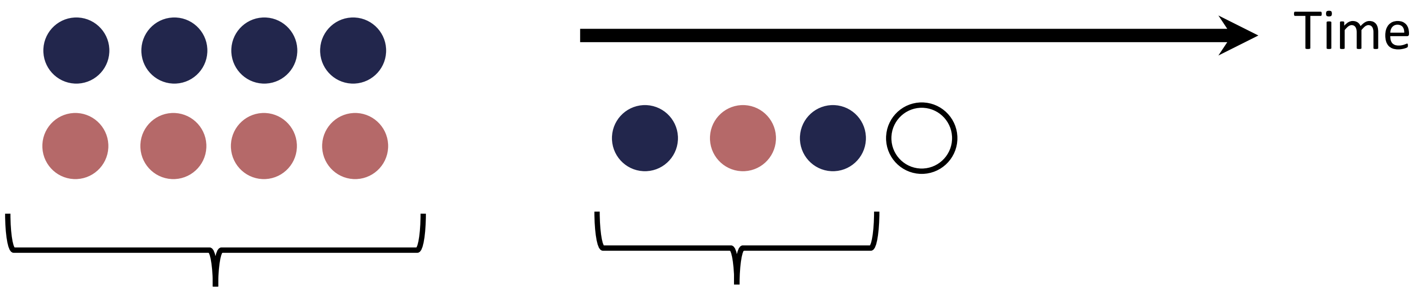
Classification based on the **Minimum Distance to Mean** (MDM)



Mean of class ***a***

Mean of class ***b***

Use data from the **source dataset** to classify the trials from the **target dataset**



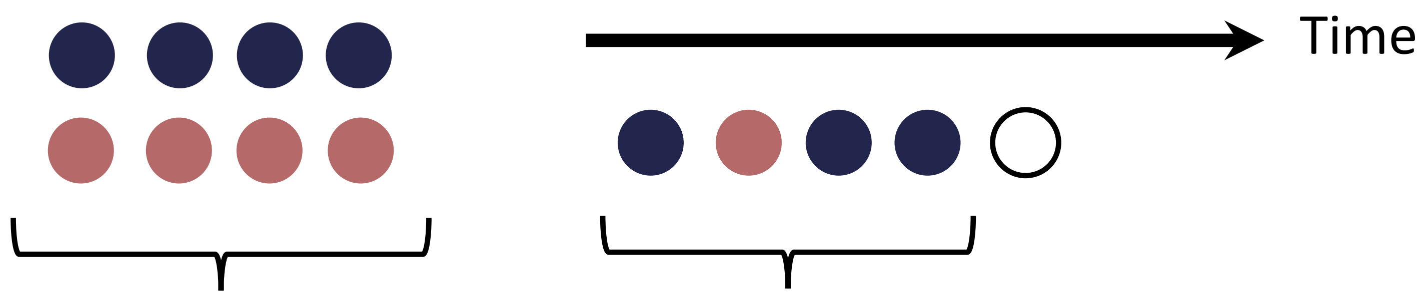
Source Dataset
(all labels known)

Target Dataset
(labels available sequentially)

We use **spatial covariance matrices** as discriminant features

Data points belong to a **Riemannian manifold**

Use data from the **source dataset** to classify the trials from the **target dataset**



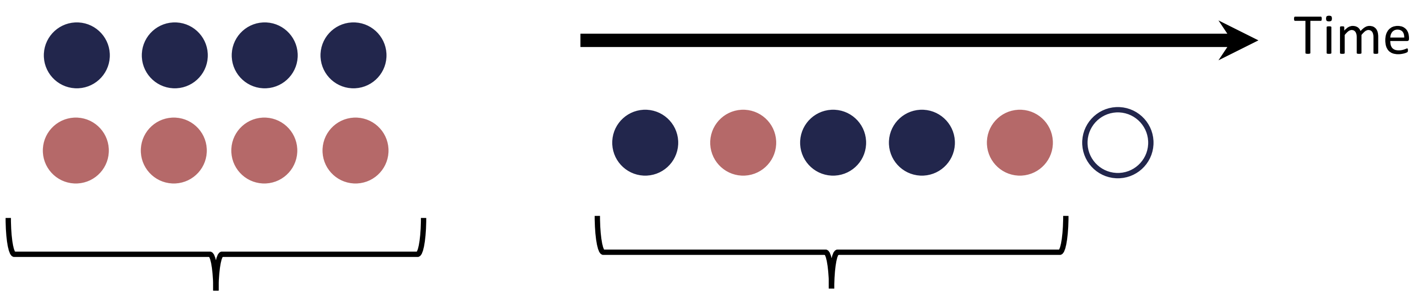
Source Dataset
(all labels known)

Target Dataset
(labels available sequentially)

We use **spatial covariance matrices** as discriminant features

Data points belong to a **Riemannian manifold**

Use data from the **source dataset** to classify the trials from the **target dataset**



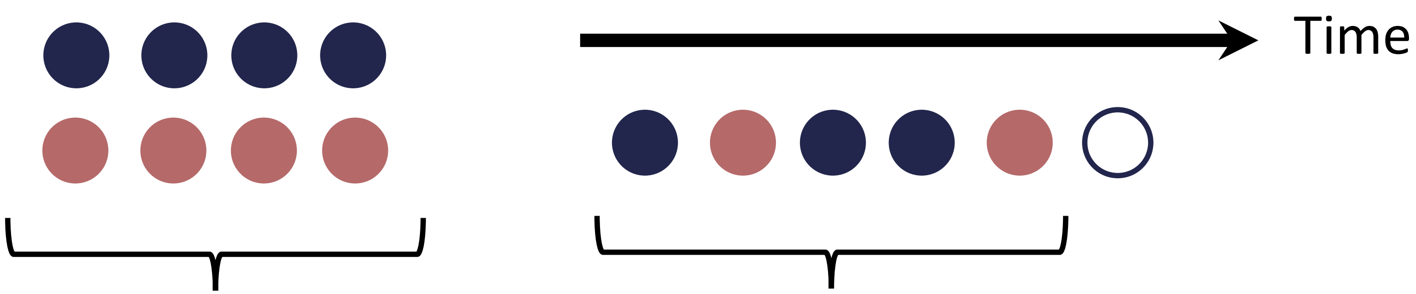
Source Dataset
(all labels known)

Target Dataset
(labels available sequentially)

We use **spatial covariance matrices** as discriminant features

Data points belong to a **Riemannian manifold**

Use data from the **source dataset** to classify the trials from the **target dataset**



Source Dataset
(all labels known)

Target Dataset
(labels available sequentially)

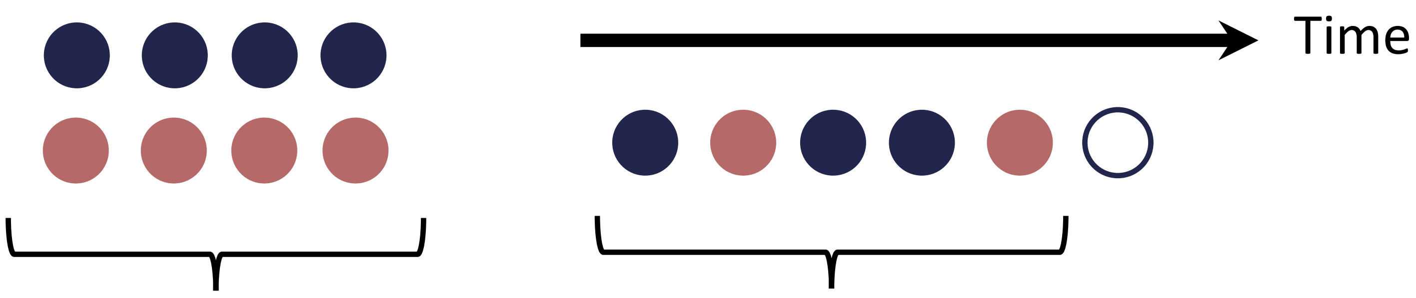
We use **spatial covariance matrices** as discriminant features

Data points belong to a **Riemannian manifold**

We do Transfer Learning via **distribution matching**

1) Fit classifier *C* to source dataset
2) Match target and source distributions
3) Classify unknown trial using *C*

Use data from the **source dataset** to classify the trials from the **target dataset**



Source Dataset
(all labels known)

Target Dataset
(labels available sequentially)

We use **spatial covariance matrices** as discriminant features

Data points belong to a **Riemannian manifold**

We do Transfer Learning via **distribution matching**

1) Fit classifier *C* to source dataset
2) Match target and source distributions
3) Classify unknown trial using *C*

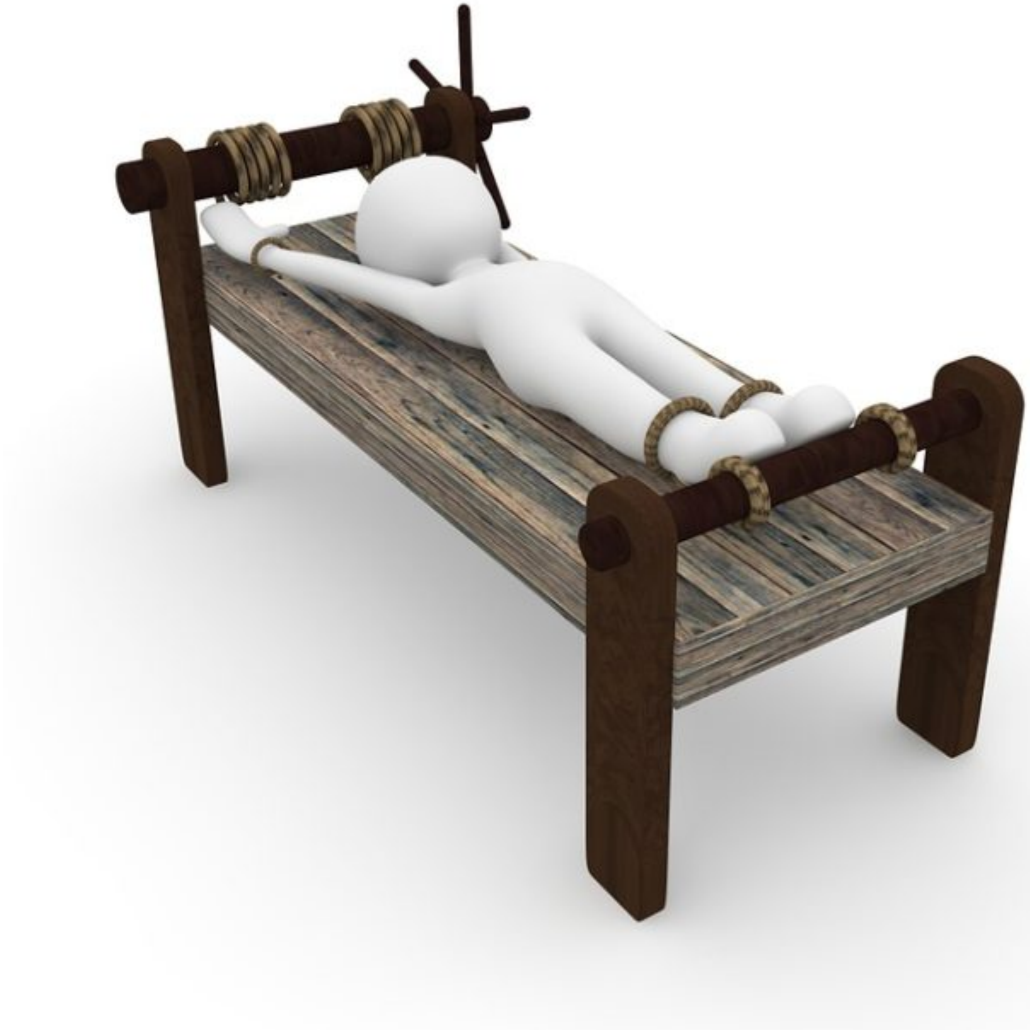
Match statistical distributions by applying geometrical **transformations** to the data points !

gipsa-lab

Outline

The **Procrustes bed** appears in Greek mythology

(The word *Procrustean* is used to describe situations where different lengths or sizes or properties are **fitted** to a **standard**)

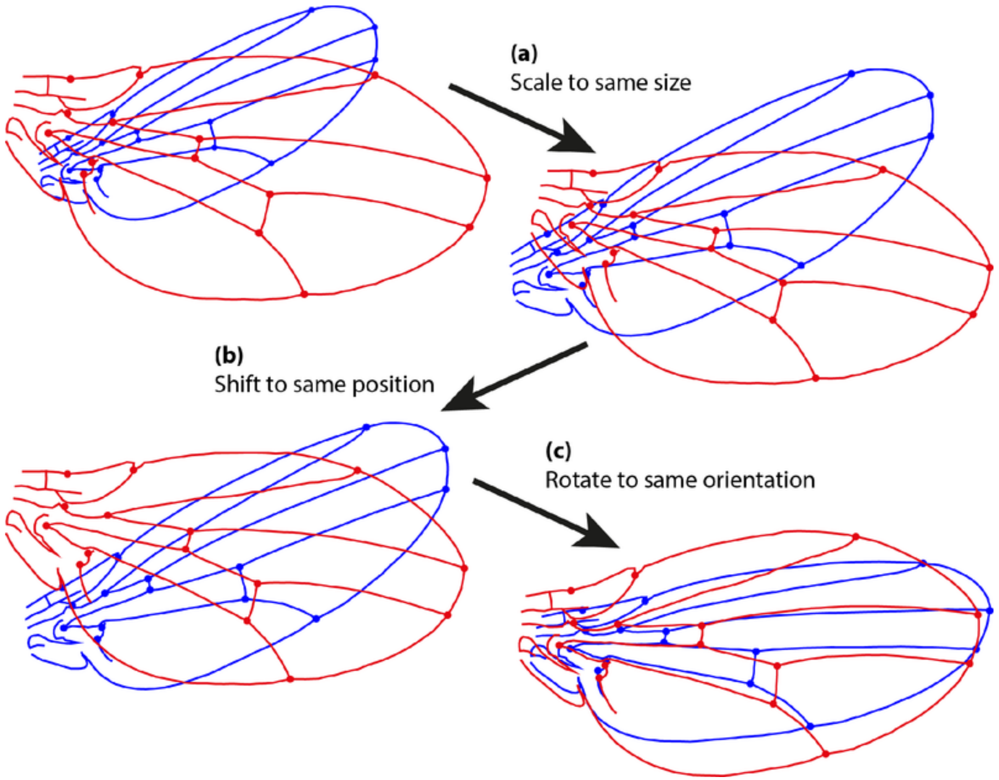


The **Procrustes bed** appears in Greek mythology

(The word *Procrustean* is used to describe situations where different lengths or sizes or properties are **fitted** to a **standard**)

Procrustes **analysis** is commonly used in statistical shape analysis to and consists of matching shapes via

- Translation
- Stretching
- Rotation



gipsa-lab

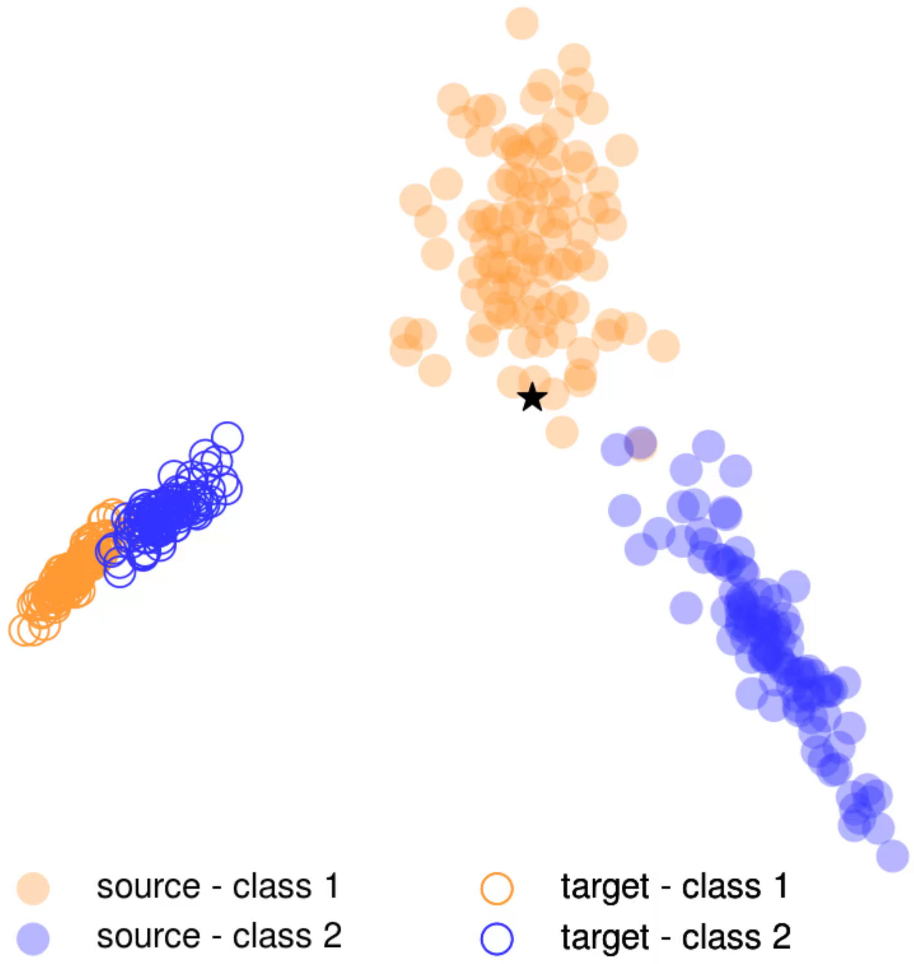
Each dot represents a **trial** in the dataset

The **Procrustes bed** appears in Greek mythology

(The word *Procrustean* is used to describe situations where different lengths or sizes or properties are **fitted** to a **standard**)

Procrustes **analysis** is commonly used in statistical shape analysis to and consists of matching shapes via

We propose a PA in the SPD manifold to **match** the **shape** of the statistical **distributions** of sessions

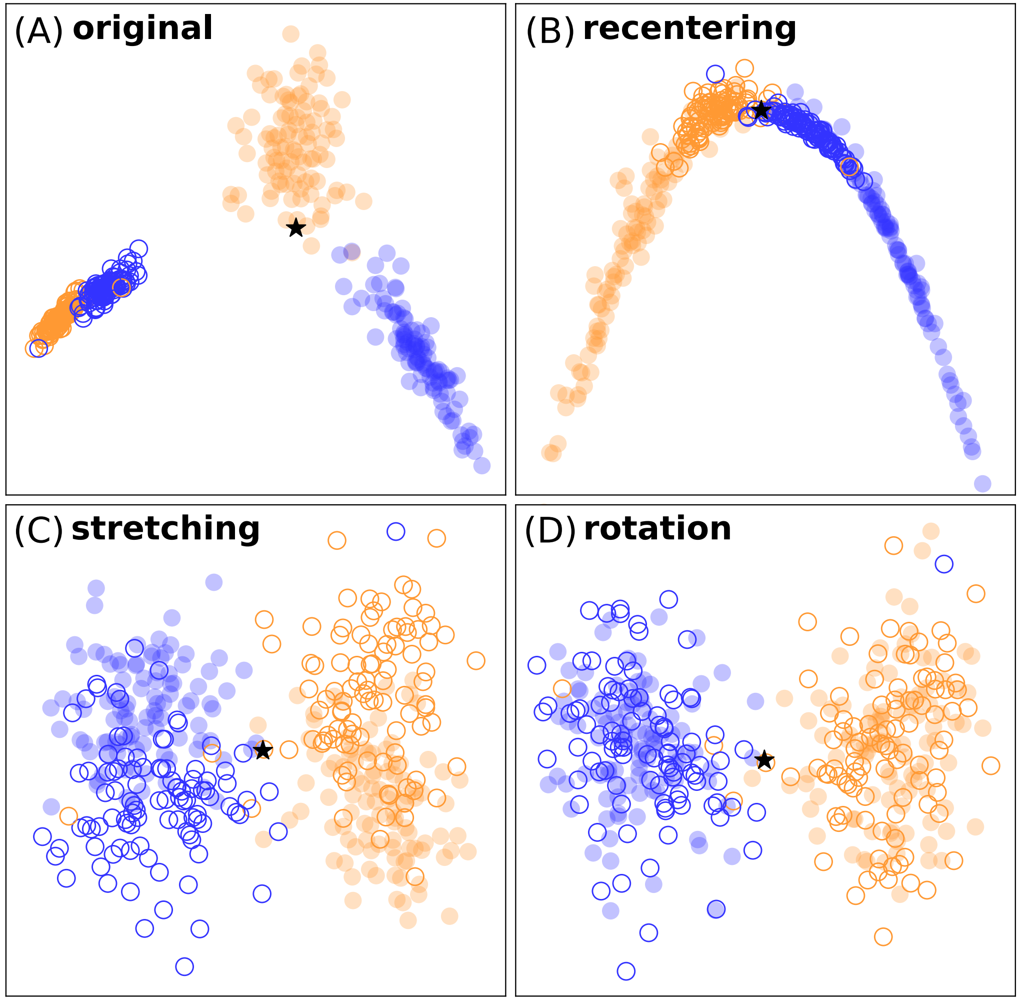


The **Procrustes bed** appears in Greek mythology

(The word *Procrustean* is used to describe situations where different lengths or sizes or properties are **fitted** to a **standard**)

Procrustes **analysis** is commonly used in statistical shape analysis to and consists of matching shapes via

We propose a PA in the SPD manifold to **match** the **shape** of the statistical **distributions** of sessions



The transformations have to respect the **intrinsic geometry** of the SPD manifold

Suppose we have a set of *K* matrices $\{C_1, \ldots, C_K\}$ and their geometric mean is M

RECENTERING

$$C_i^{(\mathrm{rct})} = M^{-1/2} C_i M^{-1/2}$$

STRETCHING

$$C_i^{(\mathrm{str})} = \left(C_i^{(\mathrm{rct})}\right)^p$$

ROTATION

$$C_i^{(\mathrm{rot})} = U C_i^{(\mathrm{str})} U^T$$ with $UU^T = \mathbf{I}$

Outline

Introduction

Procrustes Analysis

Results

Concluding remarks

Four datasets on three BCI paradigms

Publicly Available

Physionet EEG MI database	(109 subjects, 40 trials, 2 classes)
GigaDB EEG MI database	(37 subjects, 200 trials, 2 classes)
SSVEP Chevallier database	(12 subjects, 24 trials, 3 classes, 2 sessions)
P300 Brain Invaders database	(24 subjects, 576 trials, 2 classes)

For each dataset, Transfer Learning was always done in the **cross-subject** case

Cross-subject accuracy

Get the accuracies in classification when :

(Consider an **increasing** number of available points in the target dataset)

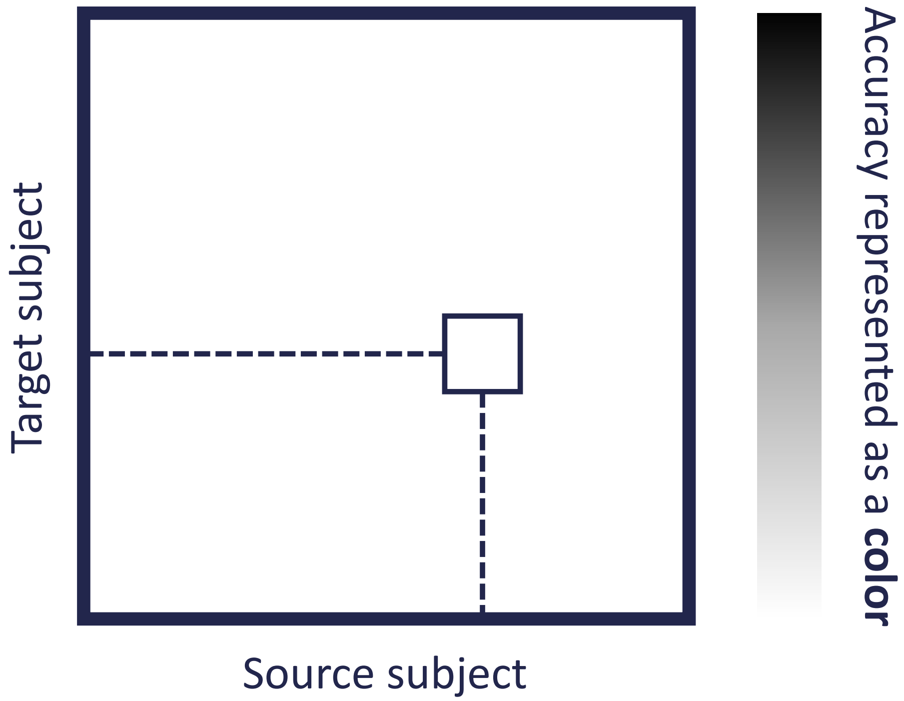
Compare the results for **three** methods :

ORIGINAL

RECENTERING

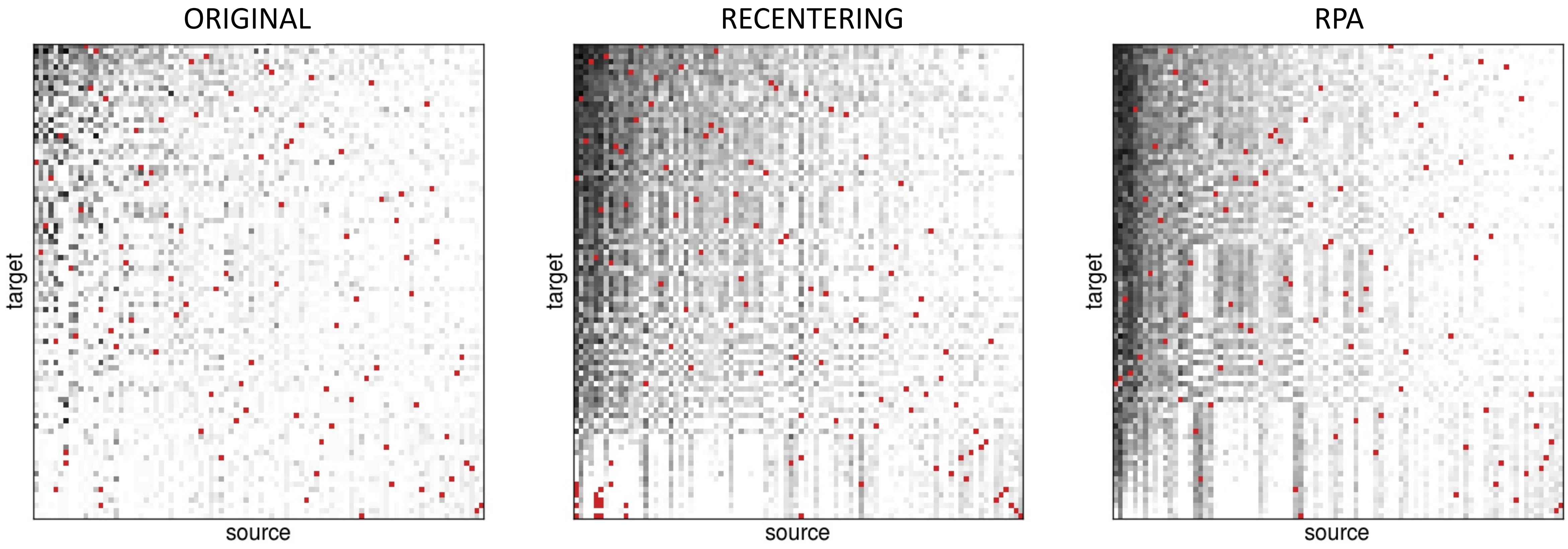
RPA

1) Classifier *C* is fitted to **source** subject dataset
2) Match target and source statistical distributions
3) Classify unknown trials from **target** dataset



Cross-subject accuracy – Physionet MI

\# of data points in the target dataset : **1**

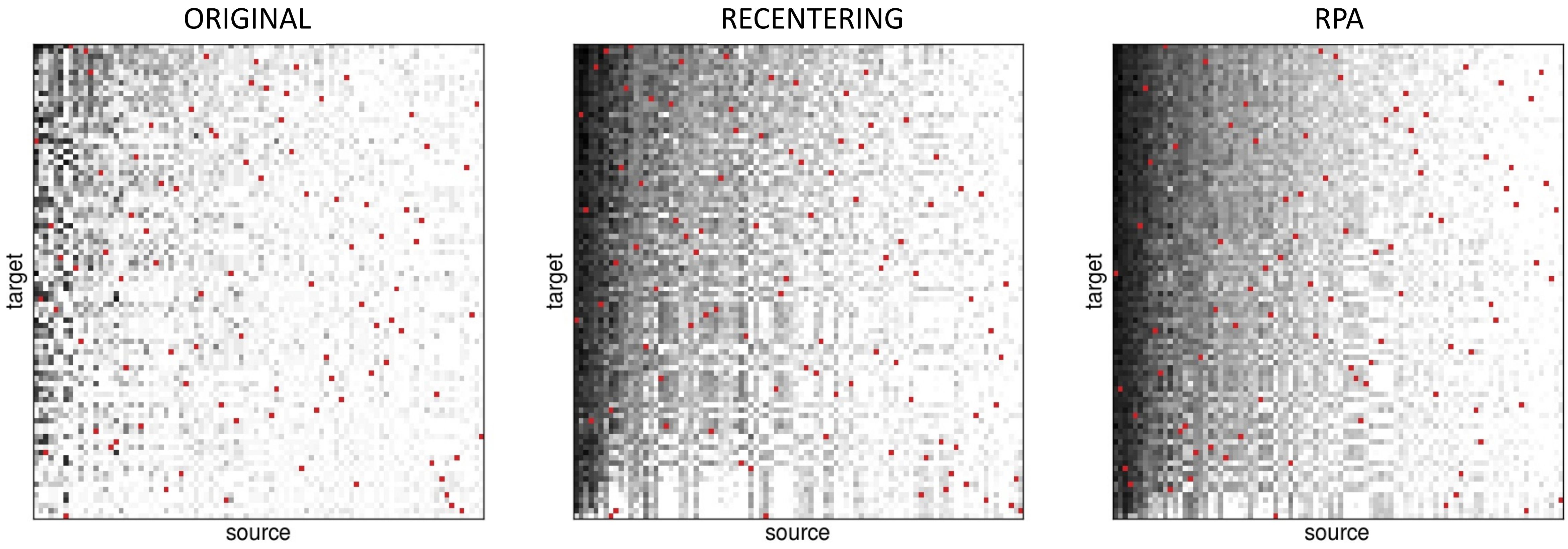


(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

Source subject = Target subjet

Cross-subject accuracy – Physionet MI

of data points in the target dataset : **5**



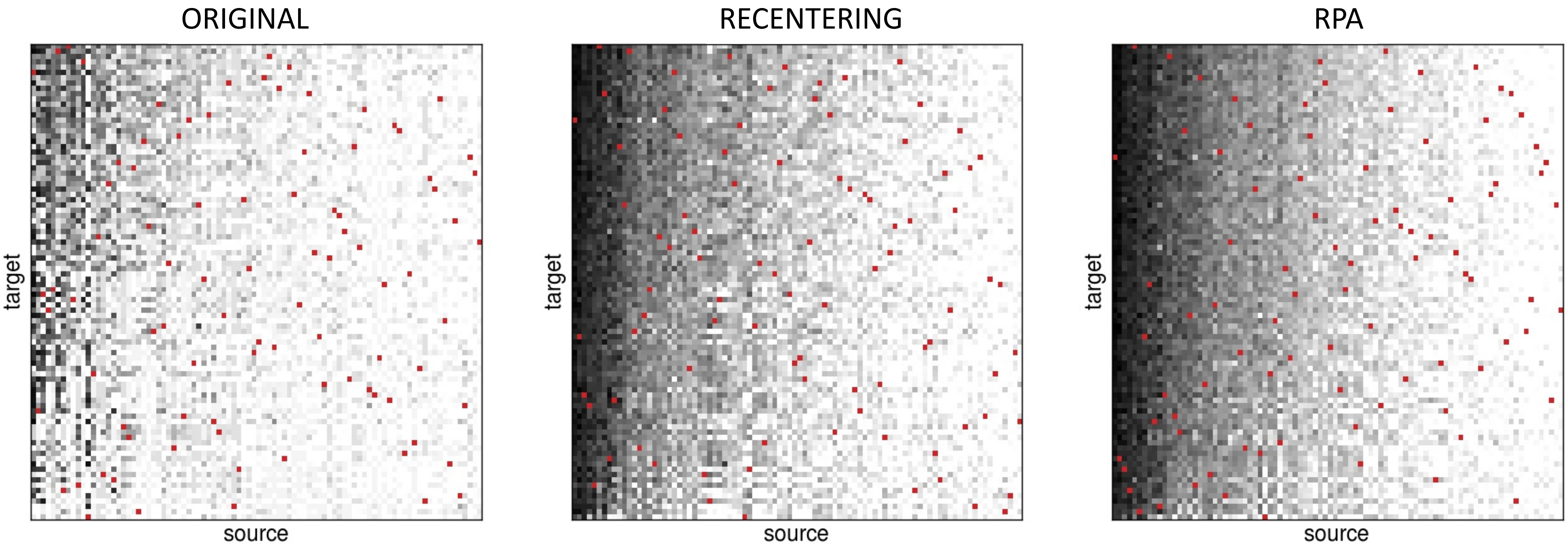
(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

Source subject = Target subjet

gipsa-lab

Cross-subject accuracy – Physionet MI

\# of data points in the target dataset : **10**

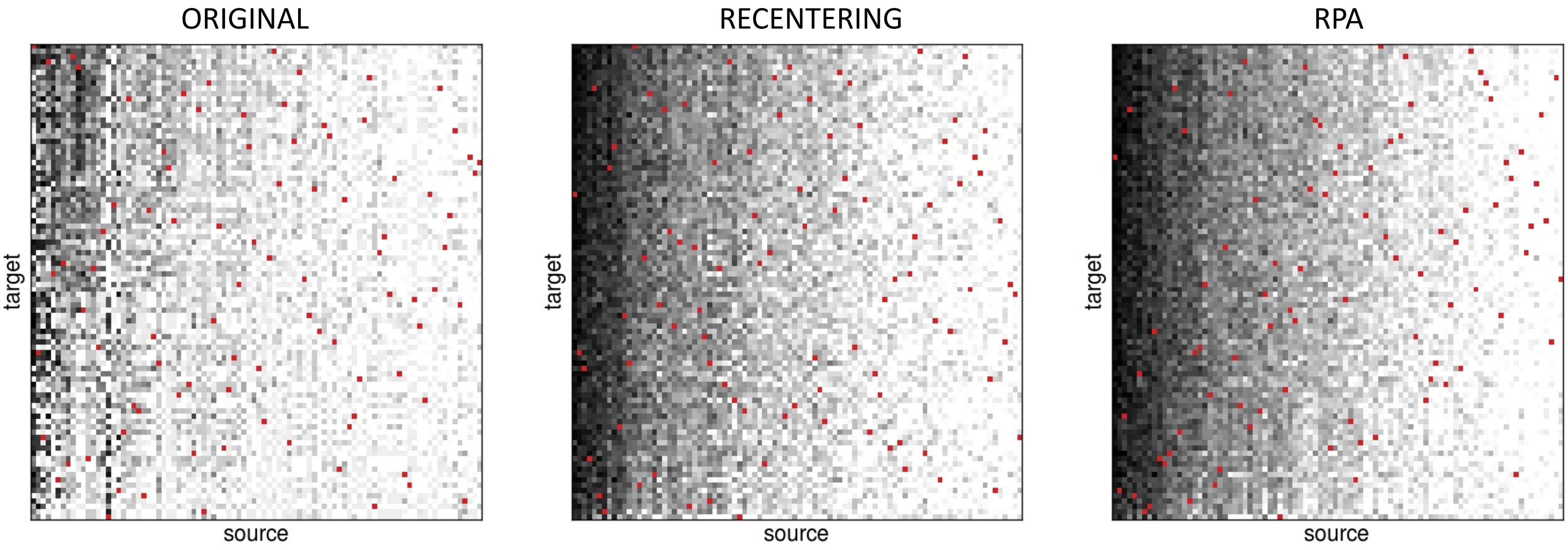


(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

Source subject = Target subjet

Cross-subject accuracy – Physionet MI

of data points in the target dataset : **15**

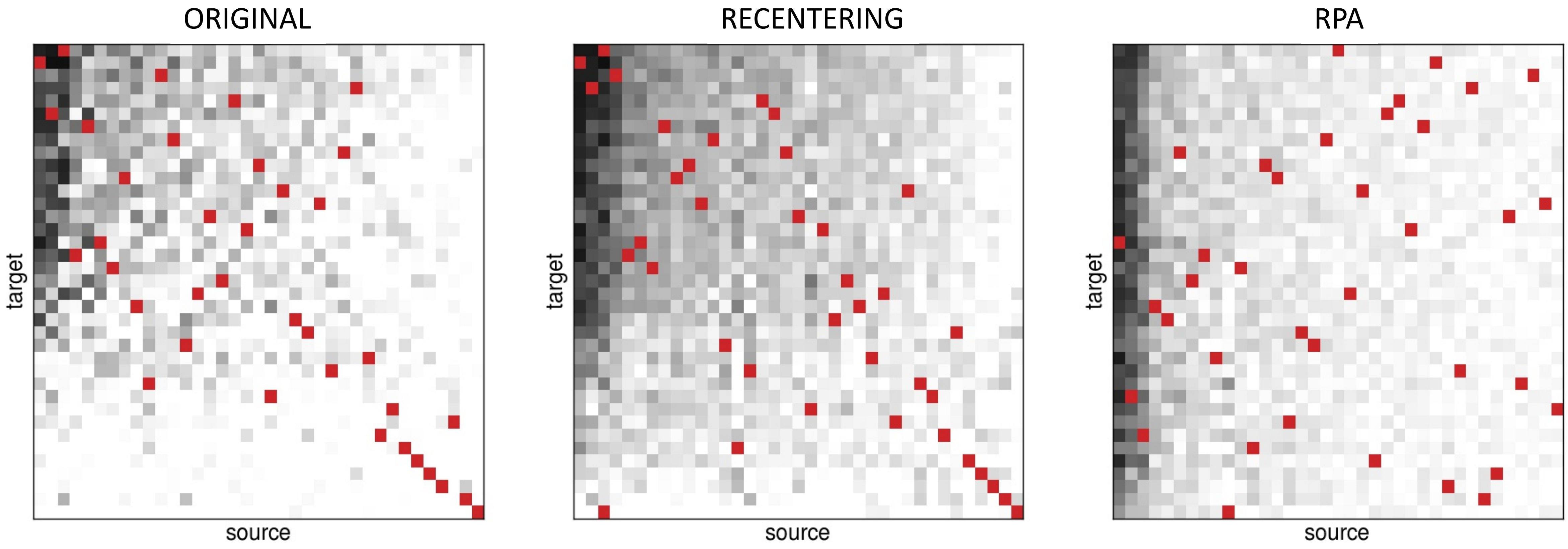


(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

Source subject = Target subjet

Cross-subject accuracy – GigaDB MI

of data points in the target dataset : **1**

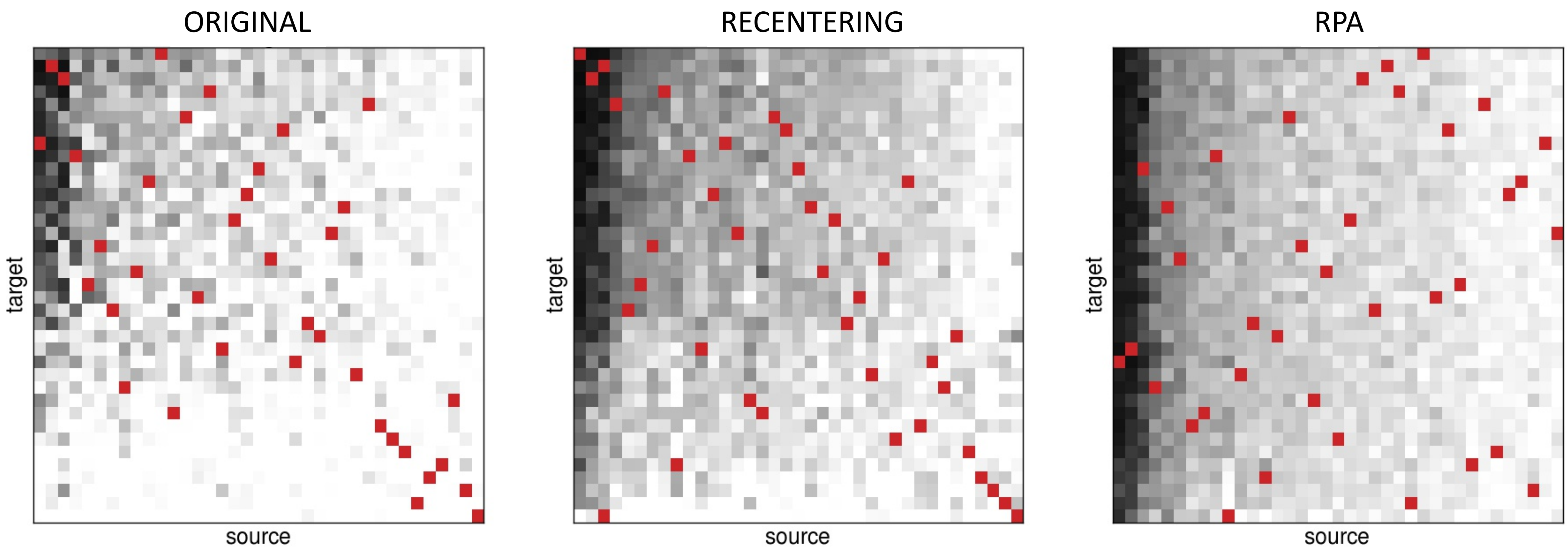


(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

Source subject = Target subjet

Cross-subject accuracy – GigaDB MI

of data points in the target dataset : **5**



(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

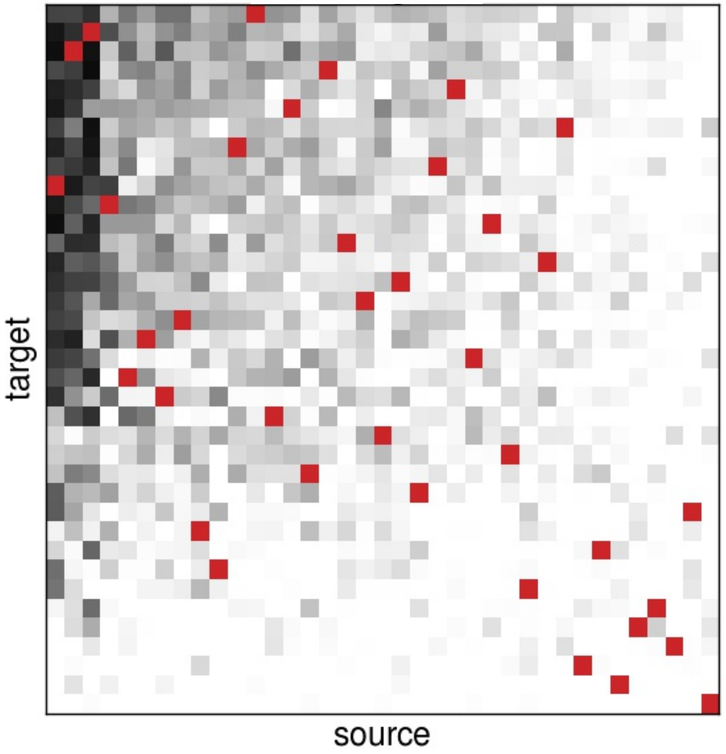
Source subject = Target subjet

gipsa-lab

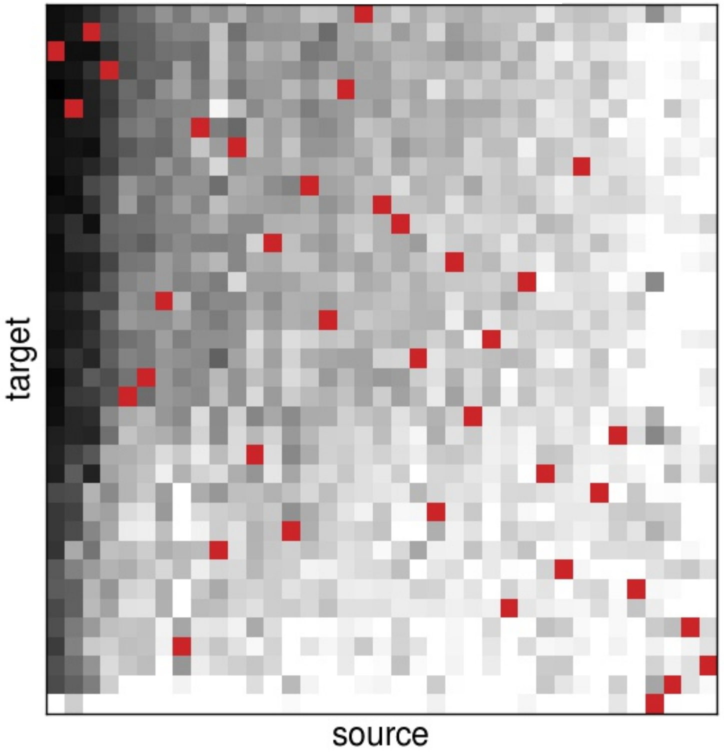
Cross-subject accuracy – GigaDB MI

of data points in the target dataset : **10**

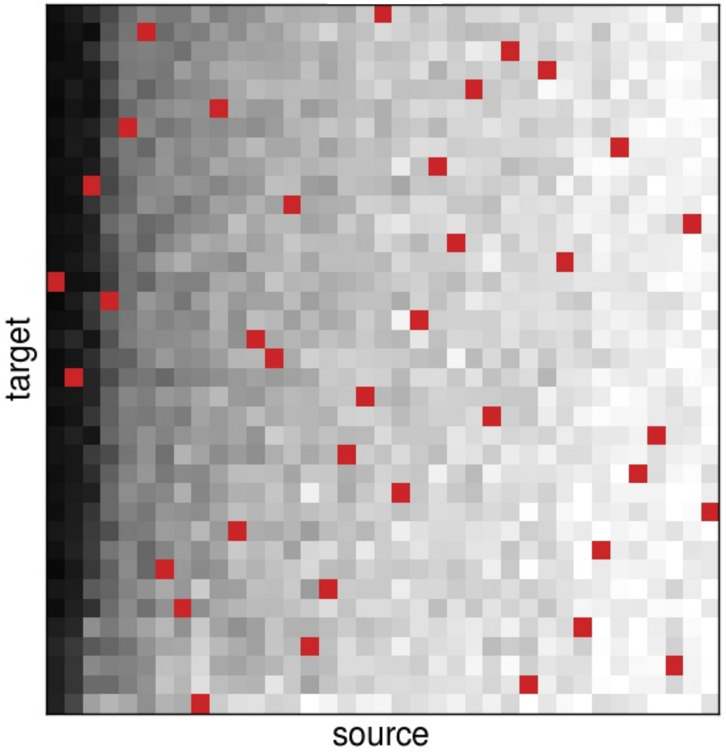
ORIGINAL



RECENTERING



RPA



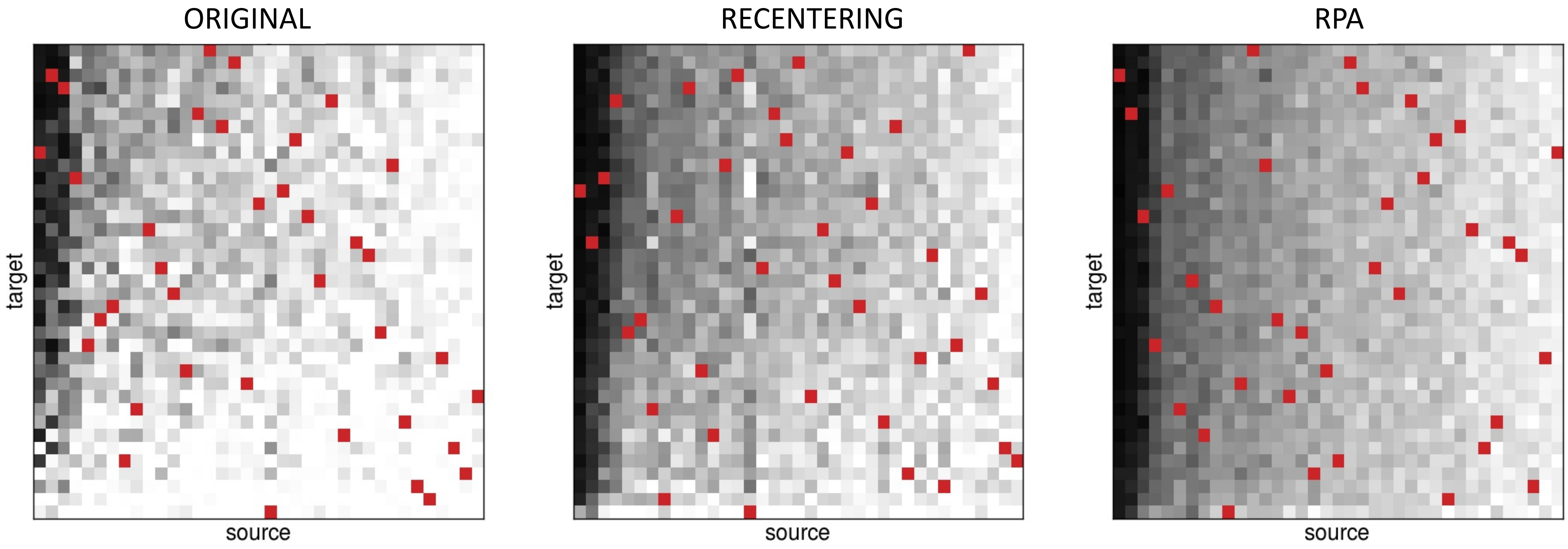
(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

Source subject = Target subjet

gipsa-lab

Cross-subject accuracy – GigaDB MI

of data points in the target dataset : **25**



(Colormap : **white** indicates 0.5 accuracy and **black** is 1.0)

Source subject = Target subjet

Cross-subject accuracy

MI-Physionet	MEAN ACCURACY		
N	direct	recent.	RPA
1	0.52	0.55	0.55
5	0.53	0.59	0.61
10	0.54	0.61	0.63
15	0.55	0.63	0.64

MI-GigaDB	MEAN ACCURACY		
N	direct	recent.	RPA
1	0.55	0.58	0.55
5	0.55	0.60	0.61
10	0.56	0.62	0.64
25	0.58	0.65	0.67

SSVEP	MEAN ACCURACY		
N	direct	recent.	RPA
1	0.59	0.59	0.61
2	0.64	0.65	0.68
4	0.71	0.72	0.76
6	0.75	0.77	0.80

P300	MEAN AUC		
N	direct	recent.	RPA
6	0.49	0.59	0.55
12	0.50	0.57	0.57
32	0.52	0.54	0.57
48	0.53	0.54	0.56

Outline

Our method works for **any** BCI paradigm : it depends only on the choice of SPD matrices

Our method works for **any** BCI paradigm : it depends only on the choice of SPD matrices

In every dataset there are **good** and **bad** source subjects. How to deal with this ?

Our method works for **any** BCI paradigm : it depends only on the choice of SPD matrices

In every dataset there are **good** and **bad** source subjects. How to deal with this ?

Our method could be used with **boosting** and other strategies for **combining** classifiers

Our method works for **any** BCI paradigm : it depends only on the choice of SPD matrices

In every dataset there are **good** and **bad** source subjects. How to deal with this ?

Our method could be used with **boosting** and other strategies for **combining** classifiers

Some **interesting** works related to ours

Zanini et al. – *"Transfer Learning: a Riemannian geometry framework with applications to Brain-Computer Interfaces"* (2018)

Gayraud et al. – *"Optimal Transport Applied to Transfer Learning"* (2017)

Arvaneh et al. – *"EEG data space to reduce intersession nonstationarity in brain-computer interface"* (2013)

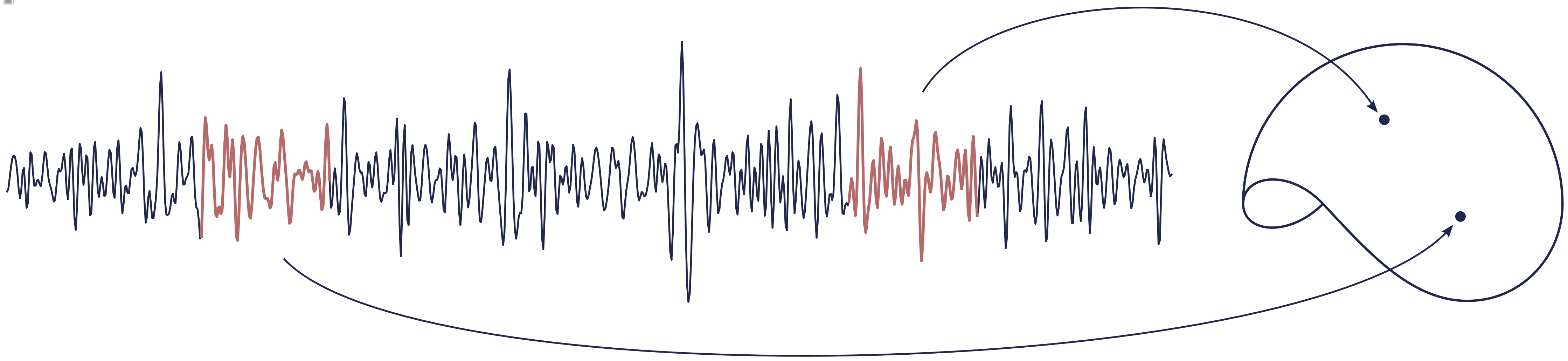
Thank you for your attention :-)

Questions ?

Riemannian Procrustres Analysis

Transfer Learning for Brain Computer Interfaces

Pedro L. C. Rodrigues   Marco Congedo   Christian Jutten



Presentation for the Machine Learning group at GIPSA-lab

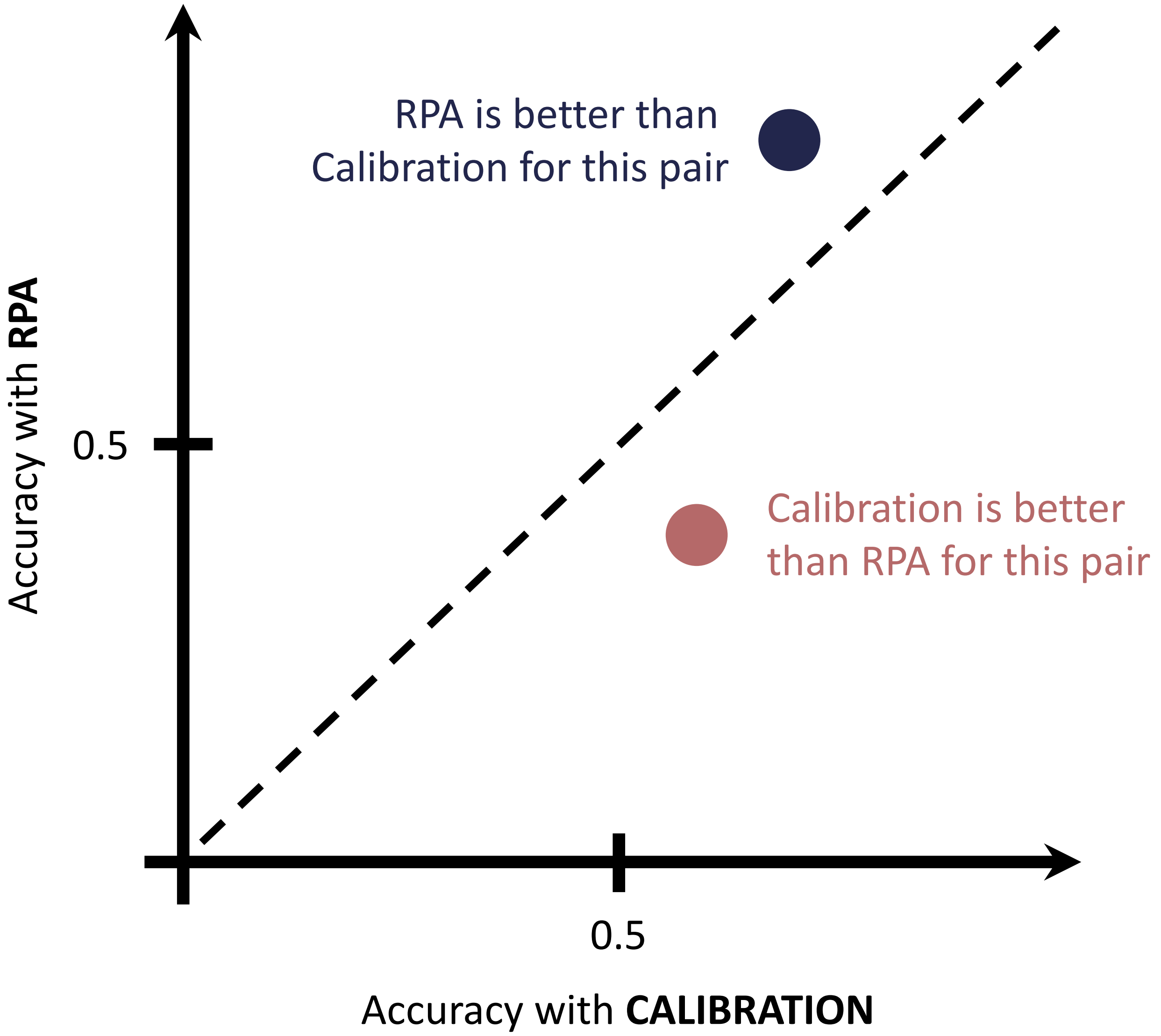
22th May 2018, Grenoble, France

Better than calibration ?

Compare RPA to **Calibration**

(Consider an **increasing** number of available points in the target dataset)

Use the Wilcoxon Paired statistical test to check if one method is **superior** to the other in **average** (for all pairs of subjects)

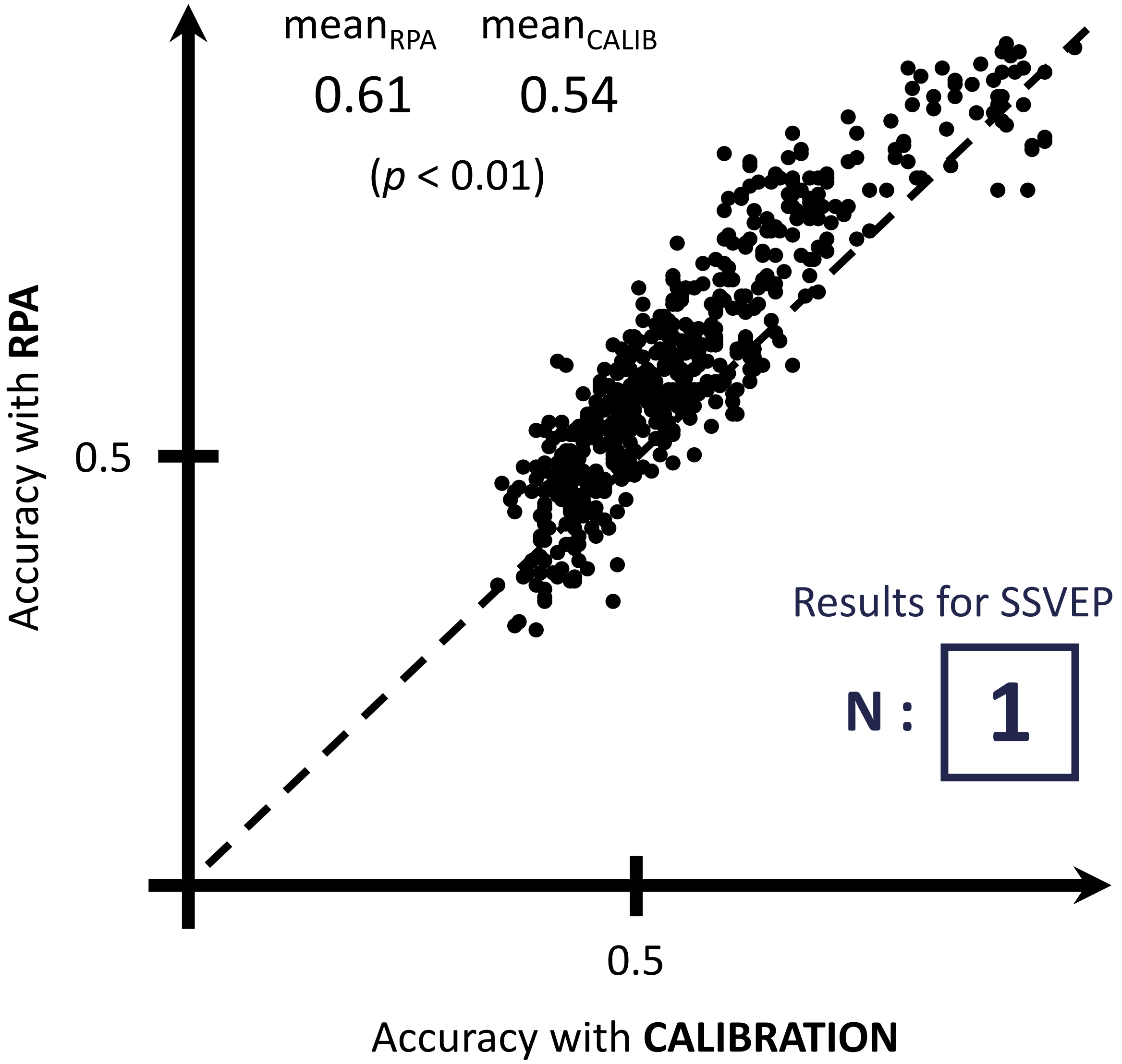


Better than calibration ?

Compare RPA to **Calibration**

(Consider an **increasing** number of available points in the target dataset)

Use the Wilcoxon Paired statistical test to check if one method is **superior** to the other in **average** (for all pairs of subjects)

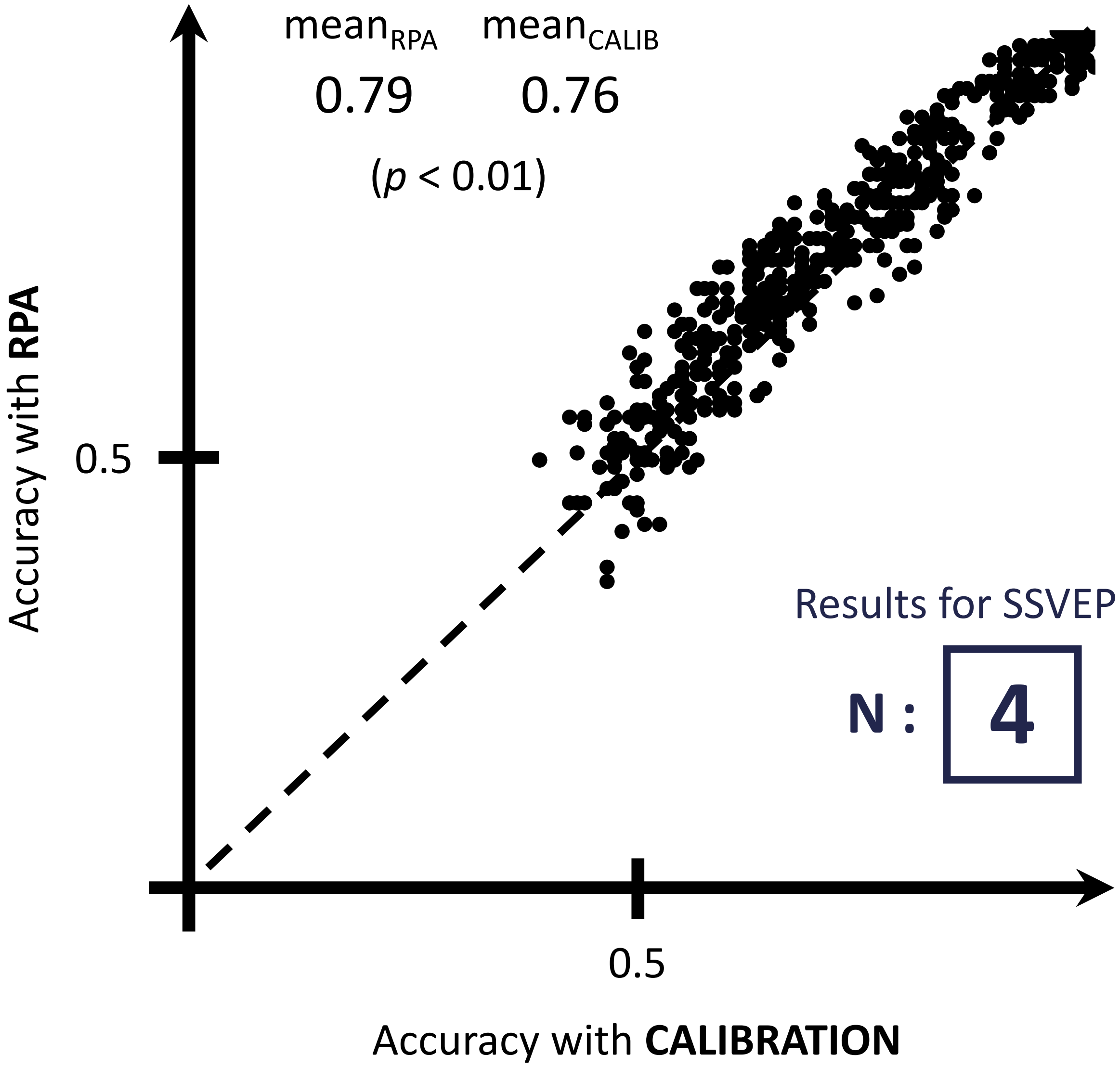


Better than calibration ?

Compare RPA to **Calibration**

(Consider an **increasing** number of available points in the target dataset)

Use the Wilcoxon Paired statistical test to check if one method is **superior** to the other in **average** (for all pairs of subjects)



Results on the GIGADB dataset (Motor Imagery, Two classes)