Reporting results with confidence II

Multivariate approach

Aalto-yliopisto
Aalto-universitetet
Aalto University

Marijn van Vliet

Practical MEEG 3rd Edition

MNE MEG + EEG ANALYSIS & VISUALIZATION

31 October 2025

Summary

1. Uses for multivariate analysis
2. What machine models do: regression & classification
3. The curse of dimensionality, overfitting, & cross-validation
4. Spatial filtering, decoding across time
5. Determining statistical significance of the result
6. Interpreting the model: weights vs patterns
7. Representational spaces: tracking information content

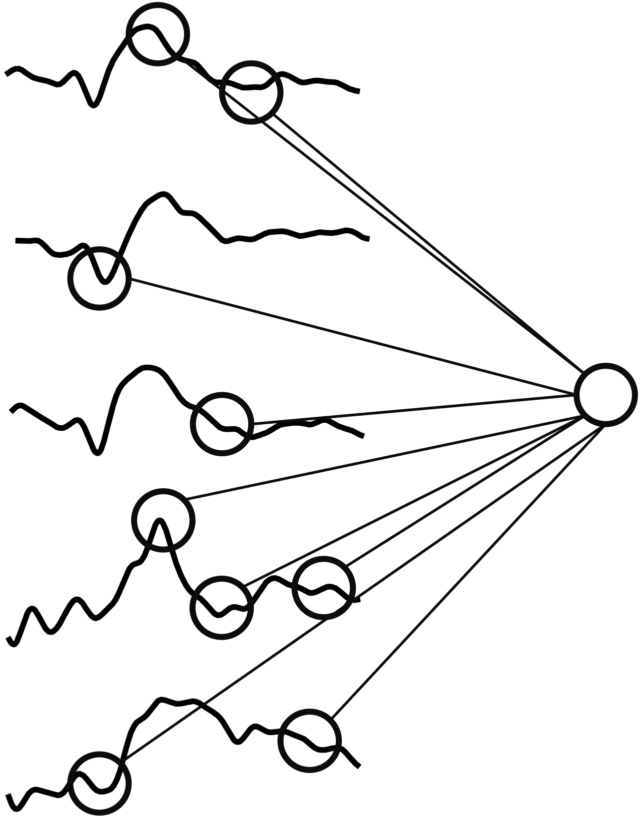
Summary

1. **Uses for multivariate analysis**
2. What linear machine models do: regression & classification
3. The curse of dimensionality, overfitting, & cross-validation
4. Spatial filtering, decoding across time
5. Determining statistical significance of the result
6. Interpreting the model: weights vs patterns
7. Representational spaces: tracking information content

Multivariate means combining multiple data points into a single value

We have already seen some examples:

- Independent component analysis (ICA)
- Source estimation
- Averaging across a region of interest (ROI)
- General linear model (GLM) with design matrix



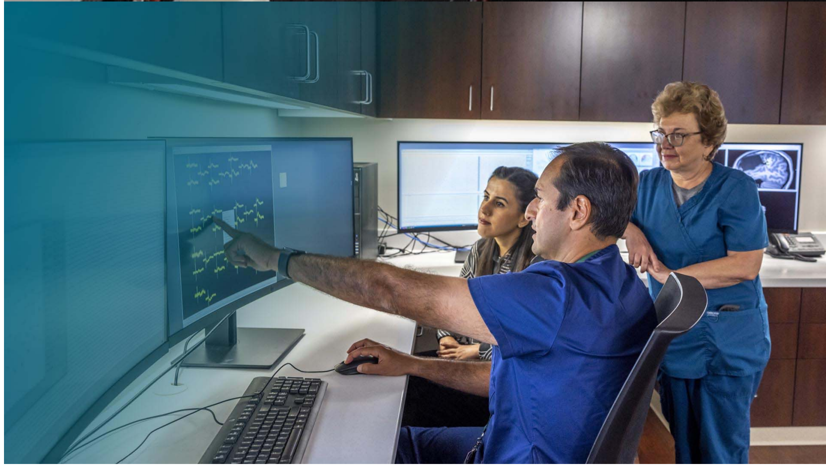
In this lecture we will focus on:

- Machine learning
  - Estimating the best way to combine data points to achieve some stated goal.

Applications of machine learning in EEG/MEG data analysis

Automated decision making:

- Brain-computer interfaces (BCI)
- Biomarkers / diagnostics



Applications of machine learning in EEG/MEG data analysis

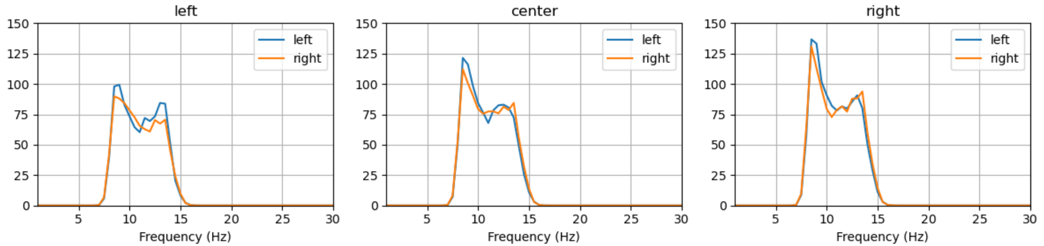
Automated decision making:

- Brain-computer interfaces (BCI)
- Biomarkers / diagnostics

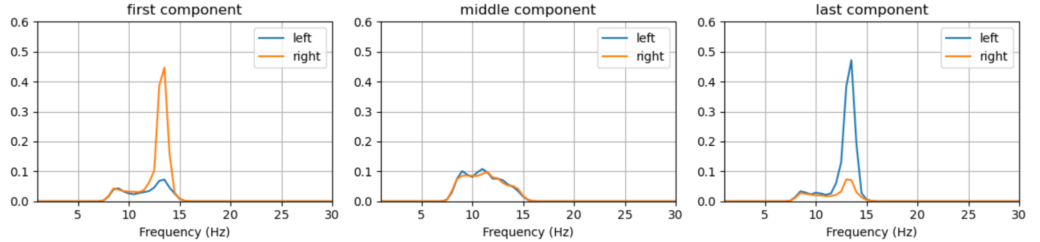
Filtering

Isolating informative signal components

Imagined movement. Three EEG electrodes:



After filtering with common spatial patterns (CSD):



source: BCI competition IV data, Berlin BCI group

Applications of machine learning in EEG/MEG data analysis

Automated decision making:

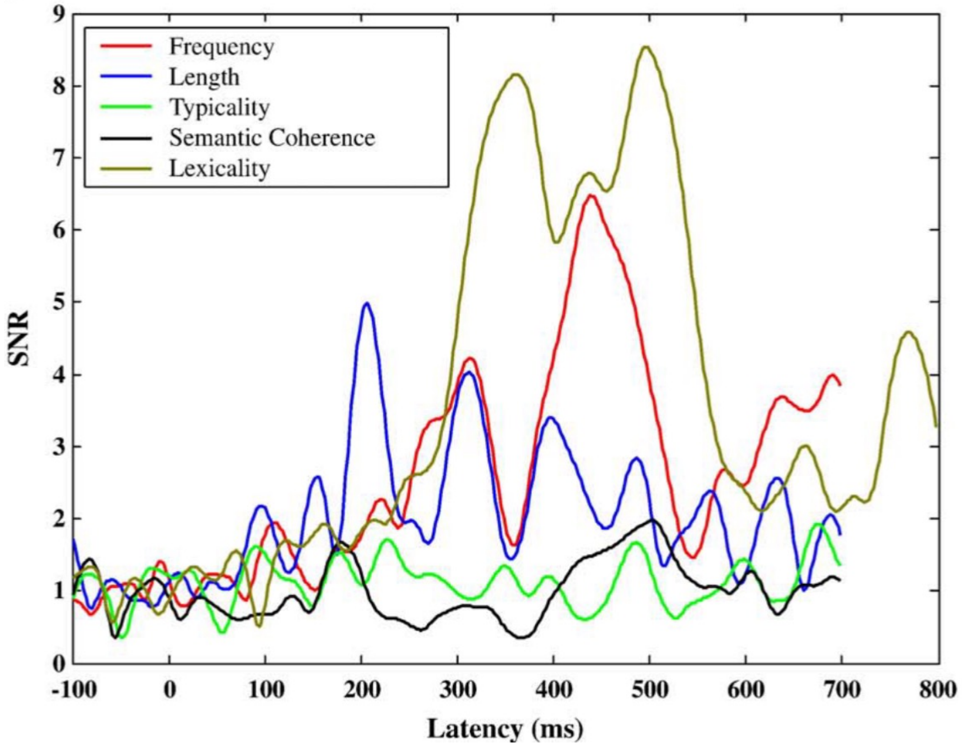
- Brain-computer interfaces (BCI)
- Biomarkers / diagnostics

Filtering

Isolating informative signal components

Information-based analysis

When/where is information about X present in the signal?

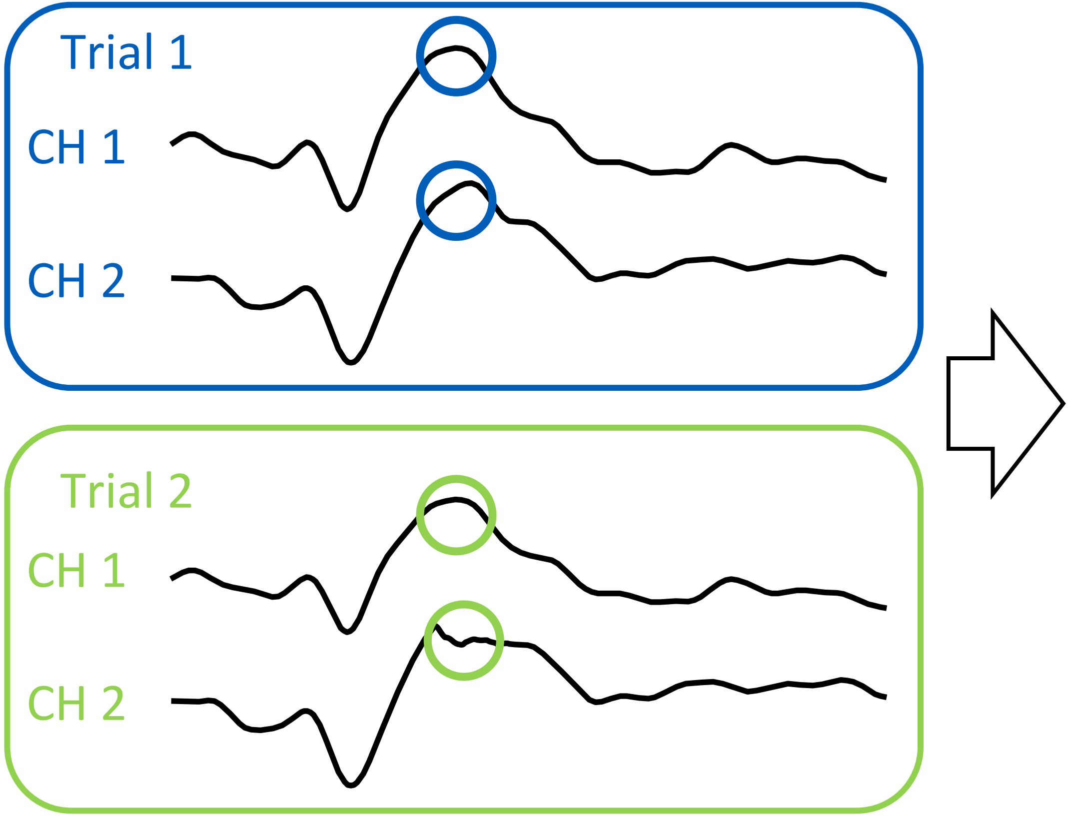


Hauk et al., NeuroImage 2005

Summary

1. Uses for multivariate analysis
2. **What linear machine models do: regression & classification**
3. The curse of dimensionality, overfitting, & cross-validation
4. Spatial filtering, decoding across time
5. Determining statistical significance of the result
6. Interpreting the model: weights vs patterns
7. Representational spaces: tracking information content

How machine learning sees your data



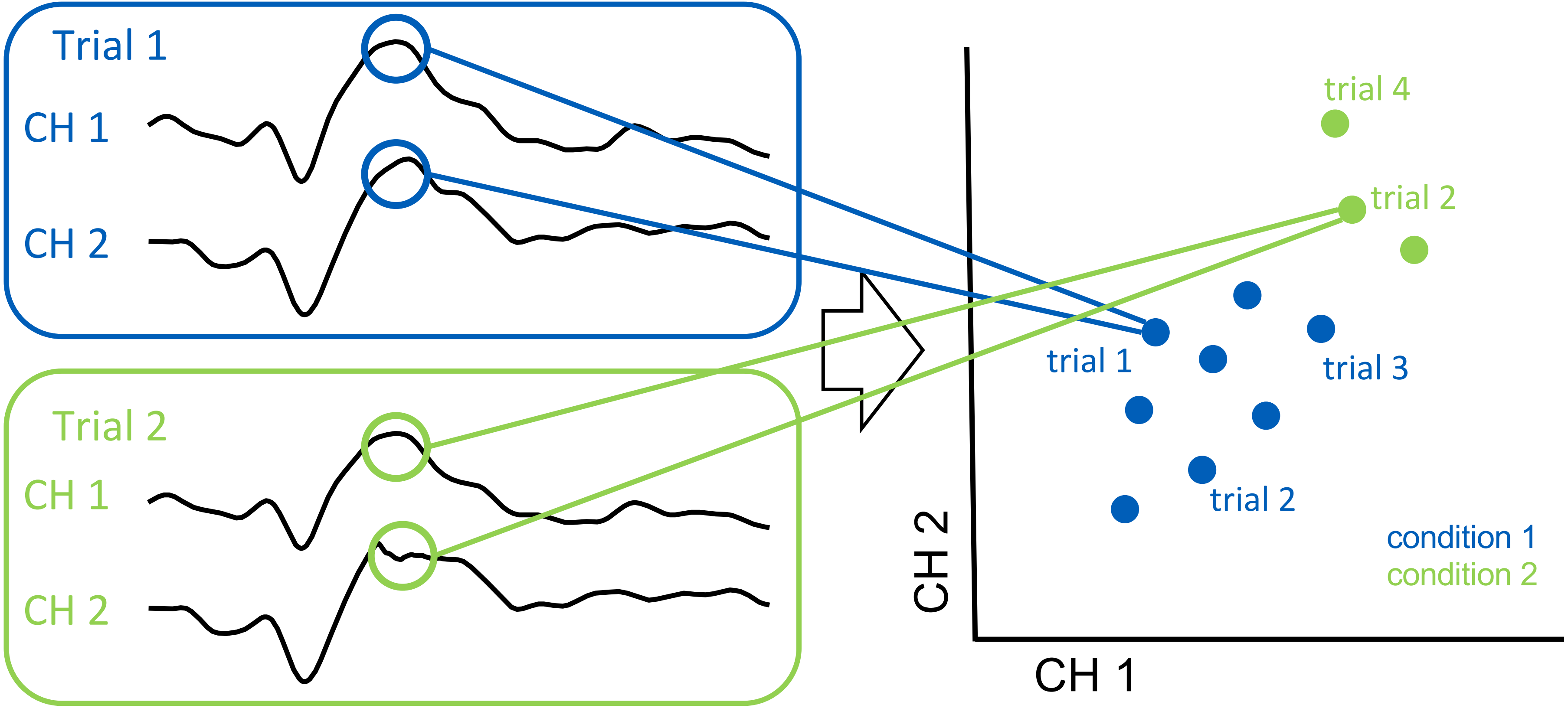
X features →

Y targets →

observations (trials) ↓

CH 1	CH 2	condition
0.1	0.3	1
0.3	0.5	2
0.2	0.6	1
1.2	1.1	1
1.7	2.0	2

How machine learning sees your data

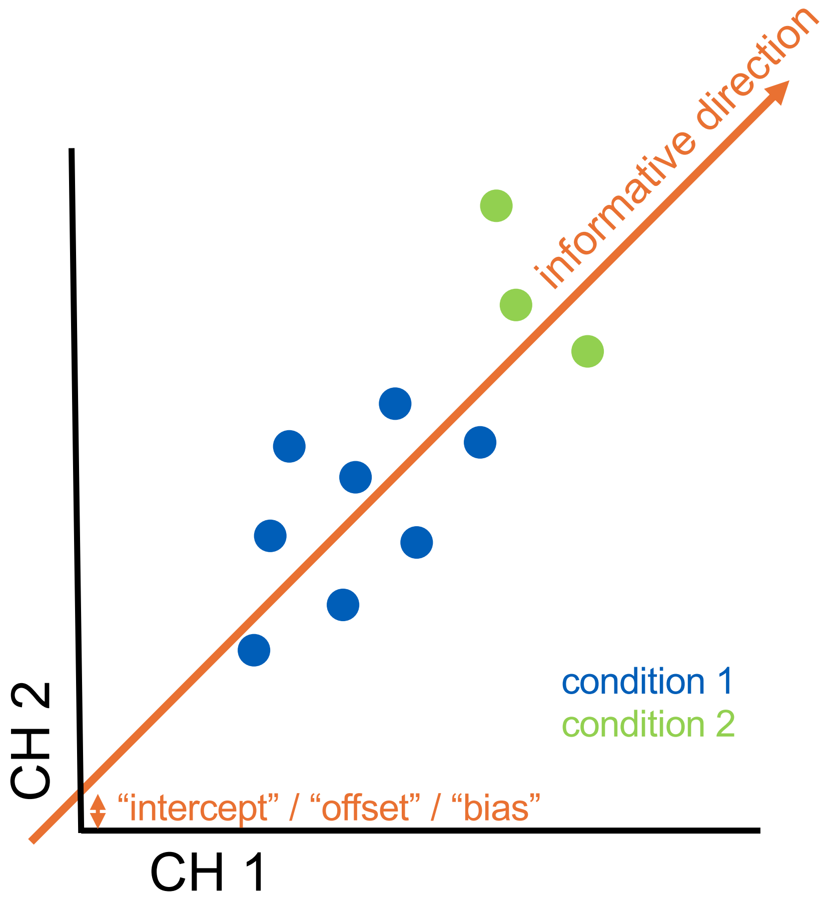


Linear regression and classification

1. Find most informative direction
   = regression weights

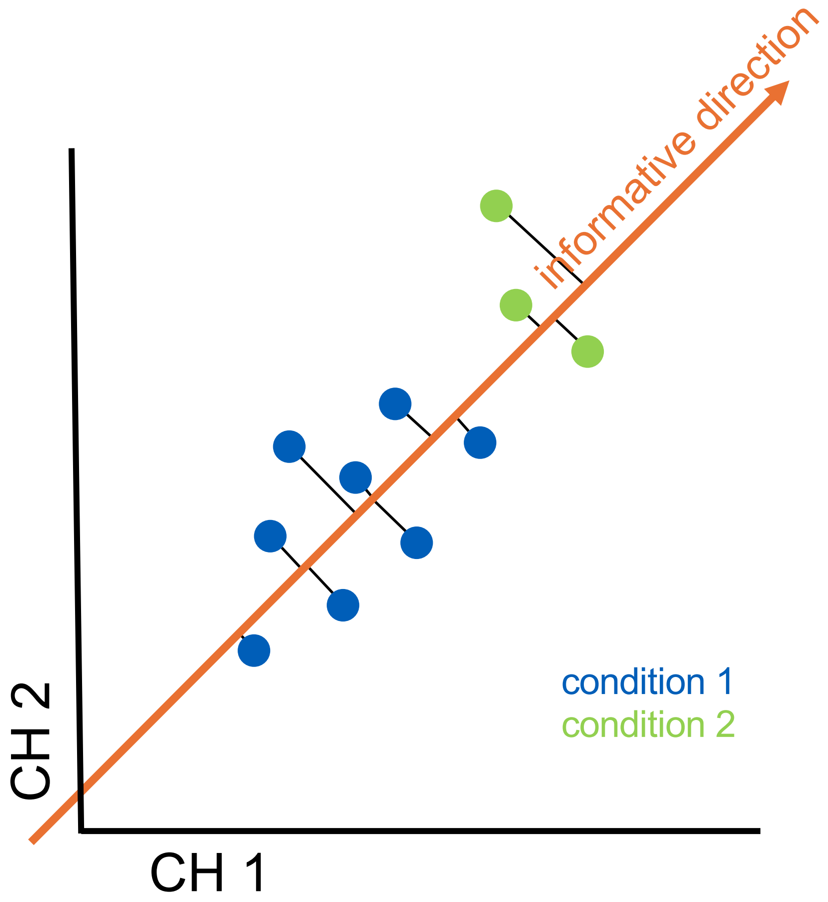
many different algorithms for this:

- Regression:
  - Ordinary Least Squares (OLS)
  - Ridge Regression
- Classification:
  - Logistic Regression (LR)
  - Support Vector Machine (SVM)
  - Linear Discriminant Analysis (LDA)
- …and many more



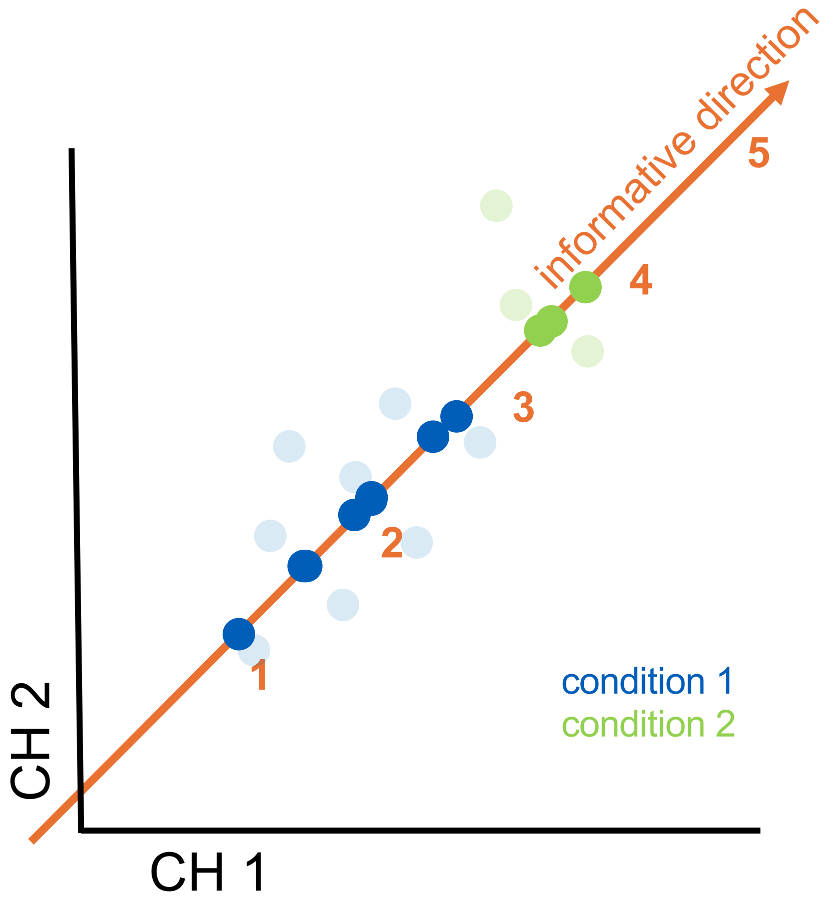
Linear regression and classification

1. Find most informative direction
   = regression weights
2. Project data onto this line



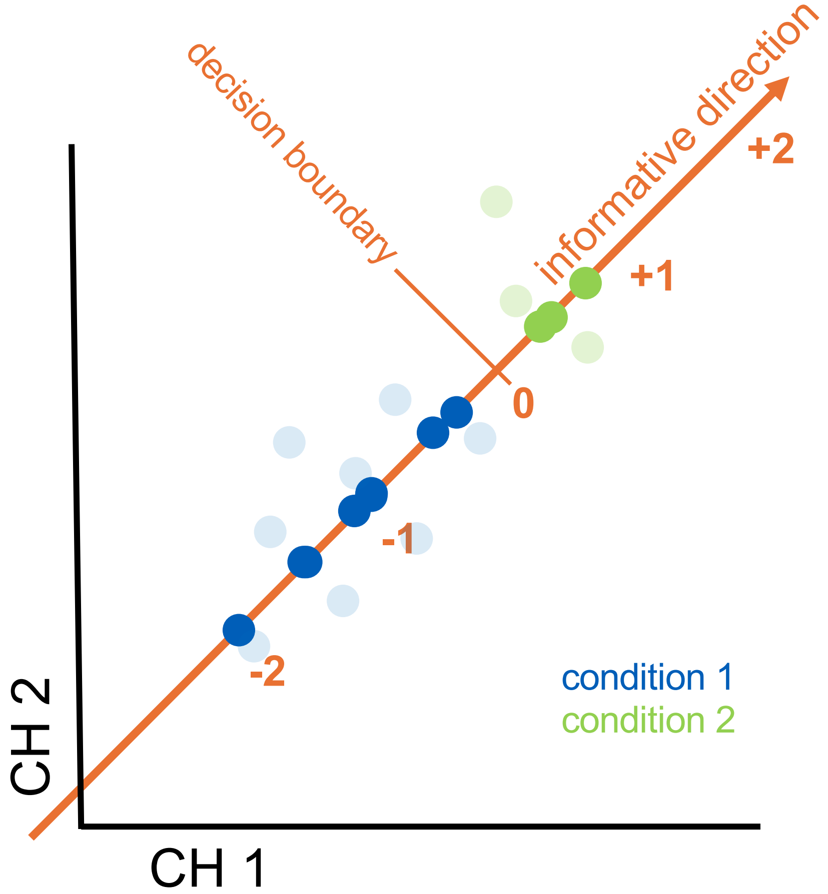
Linear regression and classification

1. Find most informative direction = regression weights
2. Project data onto this line
3. Regression result is position along the line



Linear regression and classification

1. Find most informative direction = regression weights
2. Project data onto this line
3. Regression result is position along the line
4. For binary classification: draw decision boundary, call this position “0”

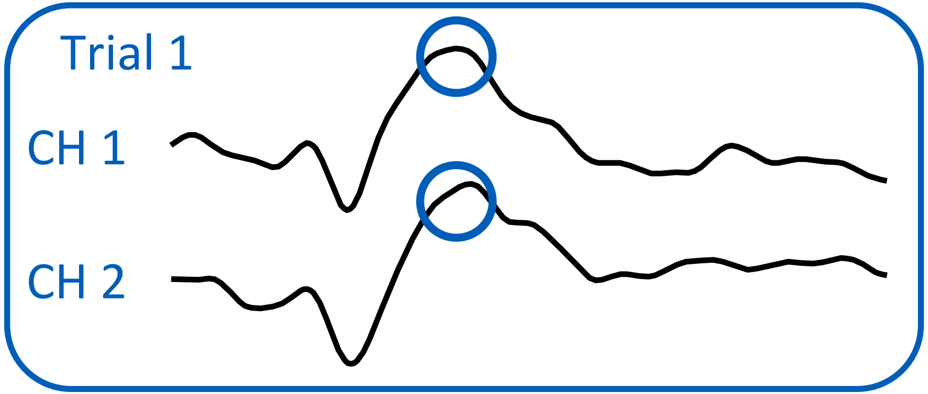


Summary

1. Uses for multivariate analysis
2. What linear machine models do: regression & classification
3. **The curse of dimensionality, overfitting, & cross-validation**
4. Spatial filtering, decoding across time
5. Determining statistical significance of the result
6. Interpreting the model: weights vs patterns
7. Representational spaces: tracking information content

The curse of dimensionality

Two channels, one time points
2 columns = 2 features = 2 dimensions



X

features →

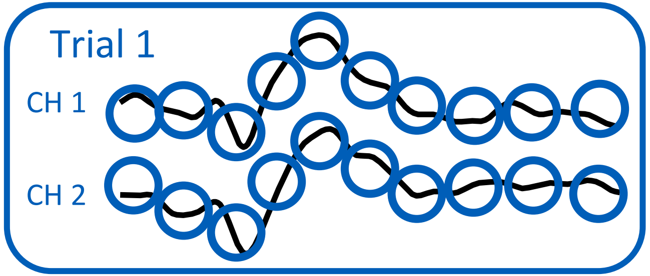
observations (trials) ↓

CH 1	CH 2
0.1	0.3
0.3	0.5
0.2	0.6
1.2	1.1
1.7	2.0

The curse of dimensionality

Two channels, ten time points

2 x 10 = 20 features/dimensions



features →

X

observations (trials) →

C1, T1	C2,T 1	C1,T 2	C2,T 2	C1,T 3	C2,T 3	C1,T 4	C2,T 4	C1,T 5	C2,T 5	C1,T 6	C2,T 6	C1,T 7	C2,T 7	C1,T 8	C2,T 8	C1,T 9	C2,T 9	C1,T 10	C2,T 10

The curse of dimensionality

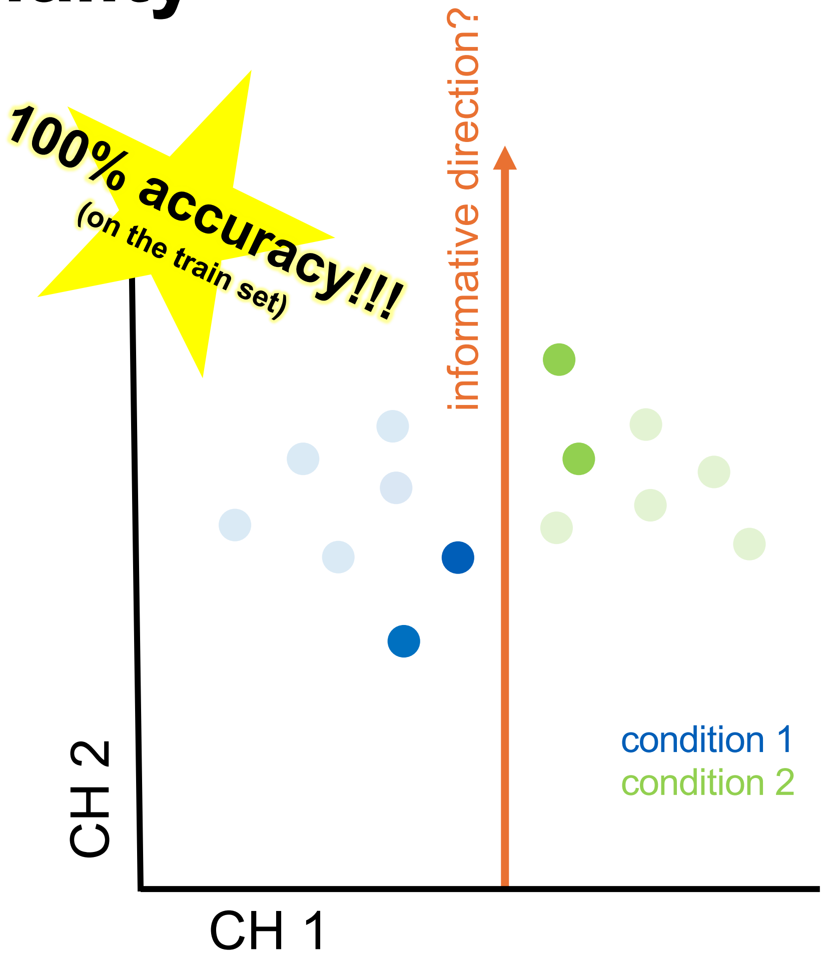
If you have few observations relative to the number of features, performance will seem amazing!

This is called “over-fitting”

Always split your data into separate “train” and “test” sets!

Cross-validation:

1. Split data into *k* sets
2. Use 1 set as test set and the other *k*-1 sets combined as train set (=“fold”)
3. Use average accuracy across all folds.



The curse of dimensionality

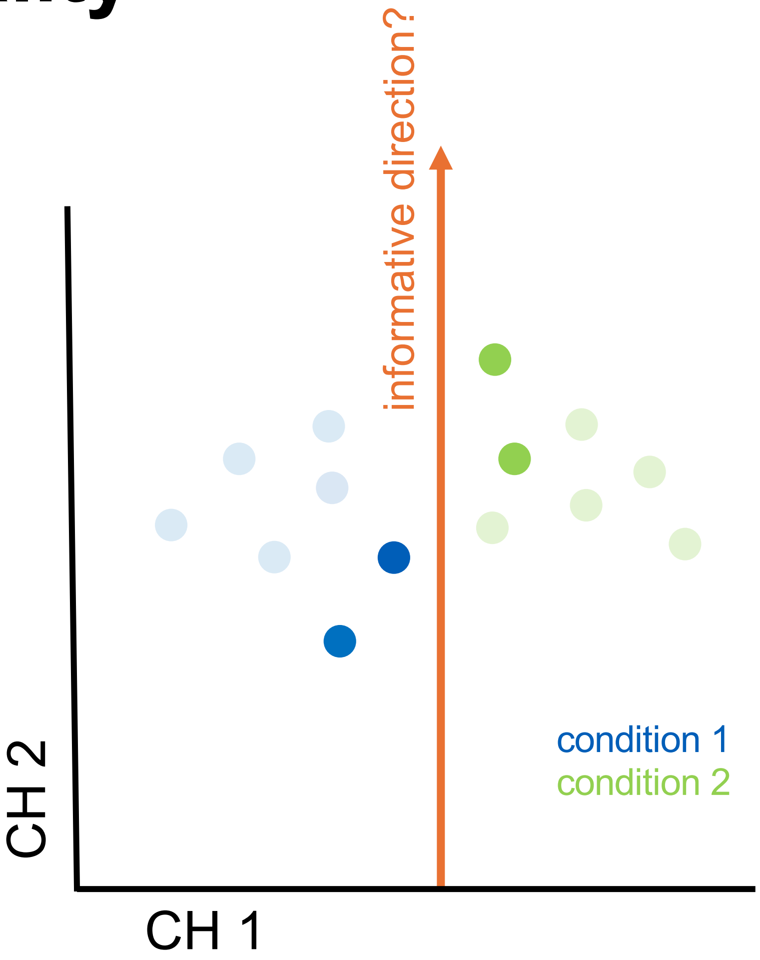
Ideally, you have many more observations than features.

In practice, we often don't:

For example:
64 channels x 100 time points = 6 400 features
we don't have >> 6 400 trials

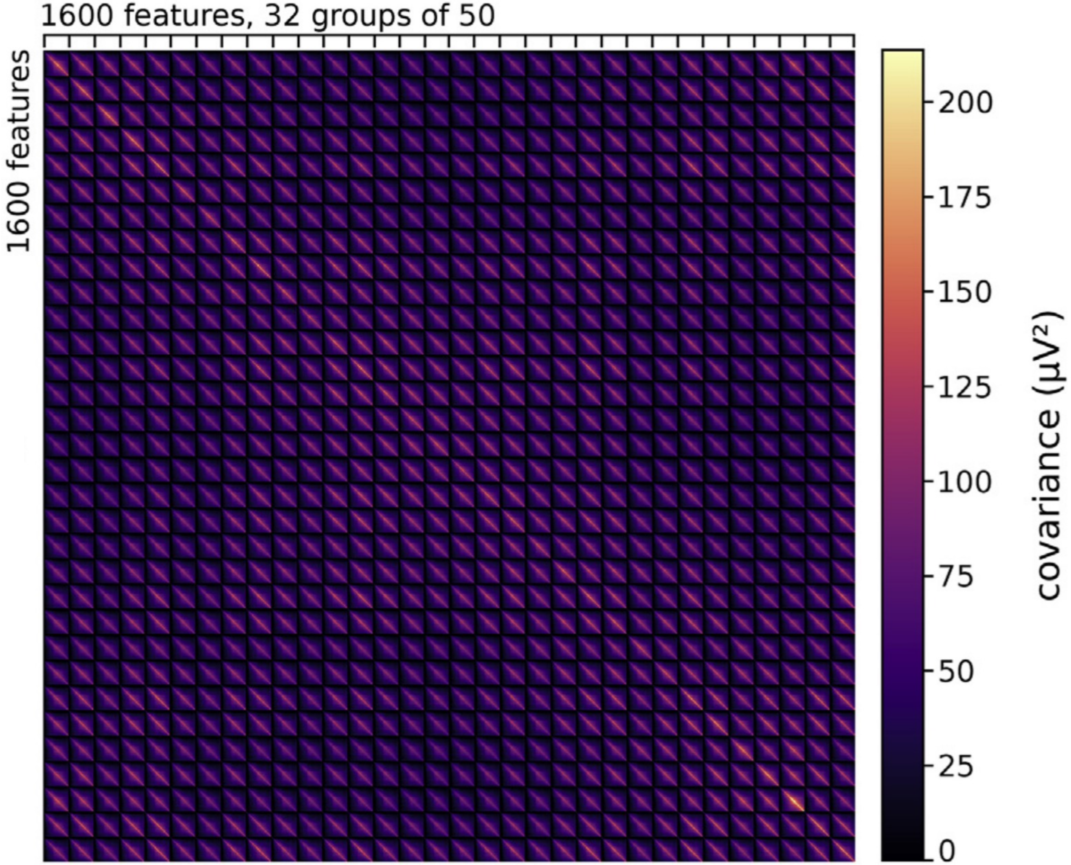
Solutions:

- Downsampling, binning
- Spatial filtering to one channel
- Use subset of data, e.g. decoding over time
- Regularization (almost always a good idea)



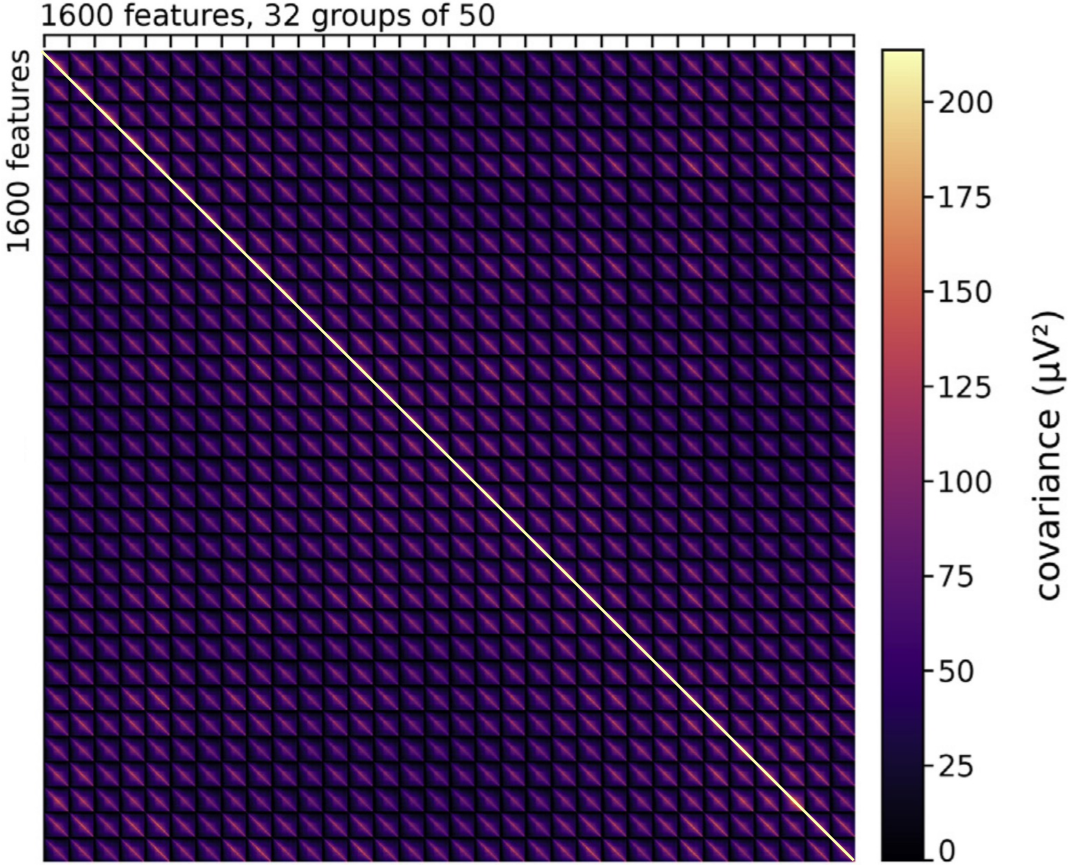
Regularization of the co-variance matrix

1 trial = 32 channels, 50 time points = 1600 features



Regularization of the co-variance matrix

1 trial = 32 channels, 50 time points = 1600 features



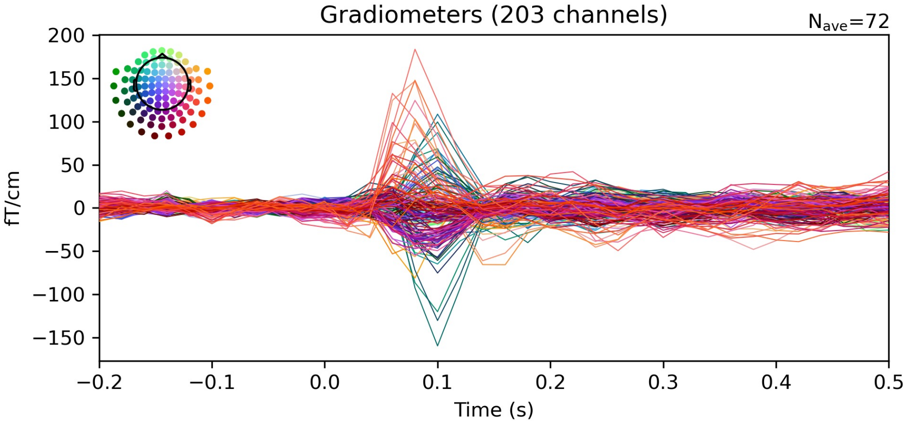
Summary

1. Uses for multivariate analysis
2. What linear machine models do: regression & classification
3. The curse of dimensionality, overfitting, & cross-validation
4. **Spatial filtering, decoding across time**
5. Determining statistical significance of the result
6. Interpreting the model: weights vs patterns
7. Representational spaces: tracking information content

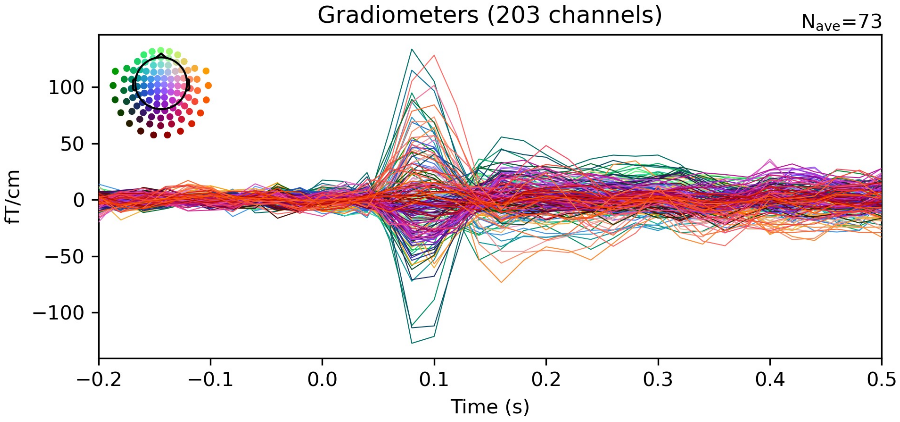
Spatial filtering

MNE-Python Sample Dataset

Auditory beep, left speaker



Auditory beep, right speaker



Goal: find a mixture of channels that is informative of the left vs right condition

Spatial filtering

X

channels →

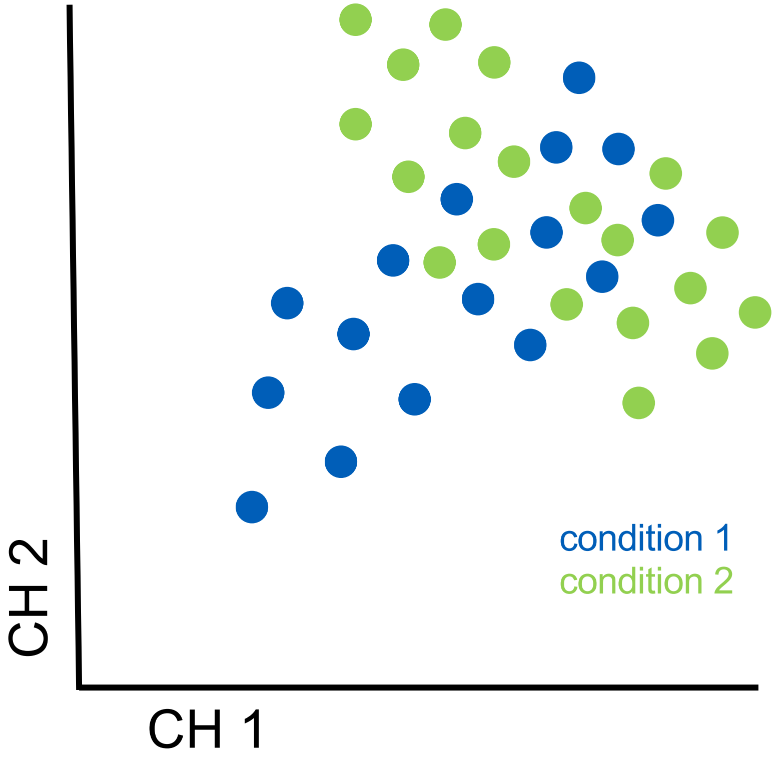
samples ↓

CH 1	CH 2	
0.1	0.3	trial 1
0.3	0.5	trial 1
0.2	0.6	trial 1
1.2	1.1	trial 1
1.7	2.0	trial 1
0.2	0.3	trial 2
0.3	0.4	trial 2
0.3	0.3	trial 2
0.4	0.6	trial 2
1.3	1.1	trial 2

Y

targets →

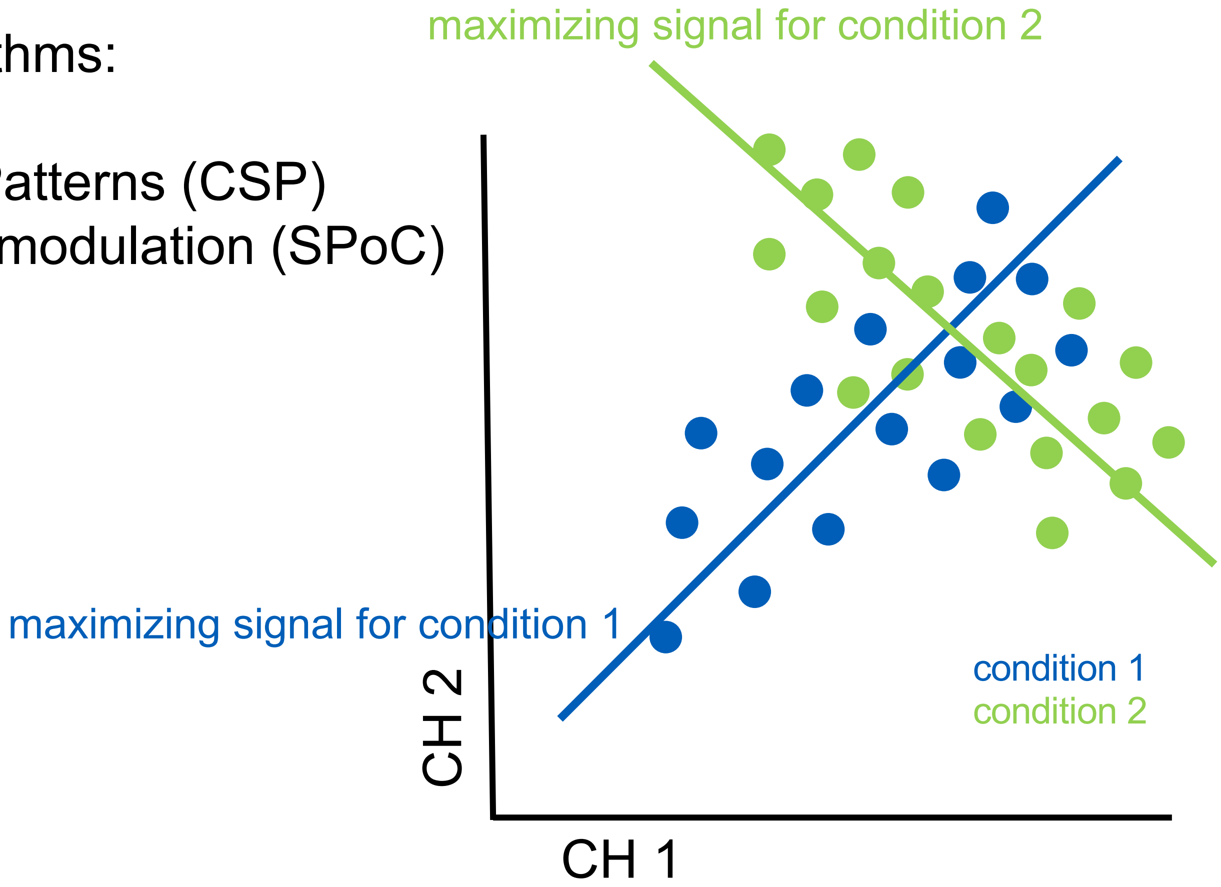
condition
1
1
1
1
1
2
2
2
2
2



Spatial filtering

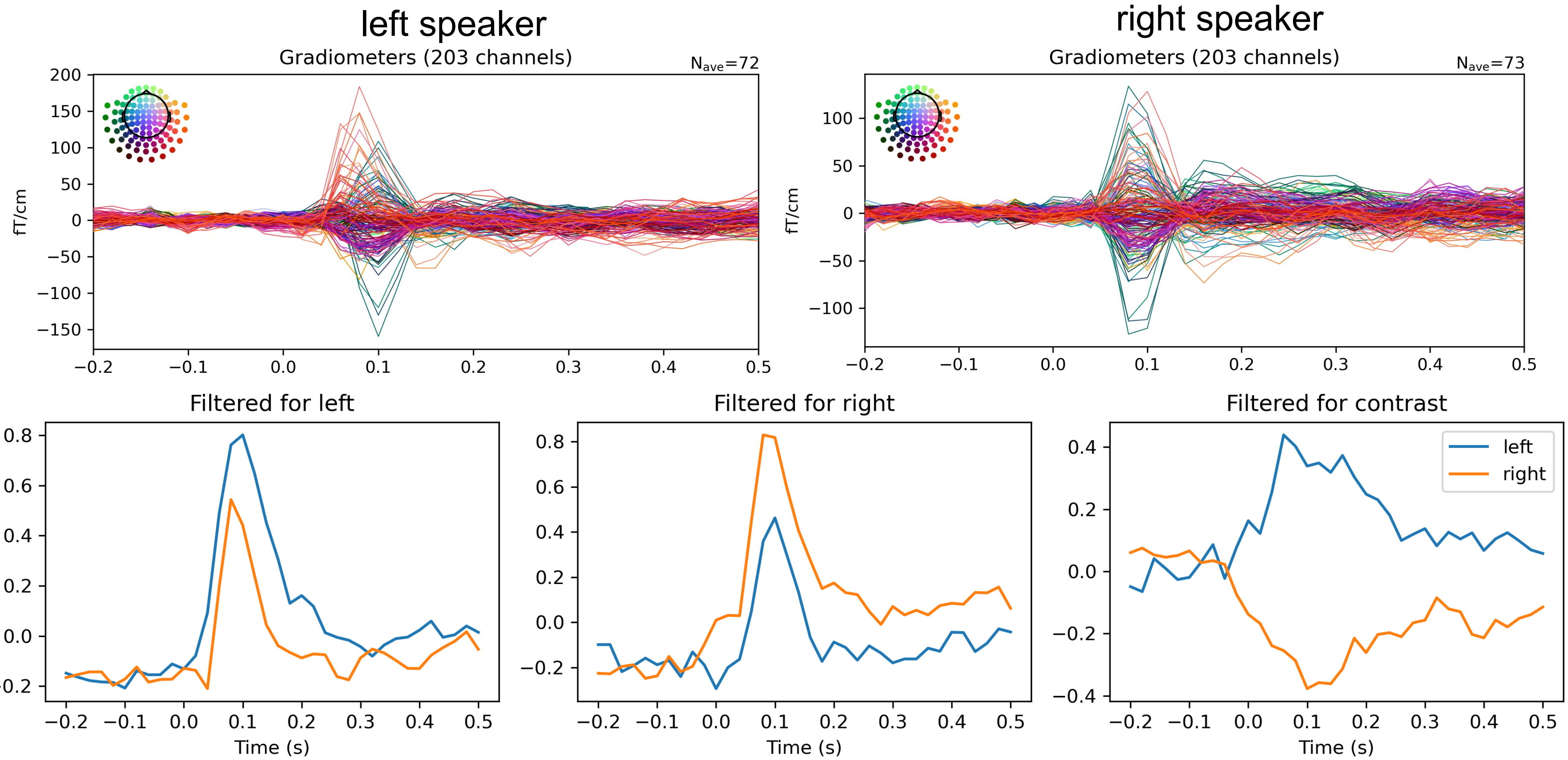
Some example algorithms:

- Common Spatial Patterns (CSP)
- Source Power Co-modulation (SPoC)
- LDA Beamformer



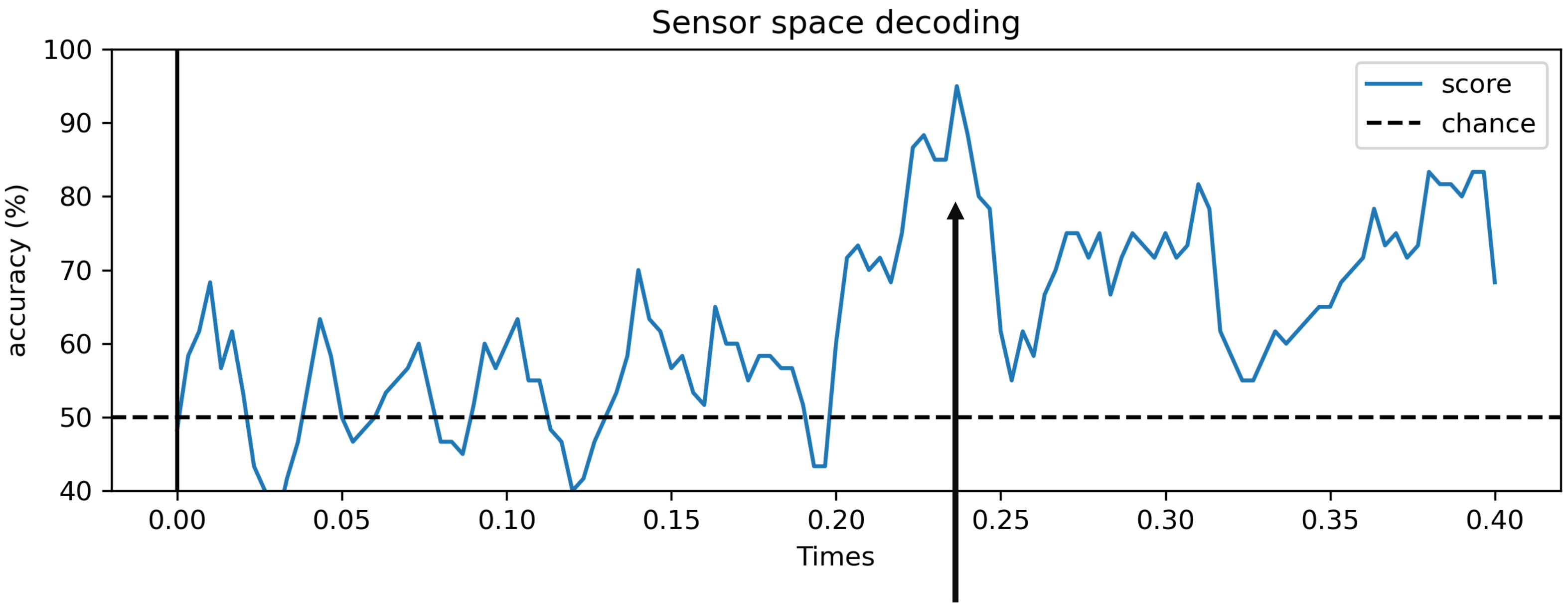
Spatial filtering

LDA beamformer



Decoding across time

- 204 channels (the gradiometers) x 1 time point
- Separate model for each time point (logistic regression)
- 5-fold cross-validation



Information distinguishing face vs scrambled

Summary

1. Uses for multivariate analysis
2. What linear machine models do: regression & classification
3. The curse of dimensionality, overfitting, & cross-validation
4. Spatial filtering, decoding across time
5. **Determining statistical significance of the result**
6. Interpreting the model: weights vs patterns
7. Representational spaces: tracking information content

Decoding the Wakeman & Henson data

- Face versus Scrambled, binary classification
- Linear Discriminant Analysis (LDA) as machine learning algorithm
- 204 channels (the gradiometers) x 121 time points
- 5-fold cross-validation
- Just for sub-01

Accuracy (mean across 5 folds): 65%
(this is terrible, a Logistic Regression classifier will do much better)

Questions:
1. Is our accuracy statistically significantly better than chance (=50%)?
2. What information did our machine learning model use?

Statistical testing: random permutations

Actual data:

features → targets →

CH 1	CH 2	condition
0.1	0.3	1
0.3	0.5	2
0.2	0.6	1
1.2	1.1	1
1.7	2.0	2

observations (trials) ↓

Randomly permuted data:

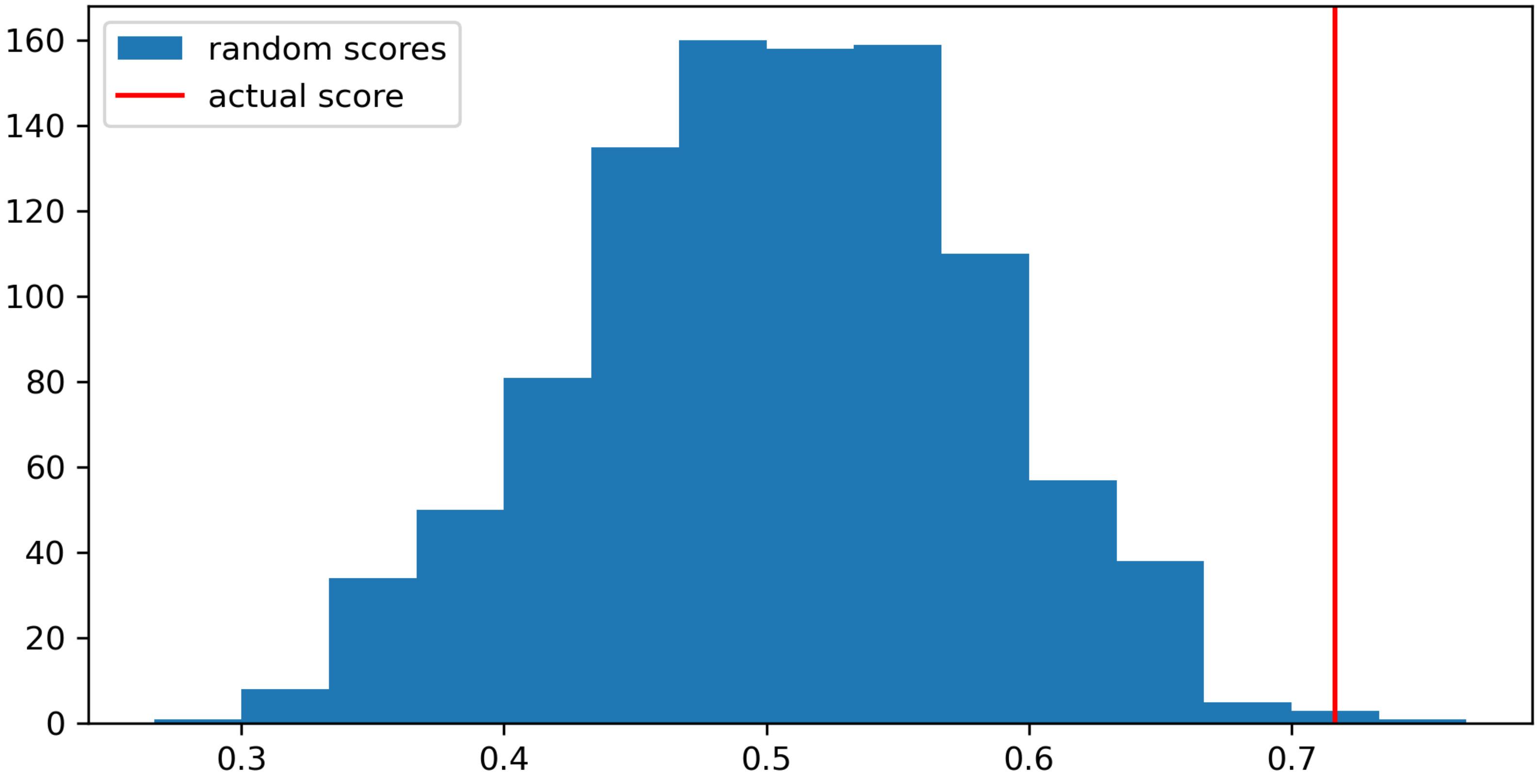
features → targets →

CH 1	CH 2	condition
0.1	0.3	2
0.3	0.5	1
0.2	0.6	1
1.2	1.1	2
1.7	2.0	1

observations (trials) ↓

shuffled

Statistical testing: random permutations



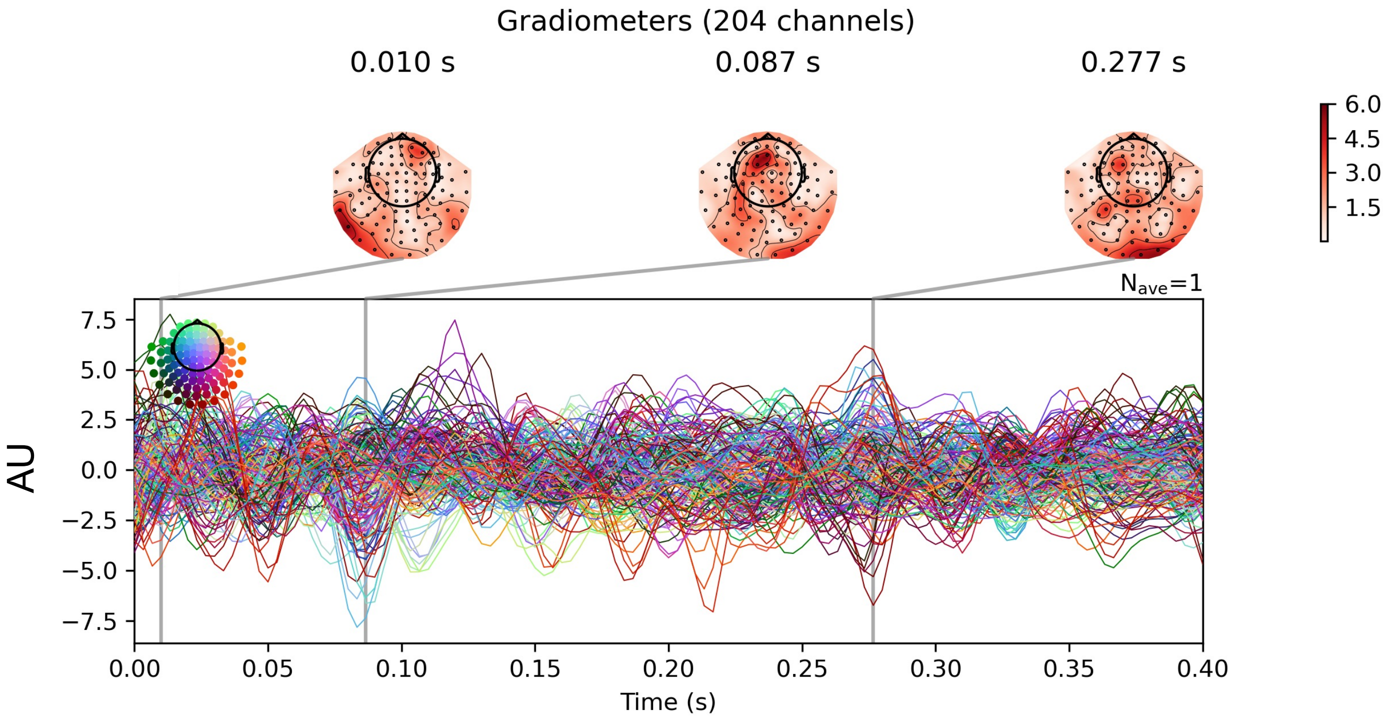
P < 0.001

Summary

1. Uses for multivariate analysis
2. What linear machine models do: regression & classification
3. The curse of dimensionality, overfitting, & cross-validation
4. Spatial filtering, decoding across time
5. Determining statistical significance of the result
6. **Interpreting the model: weights vs patterns**
7. Representational spaces: tracking information content

Visualizing regression weights

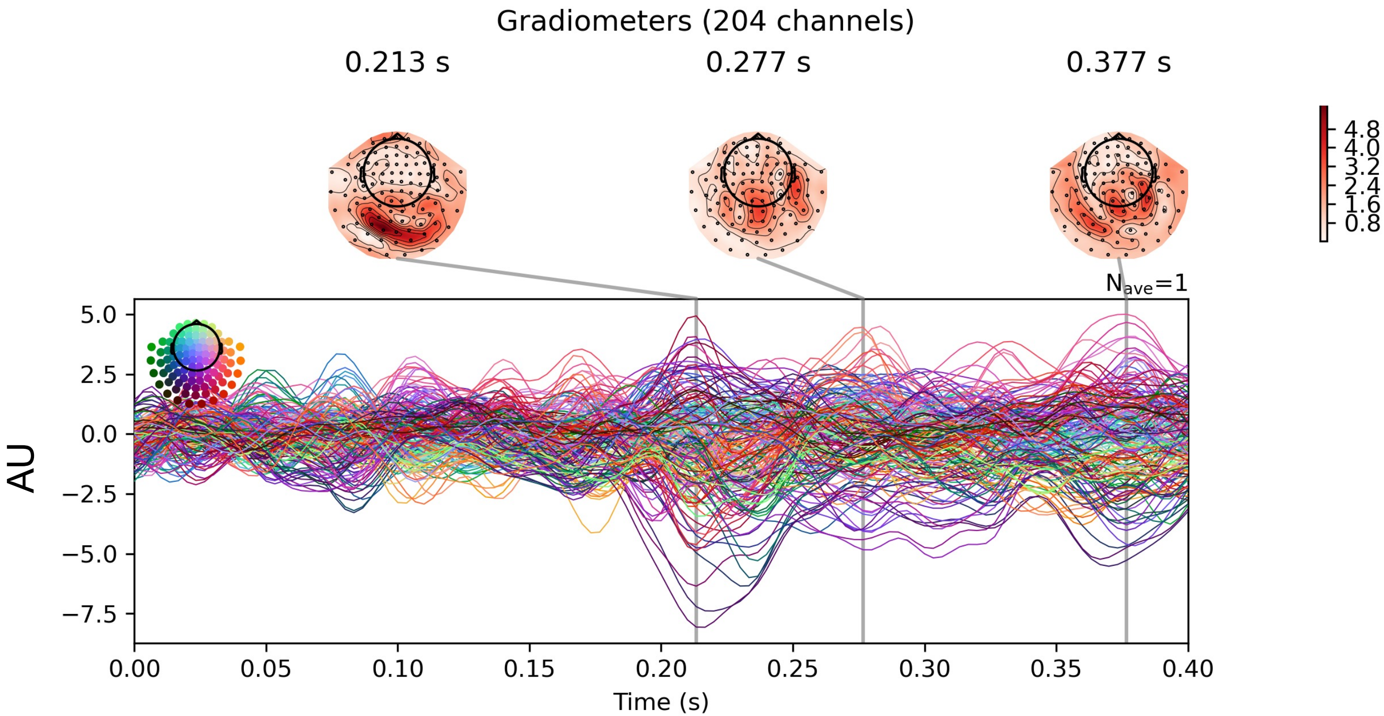
Plotting the weights as if they were an evoked response:



large weight ≠ contains useful information (face vs. scrambled)

Visualizing patterns

$P = cov(X)\ W\ cov^{-1}(Y)$ see: Haufe et al. NeuroImage (2014)

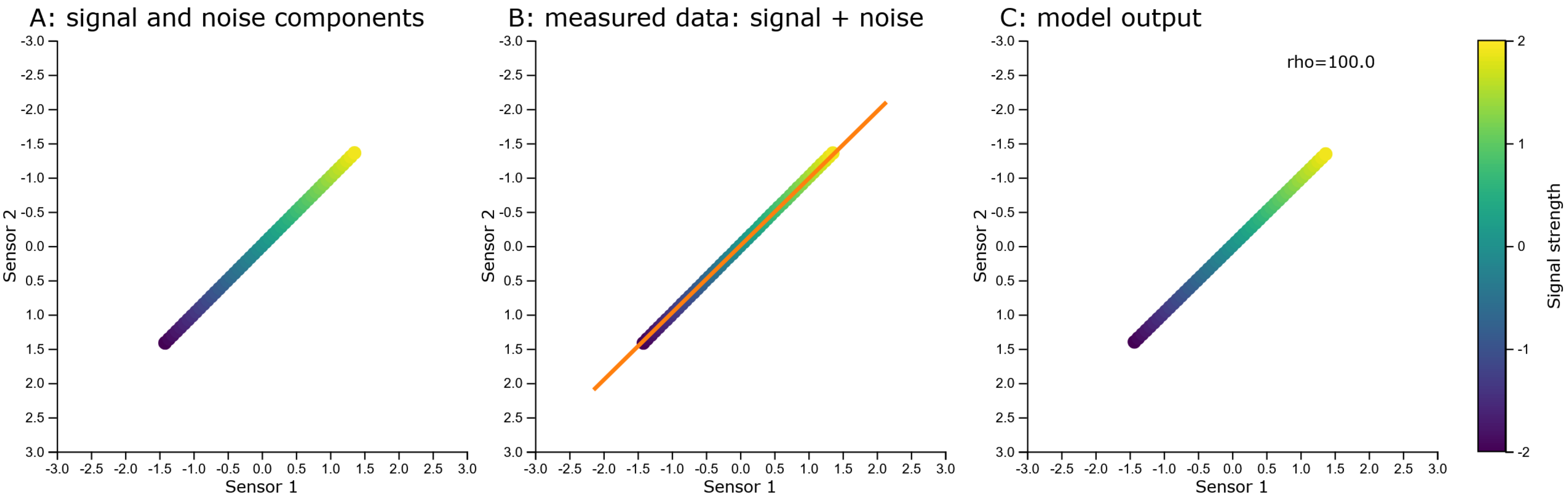


large pattern = contains useful information (face vs. scrambled)

Interactive explanation of weights vs patterns

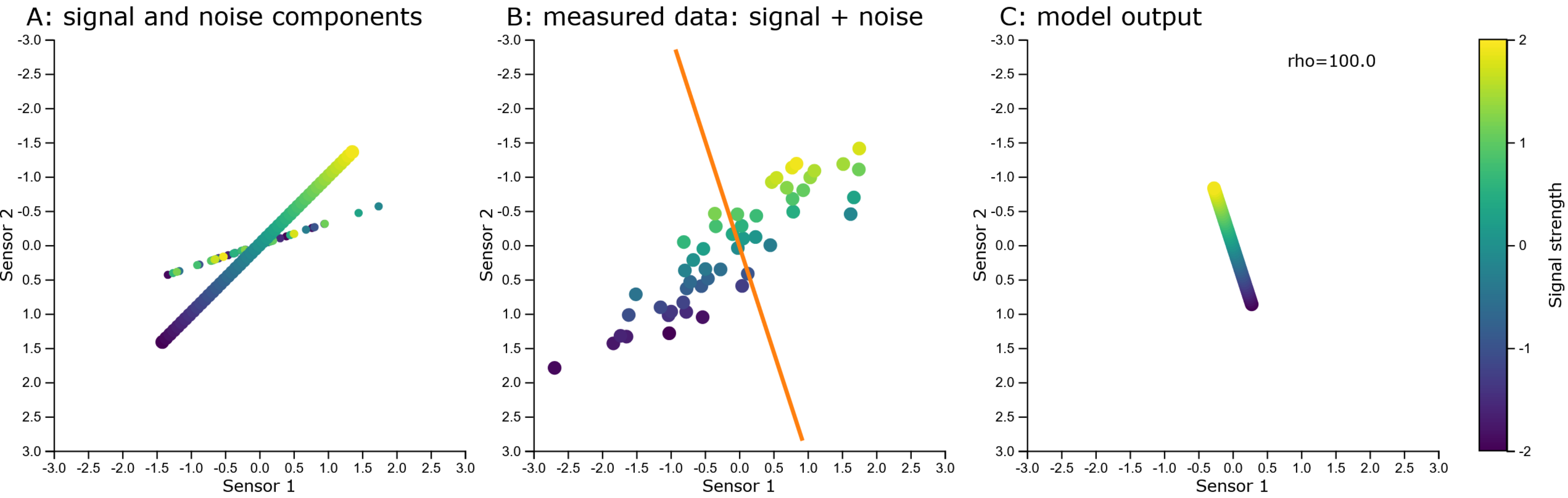
https://users.aalto.fi/~vanvlm1/posthoc/regression.html

Interactive explanation of weights vs patterns



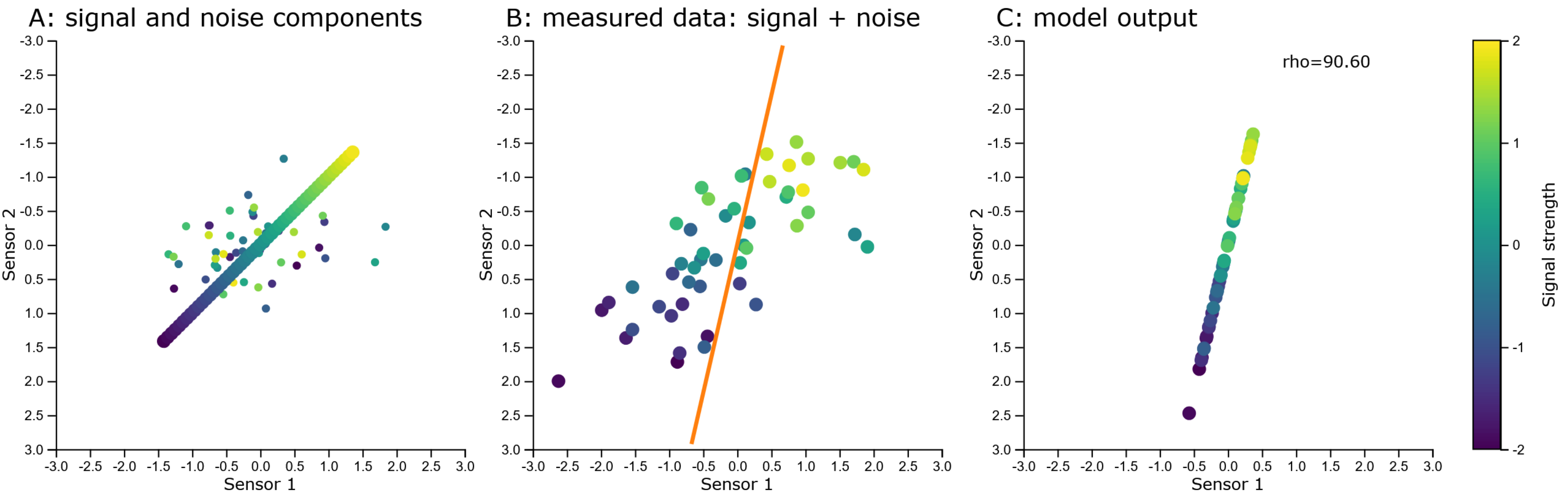
Only the informative component is present in the data

Interactive explanation of weights vs patterns



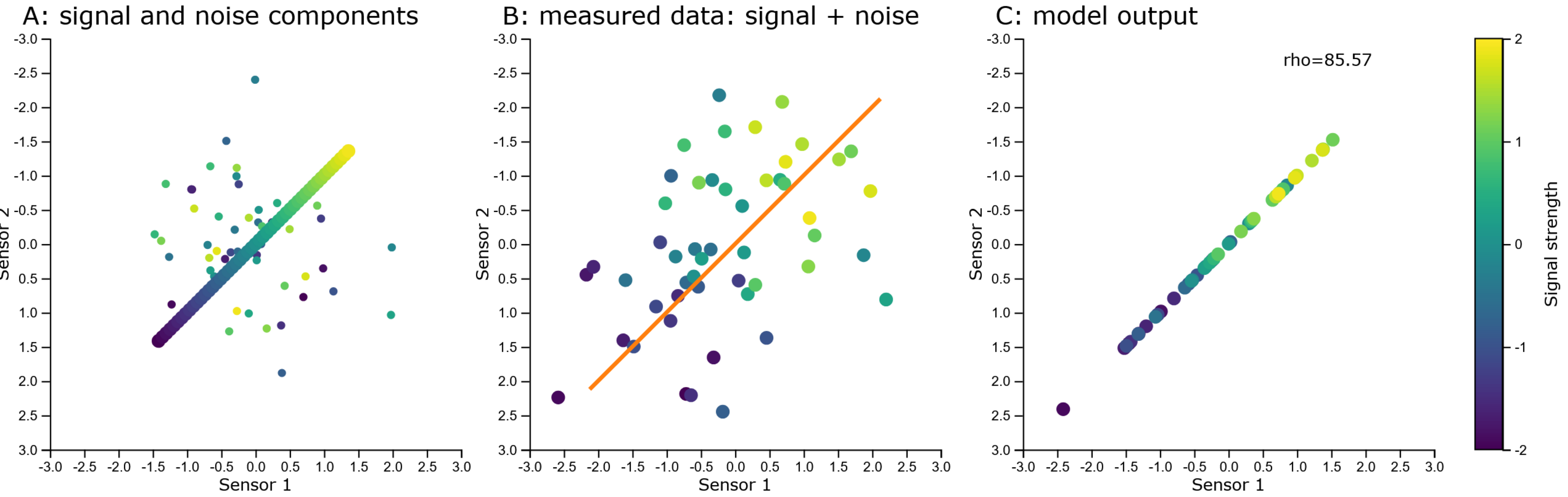
A second, uninformative brain component is also present

Interactive explanation of weights vs patterns



Multiple uninformative components are present

Interactive explanation of weights vs patterns



The uninformative components (“noise”) is spherical

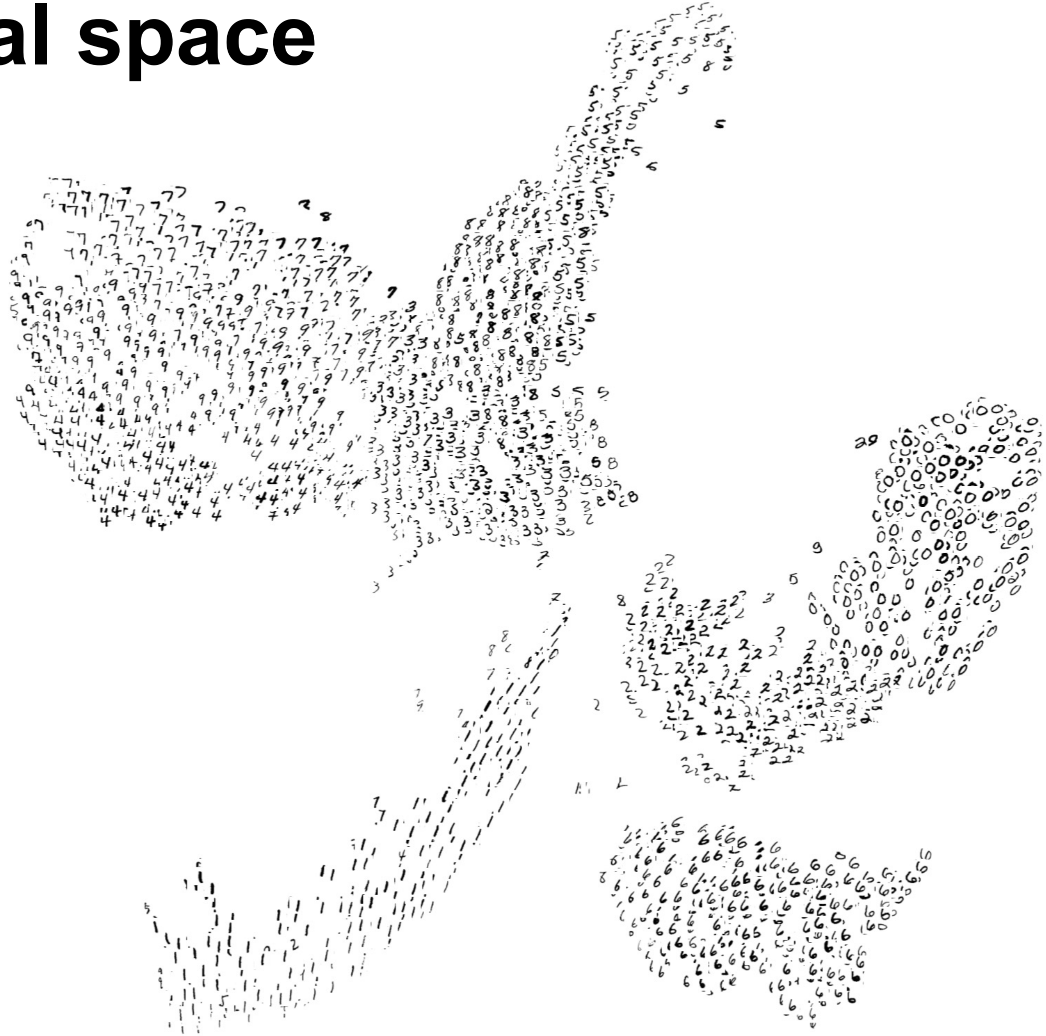
Summary

1. Uses for multivariate analysis
2. What linear machine models do: regression & classification
3. The curse of dimensionality, overfitting, & cross-validation
4. Spatial filtering, decoding across time
5. Determining statistical significance of the result
6. Interpreting the model: weights vs patterns
7. **Representational spaces: tracking information content**

Representational space

MNIST pixel space

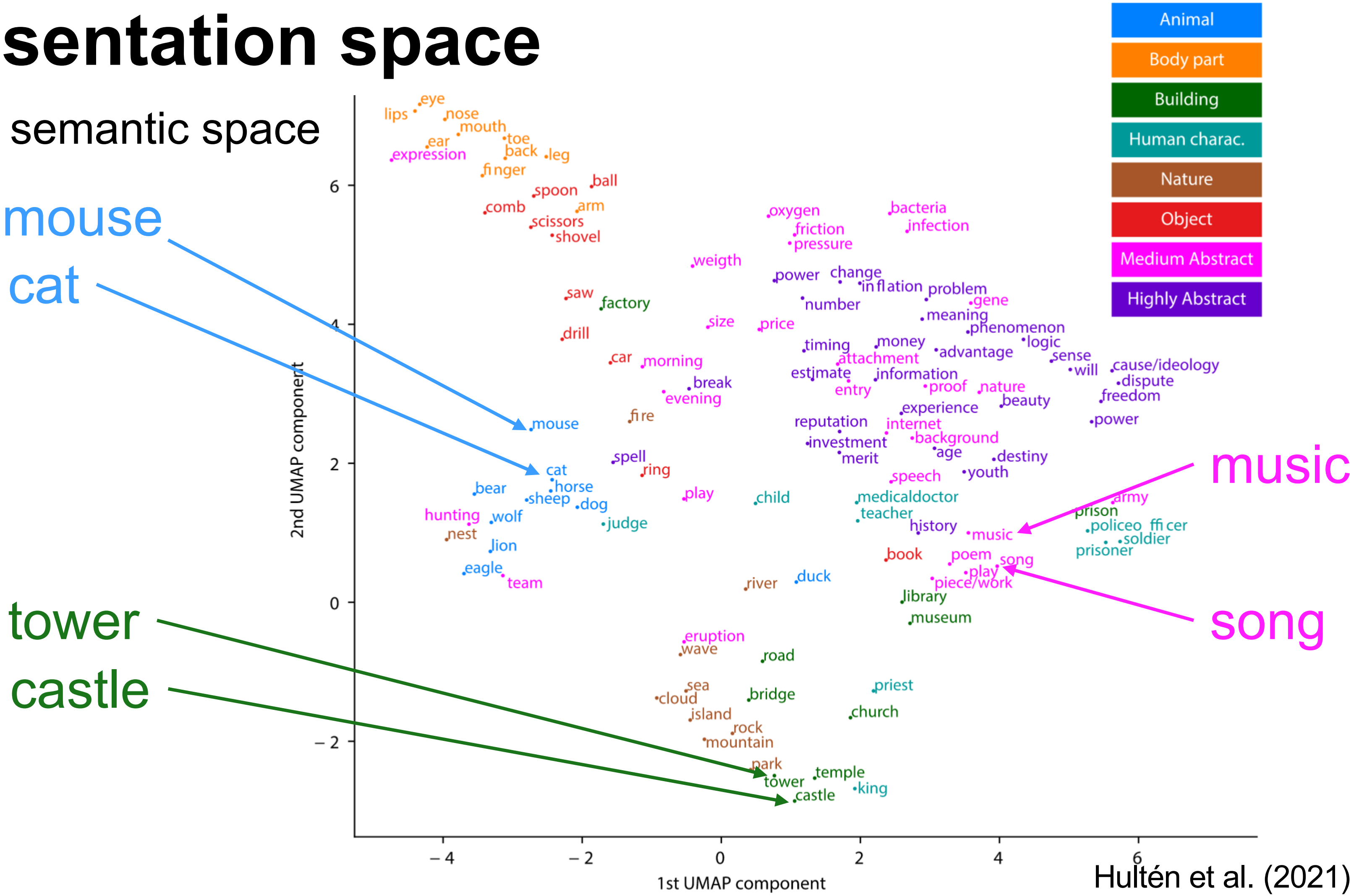
The **MNIST database** is a large dataset of handwritten digits, collected by the US Census Bureau.



https://projector.tensorflow.org

Representation space

word2vec semantic space



Hultén et al. (2021)

Mapping brain ←→ representation space

X (brain data)

features →

observations (stimuli) ↓

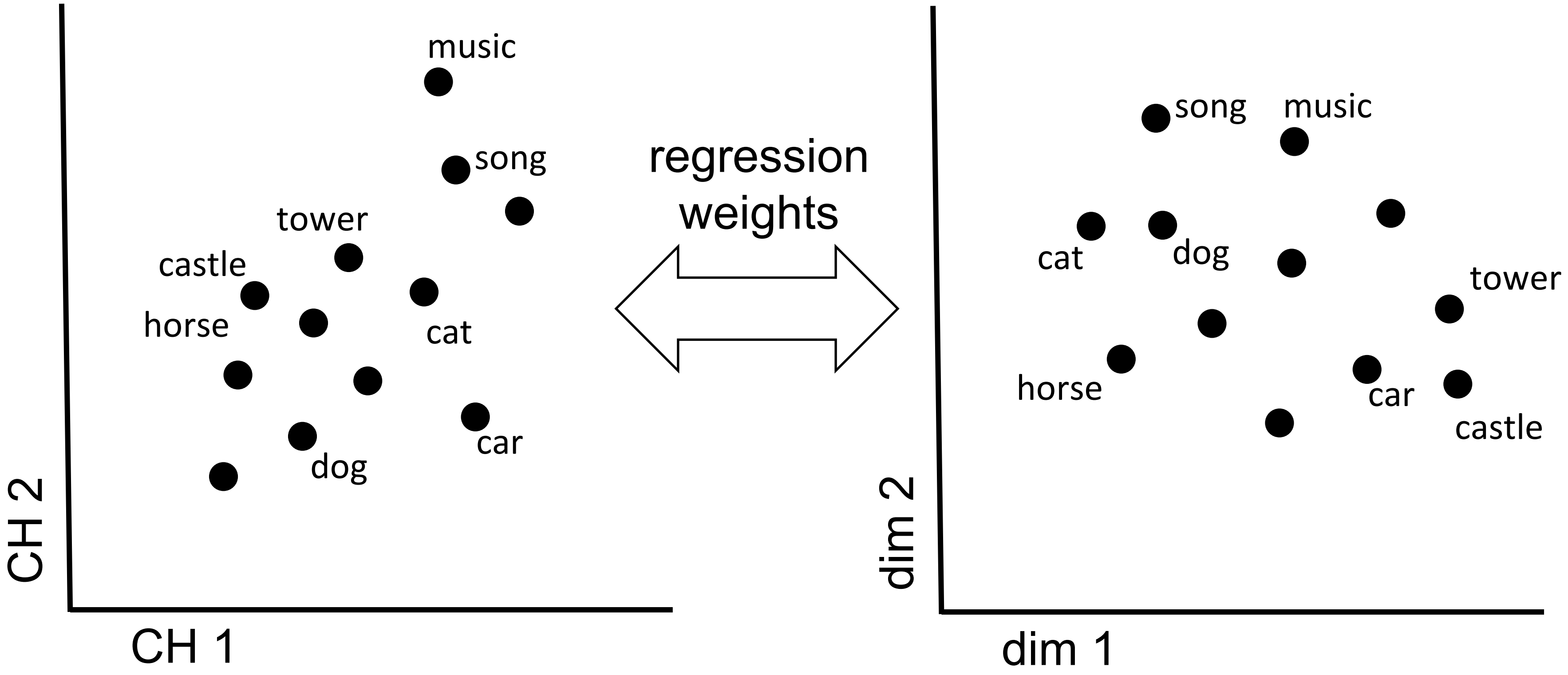
CH 1	CH 2	CH 3	CH 4	CH 5
0.1	0.3	5.2	5.1	0.9
0.3	0.5	3.7	7.6	9.5
0.2	0.6	1.2	2.1	1.0
1.2	1.1	8.3	4.2	1.2
1.7	2.0	2.9	7.6	4.3

Y (representation space)

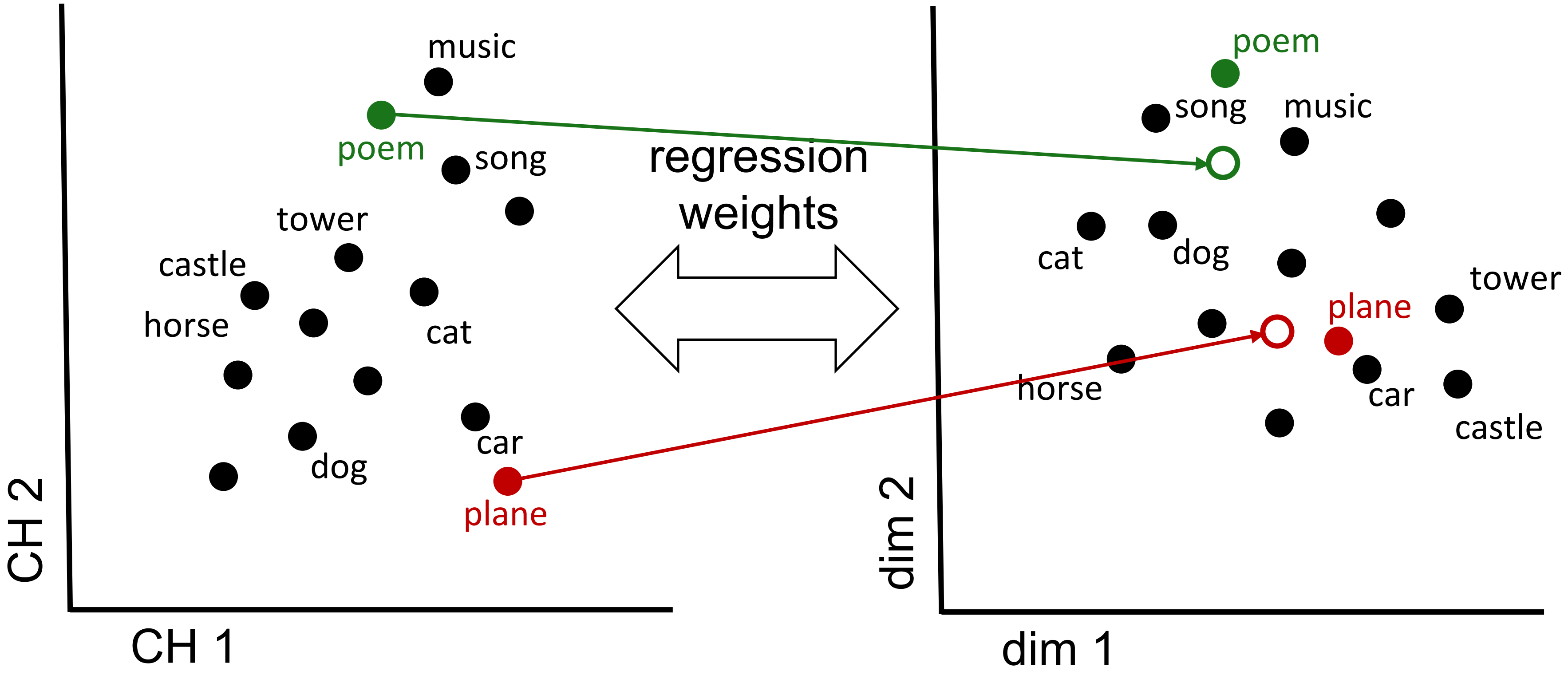
features →

dim 1	dim 2	dim 3	dim 4	dim 5
0.5	5.2	3.6	3.5	3.7
2.6	2.4	2.5	3.8	3.8
1.4	4.3	6.7	2.8	5.3
1.9	2.7	3.5	4.3	2.8
2.1	8.4	7.8	6.6	8.3

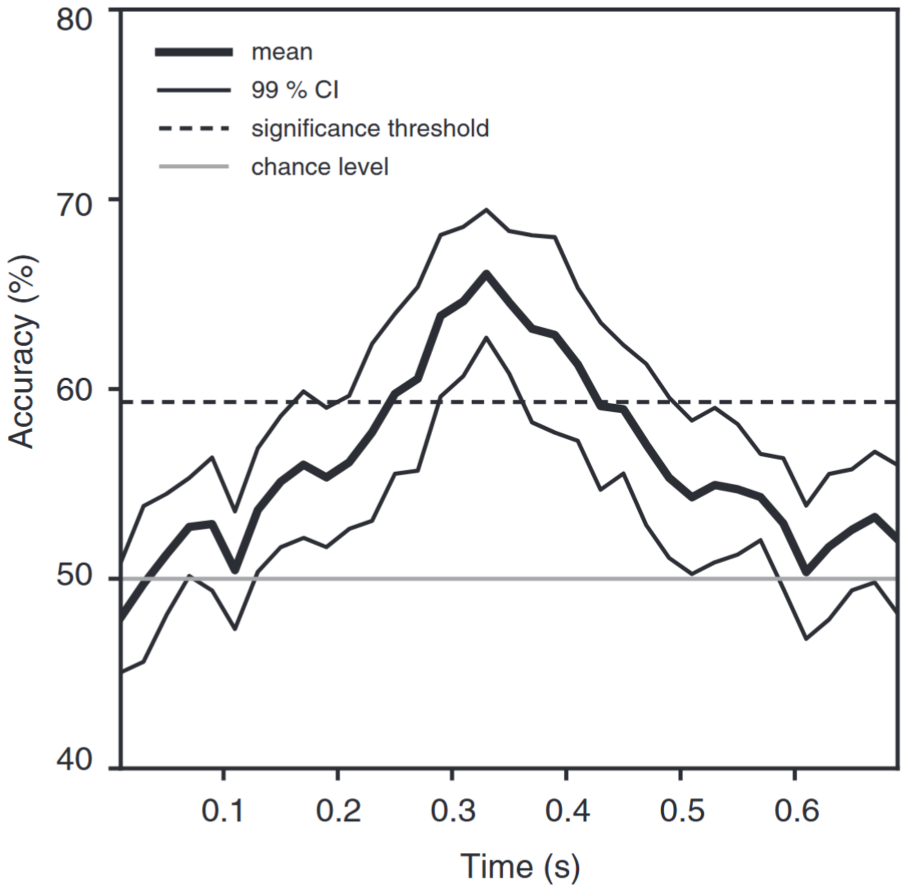
Mapping brain ←→ representation space



Mapping brain ←→ representation space

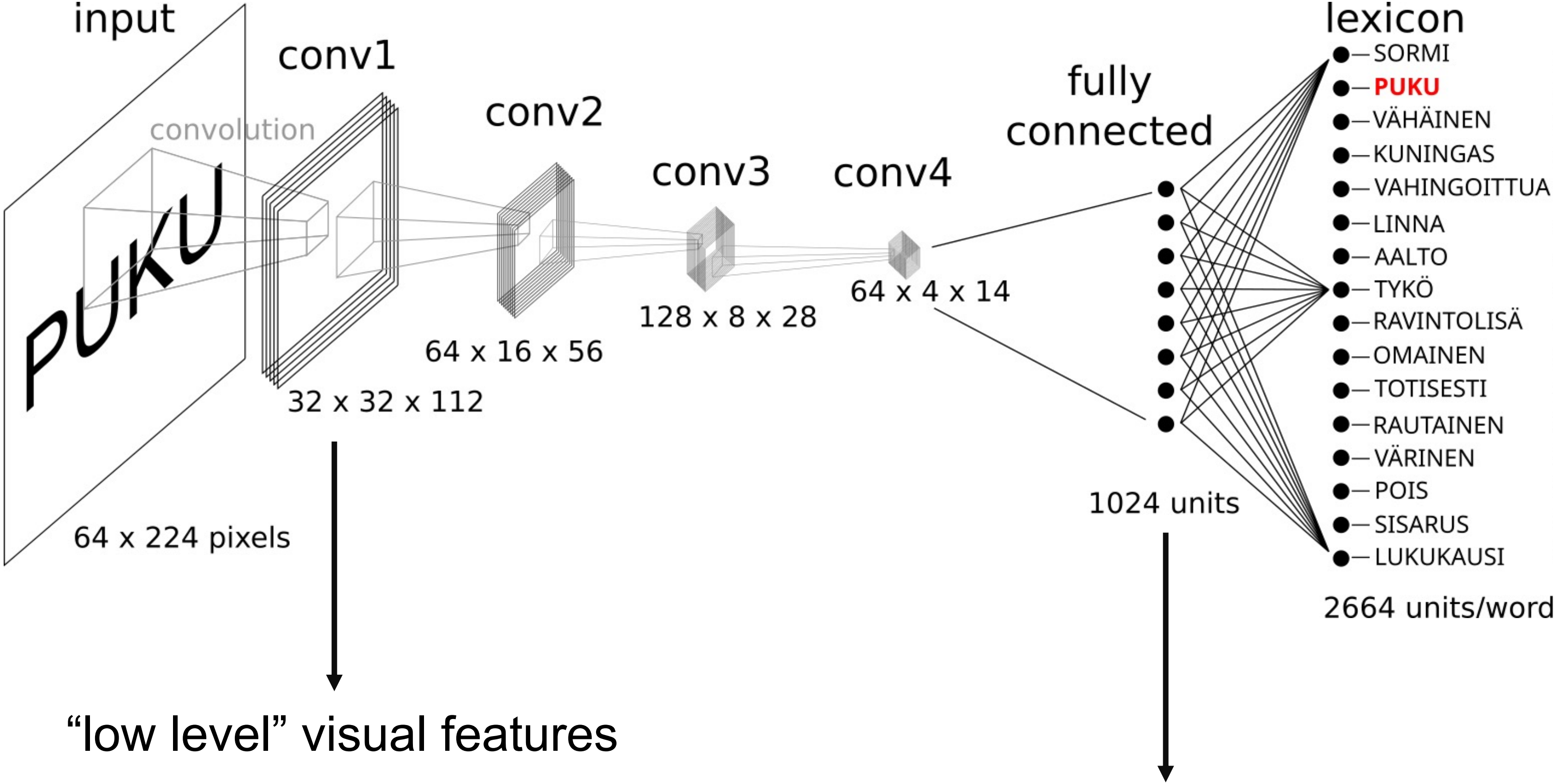


Decoding accuracy over time



Hultén et al. (2021)

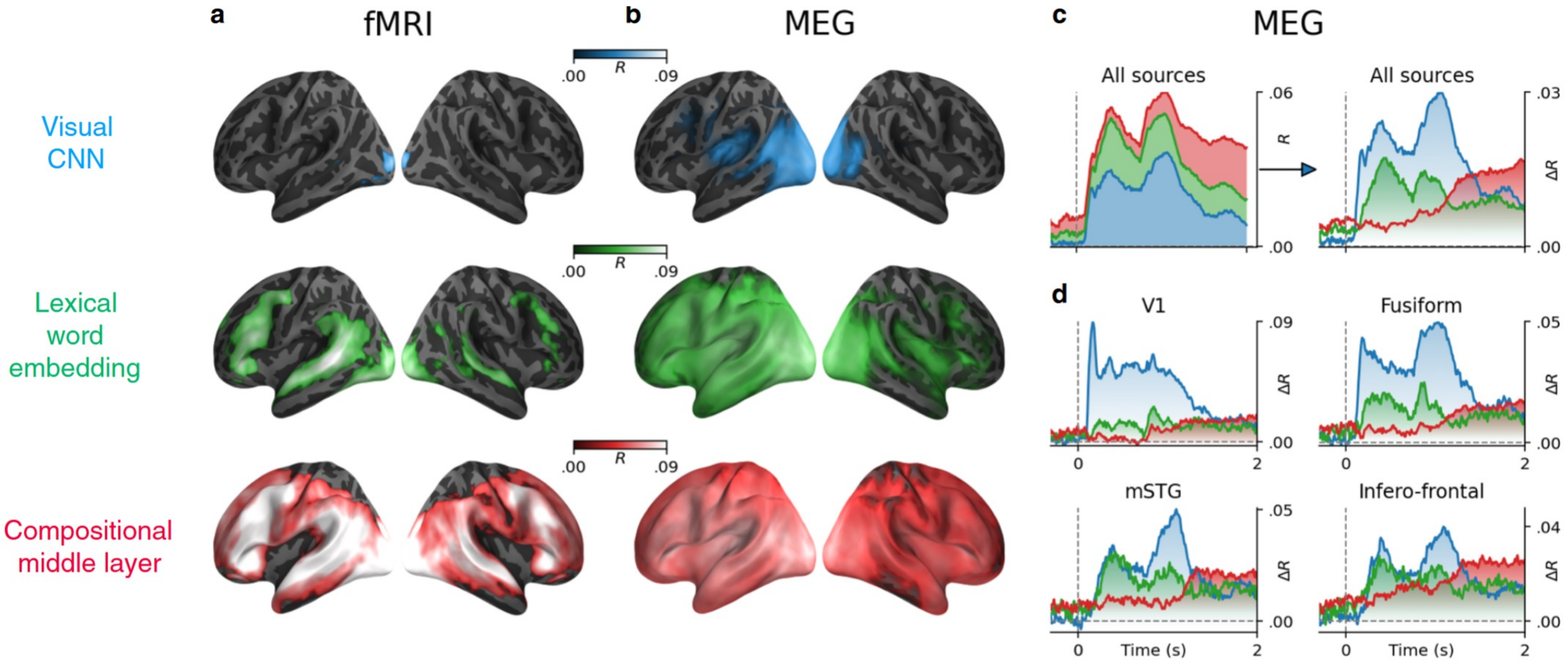
Using AI models to create representational spaces



“low level” visual features

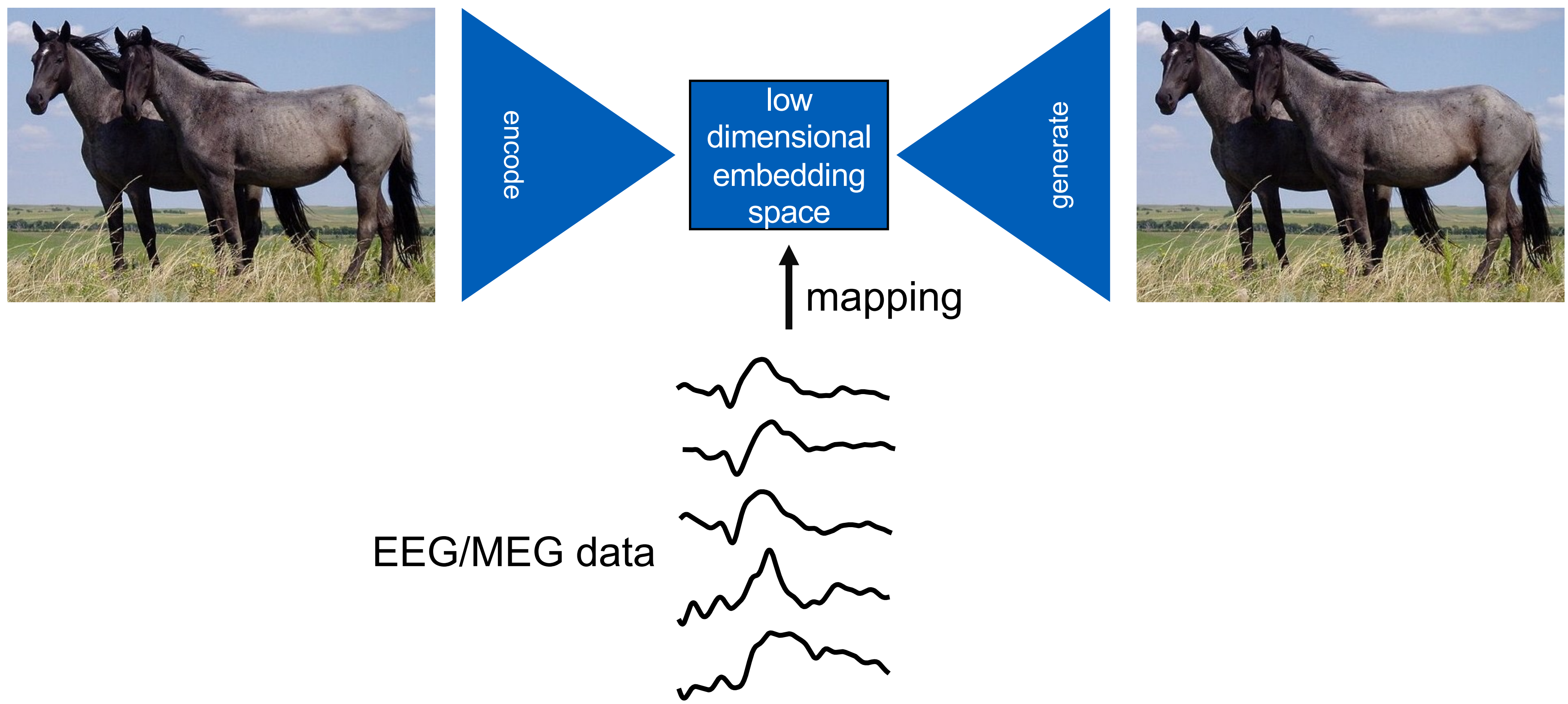
“high level” visual features

Using AI models to create representational spaces



Caucheteux & King (2022)

Mapping brain activity to the inner layer of an autoencoder



Mapping brain activity to the inner layer of an autoencoder

Benchetrit et al. (2024)

Take-aways

1. Multivariate analysis tracks **information content**
2. Machine learning sees your data as a **cloud of points**
3. More **observations** is always better, more **features** may not be
4. Always isolate your **train** and **test** sets
5. Use **random permutations** to determine statistical significance
6. Visualize and interpret **patterns**, not weights
7. You can map **brain patterns** to **embedding spaces**