Gentle Introduction to Signal Processing and Classification for Single-Trial ERP Analysis

Benjamin Blankertz

Neurotechnology Group, Technische Universität Berlin

benjamin.blankertz@tu-berlin.de

http://www.user.tu-berlin.de/blanker

24|Feb|2014

Overview of this Tutorial

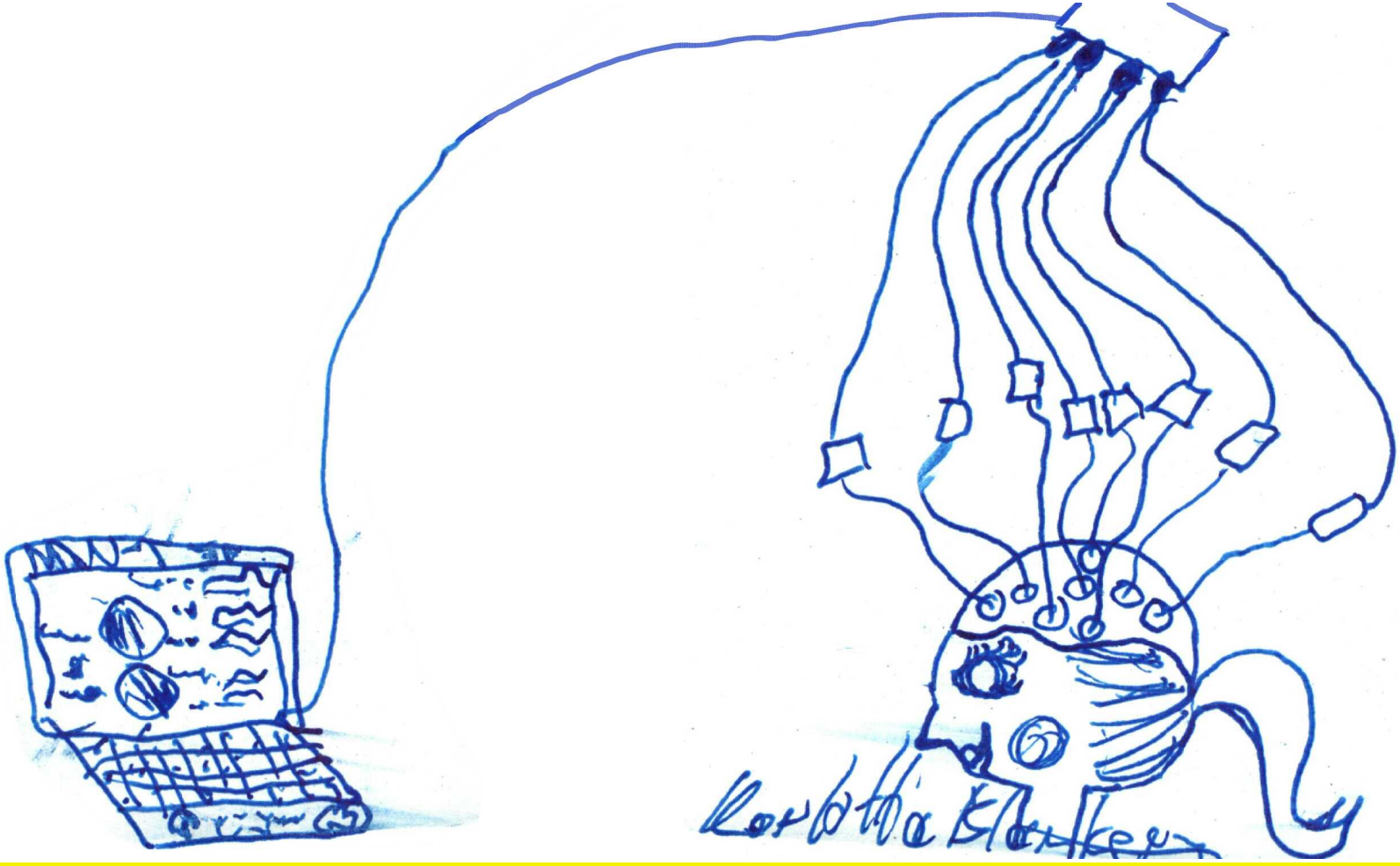
- From the Oddball Paradigm to ERP-based BCI Spellers
- From Uni- to Multivariate Features
- Classification of ERP Features
- Understanding Spatial Filters

- The Linear Model
- Illustration of Spatial Patterns and Filters
- Interpretability of Spatial Filters
- Issues in Validation

Part I

Introduction to ERP-based BCIs

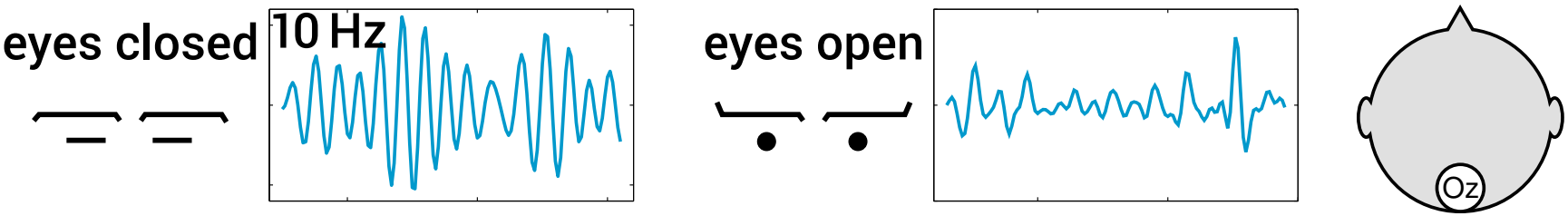
Non-Invasive Brain-Computer Interaction



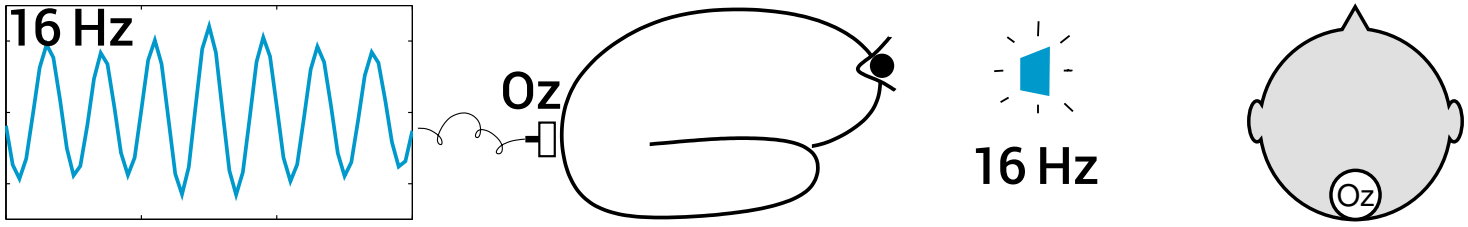
Real-time recognition of mental states of users based on brain activity for enriching human-machine interaction

Different Occurences of Neural Activity

- spontaneous oscillations, e.g., sensorimotor rhythm (SMR), or visual alpha rhythm

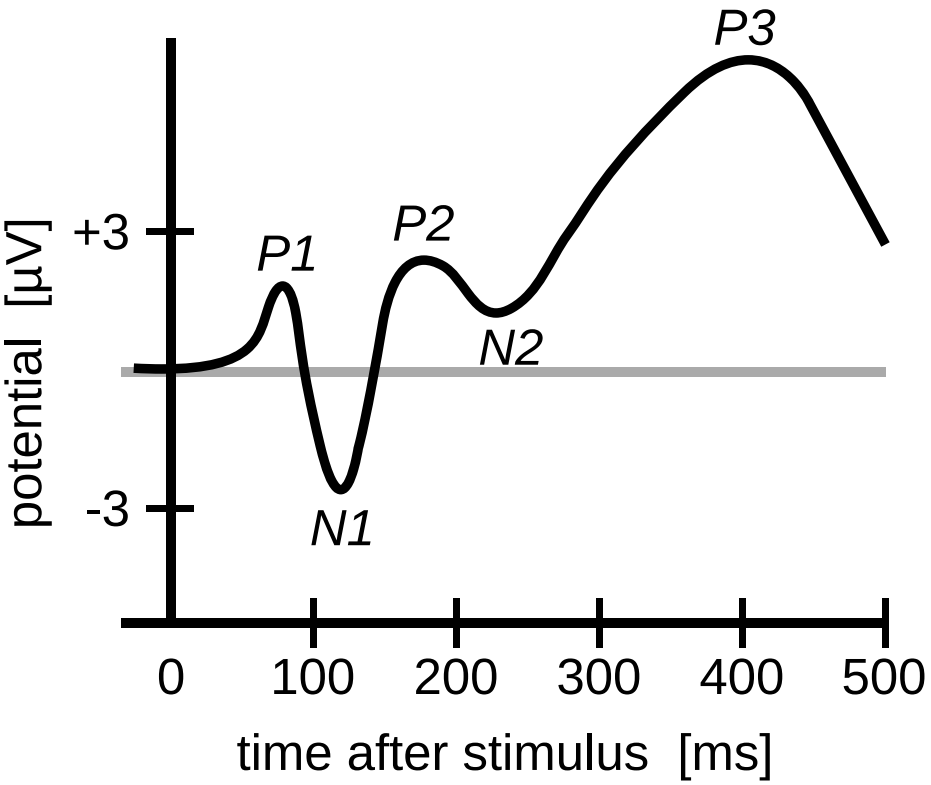


- induced oscillations, e.g., steady-state visual evoked potentials (SSVEP), auditory steady-state response (ASSR), evoked by and synchronous to a periodic external stimulus



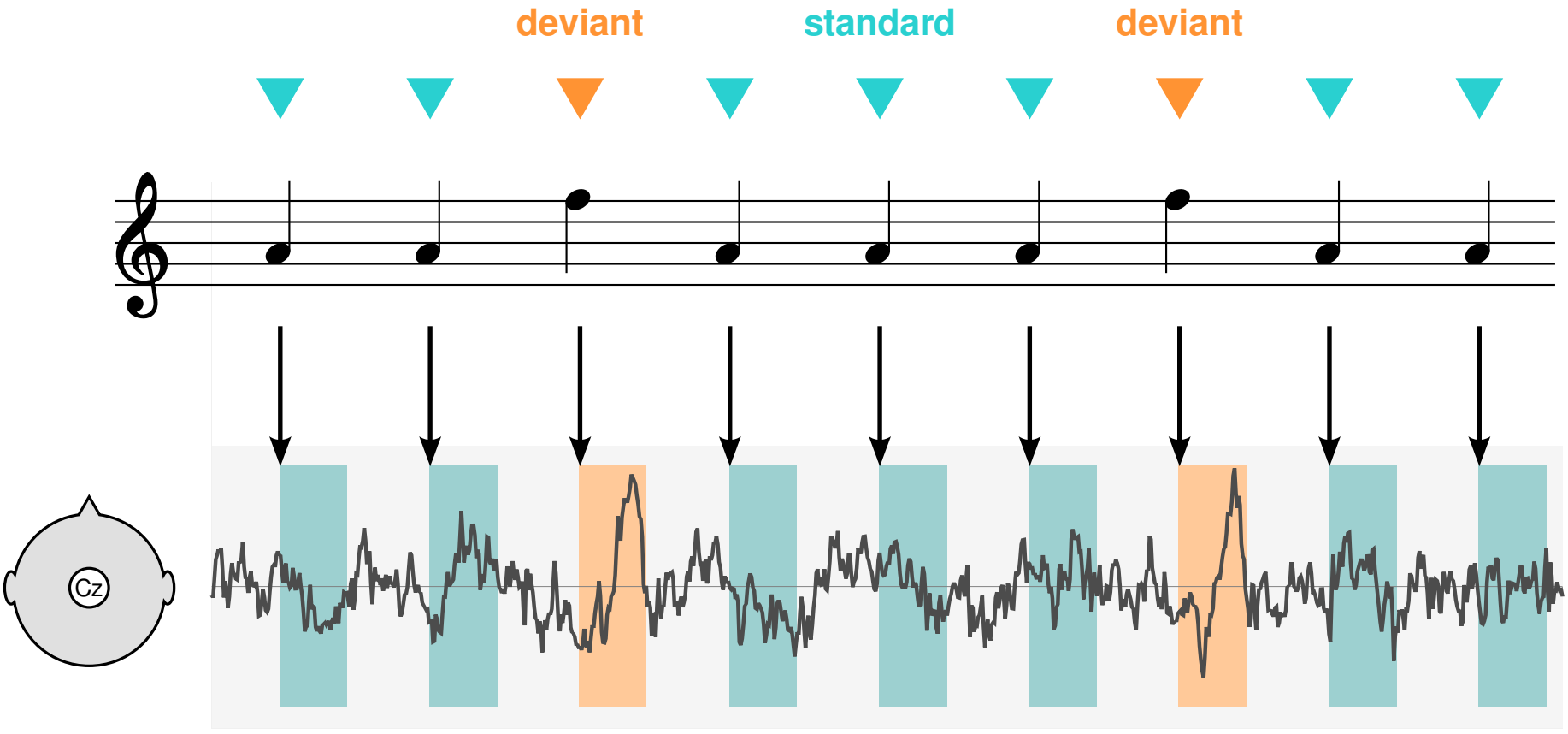
- transient activity, event-related potentials (ERPs), time-locked to an event, most often an external stimulus

Prototypical ERP



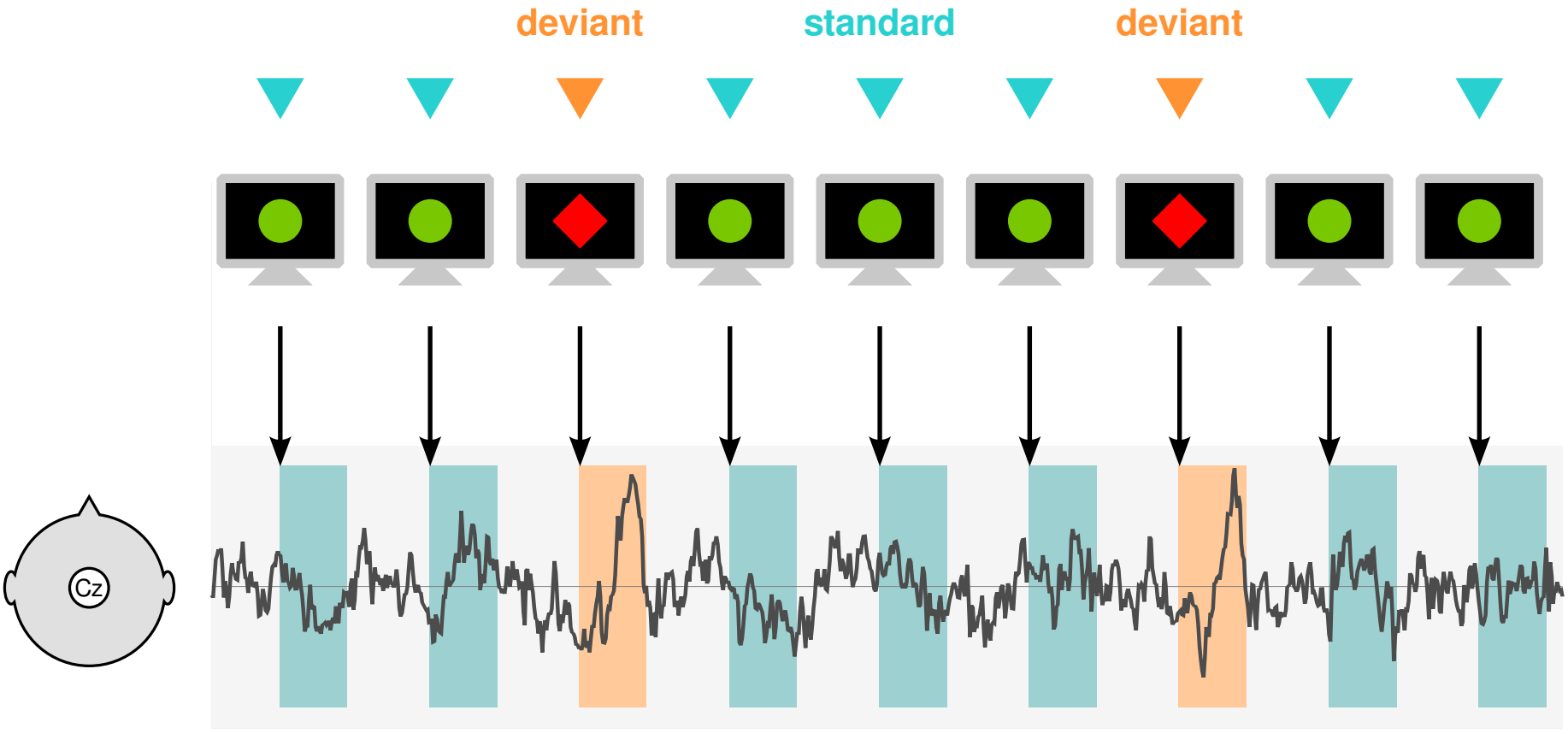
- The shown components are also labeled *P100*, *N100*, *P200*, *N200*, *P300*.
- The *P3* component is often composed of two subcomponents labeled *P3a* and *P3b* which originate from different locations in the brain.
- Be aware: sometimes negative polarity is plotted upwards.

From the Oddball Paradigm to a BCI Speller



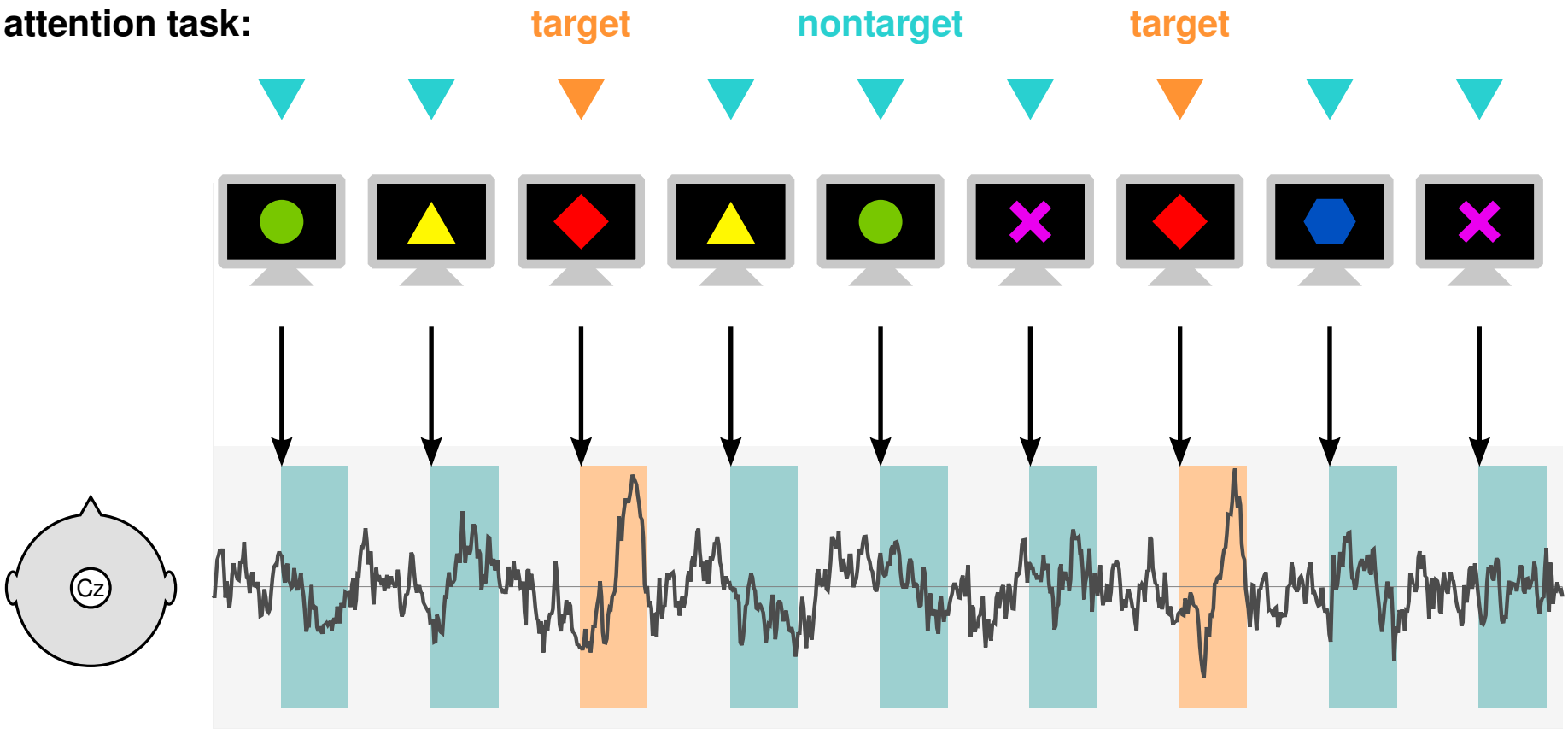
- Segments of the signal (shaded in the figure) are called *epochs* or *single-trials*.
- Typically, trials are grouped into several classes (which are, e.g., defined by experimental conditions), here *standards* and *deviants*.

From the Oddball Paradigm to a BCI Speller



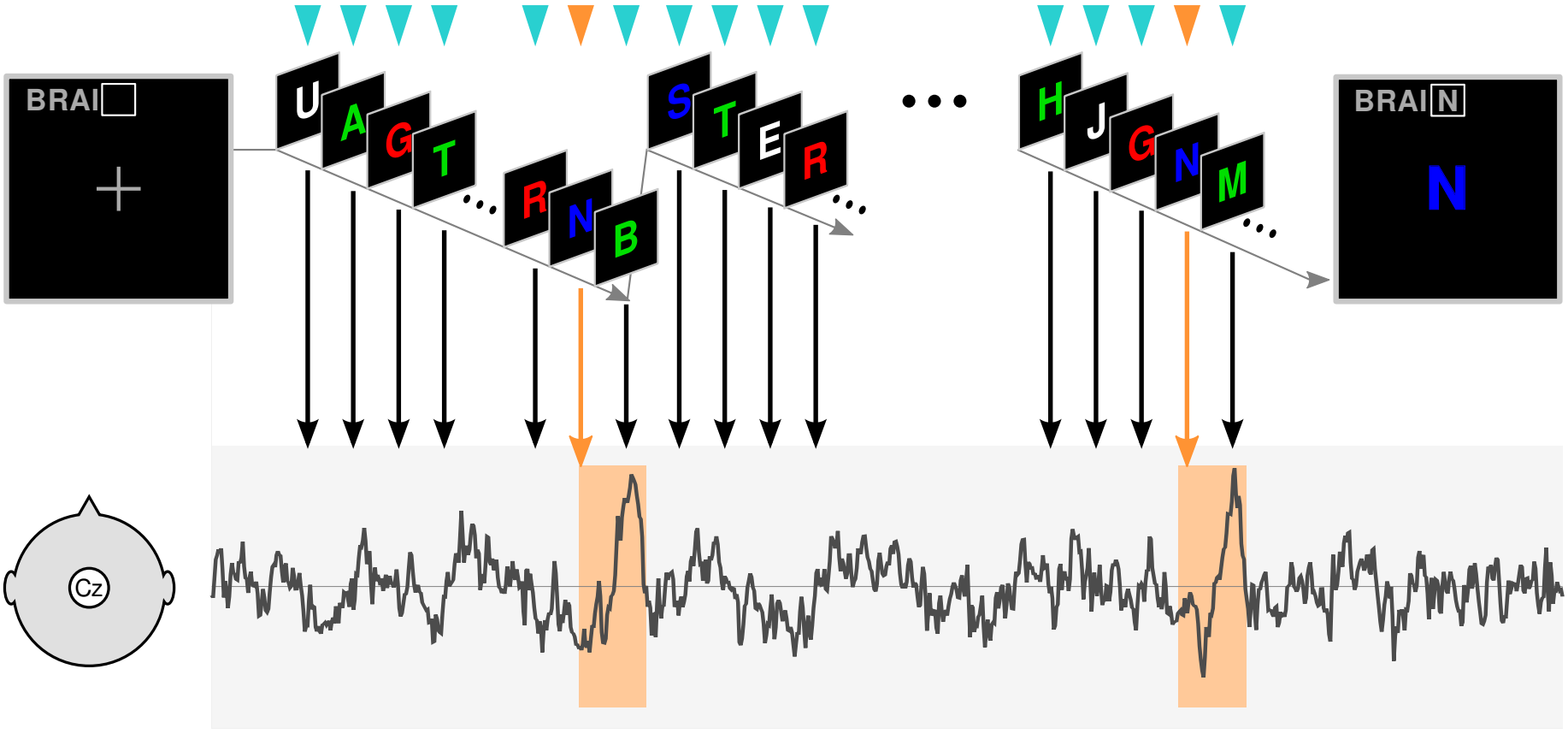
- In the visual domain it works the same.

From the Oddball Paradigm to a BCI Speller



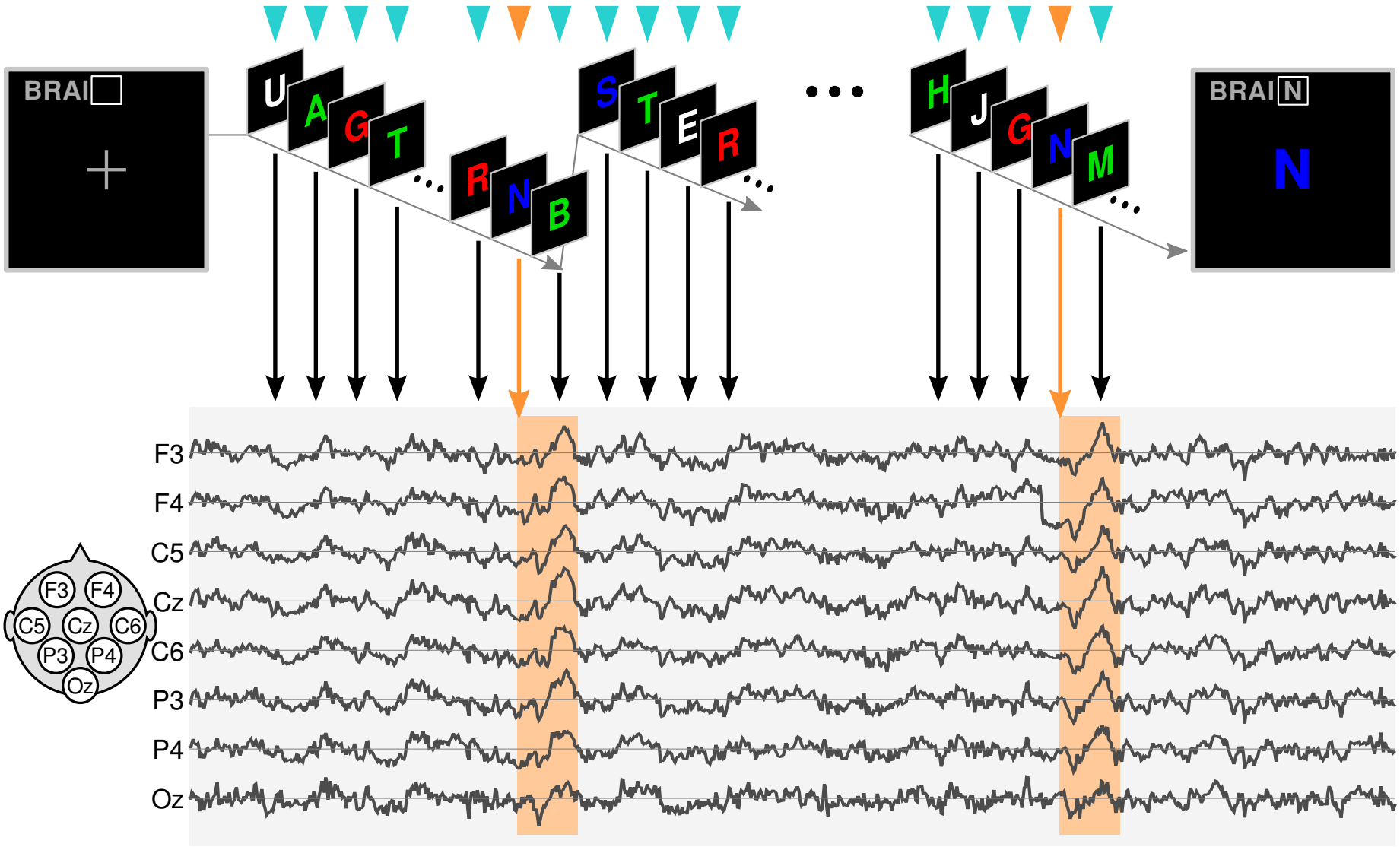
- Due to the attention task, the two classes of stimuli are also called *targets* and *nontargets*.
- Here, five stimuli are presented with the same probability. One of them is defined to be the target in an attention task.
- Thus, the class of *nontargets* is composed of various stimuli.

From the Oddball Paradigm to a BCI Speller

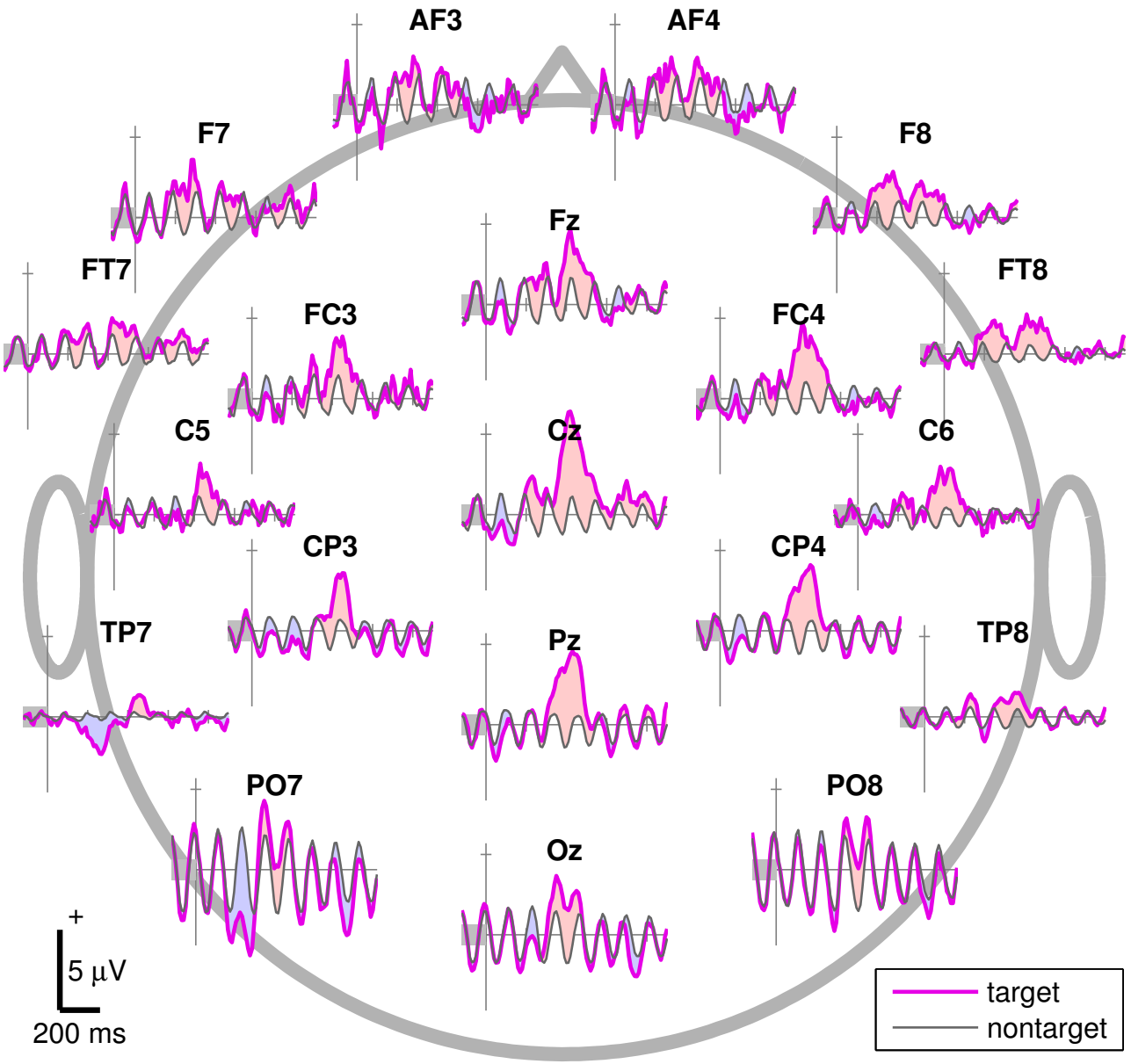


- The intended letter is the target and all others are nontargets.
- In BCI epochs are typically strongly overlapping. (Nontarget epochs are not shaded in this figure.)

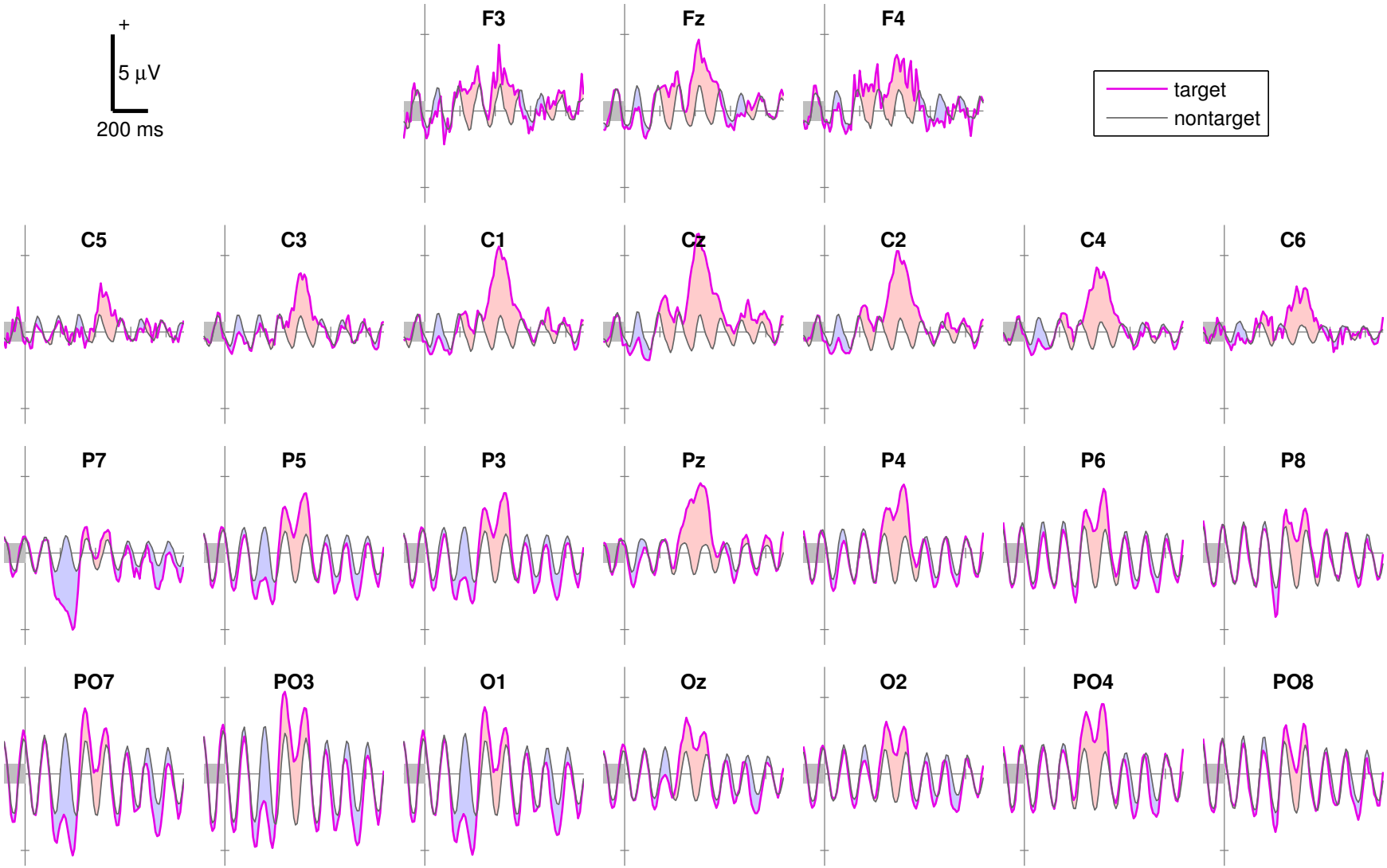
Multi-channel Epochs



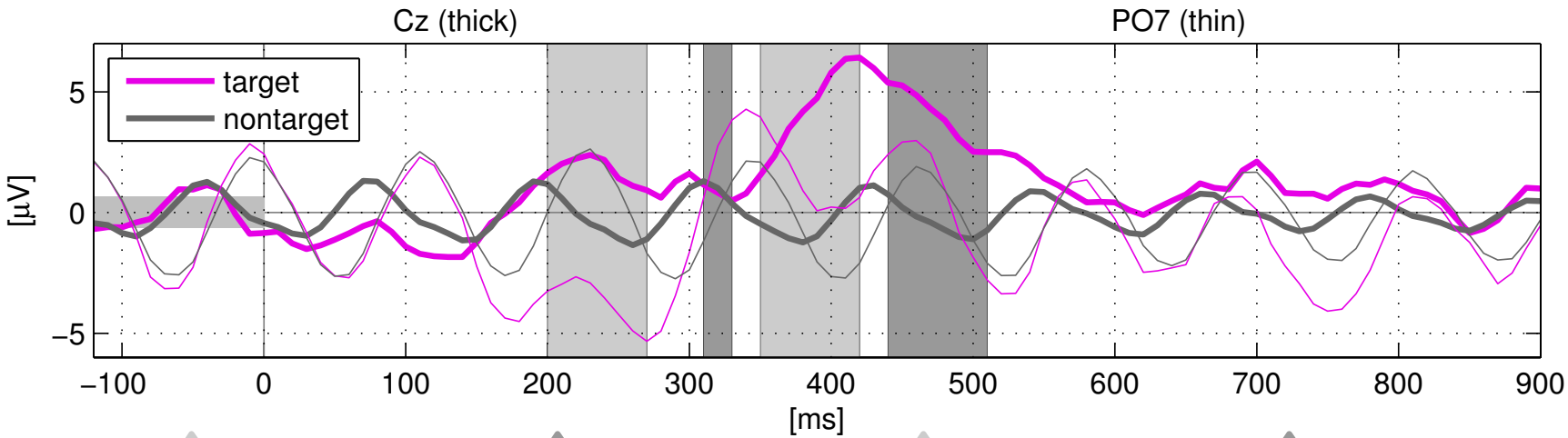
ERPs in a Head Plot

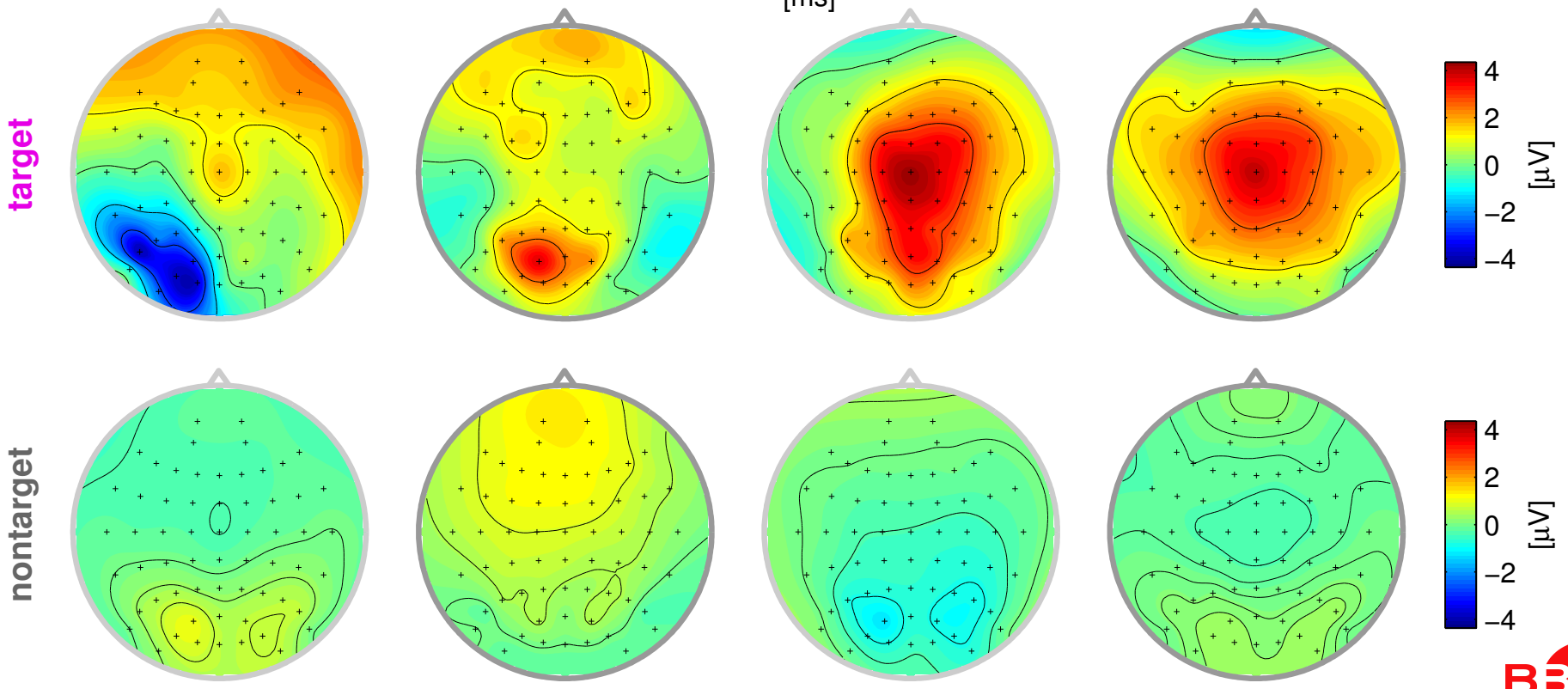


ERPs in a Grid Plot



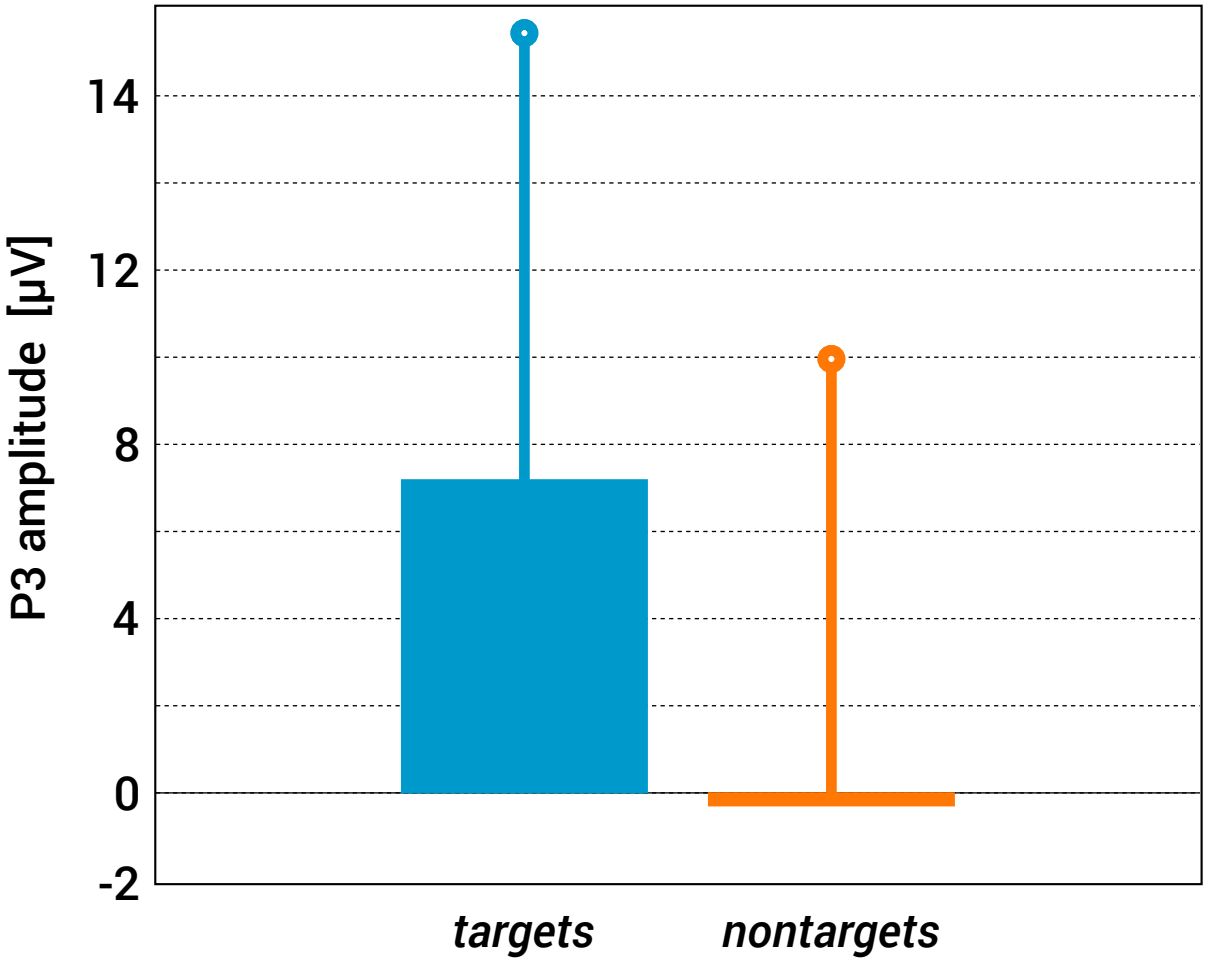
ERP Topographies





Classical Investigation of Target vs Nontarget

The classical way to compare ERPs of different conditions:



Statistics of peak amplitudes.

Part II

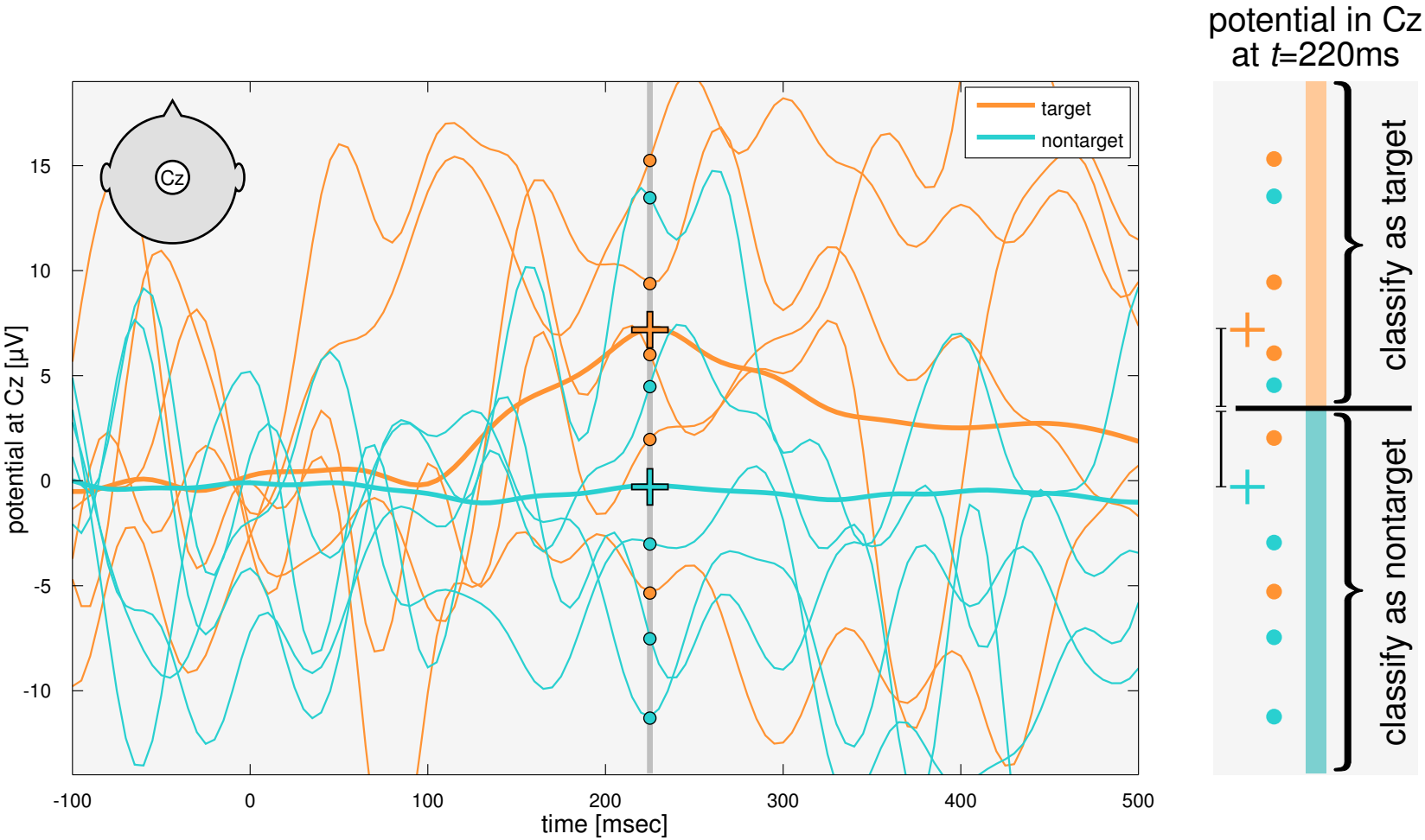
From Uni- to Multivariate Features

First, a Simple Approach to Classification

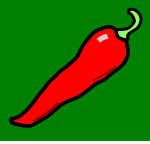
- To implement a BCI Speller, we need to distinguish target from nontarget trials.
- From the oddball literature, we conjecture that best discrimination between those classes is granted by the P3 component.
- The P3 component has its spatial focus at electrode position Cz.
- As first step for discrimination, we take as 'feature':

the amplitude at the peak time of the P3 at Cz

Univariate Features: Averages and Single-Trials



- The potential measured 220ms post-stimulus at **Cz** is a one-dimensional observation variable: a **univariate feature**.



Measures of Separability

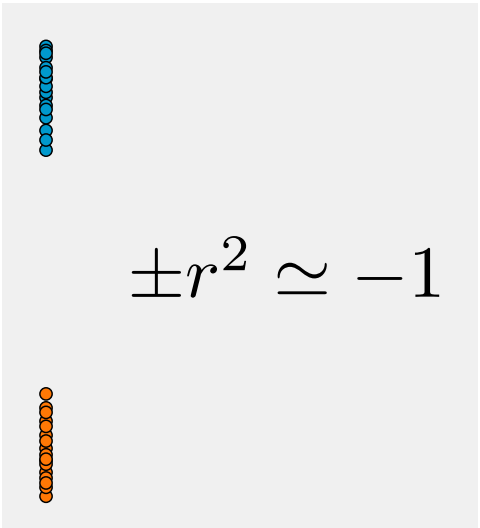
In order to assess the discriminative value of univariate features, we are interested in **measures of separability**.

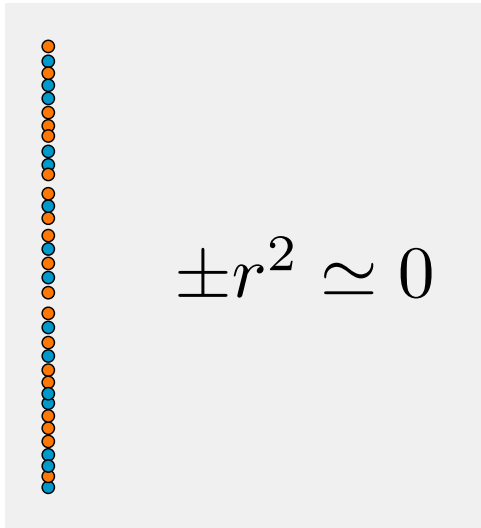
One such measure is the r^2-value, which is defined as

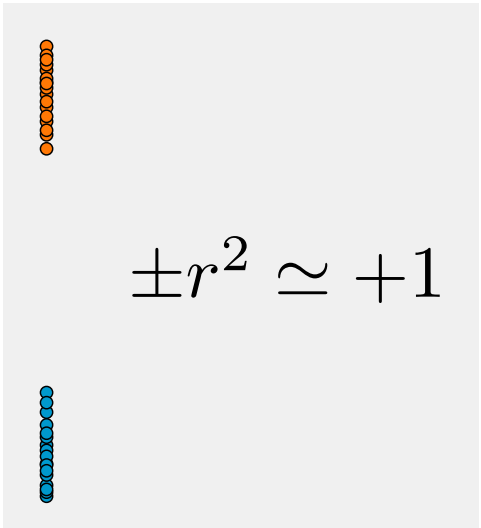
$$r^2(x, y) := \frac{N_1 \cdot N_2}{(N_1 + N_2)^2} \frac{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^2}{\operatorname{var}\langle x_i \rangle}$$

with $\mu_1 = \operatorname{mean}\langle x_i \rangle_{y_i=1}$ and $\mu_2 = \operatorname{mean}\langle x_i \rangle_{y_i=2}$ being the class means and $N_k = |\{i \,|\, y_i = k\}|$ being the number of samples of class k.

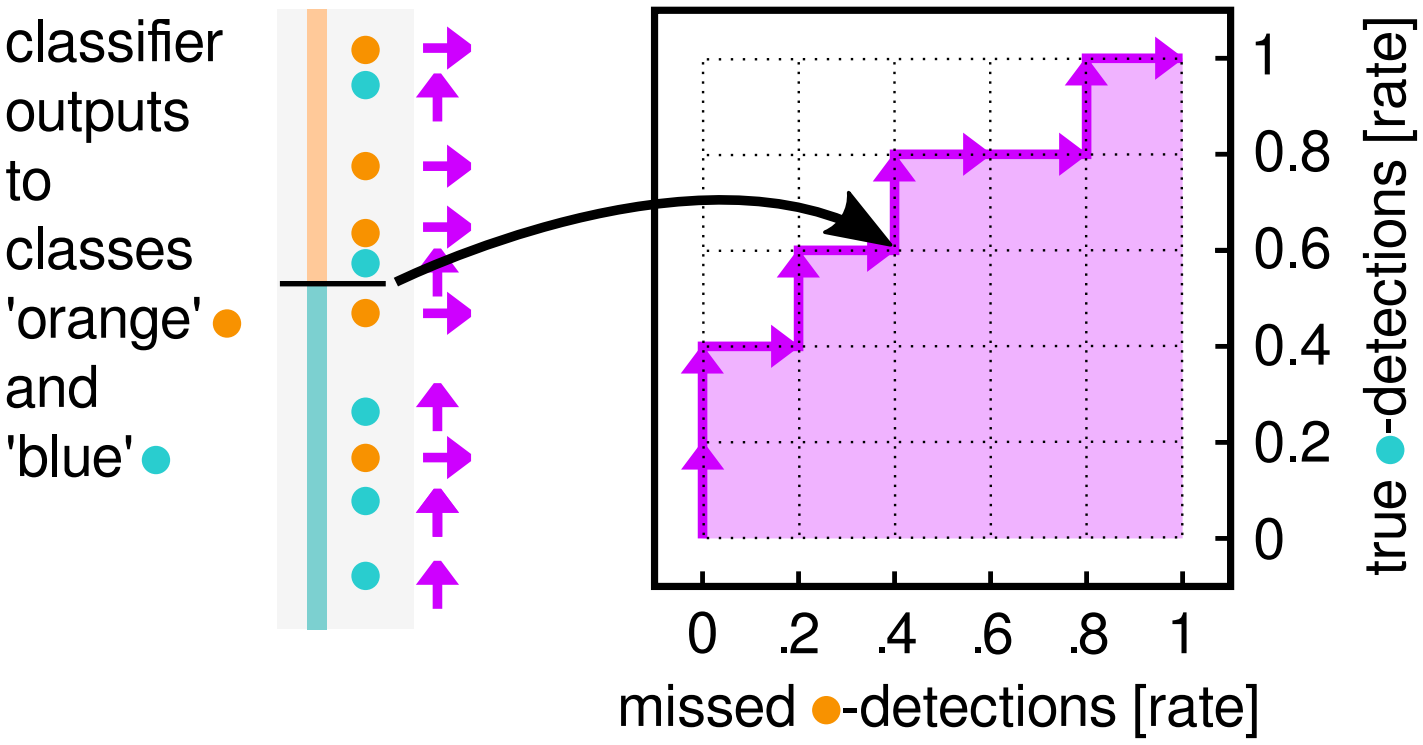
To retain a sign r^2 can be multiplied by the sign of the difference $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$.







Area under the Curve (AUC) as Measure of Seperation



area under the ROC curve:

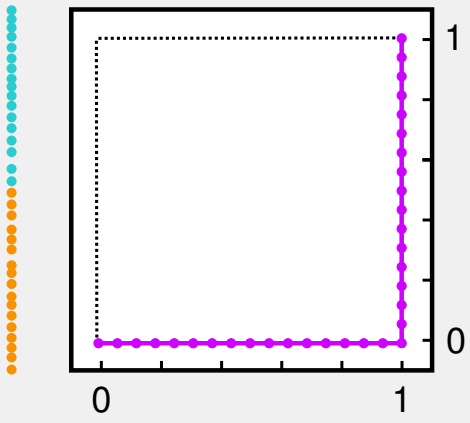
$$\text{AUC} = \frac{18}{25}$$

$= p$ that a random is ranked higher than a random

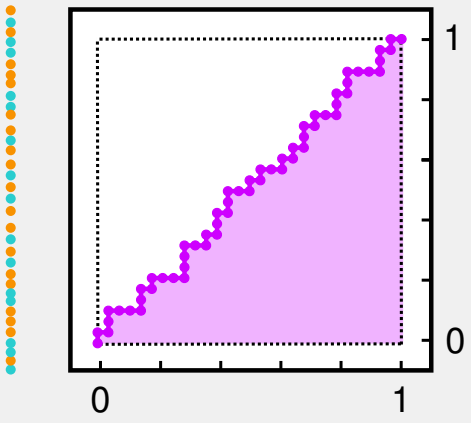
- Area Under the ROC Curve (AUC): **Measure of separation** of two univariate distributions

Examples for ROC Curves and AUC Values

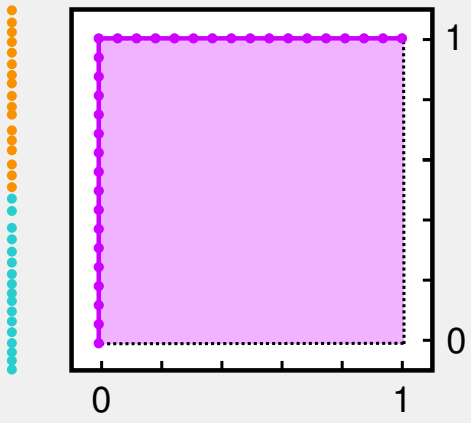
perfectly separated distributions:
AUC = 0



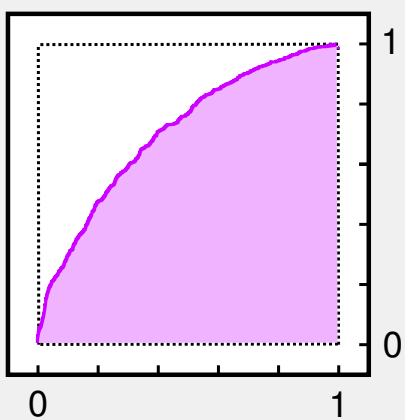
random distributions:
AUC ≈ 0.5



perfectly separated distributions:
AUC = 1

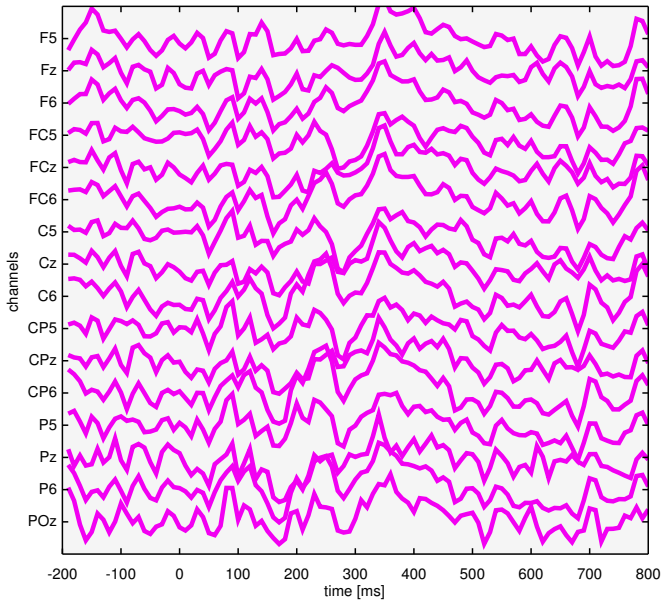


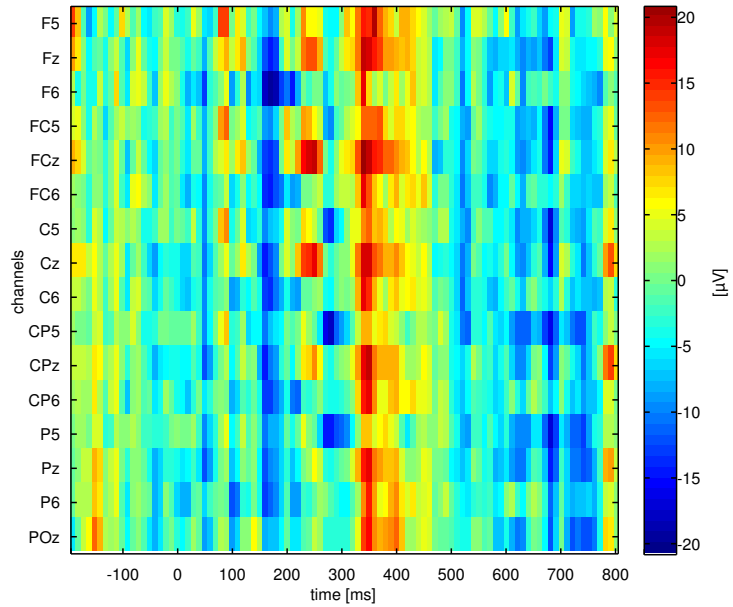
classifier outputs from our example data
AUC ≈ 0.7

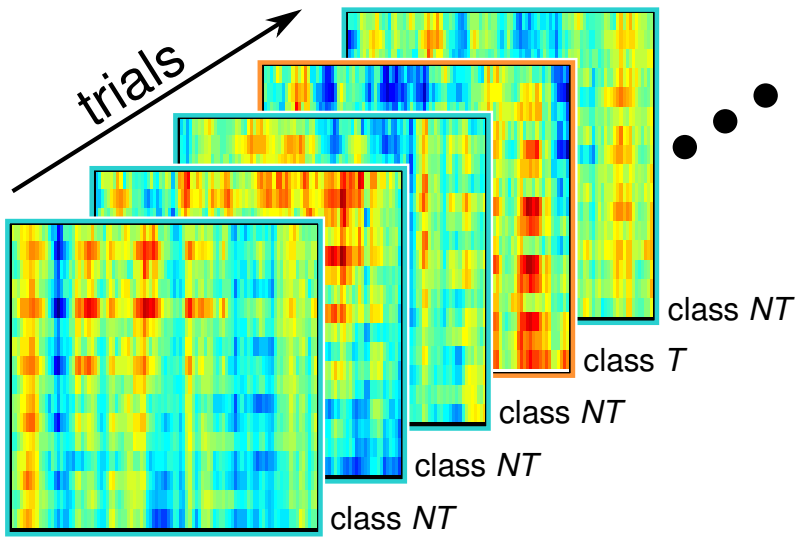


- AUC is a performance measure (like *misclassification rate*) but bias independent.

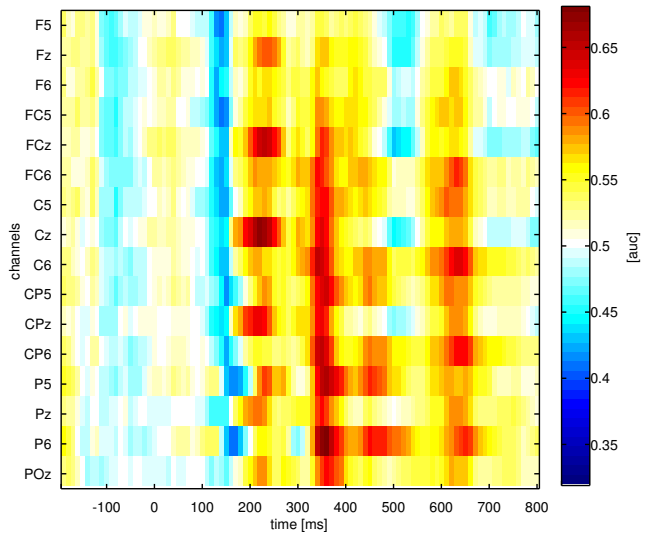
Interlude: Representation as Matrix



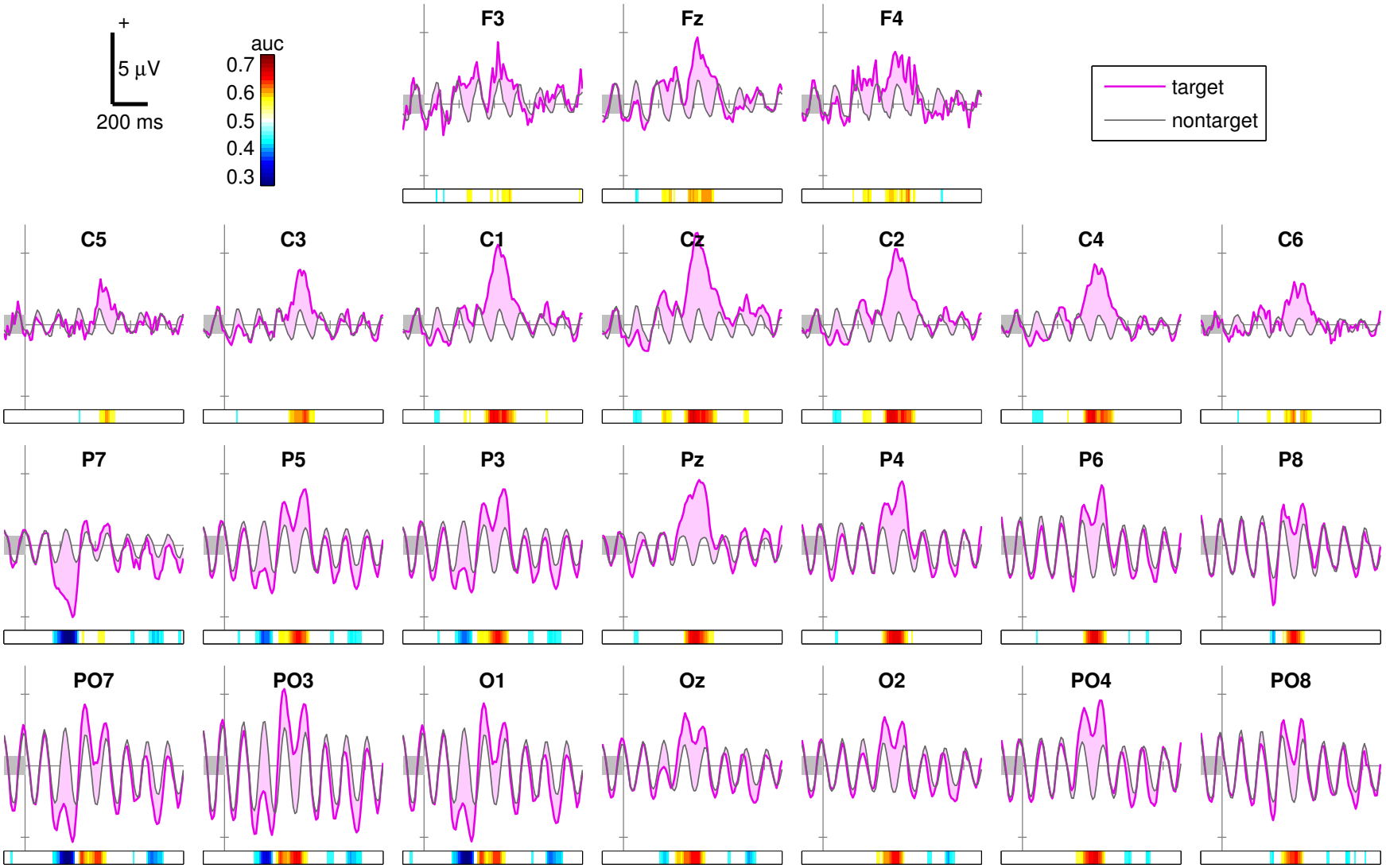




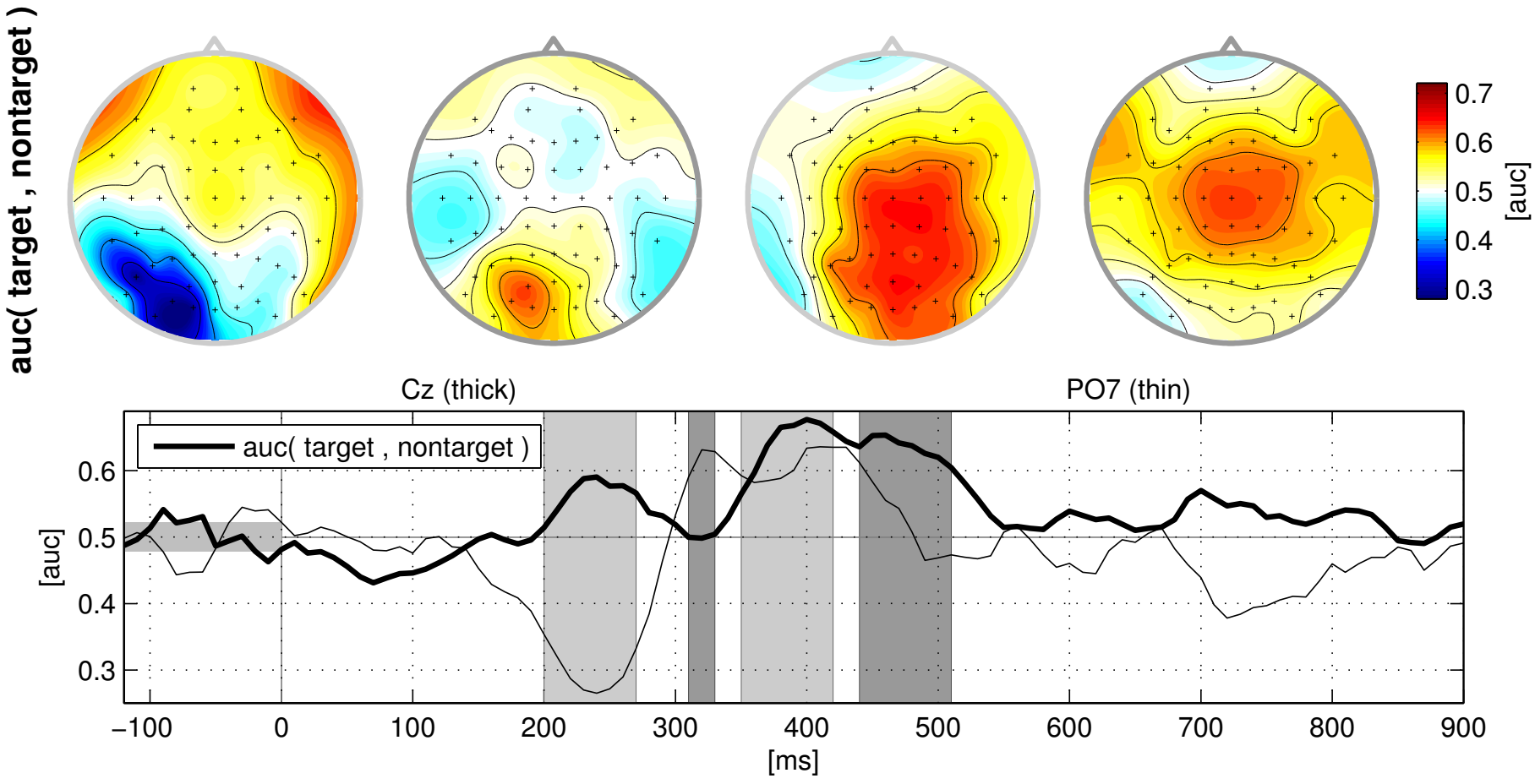
AUC across trials



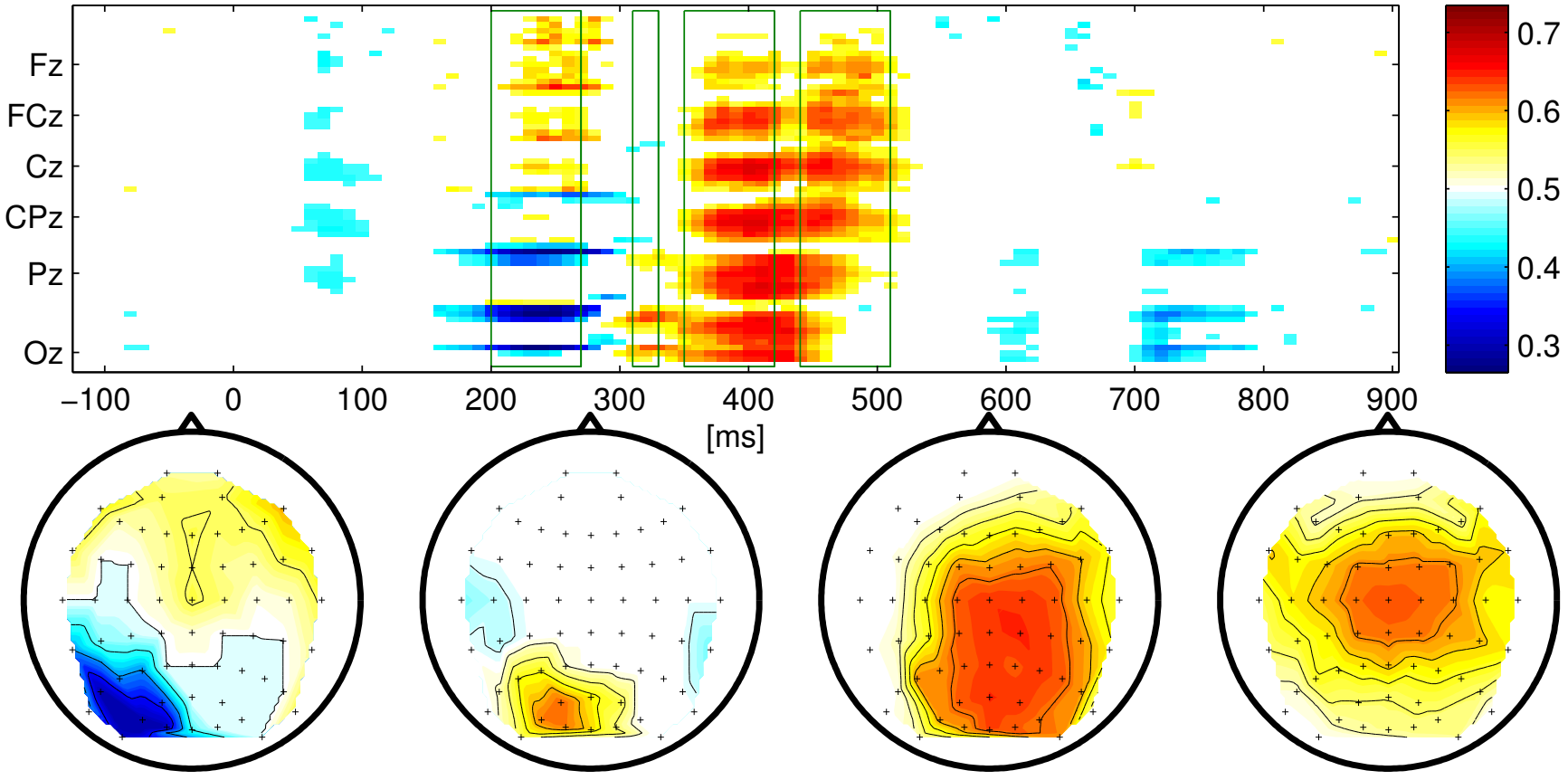
ERPs in a Grid Plot with AUC Scores



ERP Topographies of AUC Scores



AUC Matrix: Overview of Discriminative Information



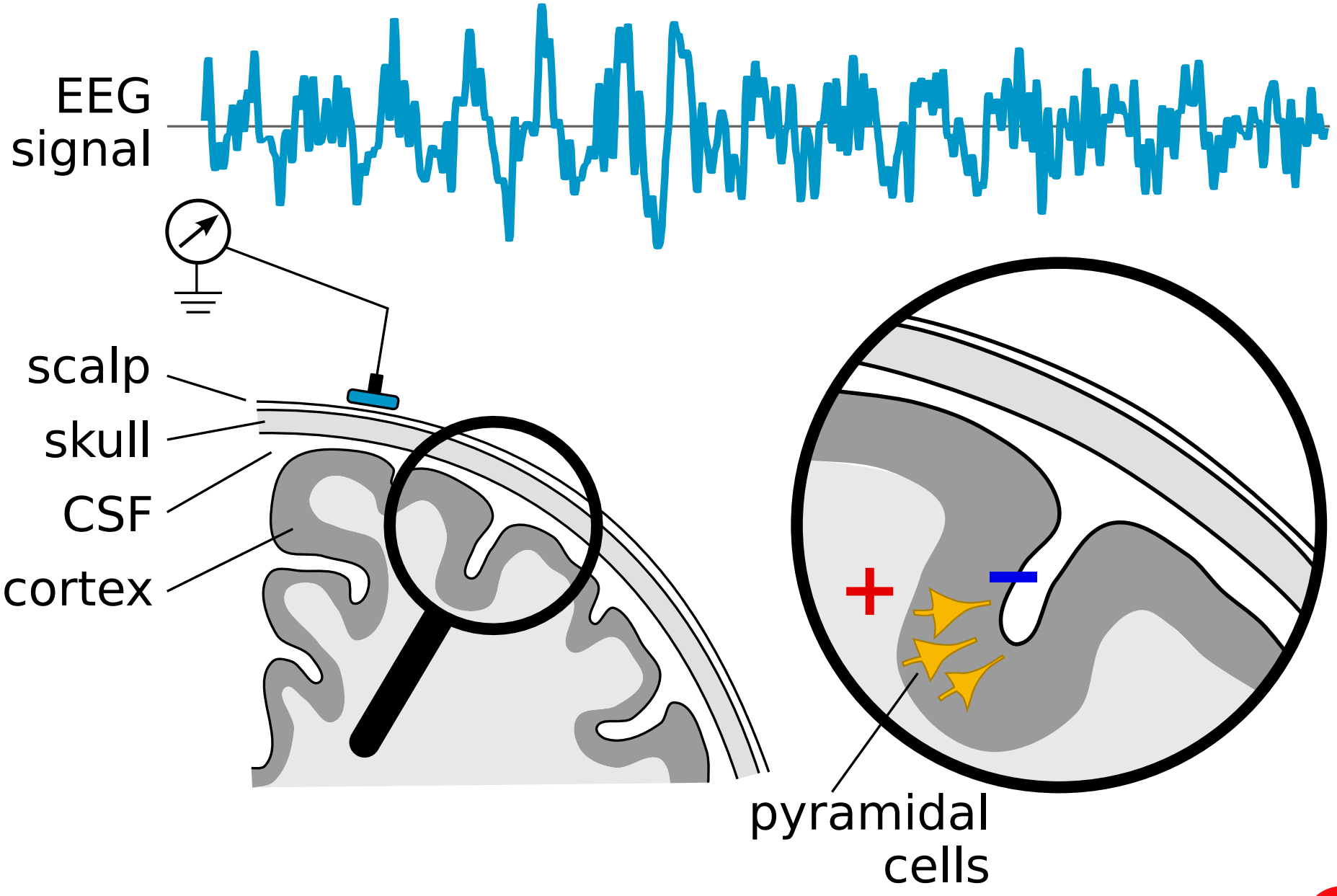
- The AUC Score matrix shows spatio-temporal evolution, and
- can be used to select meaningful time intervals (keep this in mind for later) or
- to select the most discriminative (univariate) feature.

Generation of EEG Signals

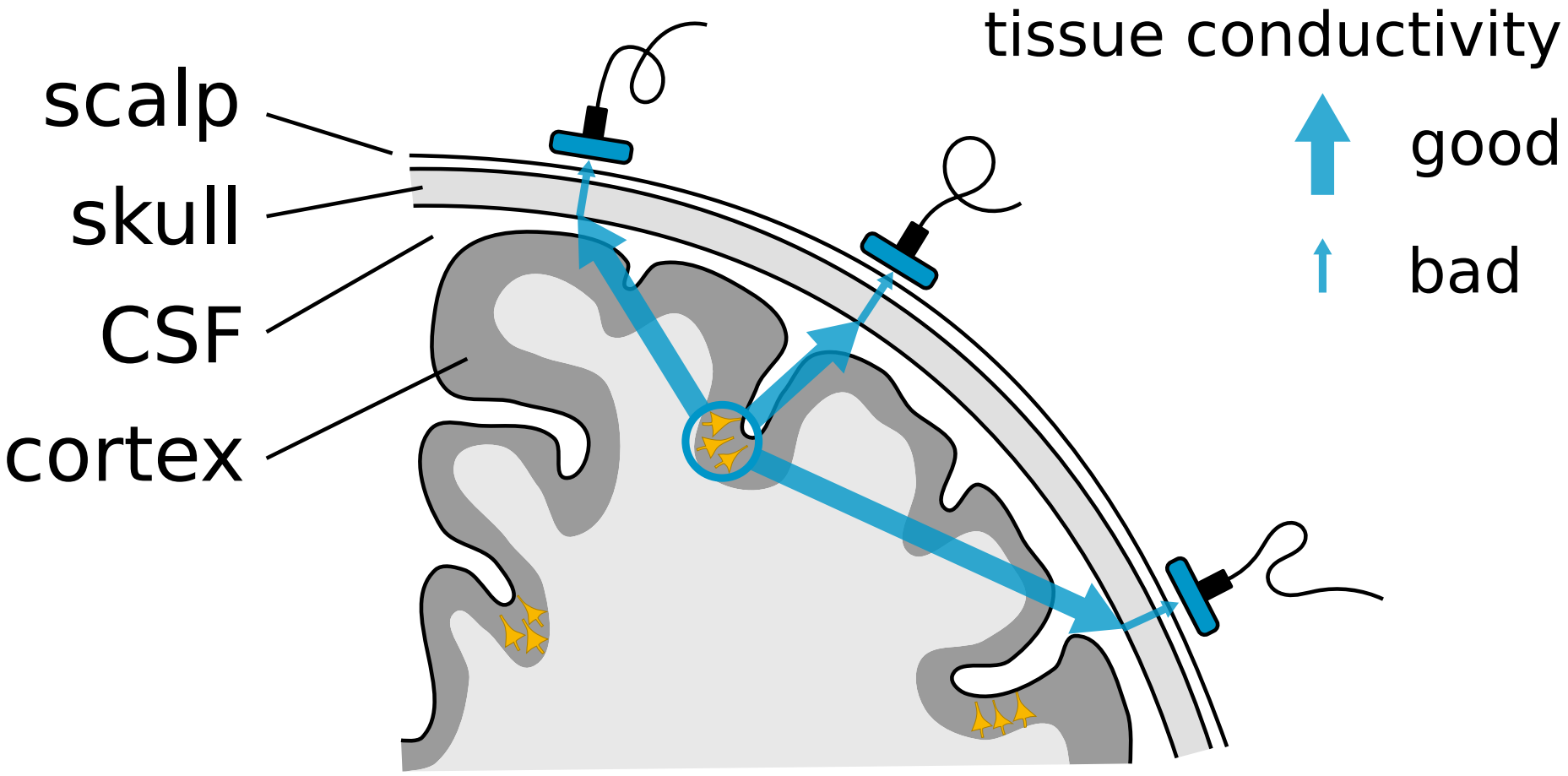
While it is certainly useful to have a measure to select the most discriminative (univariate) feature, we will see that and why it is much more beneficial to use multivariate features.

To provide background knowledge, we will make detour to the generation of EEG signals.

Generation of EEG Signals



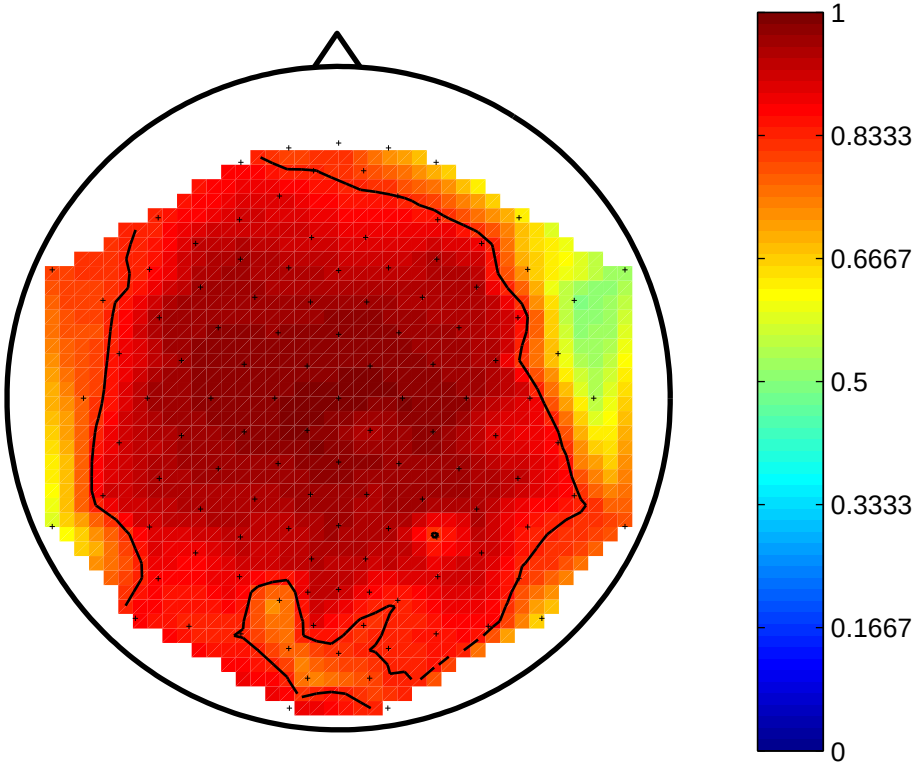
Volume Conduction in EEG



The signal arrives with almost equal intensity at different scalp locations due to the different tissue conductivities.

Mind Spatial Smearing!

- Raw EEG scalp potentials are known to be associated with a large spatial scale owing to volume conduction.
- In this typical example data set, most of the channels are highly correlated:



The map shows the correlation coefficient of each channel with channel Cz in the center.

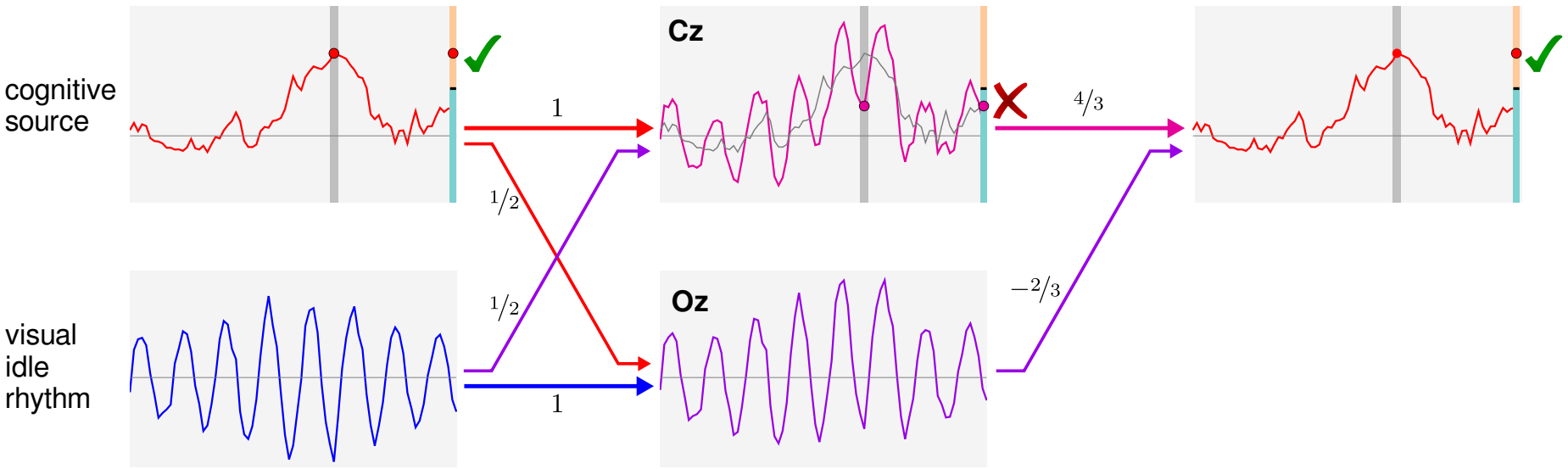
From Uni- to Multivariate Features

We have seen that a discrimination of ERPs to target and nontarget stimuli is possible based on a univariate feature.

For improved classification of EEG single-trials, we need to accumulate more information in the features.

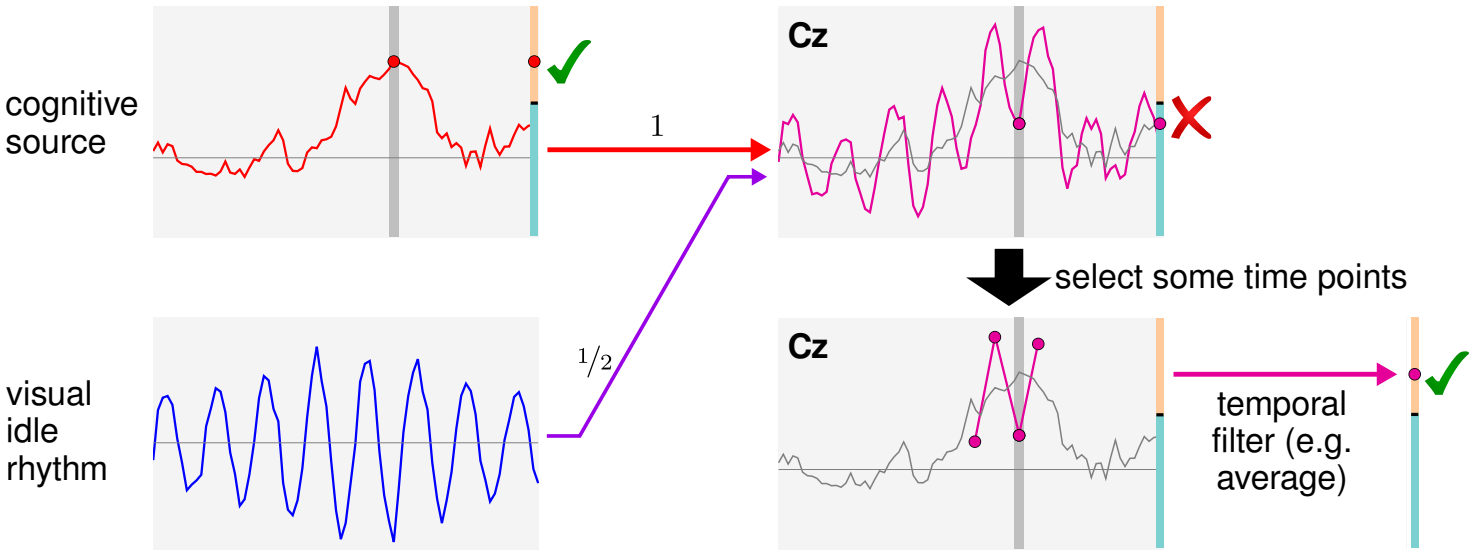
- sample ERP signals at *multiple* time points/intervals
  $\rightarrow$ **temporal feature**
- join signals from *multiple* channels
  $\rightarrow$ **spatial feature**
- do both things
  $\rightarrow$ **spatio-temporal feature**

The Virtue of Multivariate Spatial Features

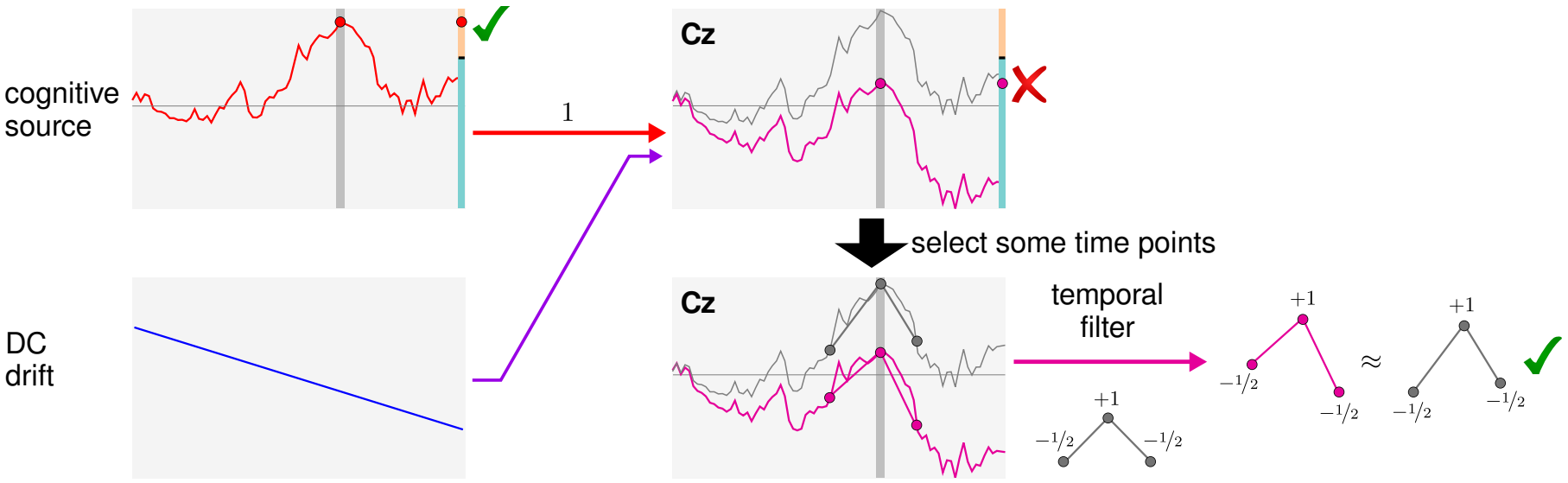


- Here, $\mathbf{w} = [4/3 \quad -2/3]^\top$ is a simple spatial filter.

The Virtue of Multivariate Temporal Features

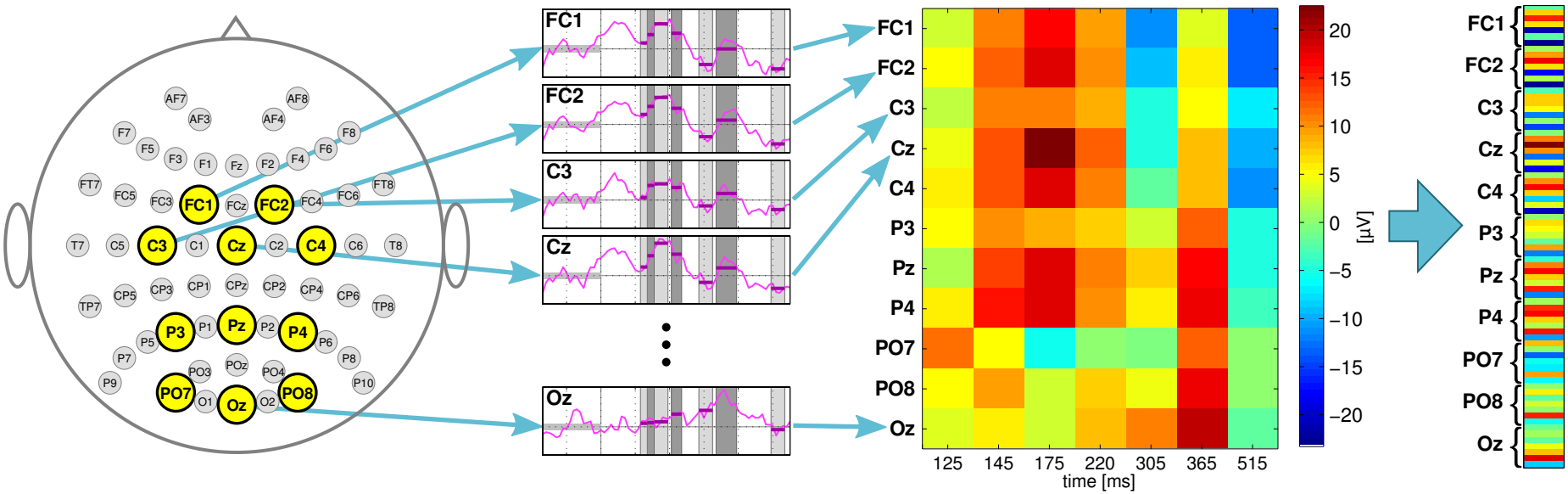


The Virtue of Multivariate Temporal Features



Extraction of Spatio-Temporal Features

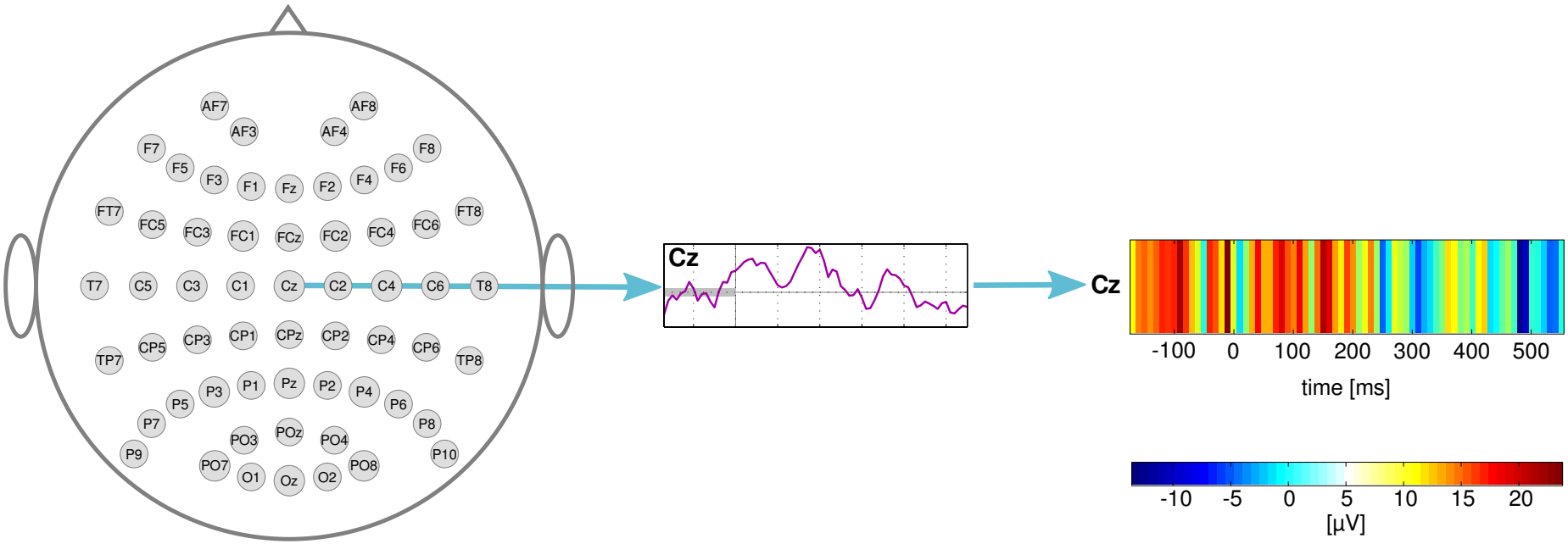
Given a set of channels and a set of time intervals, we define **spatio-temporal features** like this:



The average within each interval is calculated. These values (per interval) are concatenated for all selected channels.

Extraction of Temporal Features

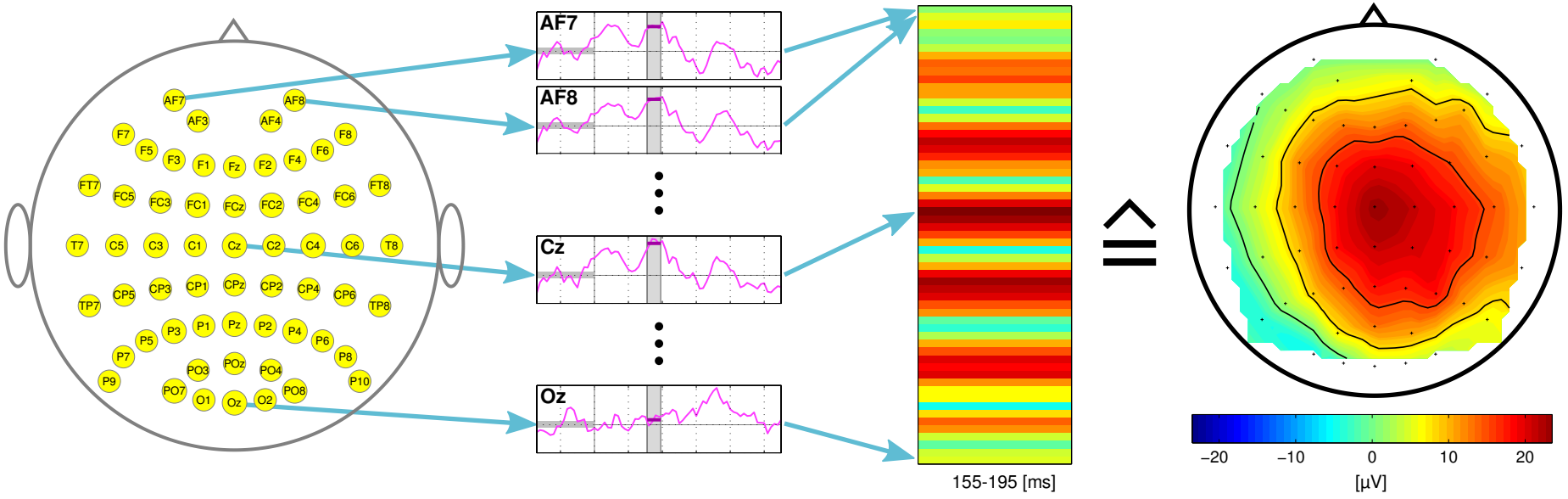
A **temporal feature** is defined by sample points in a time interval and one channel:



The dimensionality of the feature vector coincides with the number of sample points in the interval. Alternatively, several time intervals can be used, and the feature consists of the mean values within each interval.

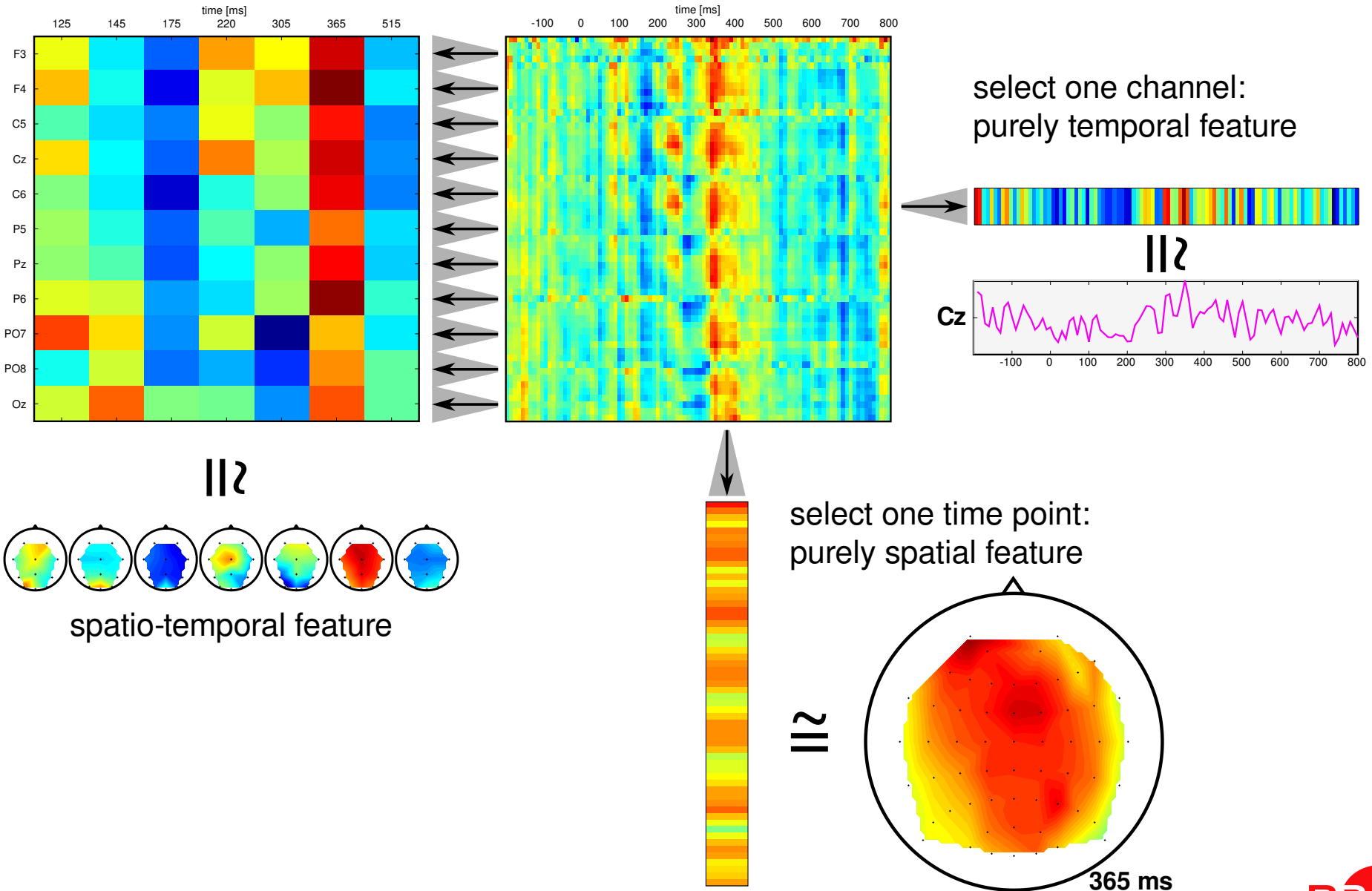
Extraction of Spatial Features

A **spatial feature** is defined by a set of channels and one time interval:



The dimensionality of the feature vector coincides with the number of (chosen) channels. The values of the feature vector are the (average) amplitude in the given time interval of the respective channel.

Overview of Multivariate ERP Features



Multivariate ERP Model

The ERP model is based on the split into time locked activity $\mathbf{p}(t)$ (assumed constant over trials) and non time locked activity $\mathbf{r}(t)$:

$$\mathbf{x}_k(t) = \mathbf{p}(t) + \mathbf{r}_k(t) \qquad \text{for trials } k = 1, \ldots, K$$

In the probabilistic view, for a fixed time t_0 both, $\mathbf{x}_k(t_0)$ and $\mathbf{r}_k(t_0)$ are (vector valued) random variables over the trials k, from which we observed K-many draws. The noise $\mathbf{r}(t_0)$ is iid distributed, and it is assumed to be Gaussian, say $\mathcal{N}(0, \mathbf{\Sigma_r})$.

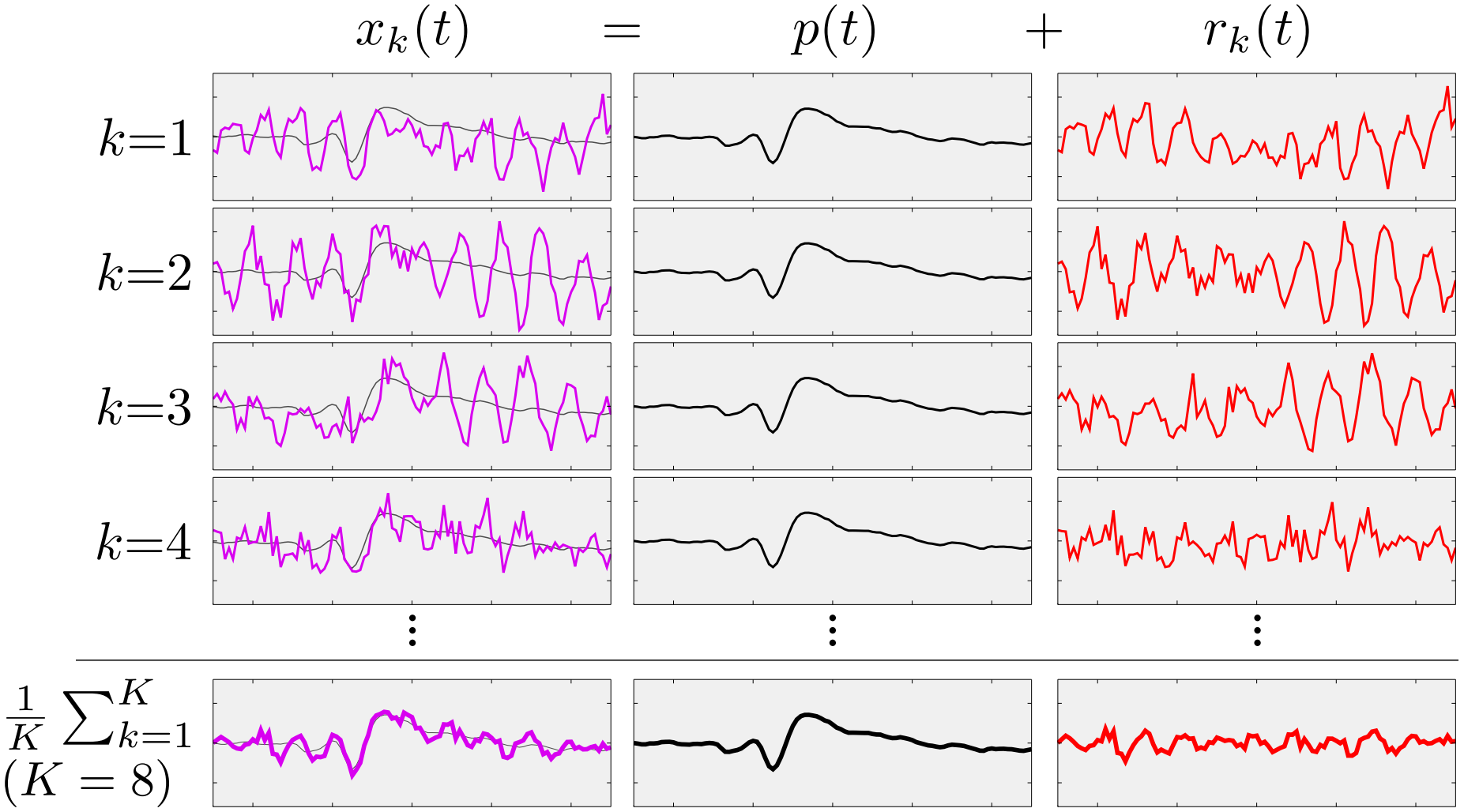
These assumptions lead to the following distribution of $\mathbf{x}_\cdot(t_0)$ across trials:

- $\boldsymbol{\mu}_\mathbf{x} = \mathbb{E}\langle \mathbf{x}_k(t_0) \rangle_k = \mathbf{p}(t_0)$
- $\mathbf{\Sigma_x} = \mathrm{Cov}\, \langle \mathbf{x}_k(t_0) \rangle_k = \mathrm{Cov}\, \langle \mathbf{r}_k(t_0) \rangle_k = \mathbf{\Sigma_r}$

This means that the distribution of $\langle \mathbf{x}_k(t_0) \rangle_k$ is $\mathcal{N}(\mathbf{p}(t_0), \mathbf{\Sigma_r})$.

Averaging across Trials

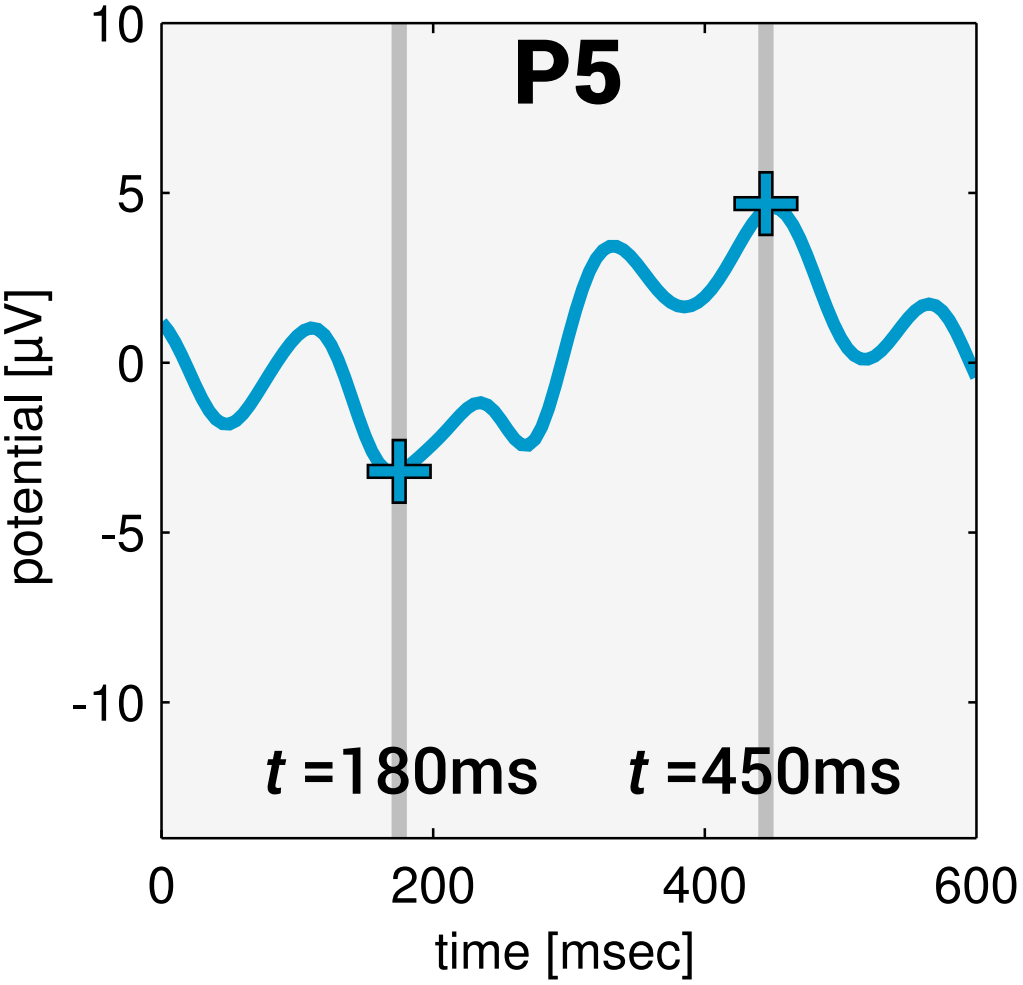
For one channel, the situation looks as follows:



Let's Start Simple: a 2D Feature

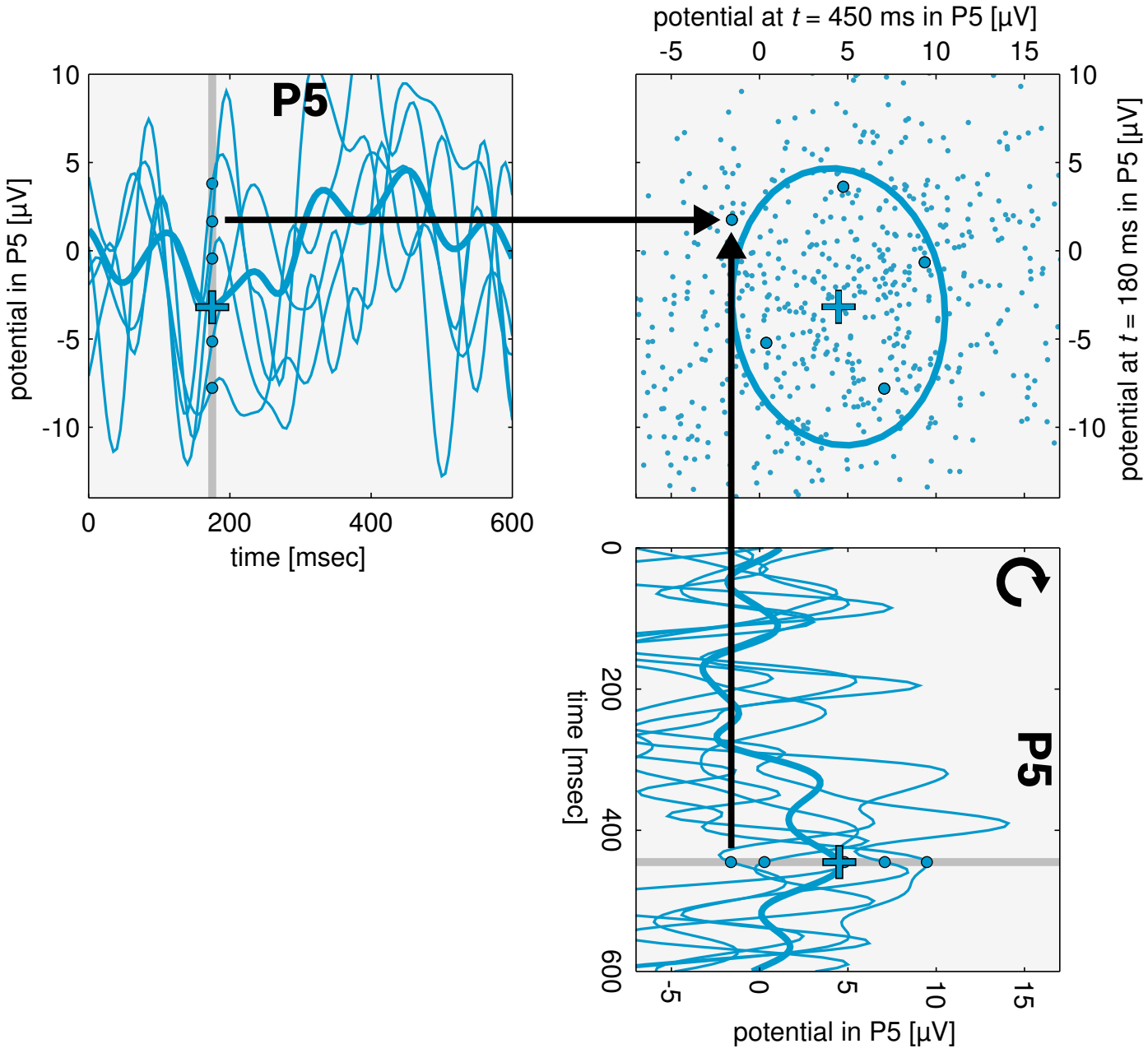
In view of classification, we are concerned with distributions.

A simple 2D feature: potential at 180ms and at 450ms in channel P5.



$$\mathbf{x} = \begin{bmatrix} x^{\mathsf{P5}}(180ms) \\ x^{\mathsf{P5}}(450ms) \end{bmatrix}$$

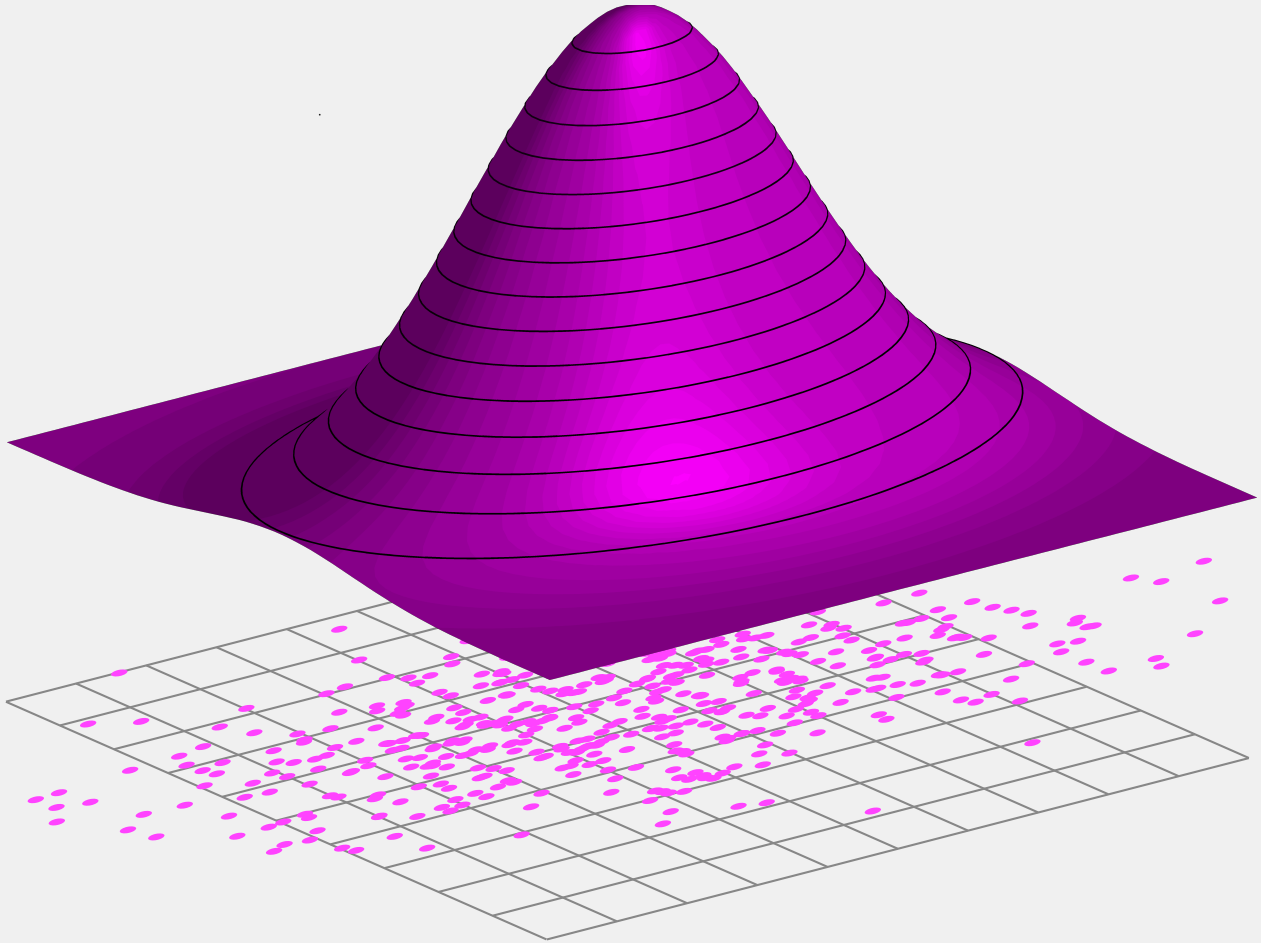
Visualizing 2D Features at Scatter Plot

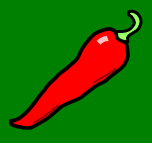


Multivariate Gaussian Distributions

(a)

$$g(\mathbf{x}) = \frac{1}{2\pi\sqrt{det(\mathbf{\Sigma})}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})^{\top}\right)$$



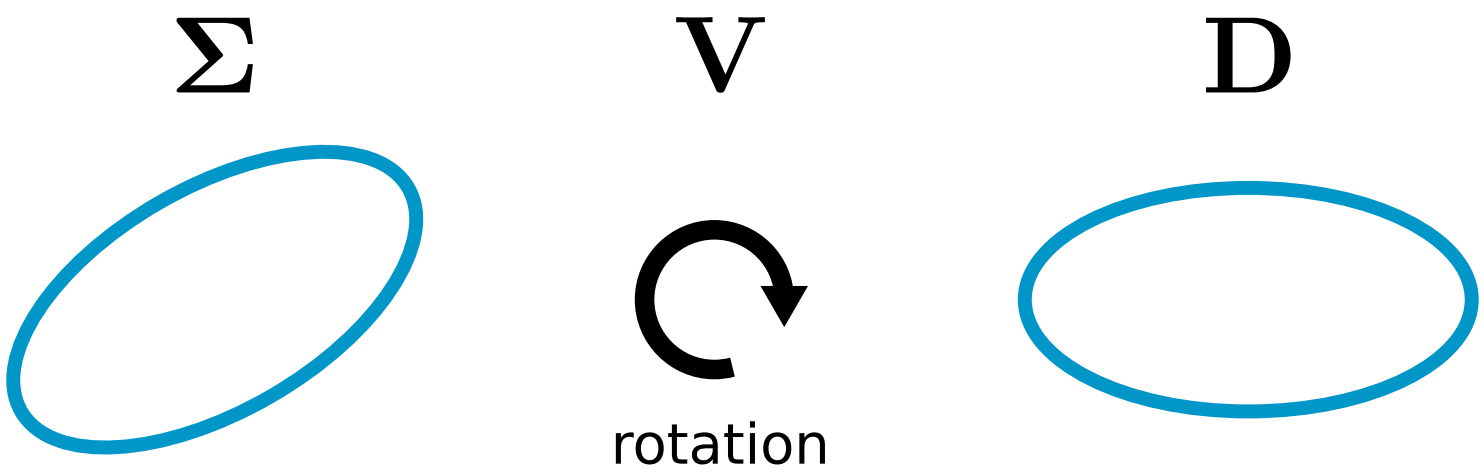


Eigenvalue Decomposition (EVD)

Given $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$ symmetric and pos. definite, there exists an orthonormal matrix $\mathbf{V} \in \mathcal{O}(m)$ of **Eigenvectors** and a diagonal matrix $\mathbf{D} \in \text{Diag}(m)$ of **Eigenvalues**, such that

$$\mathbf{\Sigma} = \mathbf{V}\mathbf{D}\mathbf{V}^\top$$

In our case, $\mathbf{\Sigma}$ is the covariance matrix of EEG signals $\mathbf{X} \in \mathbb{R}^{m \times T}$.



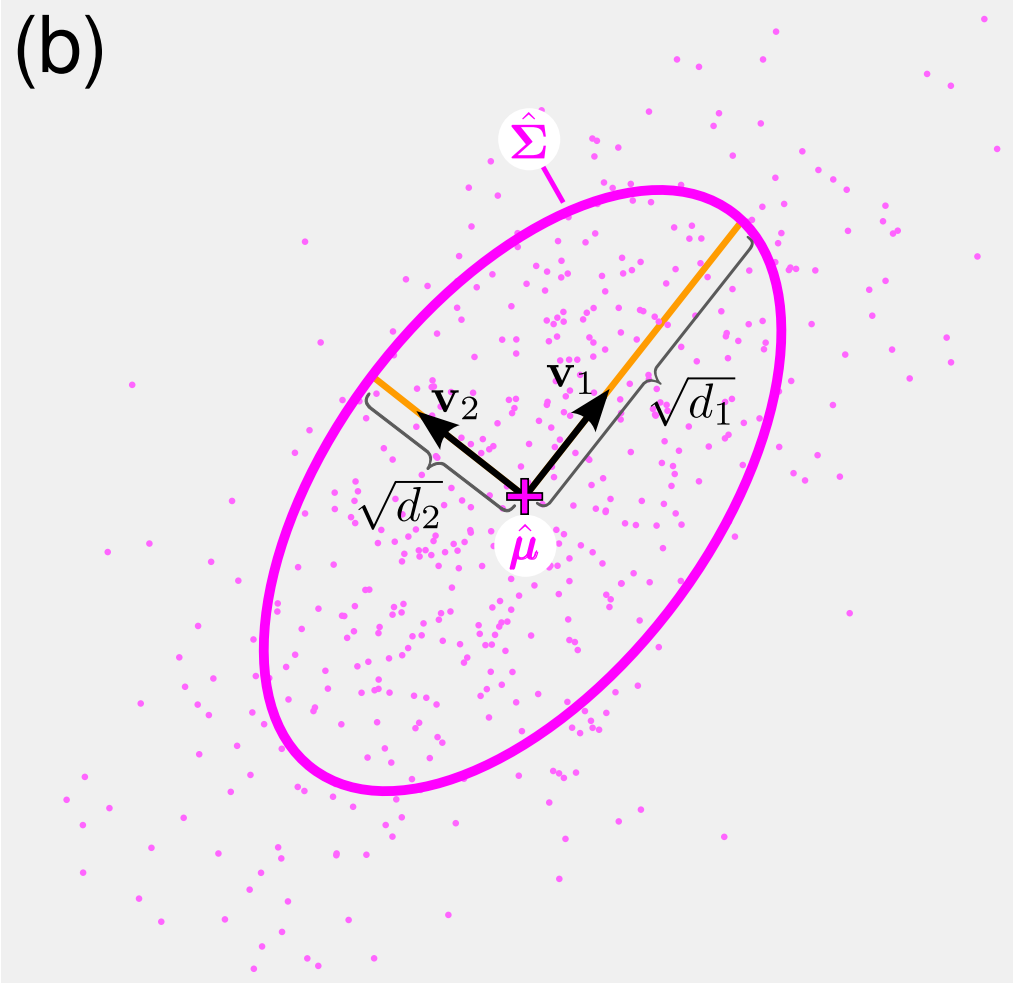
- Eigenvalues are the variance of $\mathbf{X}$ in direction of corresponding Eigenvectors.

Characterization of Gaussian Distributions

Assume samples $\mathbf{x}_1, \ldots, \mathbf{x}_K \in \mathbb{R}^p$ are modeled as $\mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$.

Eigenvalue decomposition of the covariance:

$$\hat{\boldsymbol{\Sigma}} = \mathbf{V}\mathbf{D}\mathbf{V}^\top, \quad \text{with orthonormal } \mathbf{V} \text{ and diagonal } \mathbf{D}.$$

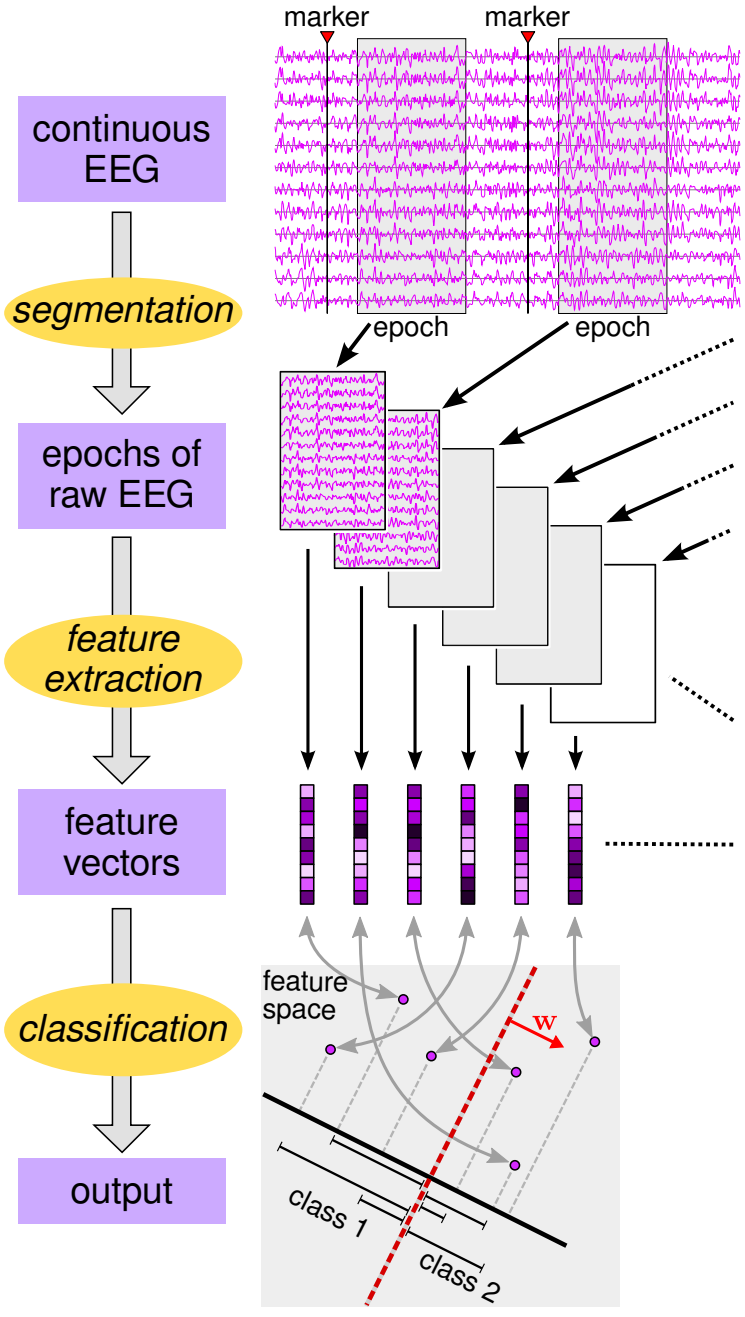


- Eigenvectors are columns of $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_p]$.
- Eigenvalues are diagonal elements d_i of $\mathbf{D}$.
- $\sqrt{d_i} = \mathrm{std}(\mathbf{v}_i^\top \mathbf{X})$
- In $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ typically $\boldsymbol{\mu}$ is considered to be the ideal true value and $\boldsymbol{\Sigma}$ noise.
- The vector of Eigenvalues is called *Eigenvalue spectrum*

Part III

Classification of ERP Features

Toward Classification for BCIs



A **classifier** is a function mapping samples of the **feature space** $\mathbb{R}^D$ to **labels**, e.g. for a binary classifier:

$$f : \mathbb{R}^D \to \{1, 2\}$$

(In our example, class labels 1 and 2 correspond to *targets* and *nontargets*.)

Often samples are mapped to $\mathbb{R}$ first, and then the label is decided based on a threshold.

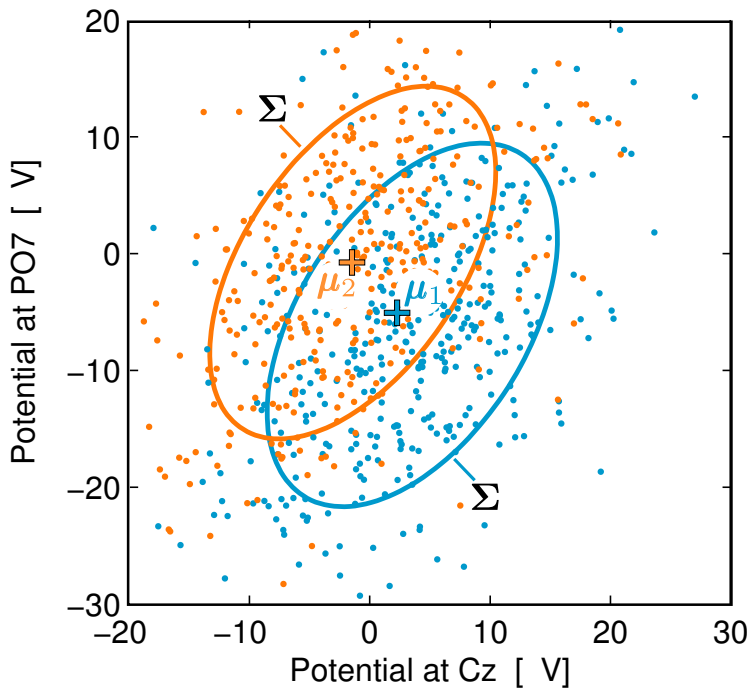
Classifiers are based on the class distributions of the samples.

Distributions of ERP Features

For classification, we have to consider the distribution of the features. According to our model (ERPs are constant across trials):

$$\mathbf{x}_k(t) = \mathbf{p}_1(t) + \mathbf{r}_k(t) \qquad \text{for trials } k \text{ of condition 1}$$
$$\mathbf{x}_k(t) = \mathbf{p}_2(t) + \mathbf{r}_k(t) \qquad \text{for trials } k \text{ of condition 2}$$

with Gaussian noise: $\mathbf{r}_\cdot(t) \sim \mathcal{N}(0, \mathbf{\Sigma})$.



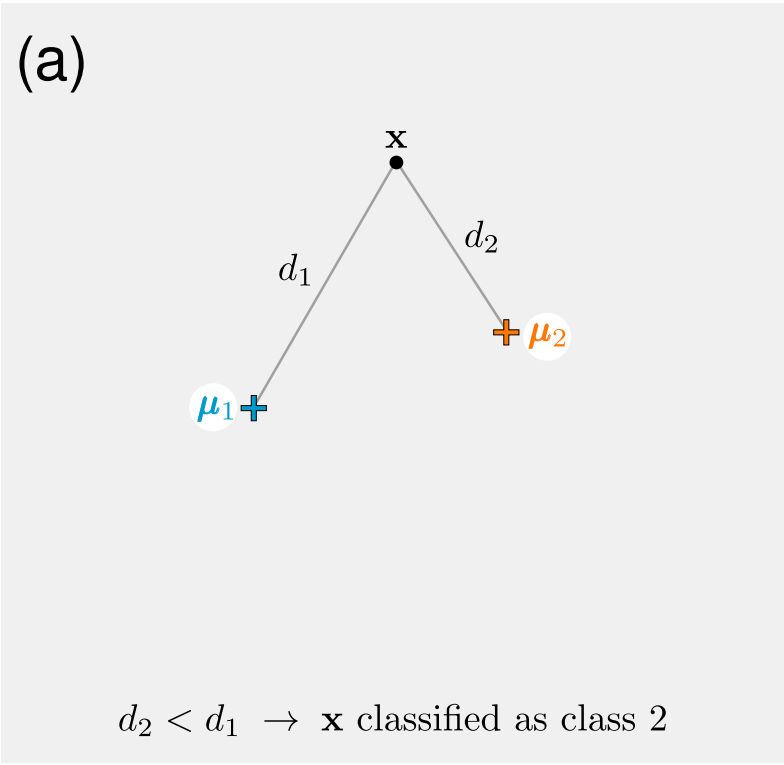
For features of ERP data:

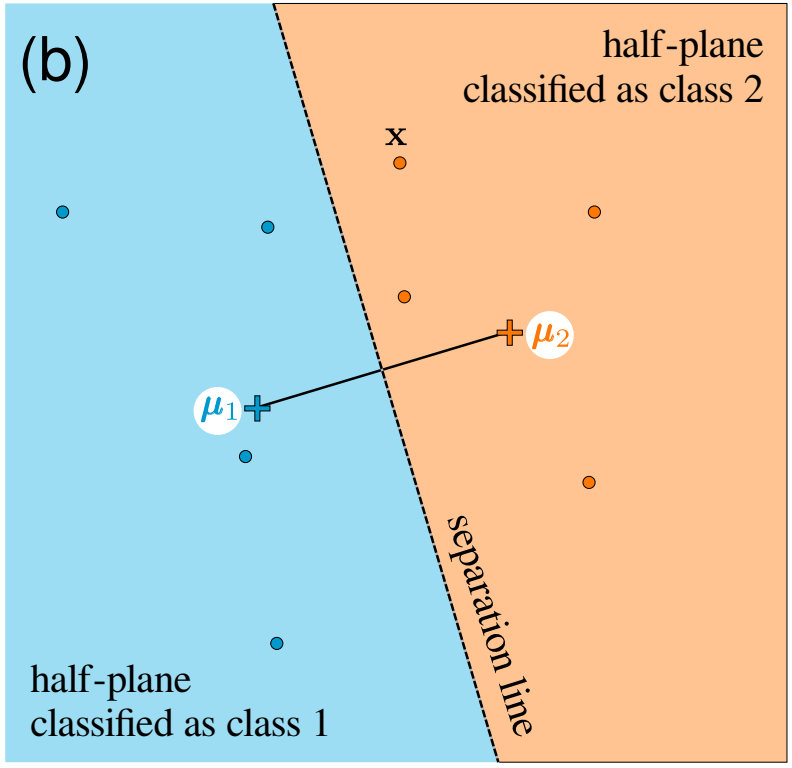
- μ_1: ERP of condition 1
- μ_2: ERP of condition 2
- $\mathbf{\Sigma}$: noise: non-phase-locked activity (independent of condition)

[Blankertz et al, NeuroImage 2011]

Nearest Centroid Classifier (NCC)

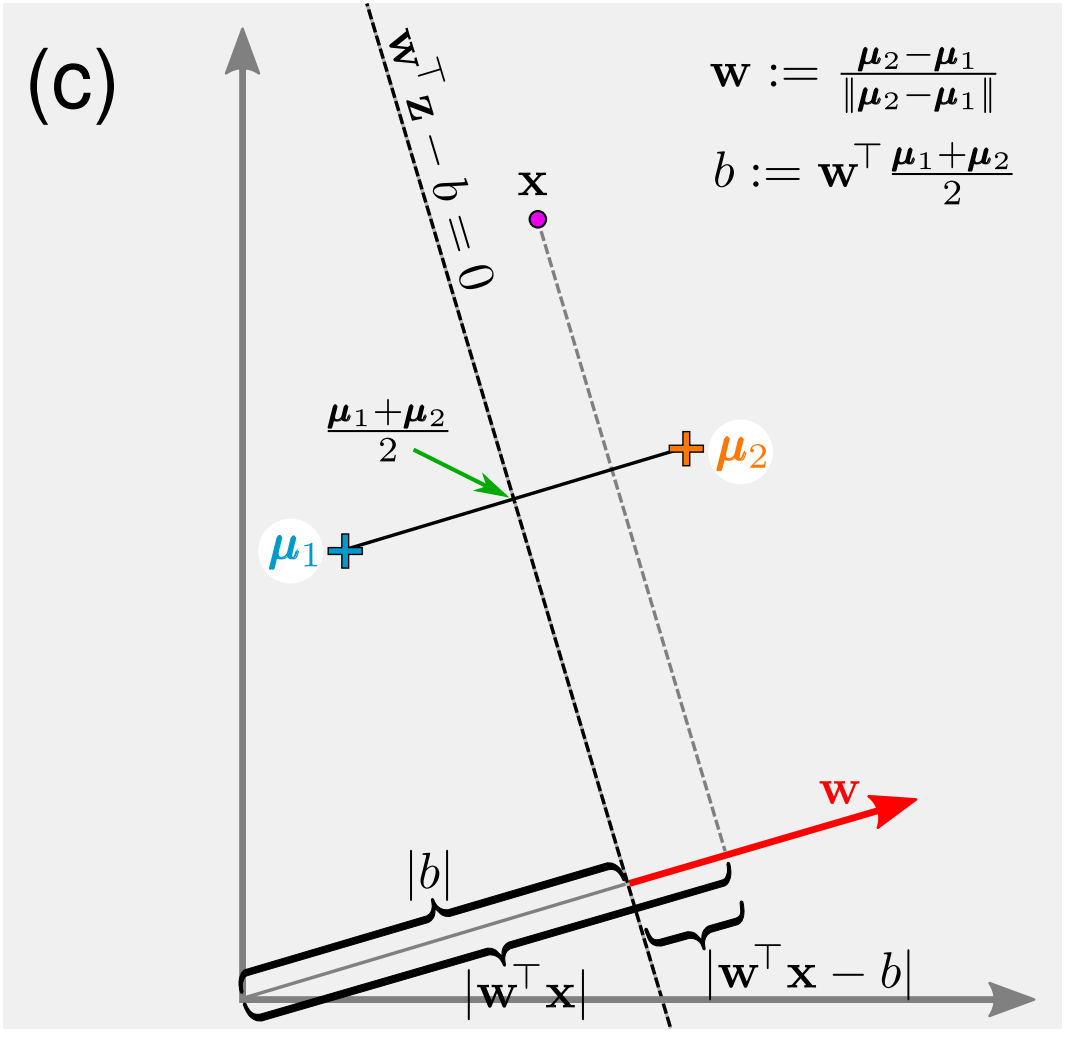
(a) Let us assume a simple setting of a classification problem with little information: Only the means (or centroids) $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ of the two distributions are known.





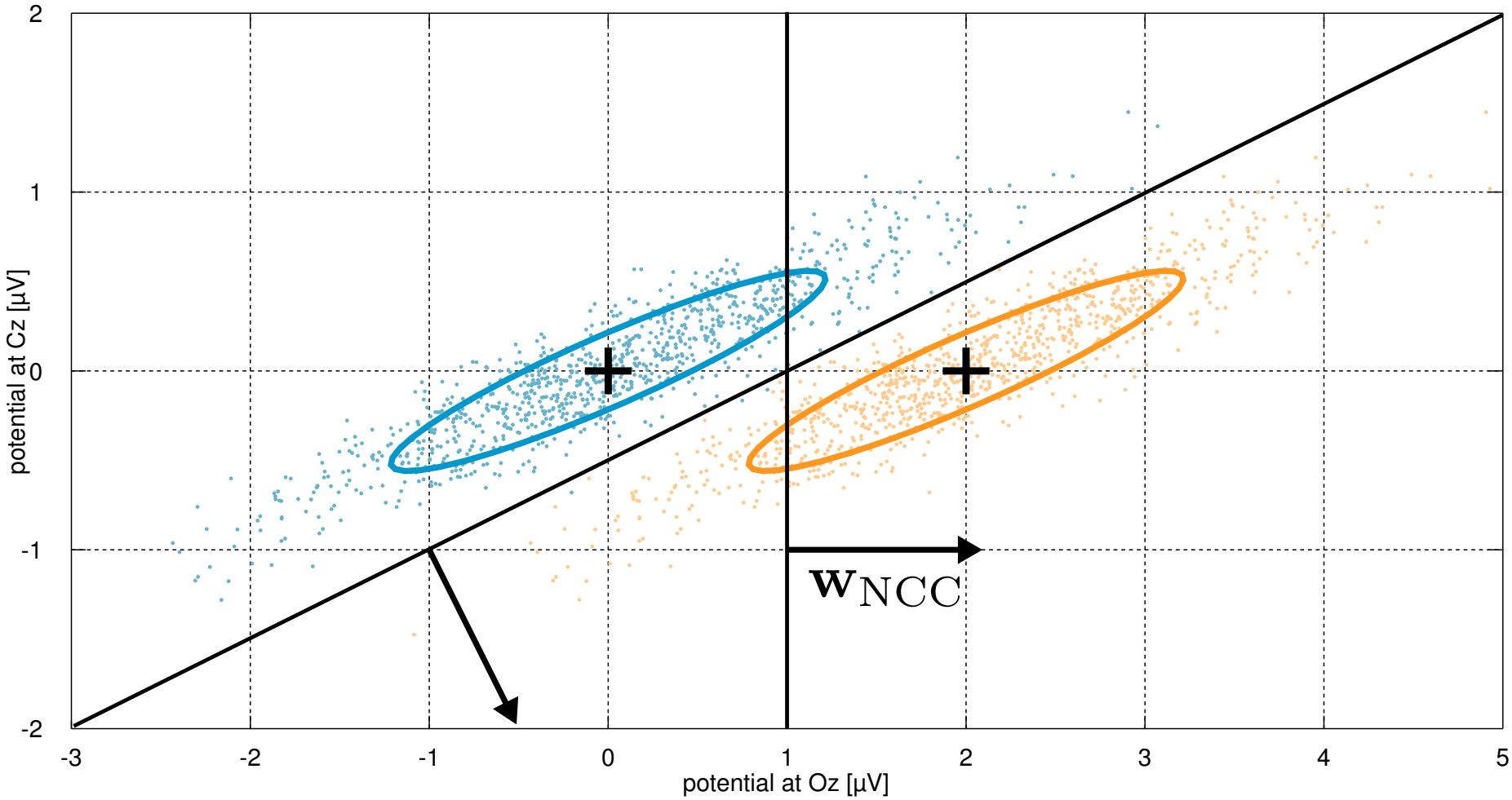
(b) This leads to a linear separation of the space with the separation line (or hyperplane in higher dimensions) intersecting perpendicularly the line connecting the centroids in the middle.

Formalization of Separating Hyperplanes



$$\mathbf{x} \quad \mapsto \quad \mathbf{w}^\top \mathbf{x} - b \quad \mapsto \quad \begin{cases} \text{class 1} & \text{if } \mathbf{w}^\top \mathbf{x} - b >= 0 \\ \text{class 2} & \text{if } \mathbf{w}^\top \mathbf{x} - b < 0 \end{cases}$$

Can We Expect NCC to Perform Well for ERP Features?



Linear Discriminant Analysis (LDA)

Using probability theory, one can derive from the following three assumptions the optimal classifier for the given class distributions.

> *Optimality means that the classifier has the minimum risk of misclassification for new samples that are drawn from these class distributions.*

1. Features of each class are Gaussian distributed.
2. Gaussians of all classes have the same covariance matrix.
3. True class distributions are known.

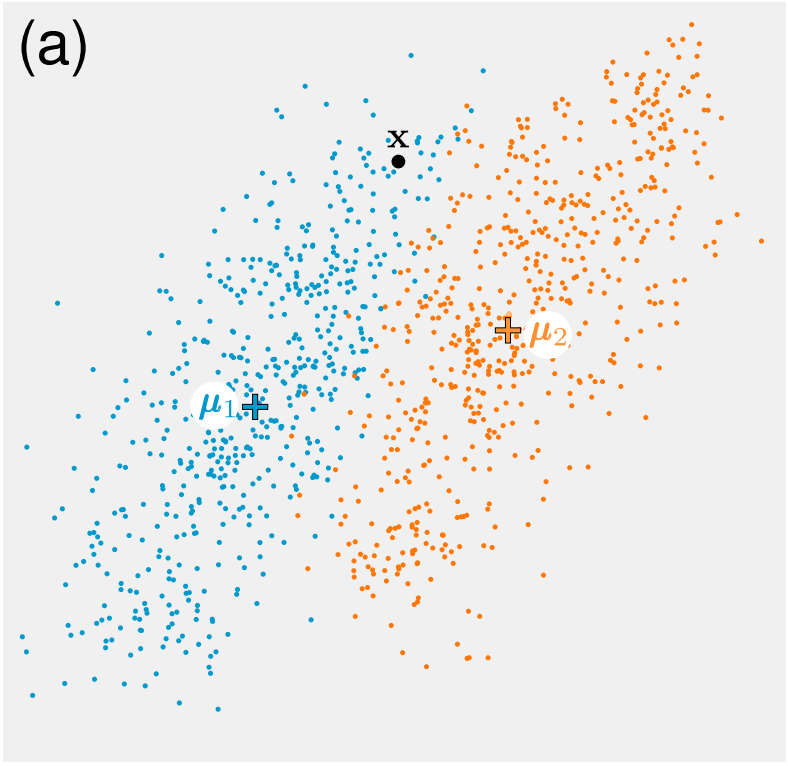
This optimal classifier is called *Linear Discriminant Analysis* (LDA) and it can be formalized in the following way: Given two Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ and $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$, LDA is defined by the normal vector

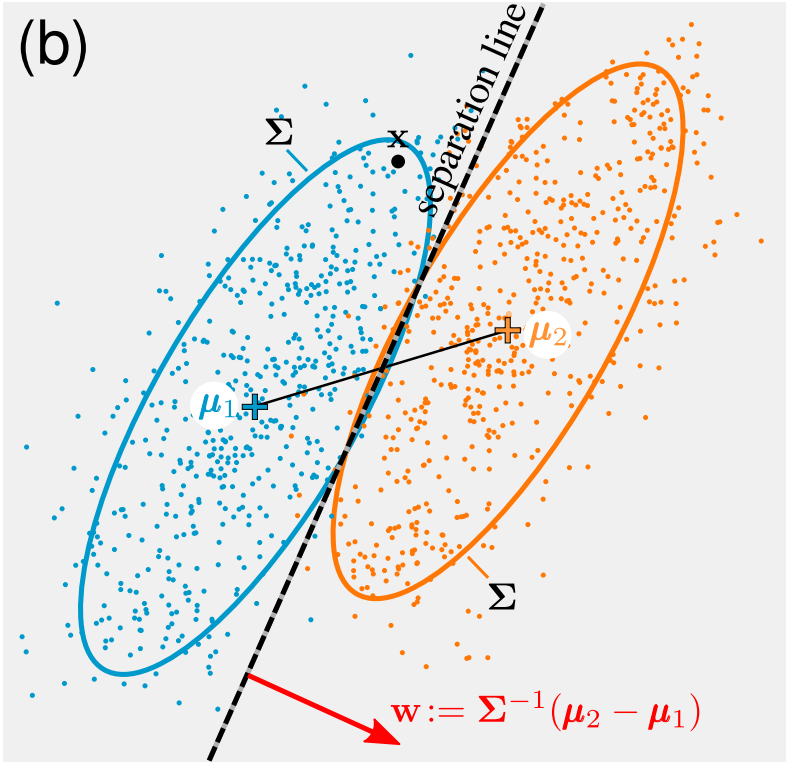
$$\mathbf{w} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \quad \text{and bias} \quad b = \mathbf{w}^\top(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)/2. \tag{1}$$

First we will look at how the LDA classification looks like and later discuss the assumptions.

Linear Disciminant Analysis

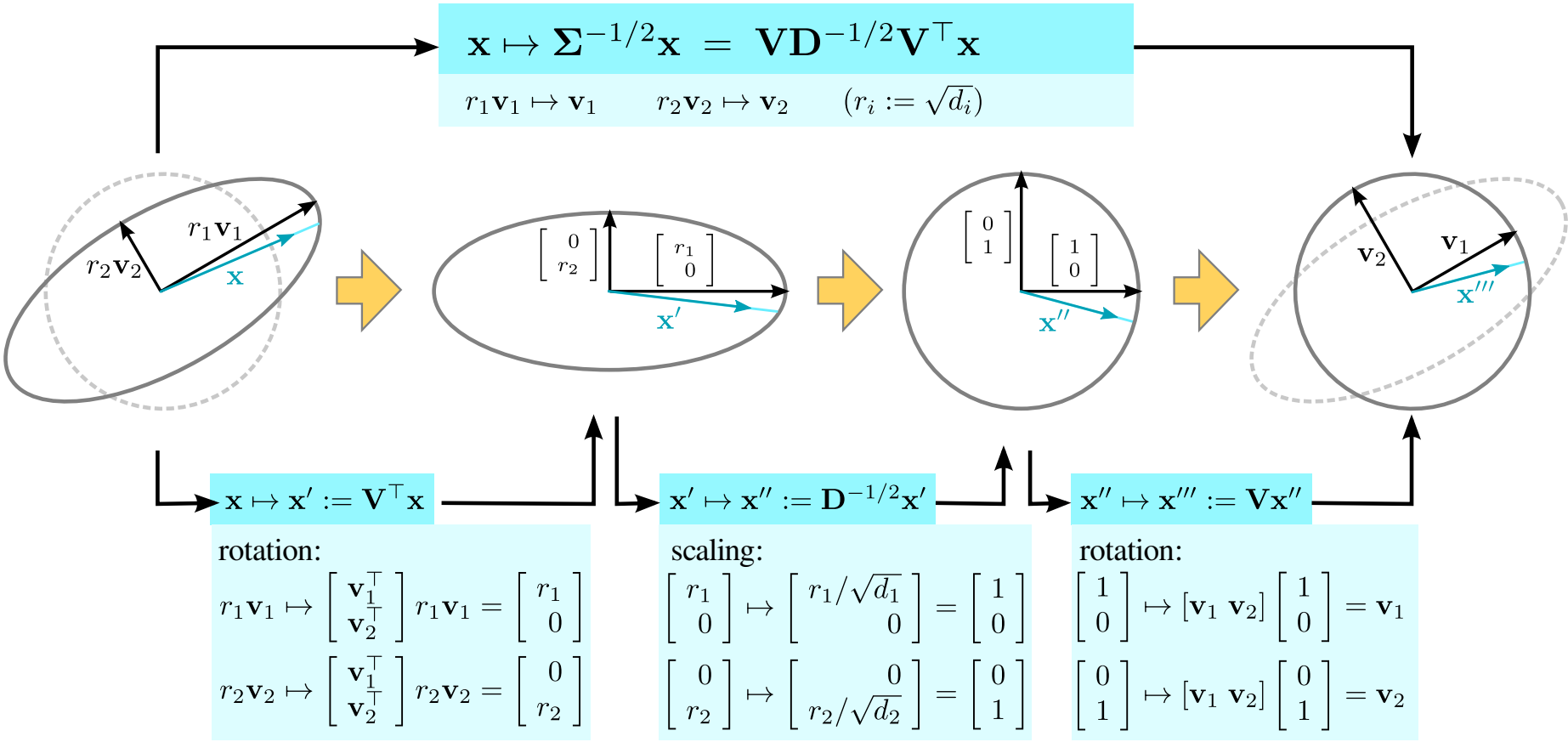
(a) Means as in the NCC example, but specific distributions are shown.





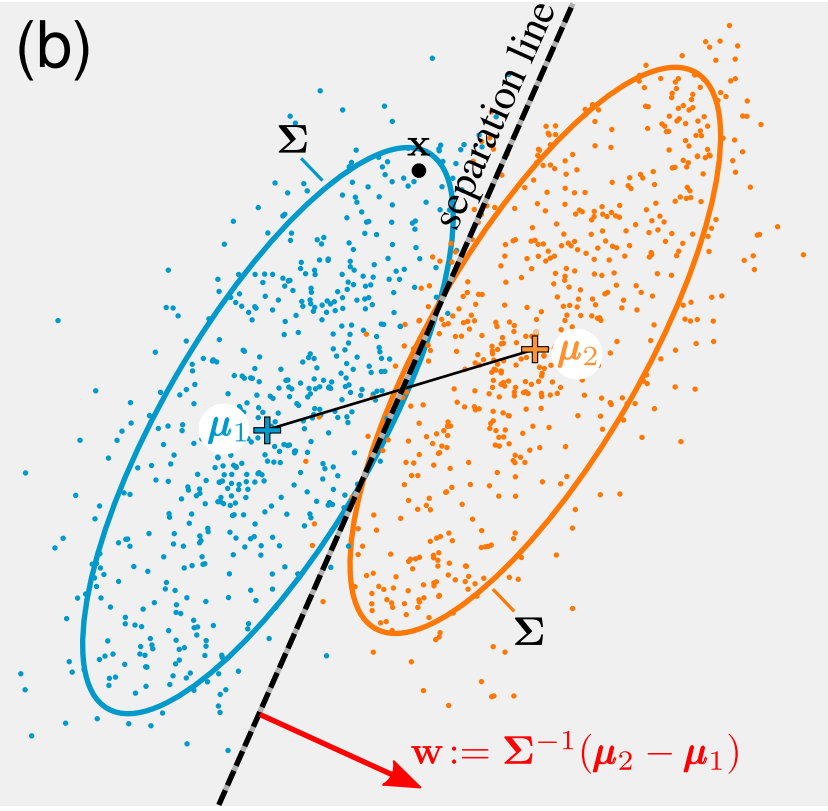
(b) In Linear Discriminant Analysis, a common covariance matrix for both classes is estimated, which describes the (class-independent) noise. Note, that $\mathbf{x}$ is classified here differently with LDA than with NCC.

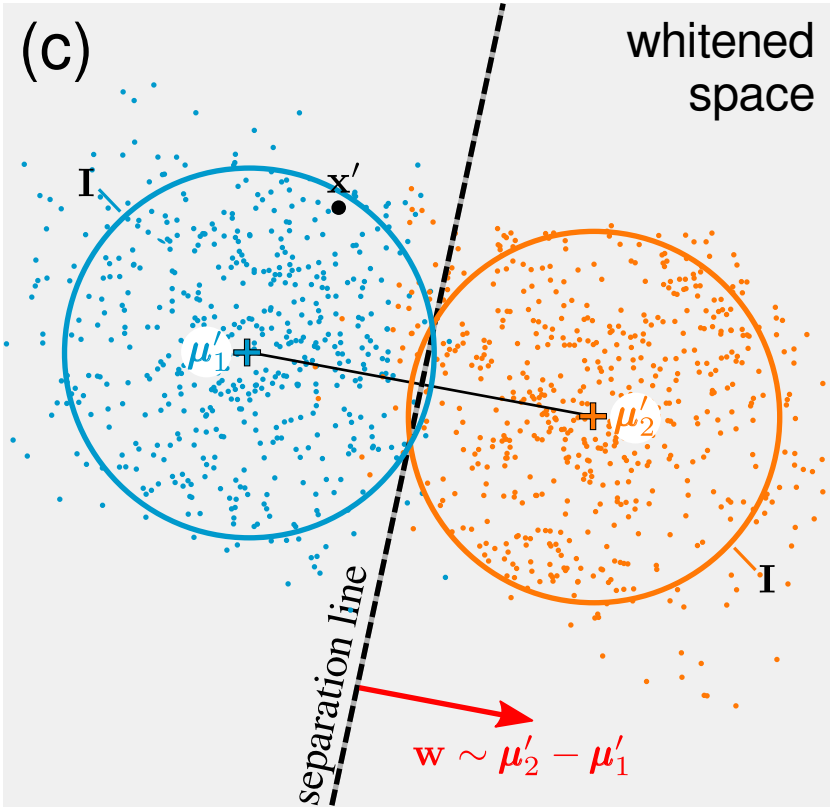
Interlude: Illustration of Whitening Transform



The whitening transform maps the space such that a Gaussian distribution with the given covariance matrix becomes a standard normal distribution, i.e., the variance in all directions is 1. It maps the ellipsoid given by the standard isodensity line of the Gaussian distribution to the unit sphere.

Correspondence between NCC and LDA





Classification with LDA in the original space is equivalent to classification with NCC in the whitened space.

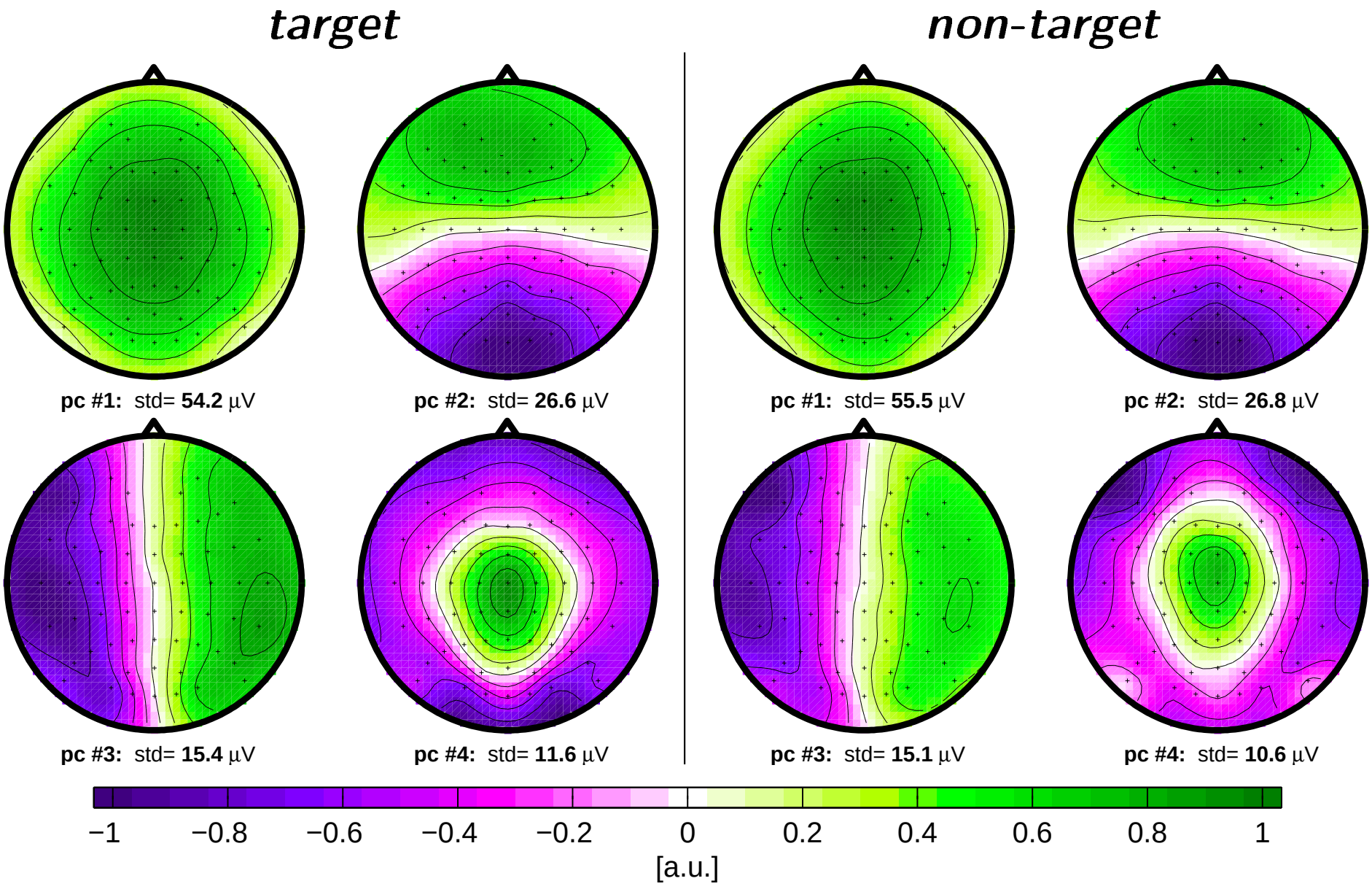
Linear Discriminant Analysis – Assumptions for Optimality

Now, we come back to the assumptions which are required to warrant optimality of the LDA classifier:

1. Features of each class are Gaussian distributed.
2. Gaussians of all classes have the same covariance matrix.
3. True class distributions are known.

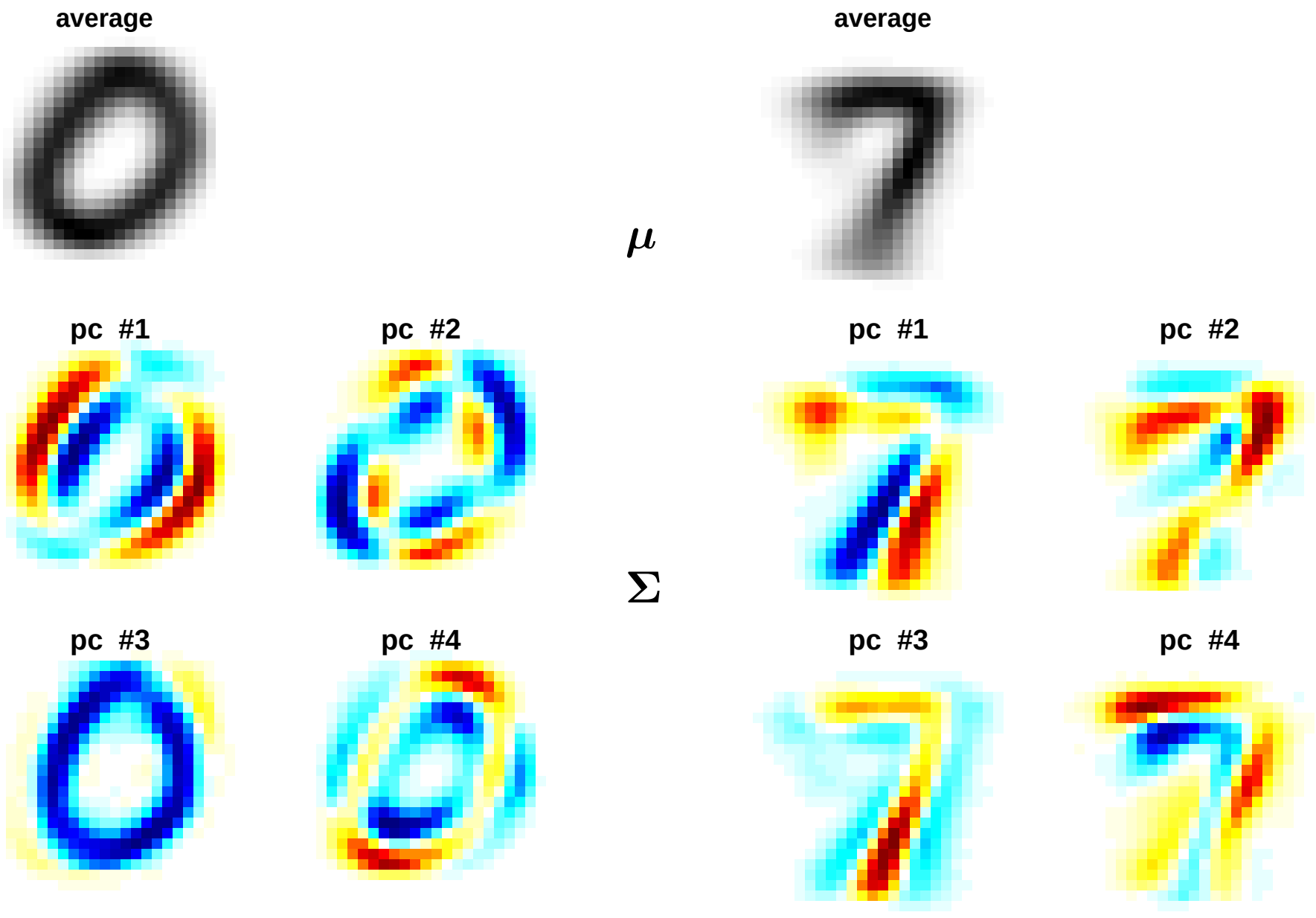
The first two assumptions have been discussed already. Further empirical results for (2) are presented on the next slides, and assumption (3) will be discussed later in this lecture.

Covariance Matrices of ERP Features



➤ Covariances of both classes look very similar.

For Comparison: Covariances in Handwritten Digits



➤ Here, covariances of both classes **do not** look similar.

Validation of Classification Procedures

To validate the performance of a classifier, one needs to have a

- **training set** on which all parameters of the model are estimated (weights of the classifier; selection of features etc.), and a
- **validation set** on which the performance is calculated.

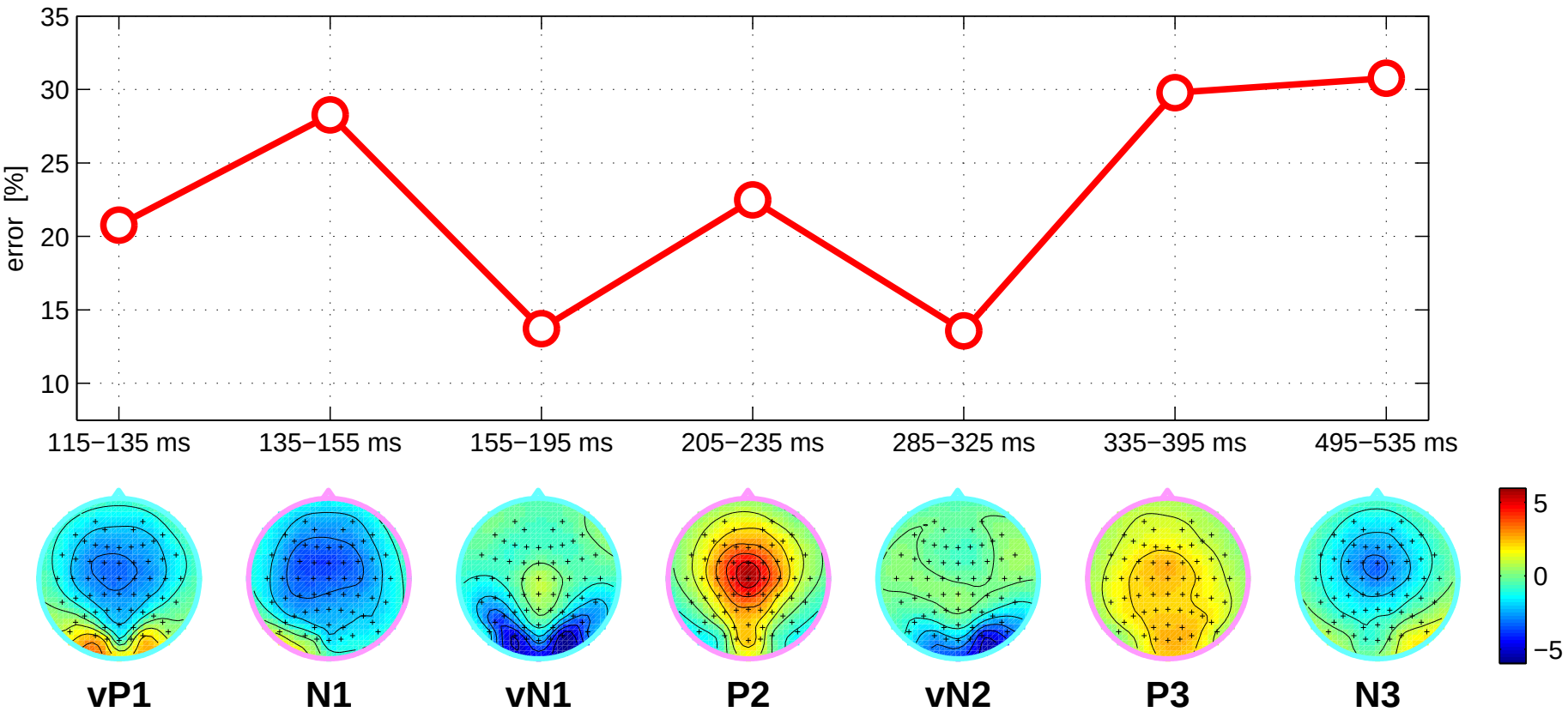
These sets of samples have to be disjoint and **INDEPENDENT**.

To that end, one can use a fixed training and validation set (e.g., first half / second half) or cross-validation.

See [Lemm et al, NeuroImage 2011] for details on validation.

Results of Classifying Spatial Features

Classifying on spatial features for various time intervals results in error rates between 14% and 31% in this example data set (visual speller):



Classification of Spatio-Temporal Features

Advancing from temporal or spatial features to *spatio-temporal* features means increasing the information.

Accordingly, a better classification performance is to be expected.

But in our example data set, the classification error **increases** from

- **14%** for the spatial feature at the best interval to
- **25%** for spatio-temporal features

when classifying with LDA.

Overfitting of LDA

When LDA was applied to high-dimensional (spatio-temporal) features, the performance broke down (result worse than on sub-features).

Given the optimality theorem, this should not happen, right?

So far, we did not discuss the third assumption:
The true distributions are known.

- This assumptions is *always* violated in non-artificial problems.
- Distribution parameters have to be estimated from given data.
- Estimated (empirical) distribution parameters necessarily deviate from the true ones.
- How much this deviation deteriorates performance is variable.

Bias in Estimating Covariance Matrices

For LDA we need estimates for the distribution parameters:

- $\hat{\boldsymbol{\mu}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k$ **empirical mean**
- $\hat{\boldsymbol{\Sigma}} = \frac{1}{K-1}\sum_{k=1}^{K}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^\top$ **emp. covariance matrix**

But, if the number of samples K is not large relative to the dimension d ($\mathbf{x} \in \mathbb{R}^d$), the estimation, in particular $\hat{\boldsymbol{\Sigma}}$, is error-prone.

This may affect classification with LDA badly.

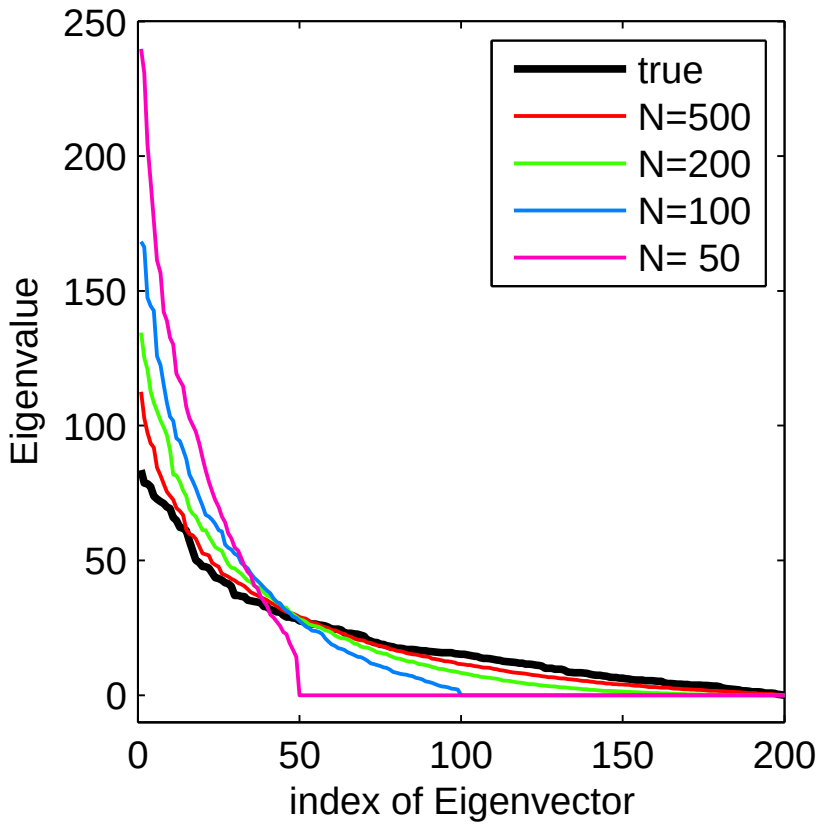
There is a systematical bias in the empirical covariance matrix:

- Large Eigenvalues of $\hat{\boldsymbol{\Sigma}}$ are too large and
- Small Eigenvalues of $\hat{\boldsymbol{\Sigma}}$ are too small

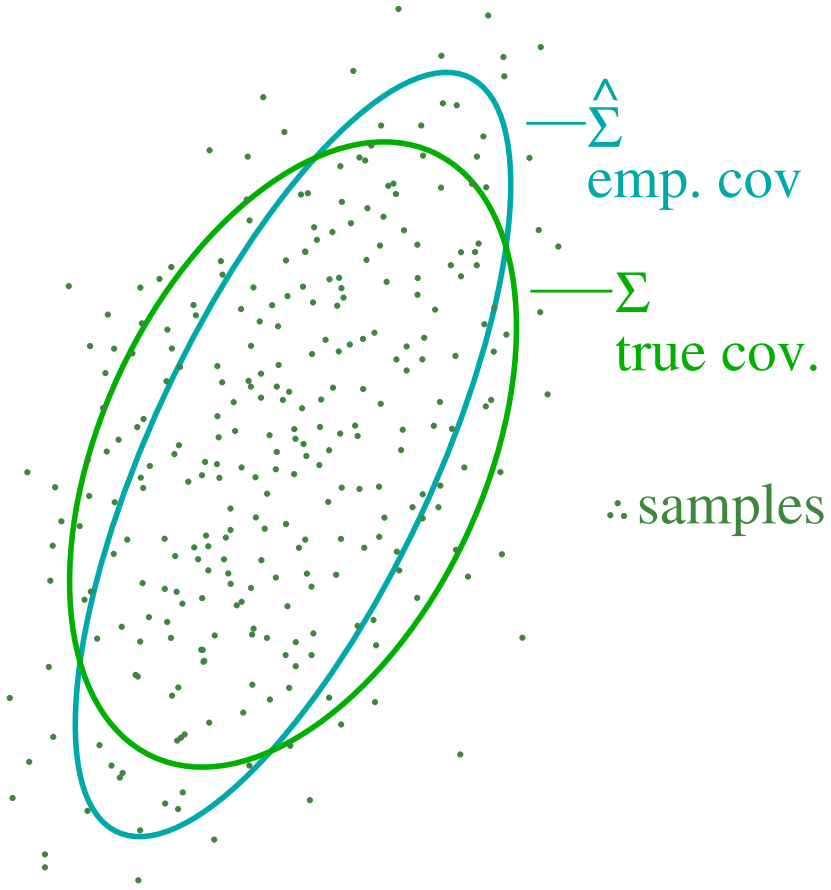
compared to those of $\boldsymbol{\Sigma}$ assuming $\mathbf{x}_1, \ldots, \mathbf{x}_K \in \mathbb{R}^d$ are drawn from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Bias in Estimating Covariances (2)

Simulation for $d = 200$:



Cartoon in 2D:

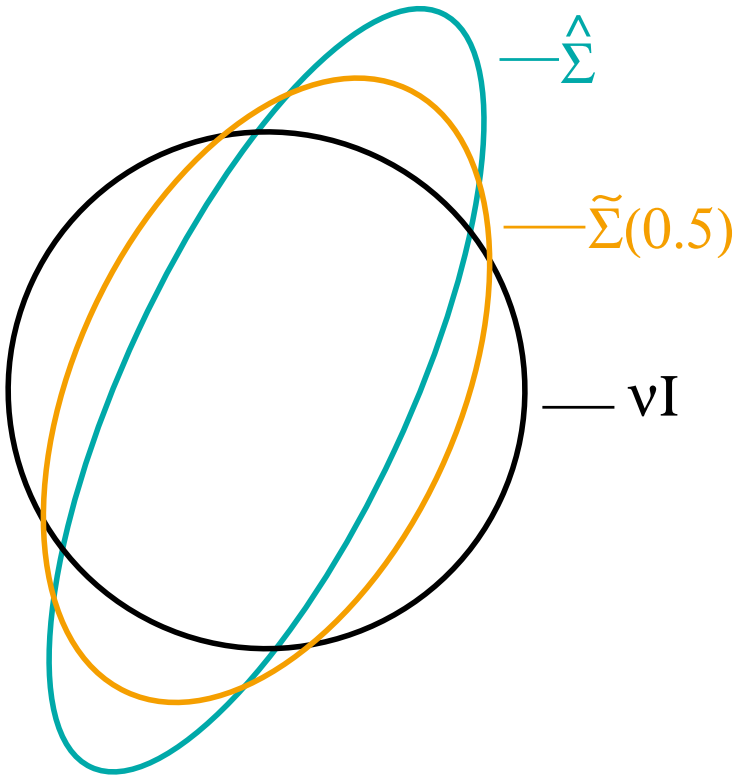


A Remedy for the Estimation Bias

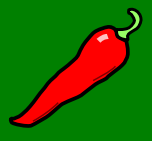
A simple way that counteracts the bias is **shrinkage**:
The empirical covariance matrix $\hat{\mathbf{\Sigma}}$ is modified to be more spherical:

$$\tilde{\mathbf{\Sigma}}(\gamma) = (1-\gamma)\hat{\mathbf{\Sigma}} + \gamma\nu\mathbf{I}$$

for a $\gamma \in [0, 1]$ and ν defined as average Eigenvalue $\mathrm{trace}(\hat{\mathbf{\Sigma}})/d$.



Next, we check that shrinkage serves the intended purpose. Covariance matrices are described by their Eigenvectors and Eigenvalues. So, we have to investigate, what happens to those, when we change over from the empirical covariance matrix $\hat{\mathbf{\Sigma}}$.



Properties of the Shrunk Covariance Matrix

From the Eigenvalue decomposition of the empirical covariance matrix $\hat{\mathbf{\Sigma}} = \mathbf{V}\mathbf{D}\mathbf{V}^\top$ with orthonormal $\mathbf{V}$ and diagonal $\mathbf{D}$, we get an Eigenvalue decomposition of $\tilde{\mathbf{\Sigma}}(\gamma) = (1-\gamma)\hat{\mathbf{\Sigma}} + \gamma\nu\mathbf{I}$ like this:

$$\begin{aligned}\tilde{\mathbf{\Sigma}}(\gamma) &= (1-\gamma)\mathbf{V}\mathbf{D}\mathbf{V}^\top + \gamma\nu\mathbf{I} \\ &= (1-\gamma)\mathbf{V}\mathbf{D}\mathbf{V}^\top + \gamma\nu\mathbf{V}\mathbf{I}\mathbf{V}^\top \\ &= \mathbf{V}\underbrace{\left((1-\gamma)\mathbf{D} + \gamma\nu\mathbf{I}\right)}_{\text{diagonal matrix}}\mathbf{V}^\top\end{aligned}$$

We see that

- $\hat{\mathbf{\Sigma}}$ and $\tilde{\mathbf{\Sigma}}(\gamma)$ have the same Eigenvectors (columns of $\mathbf{V}$)
- Extreme Eigenvalues (large/small) are shrunk/extended towards the average Eigenvalue ν as $d_i \mapsto (1-\gamma)d_i + \gamma\nu$
- $\gamma = 0$ means no shrinkage: $\tilde{\mathbf{\Sigma}}(0) = \hat{\mathbf{\Sigma}}$
- $\gamma = 1$ corresponds to spherical covariance matrices: $\tilde{\mathbf{\Sigma}}(1) = \nu\mathbf{I}$

Regularized Linear Discriminant Analysis

This technique can be used to enhance LDA to work better in the case of a low number-of-samples to dimensionality ratio. The empirical covariance matrix $\hat{\mathbf{\Sigma}}$ is replaced by a shrunk covariance matrix $\tilde{\mathbf{\Sigma}}(\gamma)$:

$$\mathbf{w}_\gamma := \tilde{\mathbf{\Sigma}}(\gamma)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

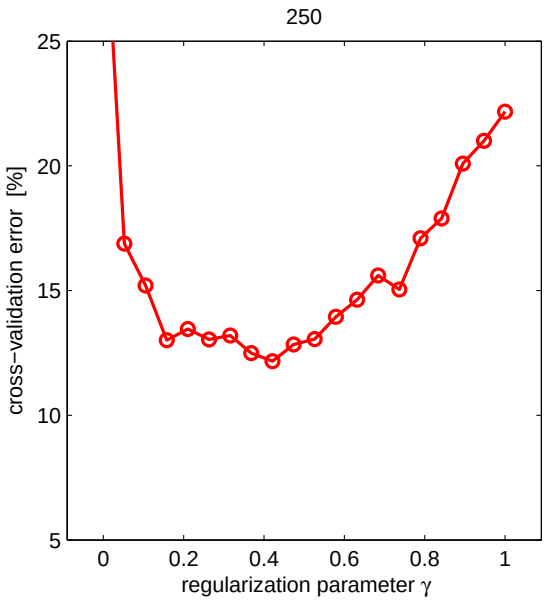
Here, γ is a hyperparameter that has to be selected between 0 and 1.

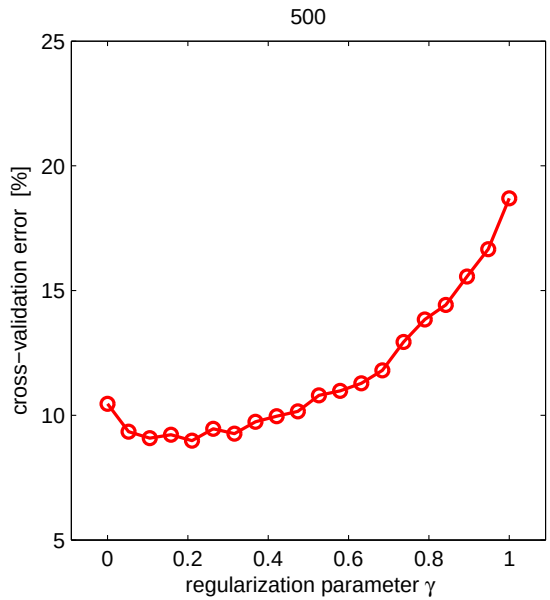
- $\gamma = 0$ yields $\mathbf{w}_0 = \hat{\mathbf{\Sigma}}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$, i.e. unregularized LDA
- $\gamma = 1$ yields $\mathbf{w}_1 = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$, i.e. NCC

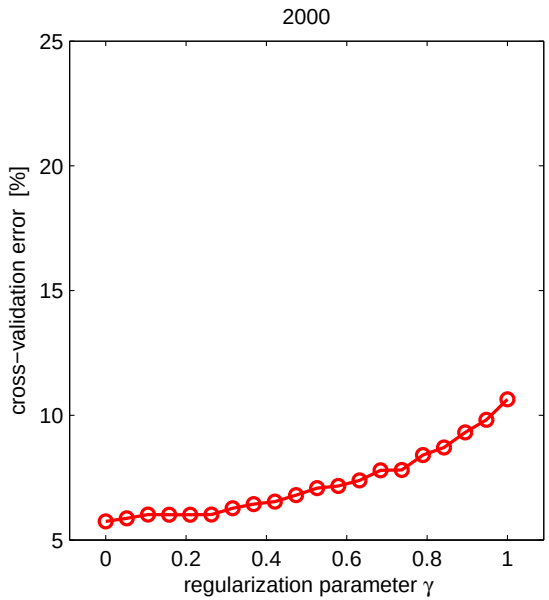
But: There is no golden rule for setting γ. A pragmatic, however time-consuming way is to use **cross-validation** for the selection.

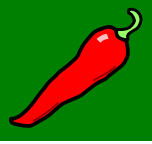
LDA with Different Shrinkage Parameters

Cross-validation results for different sizes of training data (250, 500, 2000) for different values of the shrinkage parameter γ (x-axis). Features vectors have 250 dimensions.









Optimal Selection of Shrinkage Parameter

As a (relatively) novel method for selecting the free parameter (γ) other than with cross-validation, there is an analytical method.

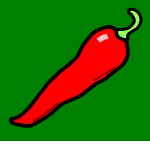
Let $\mathbf{x}_1, \ldots, \mathbf{x}_K \in \mathbb{R}^d$ be K feature vectors and let $\hat{\boldsymbol{\mu}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{x}_k$ be the empirical mean.

Aim: get a better estimate of the true covariance matrix $\boldsymbol{\Sigma}$ (especially in case $K < d$) than the sample covariance matrix

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{K-1} \sum_{k=1}^{K} (\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^\top$$

by selecting a γ in

$$\tilde{\boldsymbol{\Sigma}}(\gamma) := (1-\gamma)\hat{\boldsymbol{\Sigma}} + \gamma\nu\mathbf{I}.$$



Optimal Selection of Shrinkage Parameter

The approach of [Ledoit & Wolf, J Multivar Anal, 2004] is to minimize

$$\|\tilde{\mathbf{\Sigma}}(\gamma) - \mathbf{\Sigma}\|_F^2 \qquad \text{with } \|\cdot\|_F^2 \text{ being the Frobenius norm.}$$

We denote by $(\mathbf{x}_k)_i$ resp. $(\hat{\boldsymbol{\mu}})_i$ the i-th element of the vector $\mathbf{x}_k$ resp. $\hat{\boldsymbol{\mu}}$ and define the covariance of feature i and j in trial k:

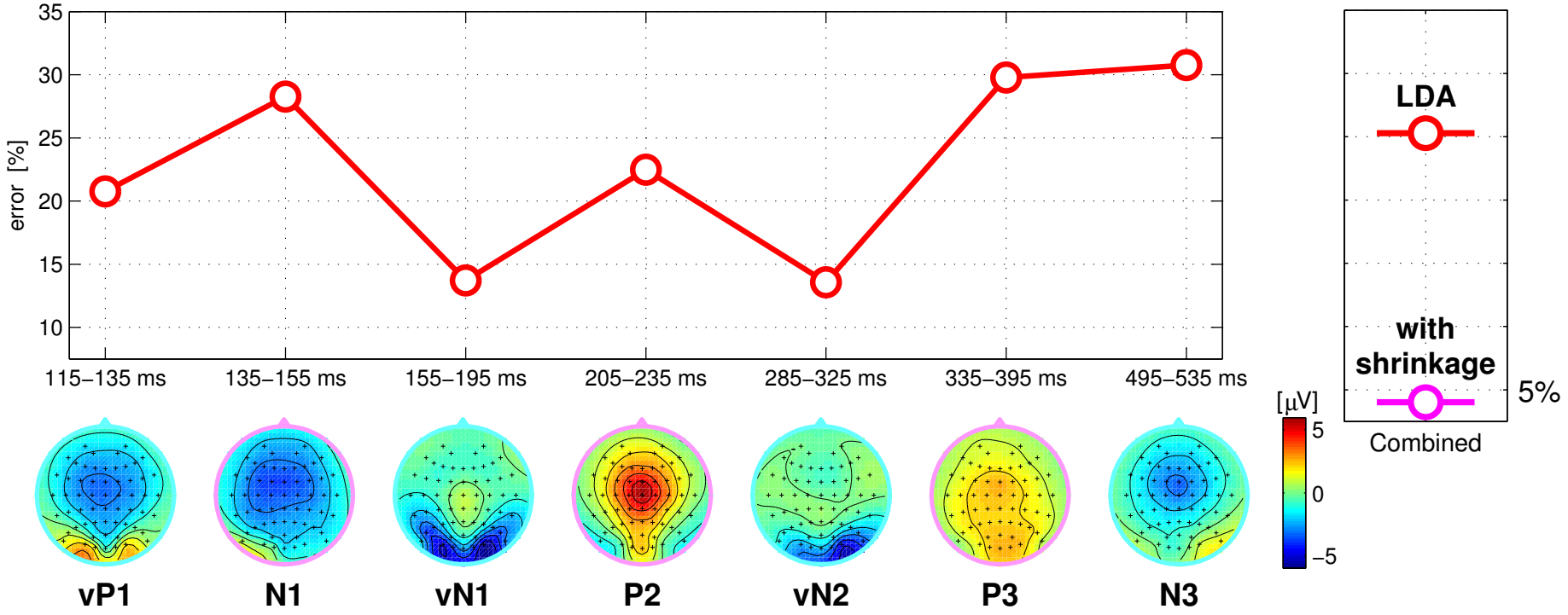
$$z_{ij}(k) = ((\mathbf{x}_k)_i - (\hat{\boldsymbol{\mu}})_i)\ ((\mathbf{x}_k)_j - (\hat{\boldsymbol{\mu}})_j)$$

Denoting by s_{ij} the element in the i-th row and j-th column of the matrix $\hat{\mathbf{\Sigma}} - \nu\mathbf{I}$, the optimal shrinkage parameter $\gamma^\star = \text{argmin}_\gamma \|\tilde{\mathbf{\Sigma}}(\gamma) - \mathbf{\Sigma}\|_F^2$ can be analytically calculated as [Schäfer & Strimmer 2005]

$$\gamma^\star = \frac{K}{(K-1)^2} \frac{\sum_{i,j=1}^{d} \text{var}_{k=1,\ldots,K}(z_{ij}(k))}{\sum_{i,j=1}^{d} s_{ij}^2}.$$

Shrinkage-LDA: use $\tilde{\mathbf{\Sigma}}(\gamma^\star)$ instead of $\hat{\mathbf{\Sigma}}$.

Classification on Single Components and Combined



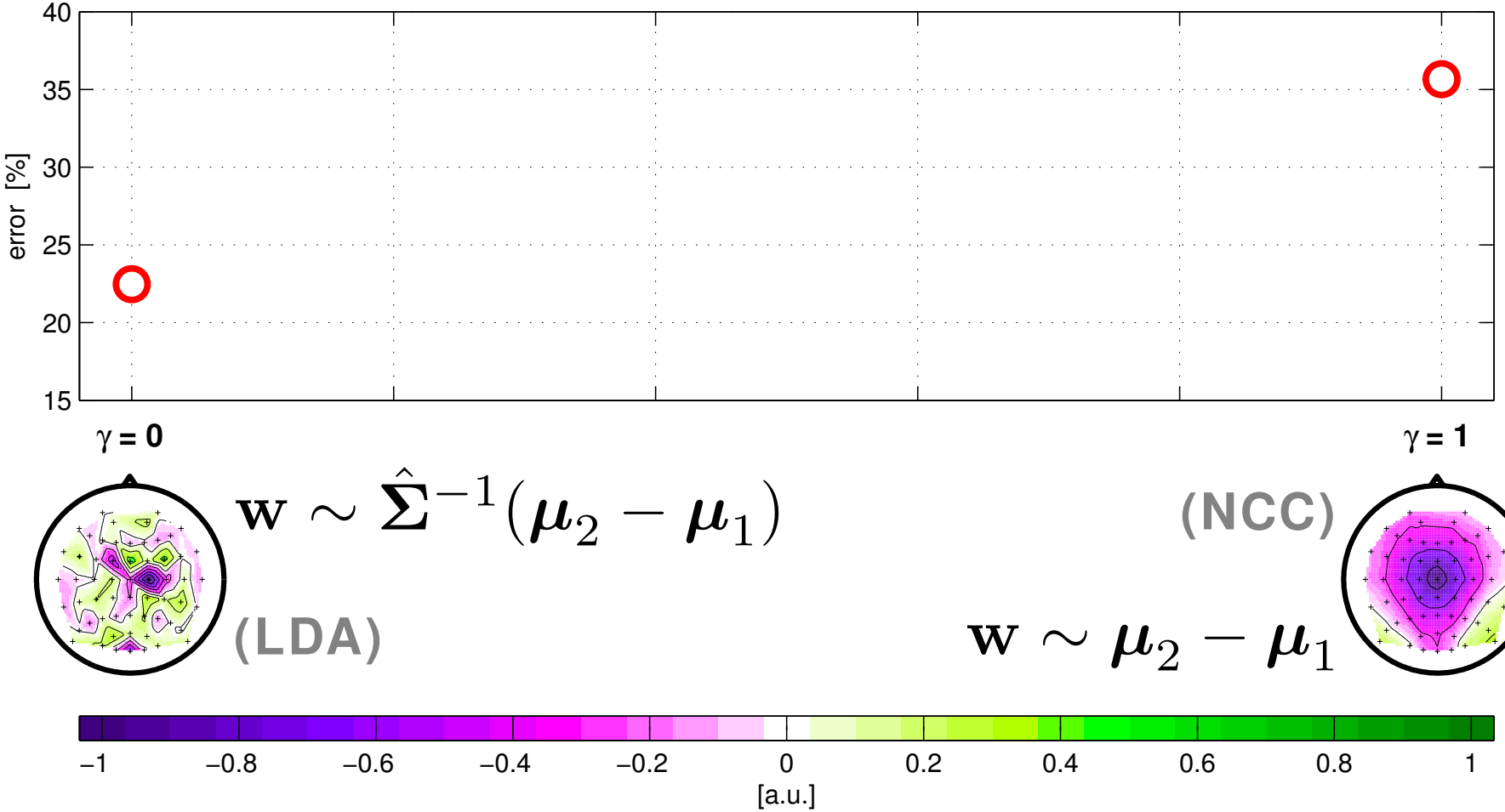
Classification (with $N = 750$ training samples) on seven different single components ($d = 55$) yields errors between **14%** and 31%.

LDA on the concatenated feature ($d = 7 \cdot 55 = 385$) performs with **25%** worse, although information is added: *overfitting*.

Shrinkage-LDA: only **4%** error. [Blankertz et al, NeuroImage 2011]

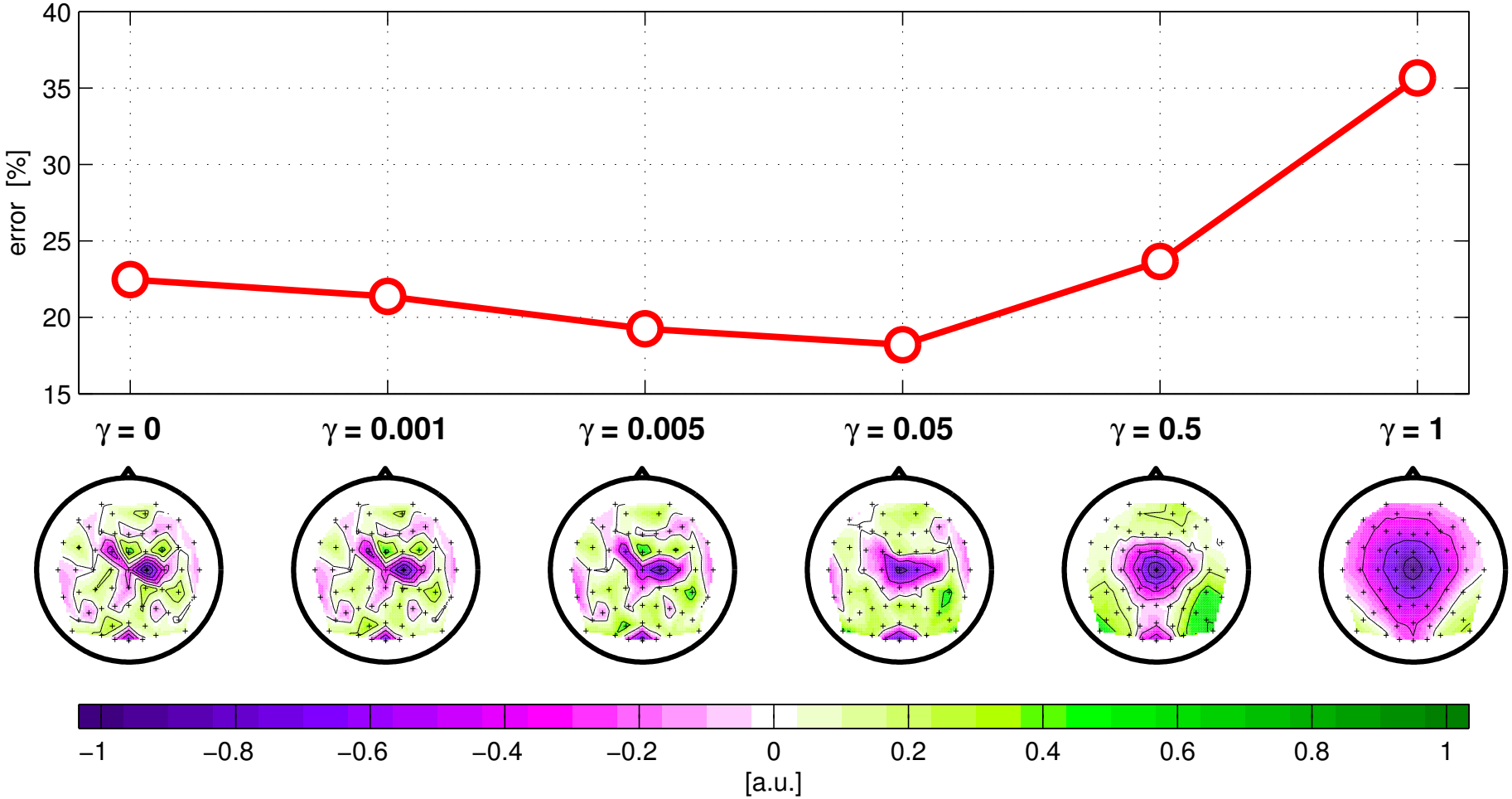
Impact of Shrinkage as Trade-off

LDA with shrinkage: $\mathbf{w} = \tilde{\mathbf{\Sigma}}(\gamma)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$;
$\tilde{\mathbf{\Sigma}}(\gamma) = (1-\gamma)\hat{\mathbf{\Sigma}} + \gamma\nu\mathbf{I}$



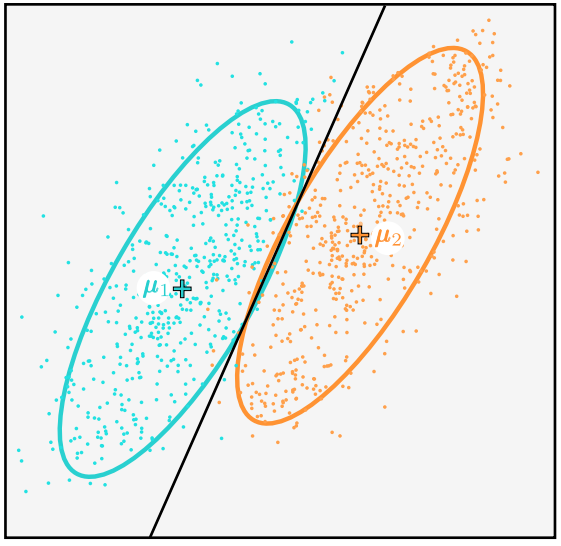
Impact of Shrinkage as Trade-off

With increasing shrinkage, the spatial filters (classifier) look smoother, but classification may degrade with too much shrinkage.

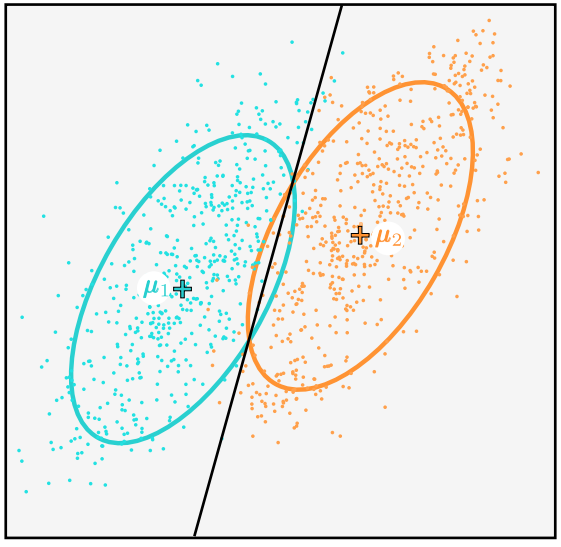


Recap: NCC, LDA and Shrinkage-LDA

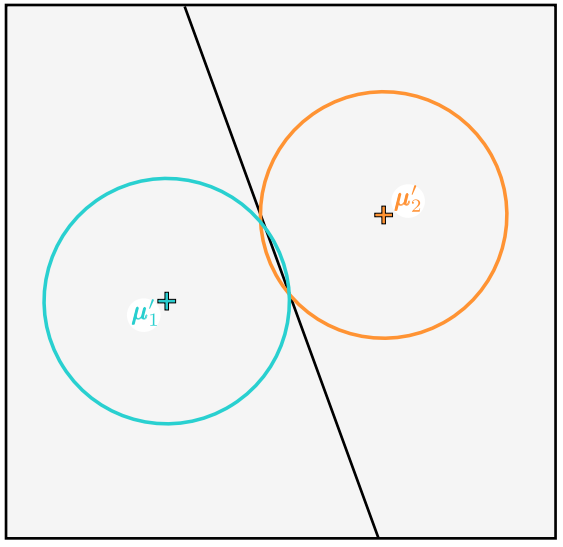
$\mathbf{w} = \hat{\mathbf{\Sigma}}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$



$\mathbf{w} = \tilde{\mathbf{\Sigma}}(\gamma)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$



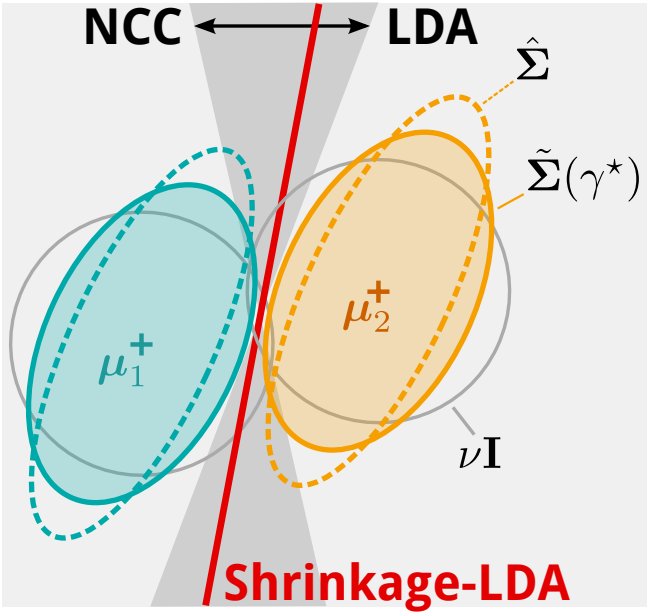
$\mathbf{w} = \boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$

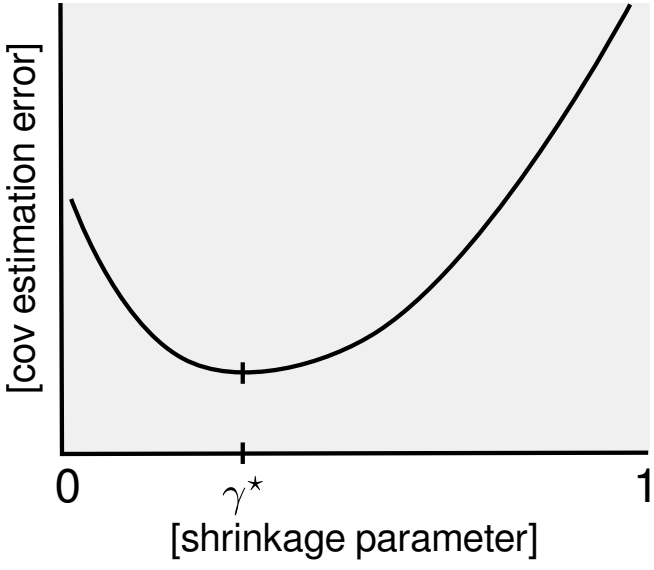


same signals of interest $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ – different spatial structure of noise $(\mathbf{\Sigma})$ or in another view: different belief in the empirical covariance matrix.

The amount of shrinkage (γ) relates to the 'believe' in the estimation of the noise.

Classification with Shrinkage-LDA at a Glance





Shrinkage-LDA hyperplane is defined by:

$$\mathbf{w} := \tilde{\mathbf{\Sigma}}(\gamma^\star)^{-1}\,(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$$

$$\tilde{\mathbf{\Sigma}}(\gamma) := (1-\gamma)\hat{\mathbf{\Sigma}} + \gamma\nu\mathbf{I}$$

Calculate optimal $\gamma^\star$ analytically:

$$\gamma^\star = \operatorname*{argmin}_{\gamma} \|\tilde{\mathbf{\Sigma}}(\gamma) - \mathbf{\Sigma}\|_F^2$$

$$= \frac{K}{(K-1)^2}\,\frac{\sum_{i,j=1}^{d} \operatorname{var}_k(z_{ij}(k))}{\sum_{i,j=1}^{d} s_{ij}^2} \quad \text{with}$$

$$z_{ij}(k) := ((\mathbf{x}_k)_i - (\hat{\boldsymbol{\mu}})_i)\,((\mathbf{x}_k)_j - (\hat{\boldsymbol{\mu}})_j)^\top$$

Selection of shrinkage parameter γ:

[Ledoit & Wolf 2004], [Schäfer & Strimmer 2005]

Tutorial on ERP classification:

[Blankertz et al, NeuroImage 2011]

Part IV

Understanding Spatial Filters

LDA as a Spatial Filter

Assume we have continuous EEG signals $\mathbf{x}(t)$, from which spatial features $\mathbf{x}_k$ (of two classes) have been extracted, e.g., amplitudes of all channels at the P3 peak latency.

Furthermore, let $\mathbf{w}$ be the weight vector of a linear classifier that was trained on those features $\mathbf{x}_k$.

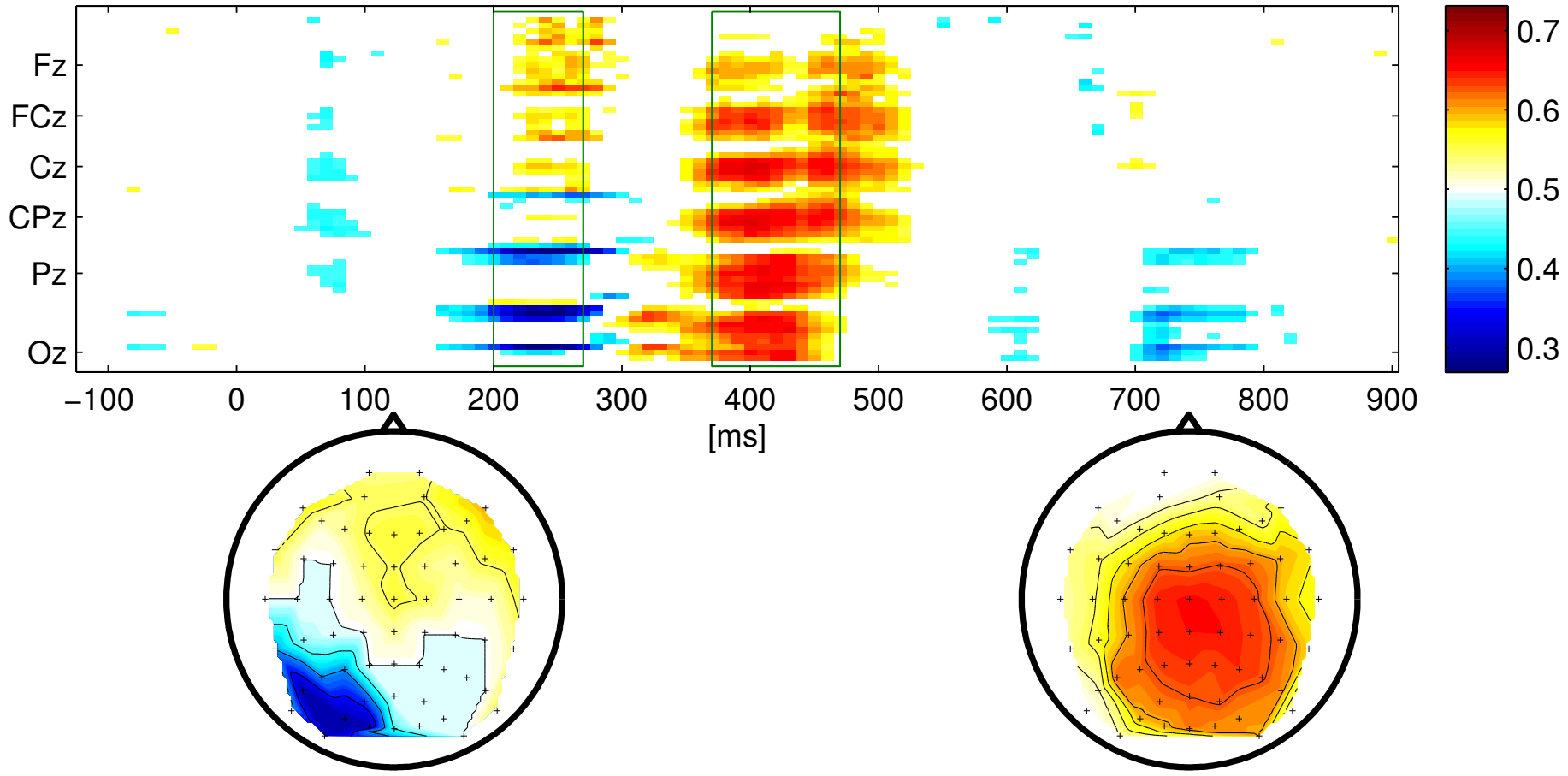
Then $\mathbf{w}$ is the filter within a discriminative (backward) model with the objective to estimate classes labels on the training data.

Note, that $\mathbf{w}$ can be applied as **spatial filter** to continuous EEG signals, like $\mathbf{x}(t)$, to extract a component:

$$y(t) := \mathbf{w}^\top \mathbf{x}(t)$$

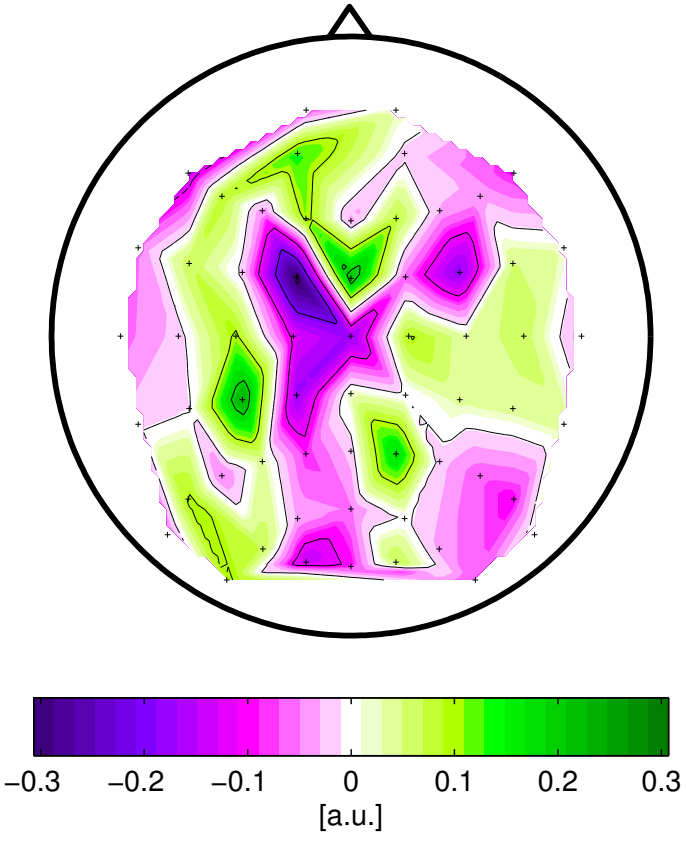
A nice application of this can be found in [Scholler et al, 2012].

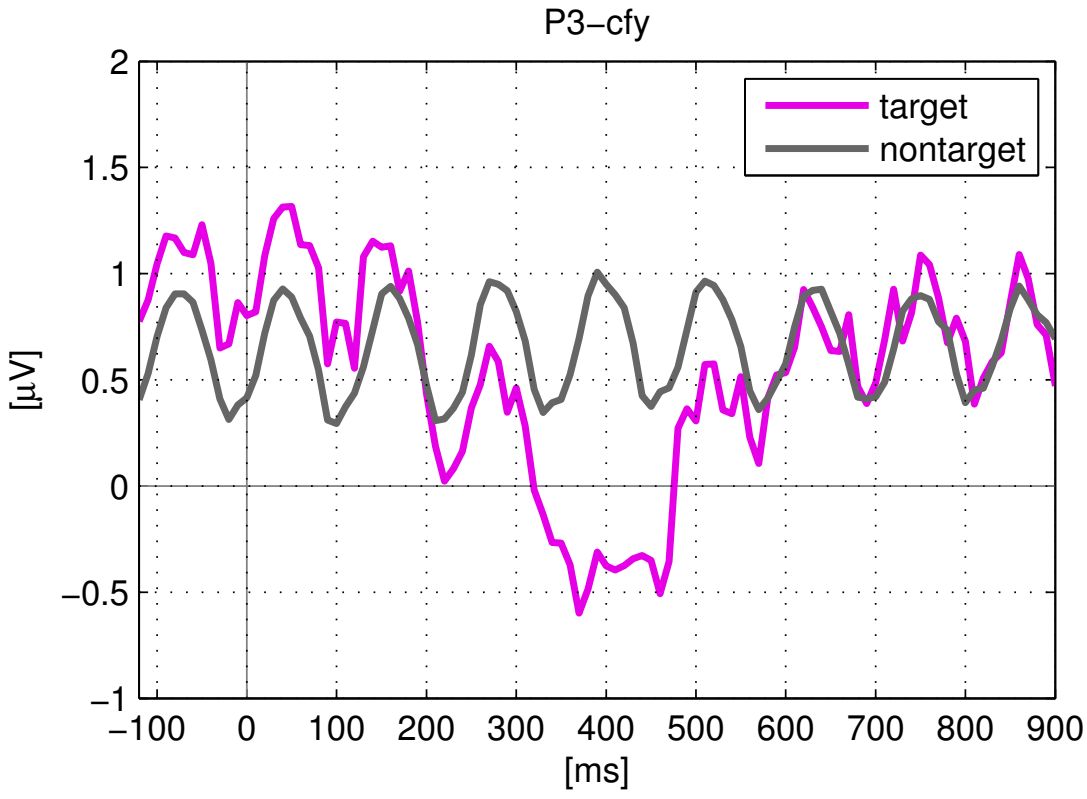
Recap: AUC matrix in RSVP Speller



In this example from a RSVP Speller experiment, the N2 interval 200-270ms and the P3 interval 370-470ms give discriminative spatial features. We train one LDA on each of those.

LDA Weight Vector as Spatial Filter – Example





This technique can be used to find ERP activity in continuous streams of data in which no timing information of event is available. (But there has to be training data with events at known time points.)

Interpretation of Spatial Filters

Let's assume we have a mixture of two sources (ignoring the noise here)

$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t) + \mathbf{a}_2 s_2(t),$$

and the task is to find a spatial filter $\mathbf{w}$ to recover s_1. Applying the (yet to be determined) filter $\mathbf{w}$ to $\mathbf{x}(t)$ yields

$$\mathbf{w}^\top \mathbf{x}(t) = \mathbf{w}^\top \mathbf{a}_1 s_1(t) + \mathbf{w}^\top \mathbf{a}_2 s_2(t).$$

To recover s_1 (i.e., to eliminate the contribution of s_2), the filter $\mathbf{w}$ needs to be chosen such that $\mathbf{w}^\top \mathbf{a}_2 = 0$: the filter $\mathbf{w}$ is orthogonal to $\mathbf{a}_2$.

In the (untypical) case of orthogonal propagation vectors ($\mathbf{a}_1^\top \mathbf{a}_2 = 0$) $\mathbf{w} = \mathbf{a}_1$ does the job: The best filter corresponds to the propagation direction of the source, i.e., a pattern.

Interpretation of Spatial Filters (2)

In the typical case ($\mathbf{a}_1^\top \mathbf{a}_2 \neq 0$), the best filter $\mathbf{w}$ to recover source s_1 also depends on the interfering source s_2, as it must be orthogonal to its propagation vector $\mathbf{a}_2$.

Example. We would like to extract

- s_1, the cognitive P300 component

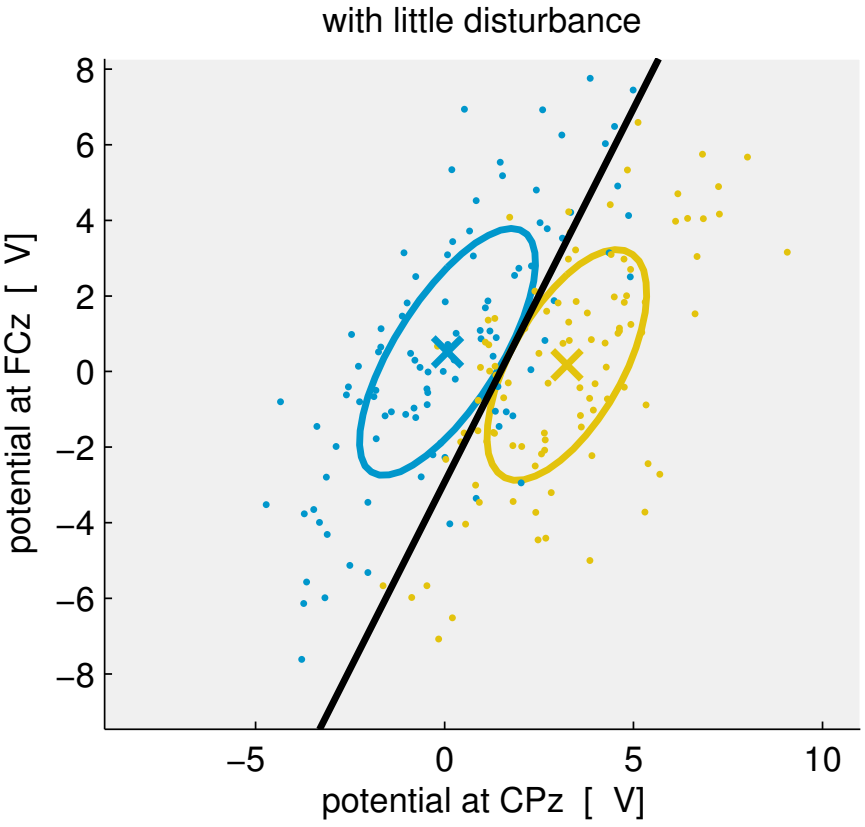
but there is interference from

- s_2, the visual area.

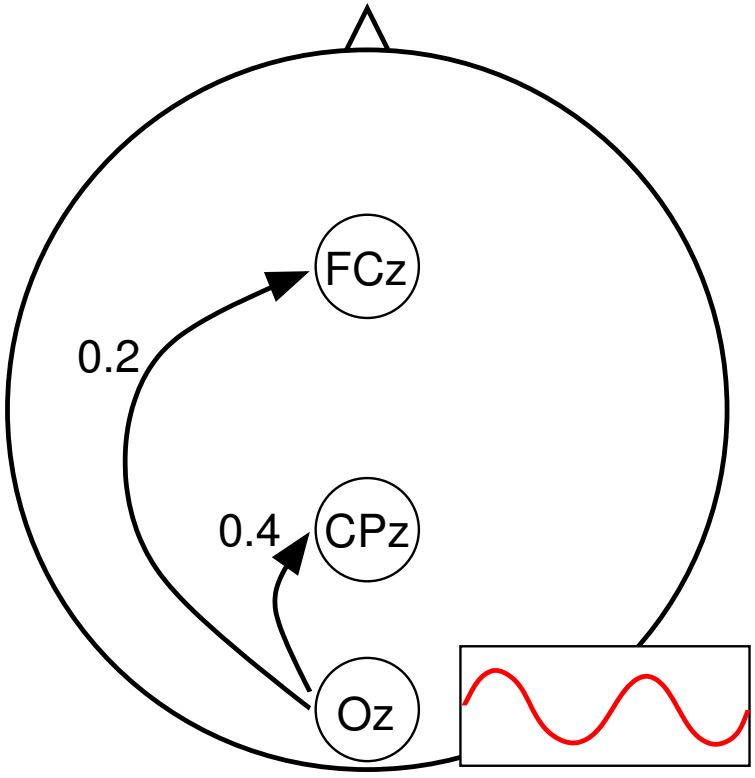
The best filter to recover the P300 component (s_1) depends also on the interfering source of the visual area (s_2). In particular, the spatial map of the filter probably shows strong weights over occipital area, although the P300 component originates from the central region.

Understanding Spatial Filters

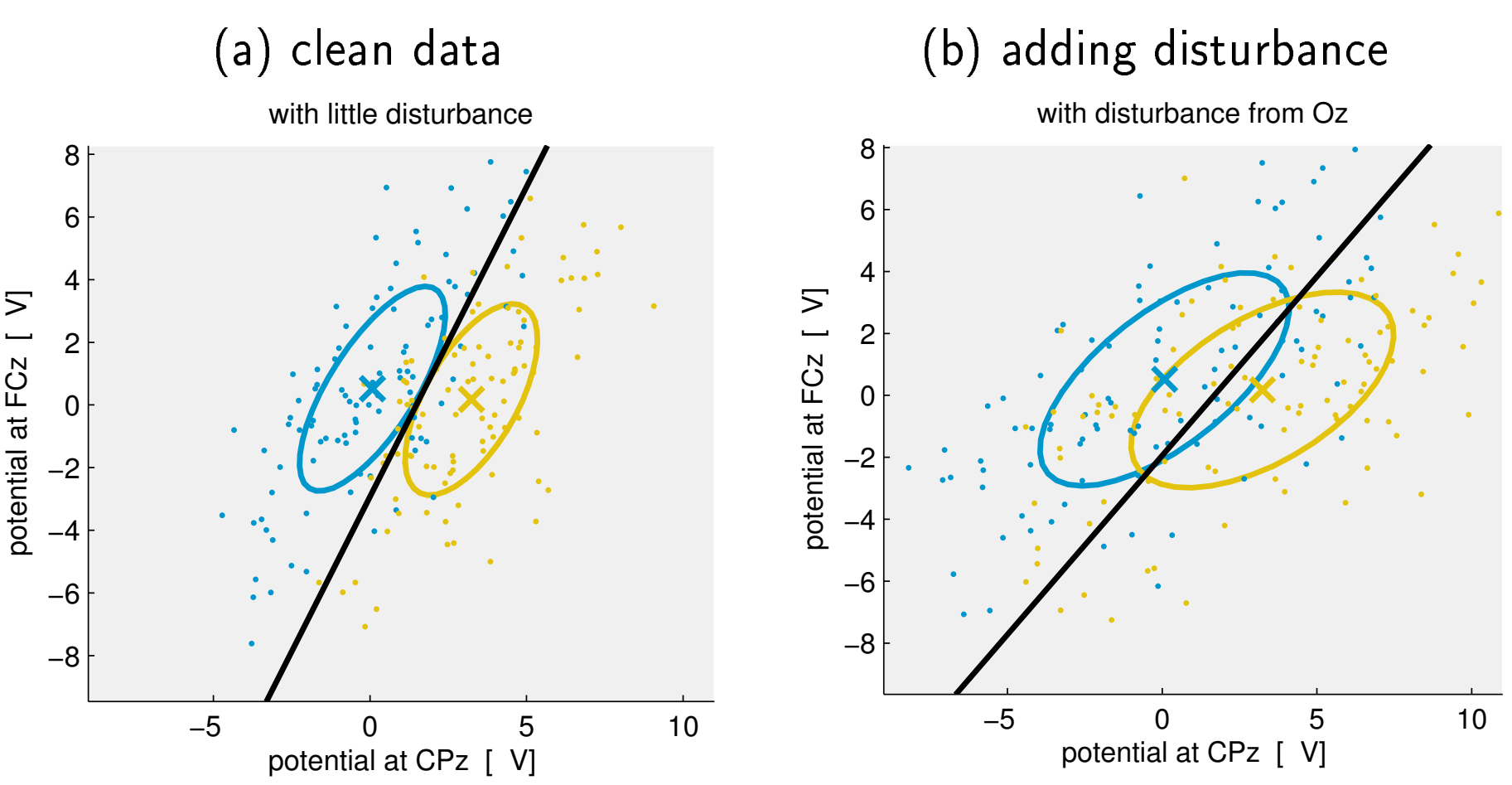
(a) clean data



(b) adding disturbance



Understanding Spatial Filters



Two channel classification of (a): 15% error, (b): 37% error

When disturbing channel Oz is added to the data (3D): 16% error. Here, channel Oz is required for good classification although itself is not discriminative.

The Blessing and Curse of Machine Learning

Machine learning provides **multivariate** techniques for the analysis of EEG data involving optimization of user-specific models.

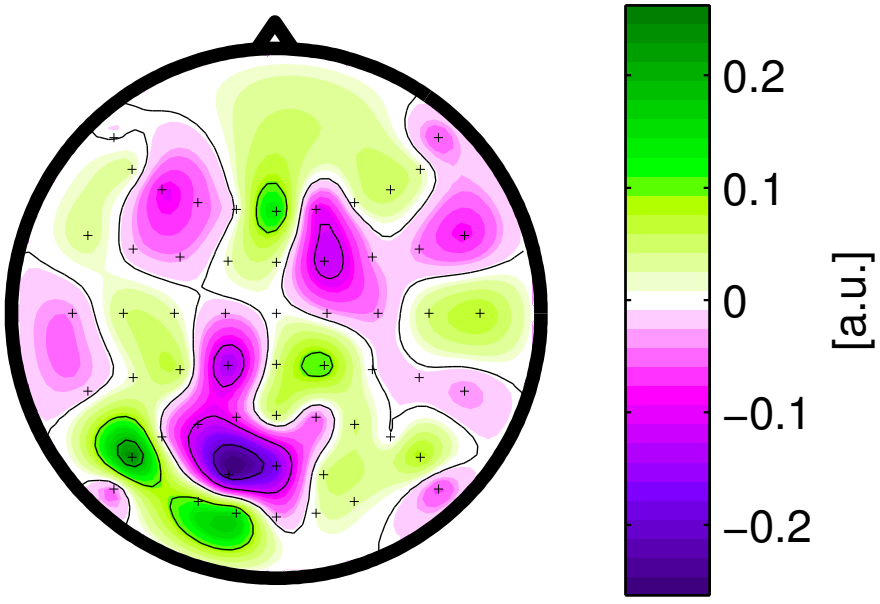
This results in a considerably **increased sensitivity** in the discovering of neural correlates.

The down side is that the **interpretation** of *where* the discriminative information originates from is not always straight forward...

... nevertheless very important. Do not use ML techniques as black box.

Interpretation of Classifier Weights?

The weights of a linear classifier can be visualized in the domain of the input features. For *spatial features*, they can be depicted as scalp topography.



LDA trained on spatial features extracted from the time interval 380–410 ms. It is tempting to interpret the prominent weights of this map wrt neurophysiology.

Temporal and spatio-temporal features suggest interpretation is an analogue way. The considerations above have shown that spatial filters cannot be interpreted in a direct way. But we will see that there is a solution [Haufe et al, 2014].

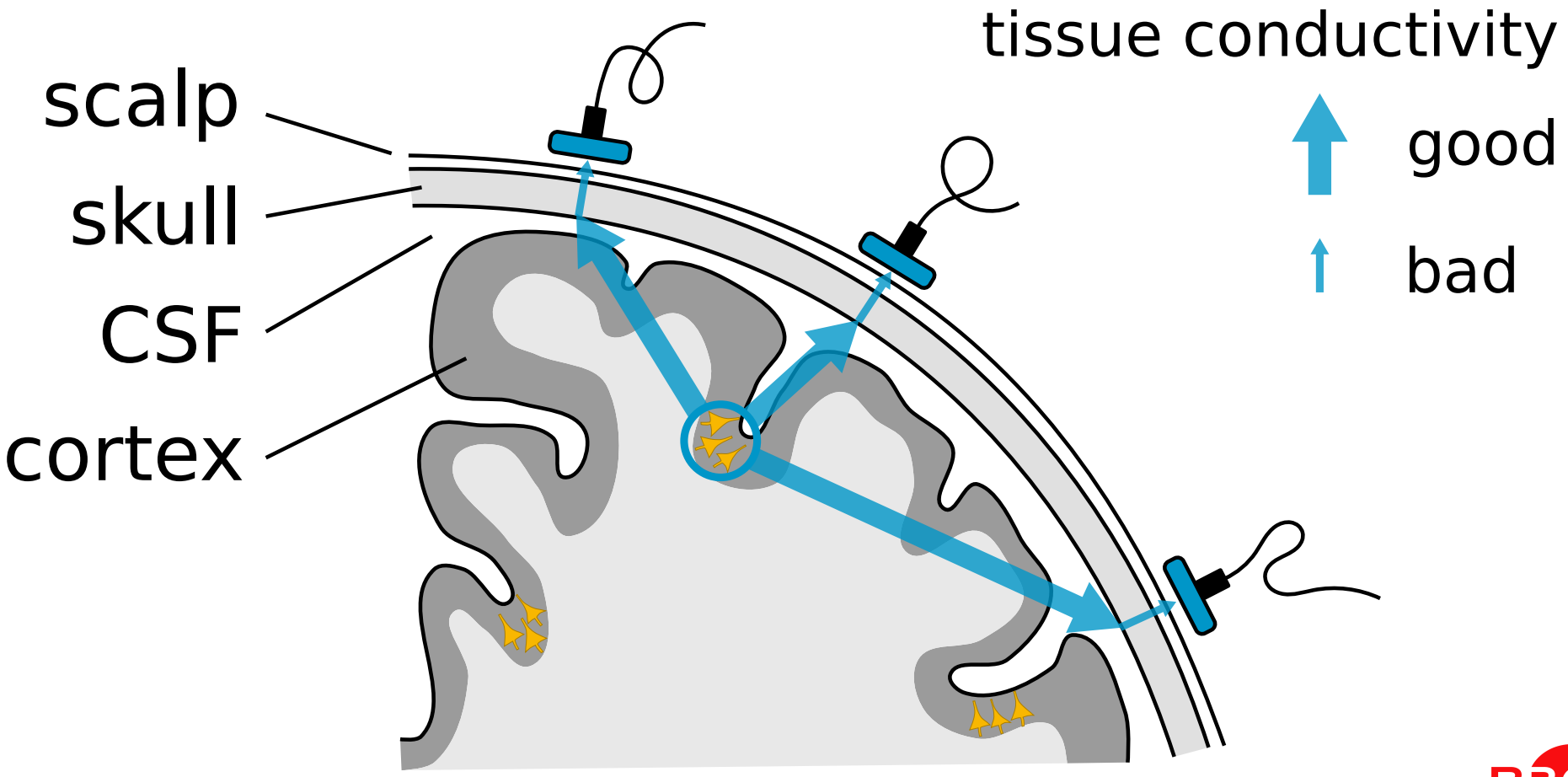
Part V

The Linear Model

Linear Model of EEG

Next, a linear model which represents the electrophysics of EEG is introduced. Although oversimplifying, this model it useful for understanding.

Remember the issue of volume conduction discussed before:

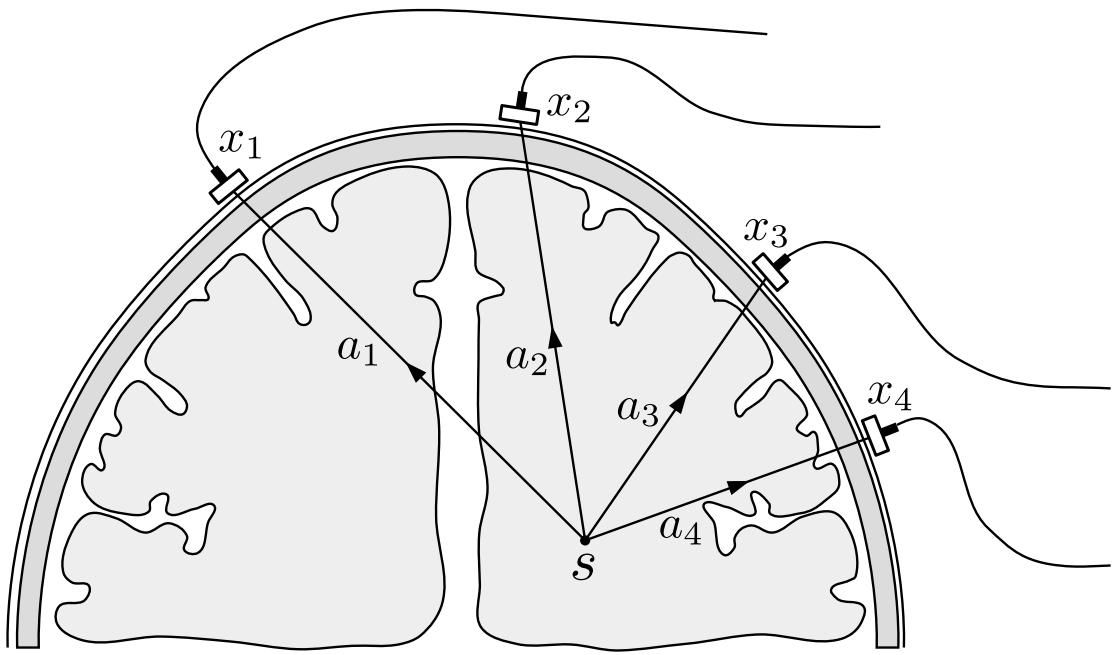


Linear Model of EEG: Propagation of Electrical Activity

- **Assumption:** The contribution of a current source $s(t)$ to the scalp potentials $\mathbf{x}(t) = [x_1, \ldots, x_P]^\top$ is linear in $s(t)$:

$$\mathbf{x}(t) = [a_1 s(t), \ldots, a_P s(t)]^\top = \mathbf{a}\, s(t)$$

- The proportionality factors in vector $\mathbf{a}$ are typically unknown and depend on the spatial distribution and orientation of the current source and the conductivity distribution of the anatomy [Parra et al, 2005].



Linear Model of EEG: Forward Model

- Now, we consider several sources with distribution vectors $\mathbf{a}_1, \ldots, \mathbf{a}_P$.
- Potentials are additive. Defining the matrix $\mathbf{A}$ as being composed of the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_P$ (i.e., $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_P]$), the **forward model** is

$$\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t)$$

- Contributions not captured by this model are considered as noise, $\mathbf{n}(t)$, typically assumed to be Gaussian distributed with mean 0.
- This gives a simple linear model representing the electrophysics of EEG:

$$\mathbf{x}(t) = \mathbf{A}\,\mathbf{s}(t) + \mathbf{n}(t)$$

Linear Model of EEG: Backward Model

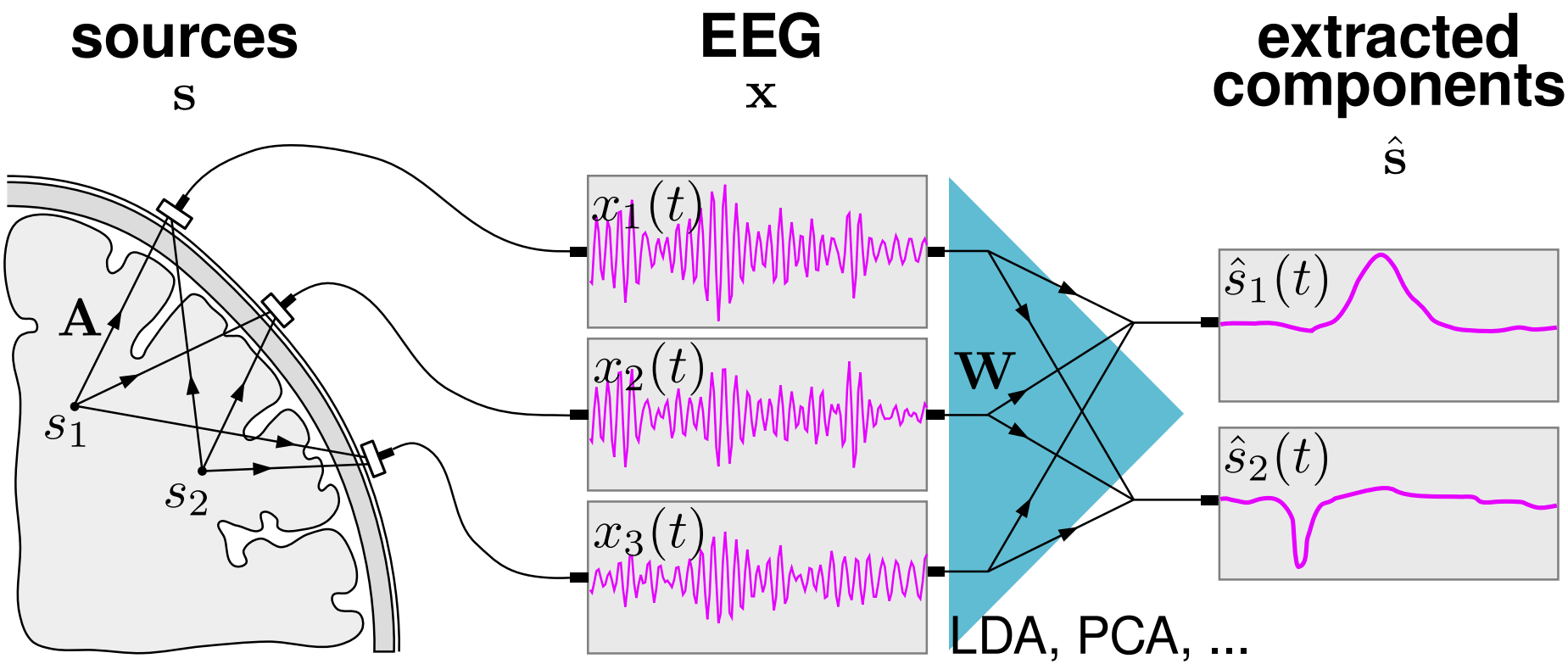
The counterpart to the forward model is the **backward model**:

$$\hat{\mathbf{s}}(t) = \mathbf{W}^\top \mathbf{x}(t)$$

The aim of the backward model *might* be the estimation of sources. If this is the case, and if the forward model $\mathbf{A}$ is known, the best choise (least mean squares estimator) is to take $\mathbf{W}^\top$ as $\mathbf{A}^+$, the pseudoinverse of $\mathbf{A}$.

However, the aim may also be the extraction of components with desired properties, e.g., a good discrimination between conditions (classification setting).

Linear Model of EEG



$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)$$

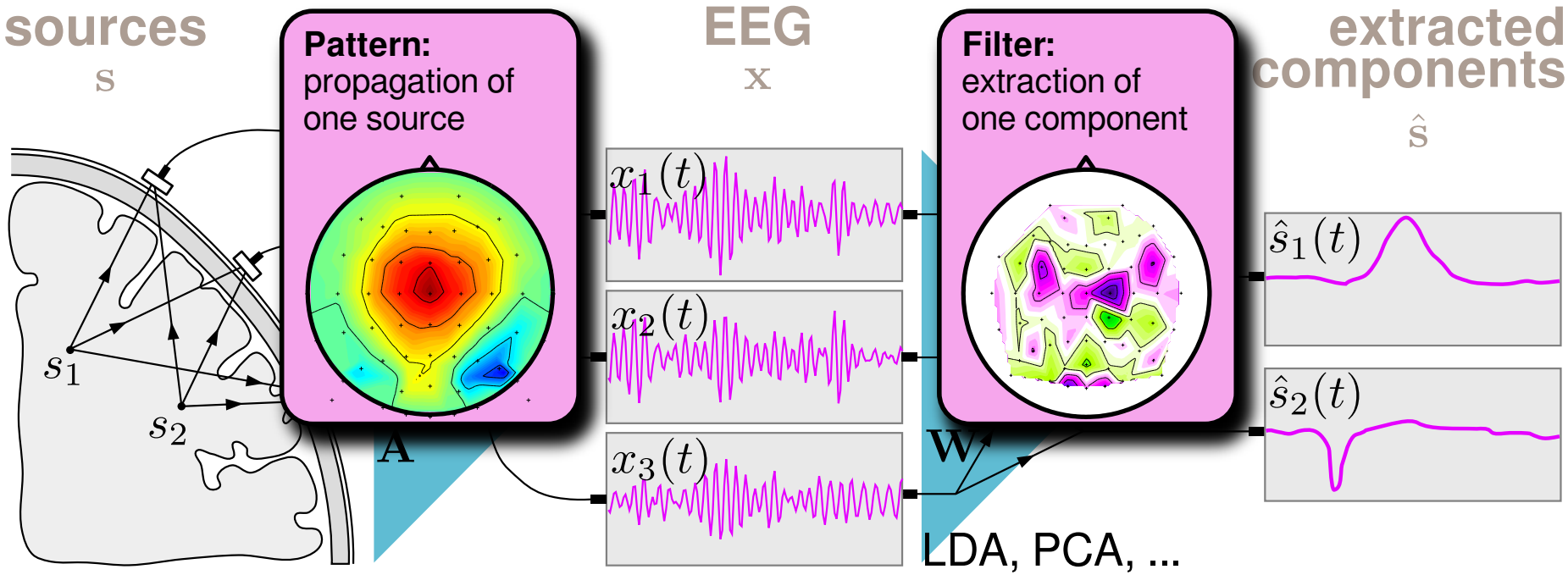
forward model

$$\hat{\mathbf{s}}(t) = \mathbf{W}^\top \mathbf{x}(t)$$

backward model

Each column of $\mathbf{A}$ is a spatial **pattern**: propagation of a source to sensors
Each row of $\mathbf{W}^\top$ is a spatial **filter**: weighting of EEG channels.

Linear Model of EEG



$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)$$

forward model

$$\hat{\mathbf{s}}(t) = \mathbf{W}^\top \mathbf{x}(t)$$

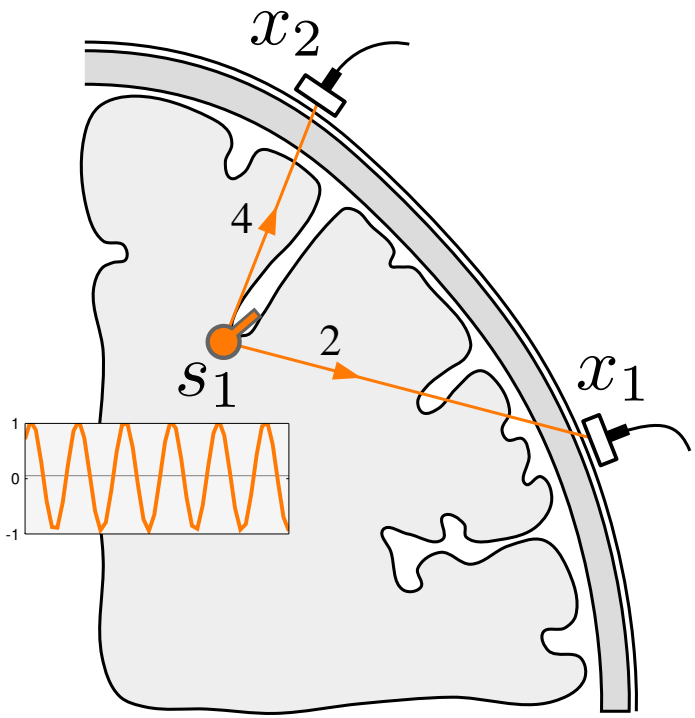
backward model

Each column of $\mathbf{A}$ is a spatial **pattern**: propagation of a source to sensors
Each row of $\mathbf{W}^\top$ is a spatial **filter**: weighting of EEG channels.

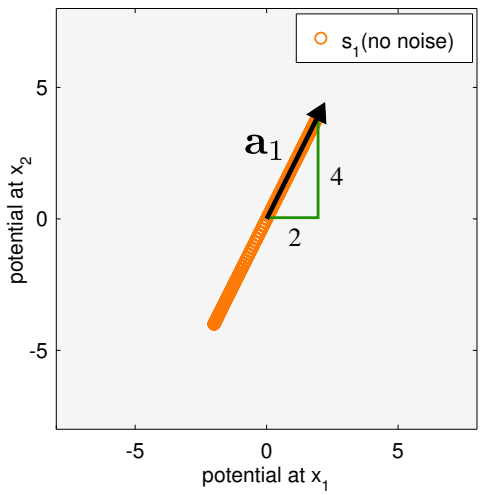
Part VI

Illustration of Spatial Patterns and Filters

Explaining Spatial Patterns and Spatial Filters

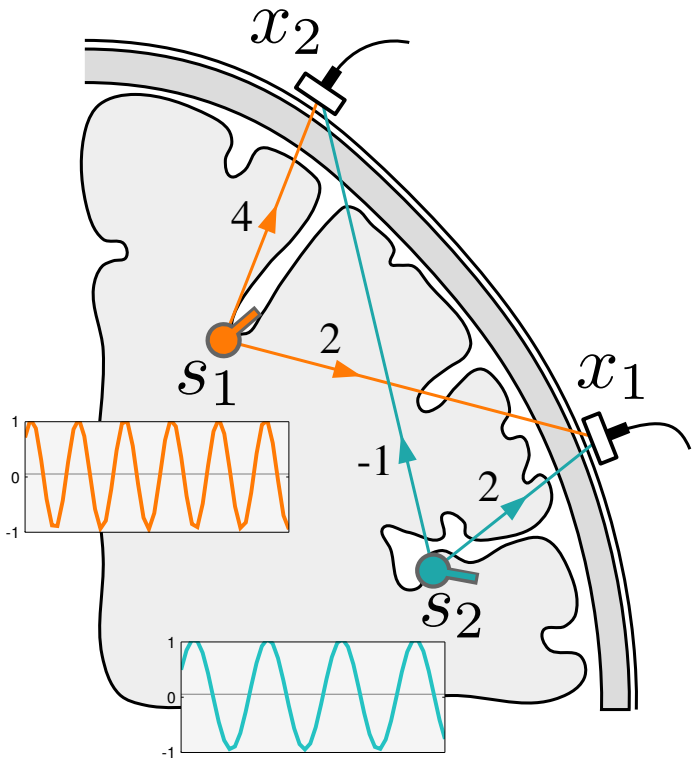


$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t)$$



$$\mathbf{a}_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

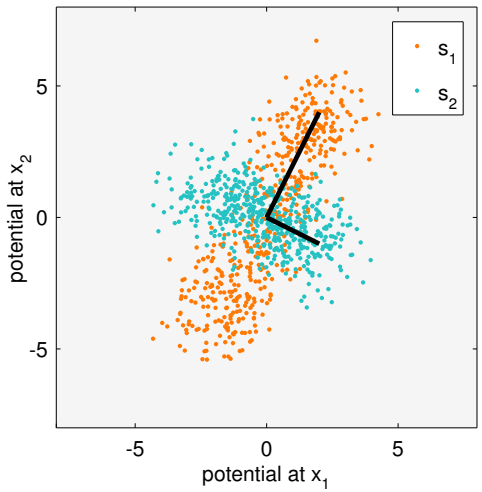
Explaining Spatial Patterns and Spatial Filters

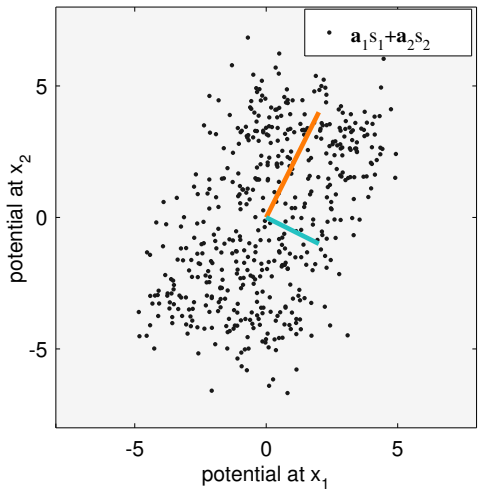


$$\mathbf{a}_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \qquad \mathbf{a}_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

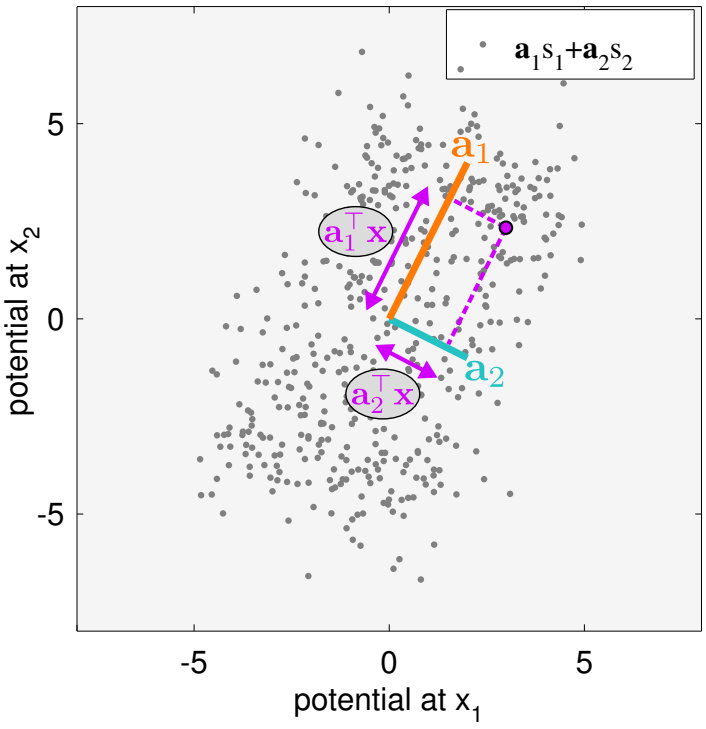
$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t) + \mathbf{n}(t)$$
$$\mathbf{x}(t) = \mathbf{a}_2 s_2(t) + \mathbf{n}(t)$$

$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t) + \mathbf{a}_2 s_2(t) + \mathbf{n}(t)$$



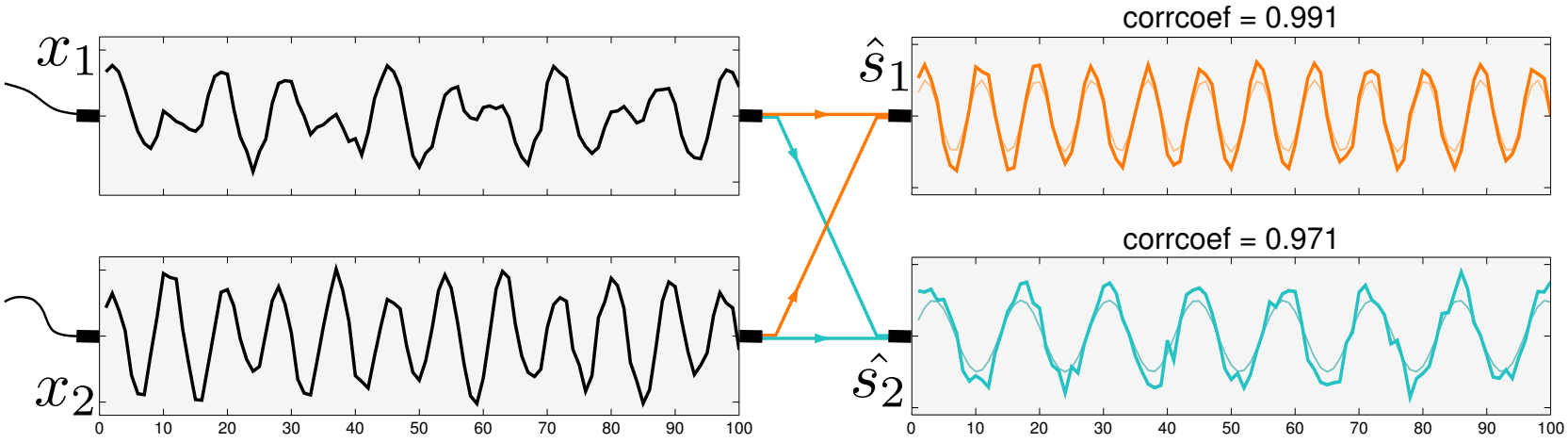


Explaining Spatial Patterns and Spatial Filters

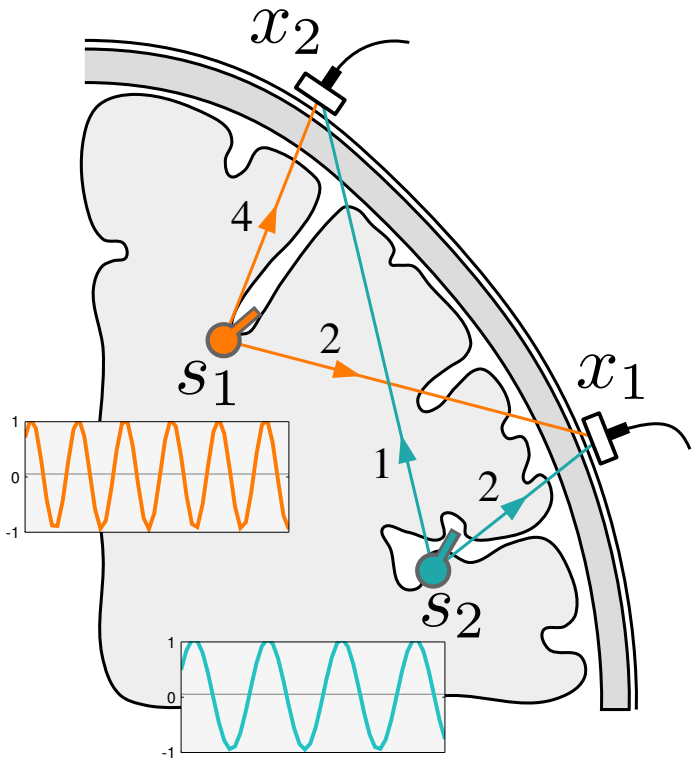


$$\hat{s}_1 = \mathbf{a}_1^\top \mathbf{x}$$
$$= \mathbf{a}_1^\top \mathbf{a}_1 s_1 + \underbrace{\mathbf{a}_1^\top \mathbf{a}_2}_{=0} s_2 + \mathbf{a}_1^\top \mathbf{n}$$
$$\sim s_1$$

$\hat{s}_2 \sim s_2$ (as above)

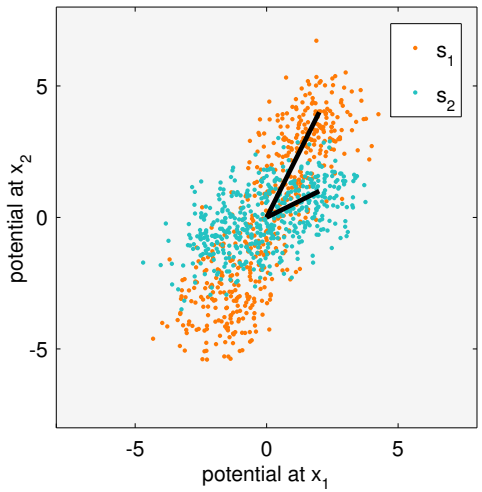


Explaining Spatial Patterns and Spatial Filters

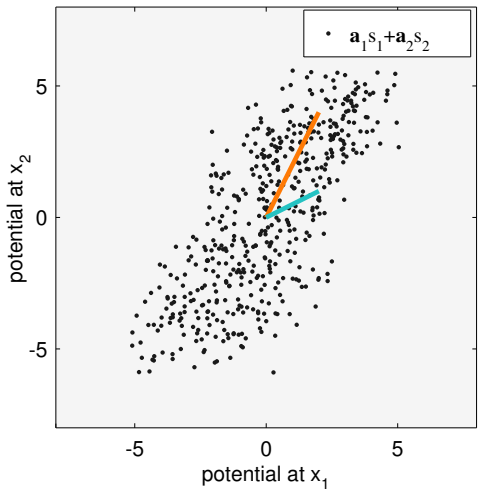


$$\mathbf{a}_1 = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \qquad \tilde{\mathbf{a}}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

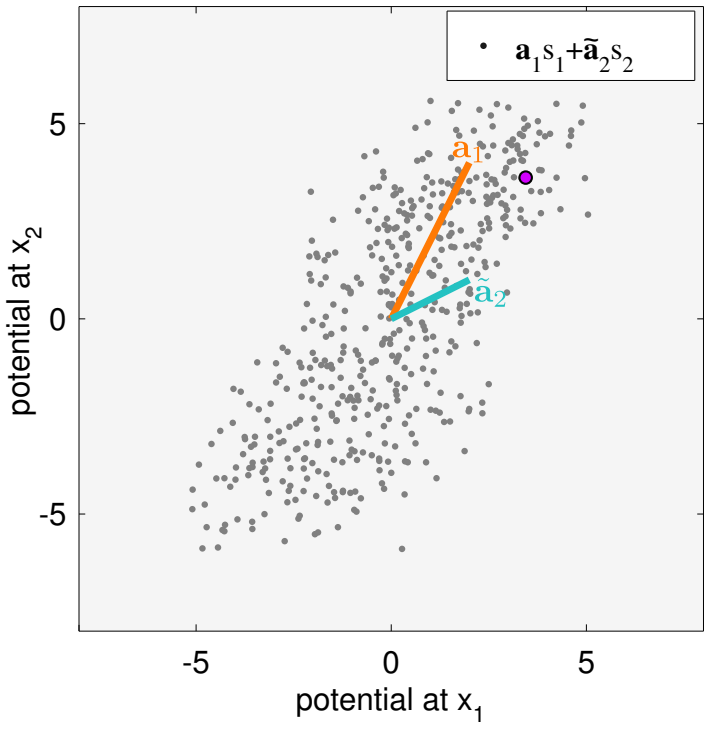
$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t) + \mathbf{n}(t)$$
$$\mathbf{x}(t) = \tilde{\mathbf{a}}_2 s_2(t) + \mathbf{n}(t)$$



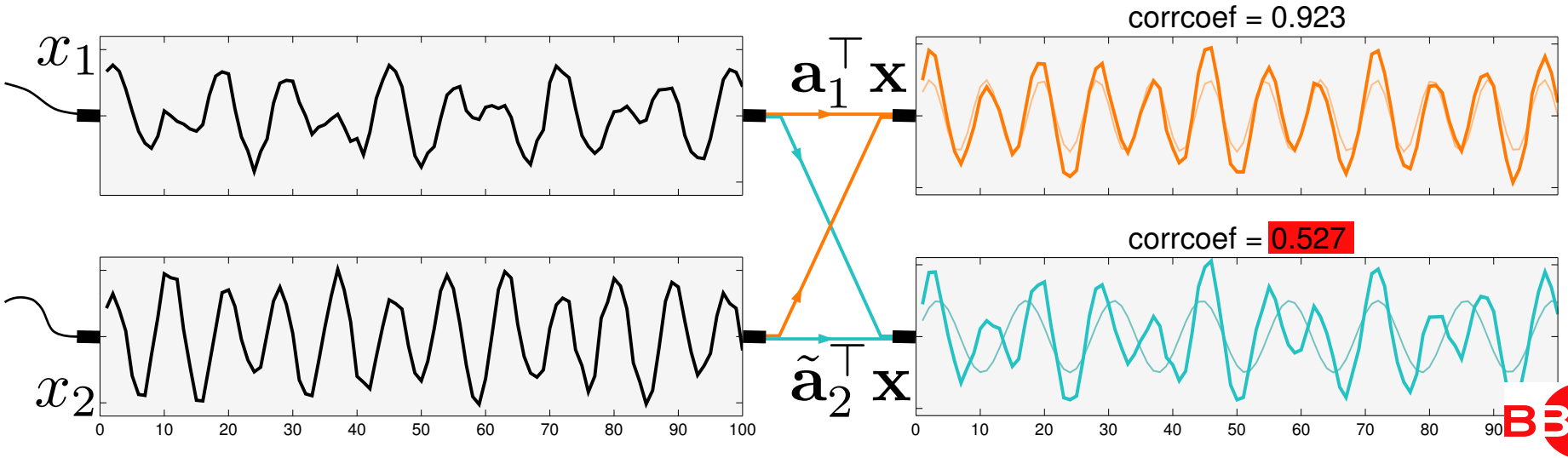
$$\mathbf{x}(t) = \mathbf{a}_1 s_1(t) + \tilde{\mathbf{a}}_2 s_2(t) + \mathbf{n}(t)$$



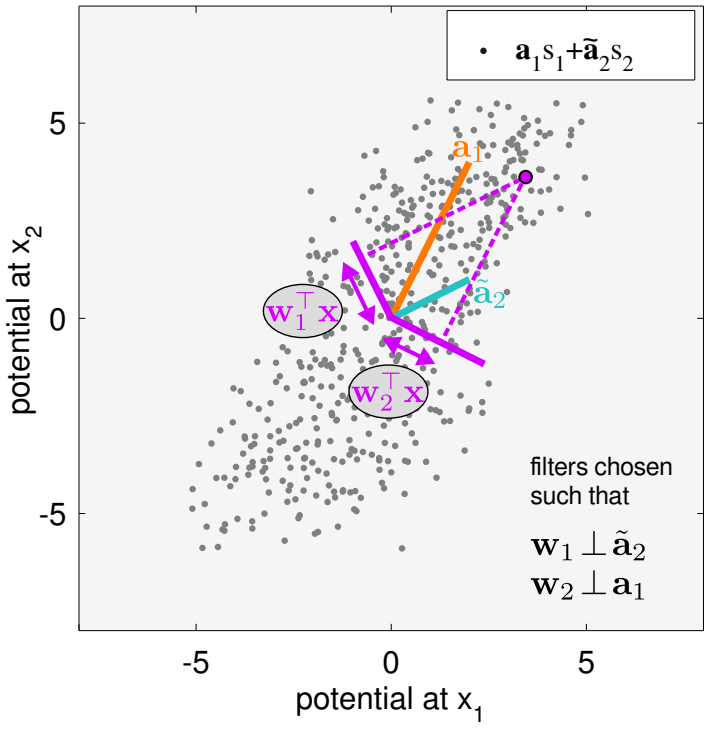
Explaining Spatial Patterns and Spatial Filters



$$\hat{s}_1 = \mathbf{a}_1^\top \mathbf{x}$$
$$= \mathbf{a}_1^\top \mathbf{a}_1 s_1 + \underbrace{\mathbf{a}_1^\top \mathbf{a}_2 s_2}_{\text{does not vanish}} + \mathbf{a}_1^\top \mathbf{n}$$

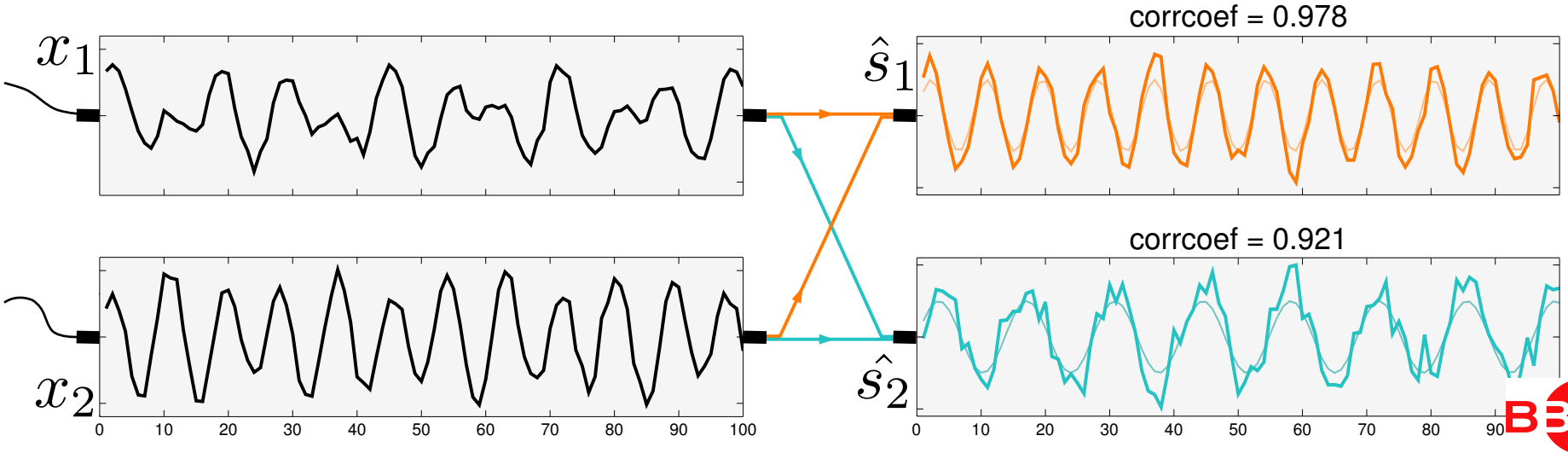


Explaining Spatial Patterns and Spatial Filters

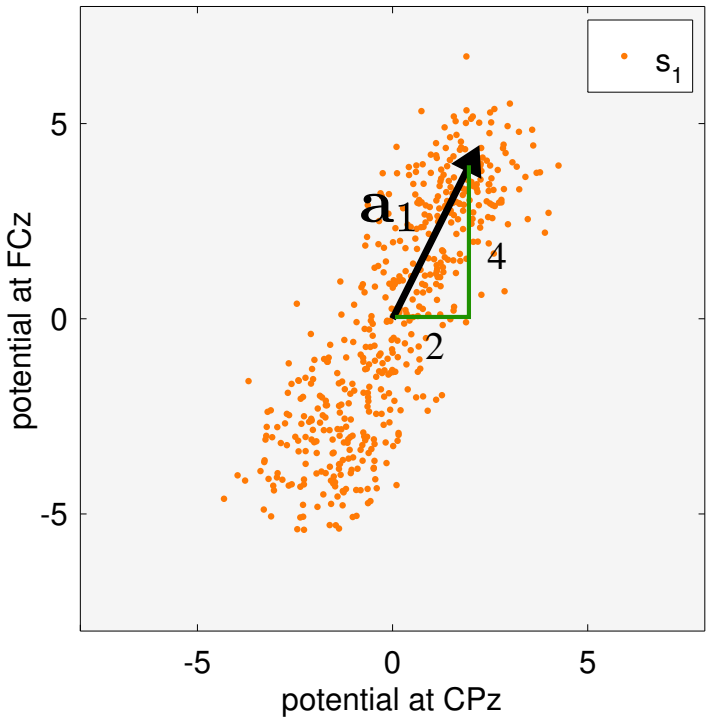


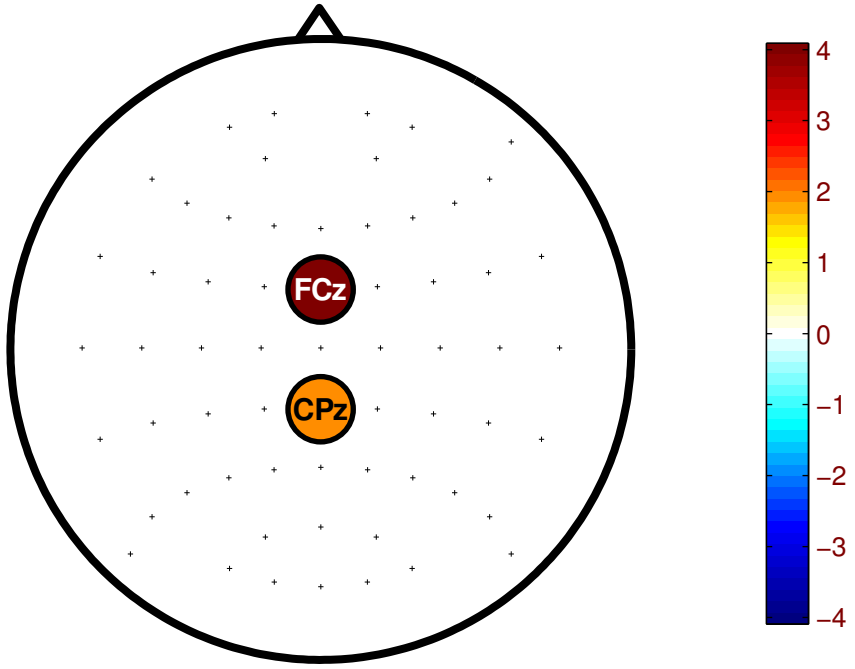
$$\hat{s}_1 = \mathbf{w}_1^\top \mathbf{x}$$
$$= \mathbf{w}_1^\top \mathbf{a}_1 s_1 + \underbrace{\mathbf{w}_1^\top \mathbf{a}_2}_{=0} s_2 + \mathbf{w}_1^\top \mathbf{n}$$
$$\sim s_1$$

$$\hat{s}_2 \sim s_2$$

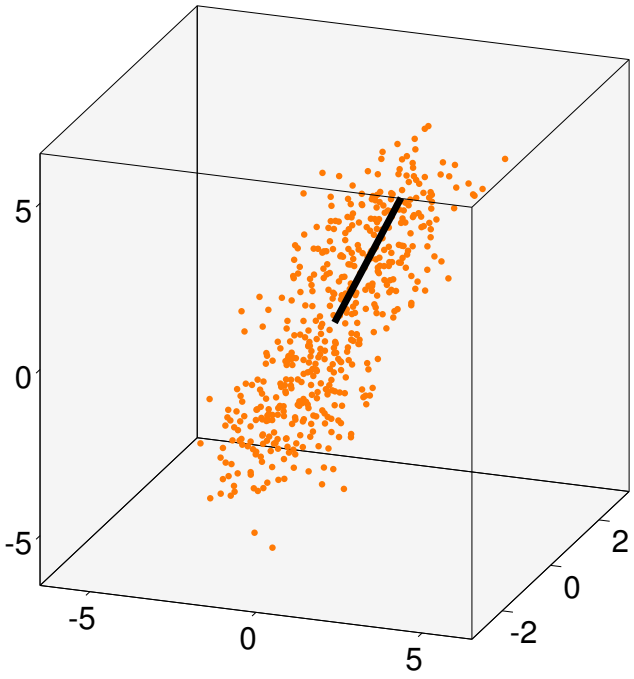


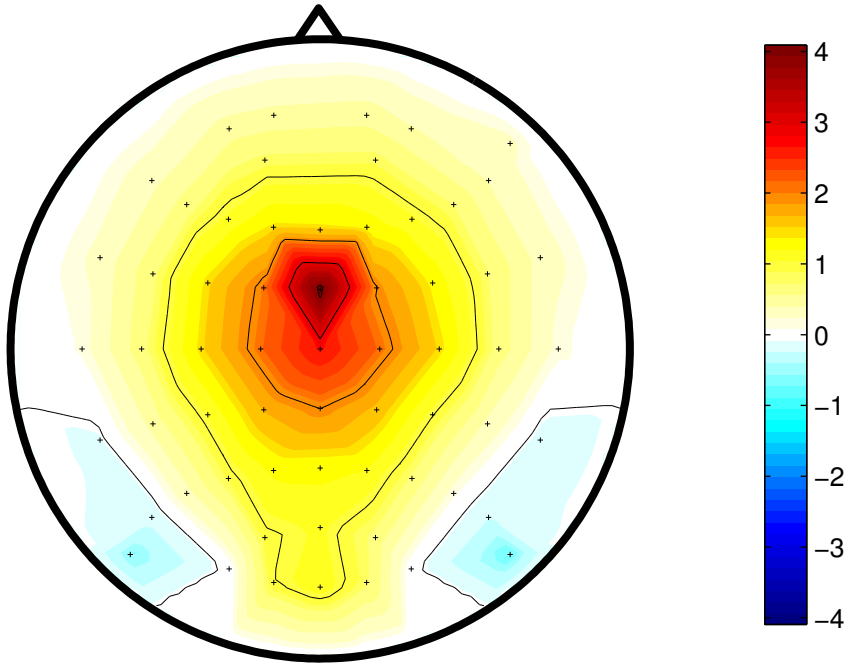
Correspondence of Vectors in Feature Space and Patterns





Correspondence of Vectors in Feature Space and Patterns





Left: High-dimensional features spaces are hard to visualize.
Right: A vector in the feature space (such as weight vectors) can be represented as a scalp topography (for spatial features).

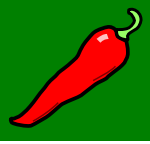
Part VII

Interpretability of Spatial Filters

The Linear Model Revisited

So far, we have seen that the spatial map of a filter (that is, e.g., obtained from a linear classifier) is difficult to interpret. Since patterns are straight forward to interpret, one is interested in finding that pattern which corresponds to a given spatial filter.

In order to derive which pattern corresponds to the filter that is obtained by a (Shrinkage-) LDA classifier from spatial features, we will consult the linear model.



Some Notions

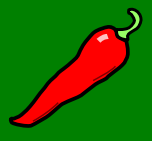
New notions:
In the following, we write $\mathbf{X}$ for $[\mathbf{x}(1), \ldots, \mathbf{x}(T)]$, $\mathbf{S}$ for $[\mathbf{s}(1), \ldots, \mathbf{s}(T)]$ and assume that the factors have zero mean.

To discriminate covariance matrices of different data matrices, we will use indices:

$$\mathbf{\Sigma_X} = \frac{1}{T-1}\mathbf{X}\mathbf{X}^\top, \qquad \mathbf{\Sigma_S} = \frac{1}{T-1}\mathbf{S}\mathbf{S}^\top$$

Remember the property of the pseudo inverse:
For source signals $\mathbf{S} \in \mathbb{R}^{K \times T}$, we assume that there are more time points T than sources K. In this case, $\mathbf{S}^+ = \mathbf{S}_R^{-1} = \mathbf{S}^\top(\mathbf{S}\mathbf{S}^\top)^{-1}$ and

$$\underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\| = \mathbf{X}\mathbf{S}^+ = \mathbf{X}\mathbf{S}^\top(\mathbf{S}\mathbf{S}^\top)^{-1}$$



Finding Patterns for given Spatial Filters

Let a filter matrix $\mathbf{W}$ be given and define $\mathbf{S} = \mathbf{W}^\top \mathbf{X}$. Then, we obtain the matrix of corresponding patterns $\hat{\mathbf{A}}$ by [Haufe et al, 2013]:

$$\hat{\mathbf{A}} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|^2 = \mathbf{X}\mathbf{S}^\top(\mathbf{S}\mathbf{S}^\top)^{-1} = \mathbf{X}\mathbf{X}^\top\mathbf{W}(\mathbf{S}\mathbf{S}^\top)^{-1}$$
$$\simeq \mathbf{\Sigma_x} \mathbf{W} \mathbf{\Sigma_s}^{-1}$$

- If $K = 1$ (in particular for LDA), we obtain

  $$\hat{\mathbf{a}} \simeq \mathbf{\Sigma_x} \mathbf{w}.$$

- If the factors are uncorrelated (e.g., PCA, ICA), we get

  $$\hat{\mathbf{A}} \simeq \mathbf{\Sigma_x} \mathbf{W}.$$

- The patterns and filters coincide if and only if additionally the observations $\mathbf{x}$ are uncorrelated

  $$\hat{\mathbf{A}} \simeq \mathbf{W}.$$

  However, this assumption is rather unrealistic for EEG due to volume conduction, as seen earlier (spatial smearing).

Part VIII

Issues in Validation

Hall of Shame in Single-Trial EEG Analysis

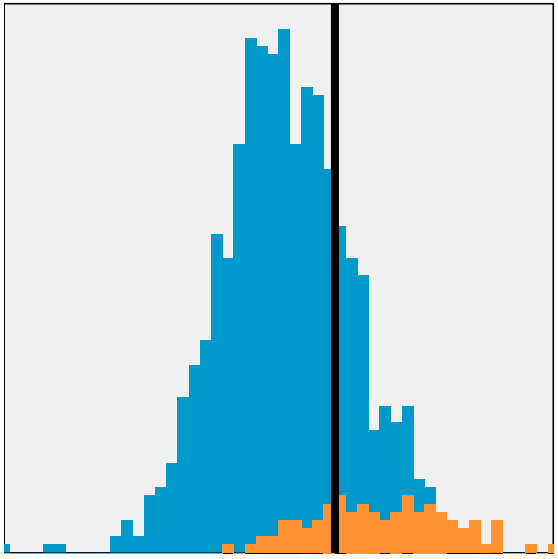
- preprocessing methods that use statistics of the whole data set like ICA, or normalization of features
  (particularly severe for methods that use label information)
- loss function not appropriate (e.g., unbalanced classes)
- artifacts/outliers are rejected from the whole data set (resulting in a simplified test set)
- features are selected on the whole data set, including trials that are later in the test set
- selection of parameters by cross validation on the whole data set and report the performance for the selected values
- non-stationarity of the data disregarded (chronological training / test data spilt vs. cross validation)
- insufficient validation for paradigms with block design

[Lemm et al, NeuroImage 2011]

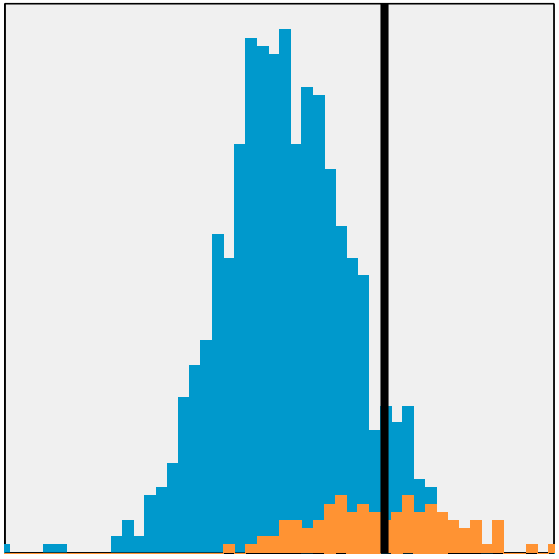
Loss Functions for Unbalanced Classes

Blue class: $N_1 = 900$ samples, orange class: $N_2 = 100$ samples.
Weighted error: $\mathrm{err}_{\mathrm{weighted}} = \frac{1}{2}\left(\mathrm{err}|_{\mathrm{class\ 1}} + \mathrm{err}|_{\mathrm{class\ 2}}\right)$

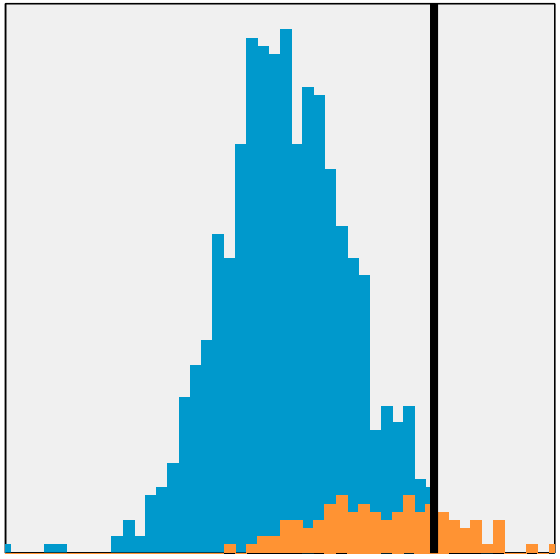
Examples of weighted and unweighted error – bias of classifier is varied:



Error rate
Unweighted: 23.6%
Weighted: 25.1%
AUC-based: 16.6%



Error rate
Unweighted: 12.8%
Weighted: 30.0%
AUC-based: 16.6%



Error rate
Unweighted: 9.5%
Weighted: 39.5%
AUC-based: 16.6%

References I

- Acqualagna, L. and Blankertz, B. (2011).
  A gaze independent speller based on rapid serial visual presentation.
  In *Conf Proc IEEE Eng Med Biol Soc*, volume 2011, pages 4560–4563.

- Blankertz, B., Lemm, S., Treder, M. S., Haufe, S., and Müller, K.-R. (2011).
  Single-trial analysis and classification of ERP components – a tutorial.
  Neuroimage, 56:814–825.

- Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., and Bießmann, F. (2014).
  On the interpretation of weight vectors of linear models in multivariate neuroimaging.
  Neuroimage, 87:96–110.

- Ledoit, O. and Wolf, M. (2004).
  A well-conditioned estimator for large-dimensional covariance matrices.
  J Multivar Anal, 88:365–411.

- Lemm, S., Blankertz, B., Dickhaus, T., and Müller, K.-R. (2011).
  Introduction to machine learning for brain imaging.
  Neuroimage, 56:387–399.

- Parra, L. C., Spence, C. D., Gerson, A. D., and Sajda, P. (2005).
  Recipes for the linear analysis of EEG.
  Neuroimage, 28(2):326–341.

- Schäfer, J. and Strimmer, K. (2005).
  A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics.
  Stat Appl Genet Mol Biol, 4:Article32.

References II

⋆ Scholler, S., Bosse, S., Treder, M. S., Blankertz, B., Curio, G., Müller, K.-R., and Wiegand, T. (2012).
Towards a direct measure of video quality perception using EEG.
IEEE Trans Image Process, 21(5):2619–2629.