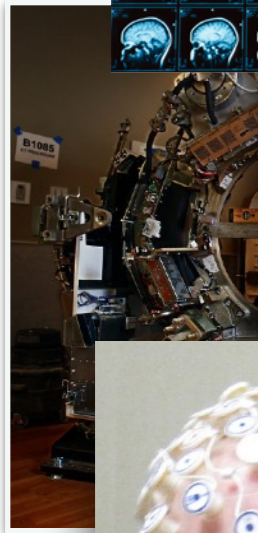
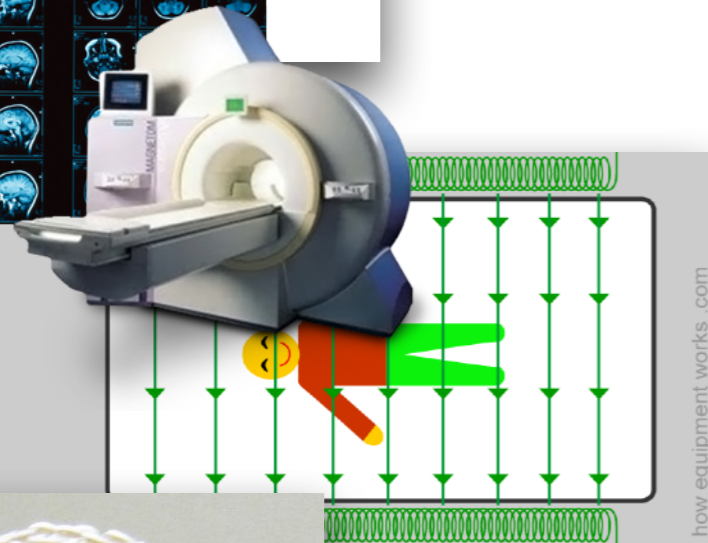
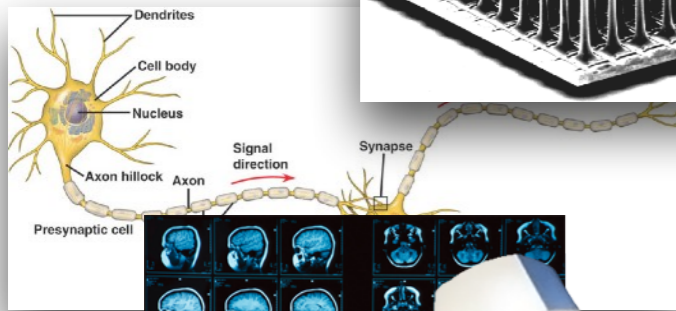
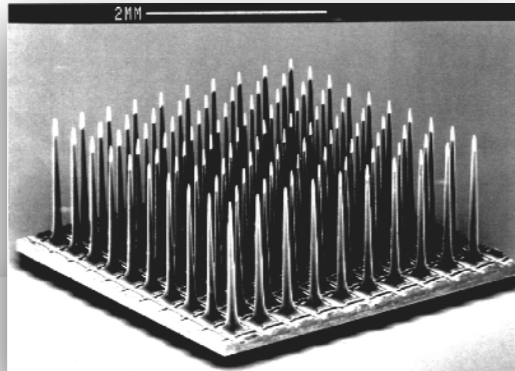


MACHINE LEARNING ON NEUROIMAGING DATA

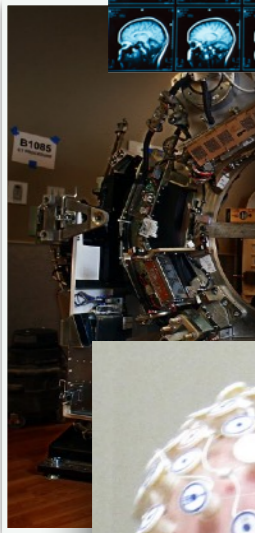
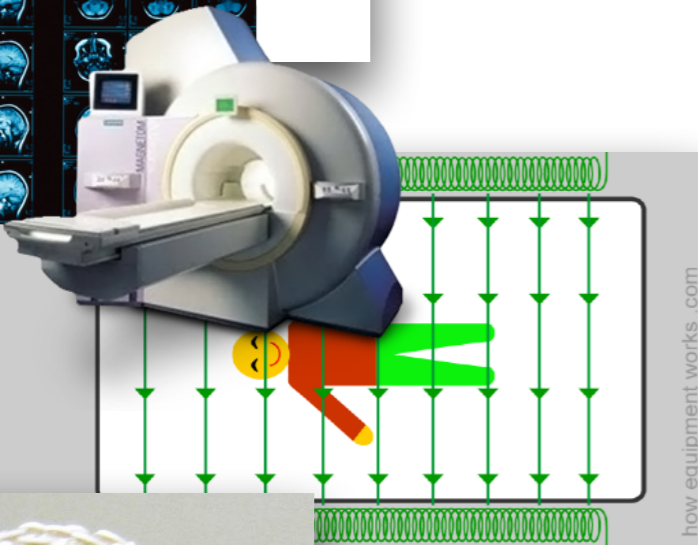
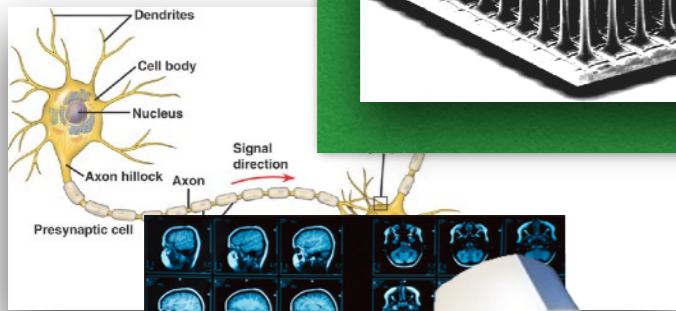
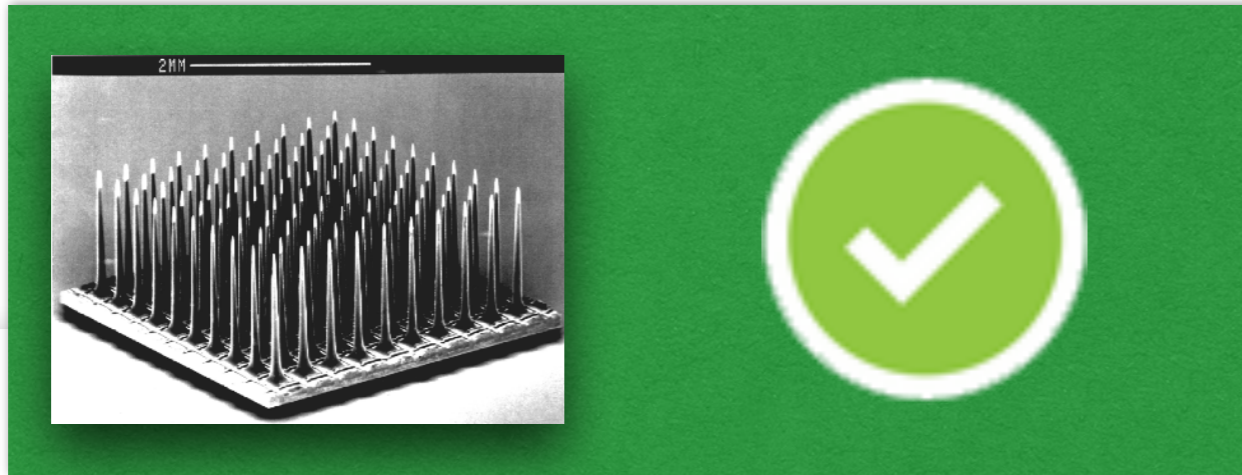
LECTURE 2: INTRODUCTION TO MACHINE LEARNING

Ilya Kuzovkin

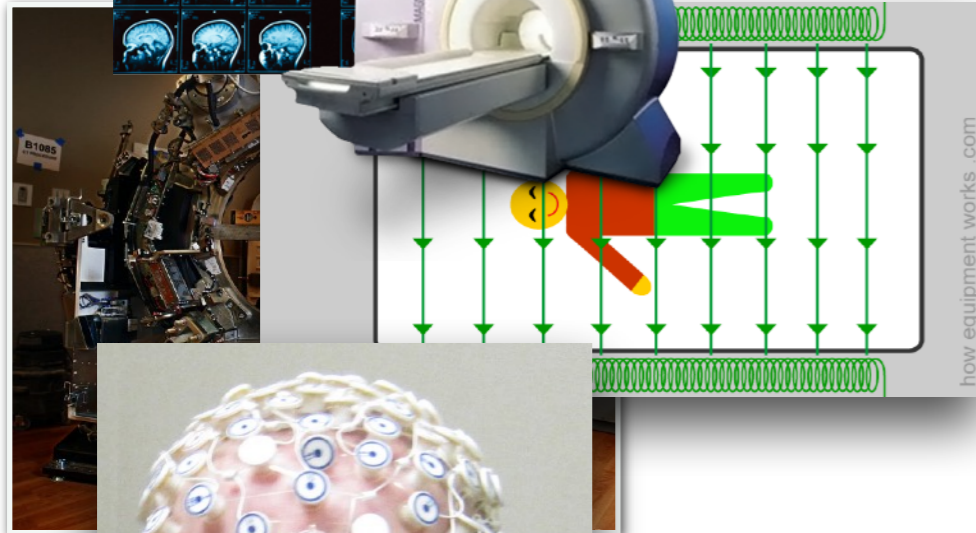
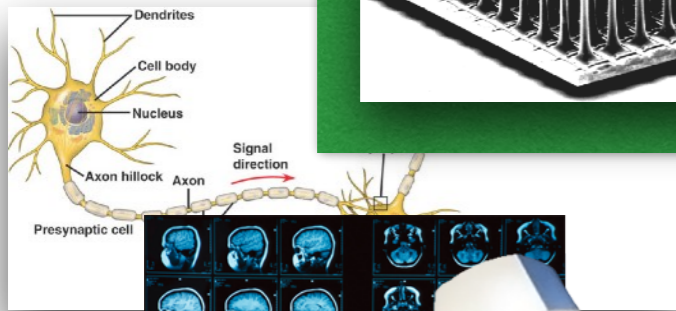
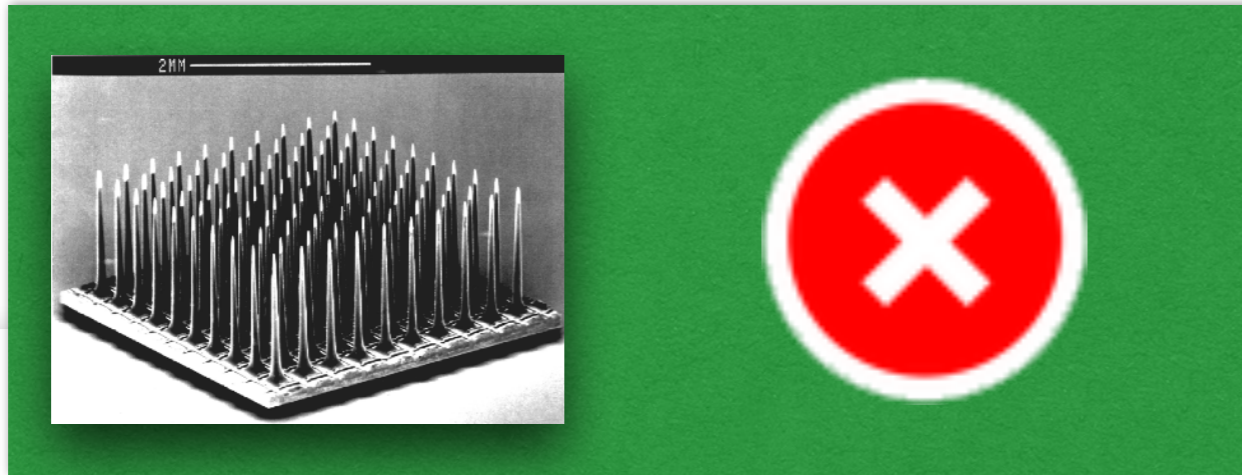
PREVIOUSLY ON SLIDES...



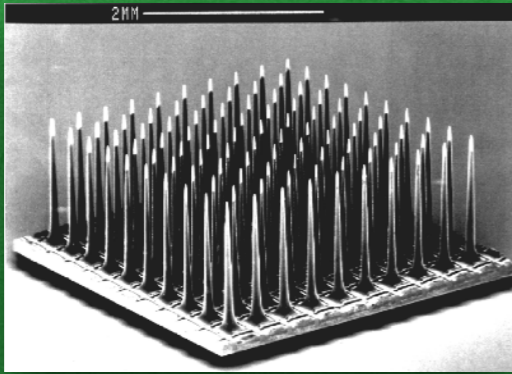
PREVIOUSLY ON SLIDES...



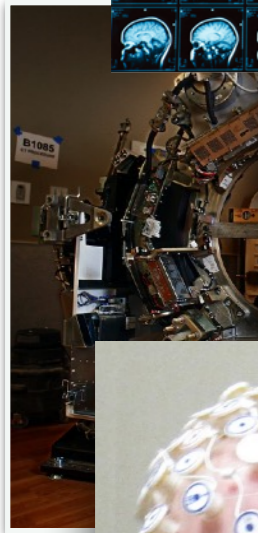
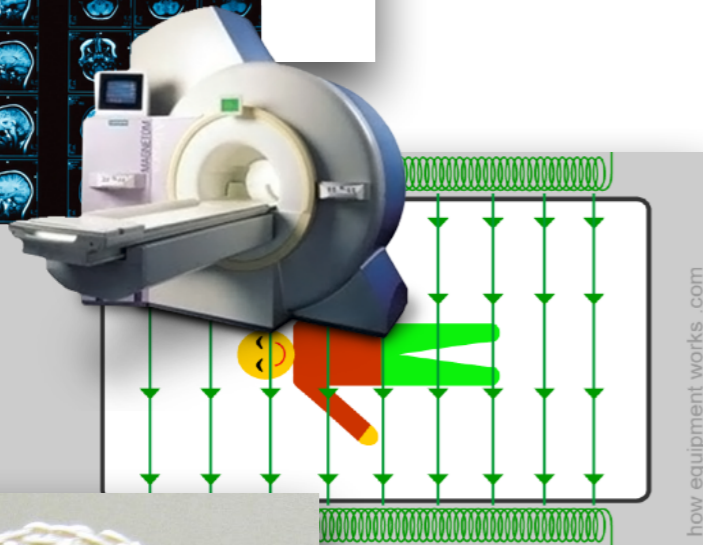
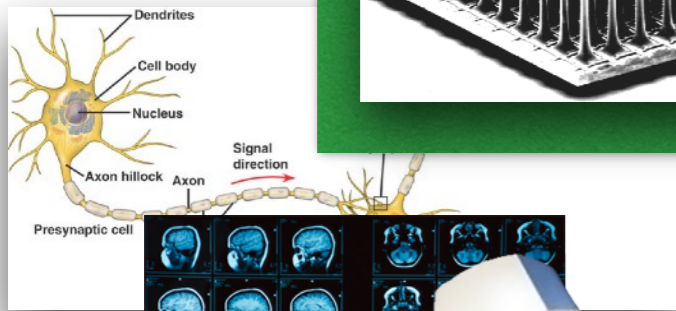
PREVIOUSLY ON SLIDES...



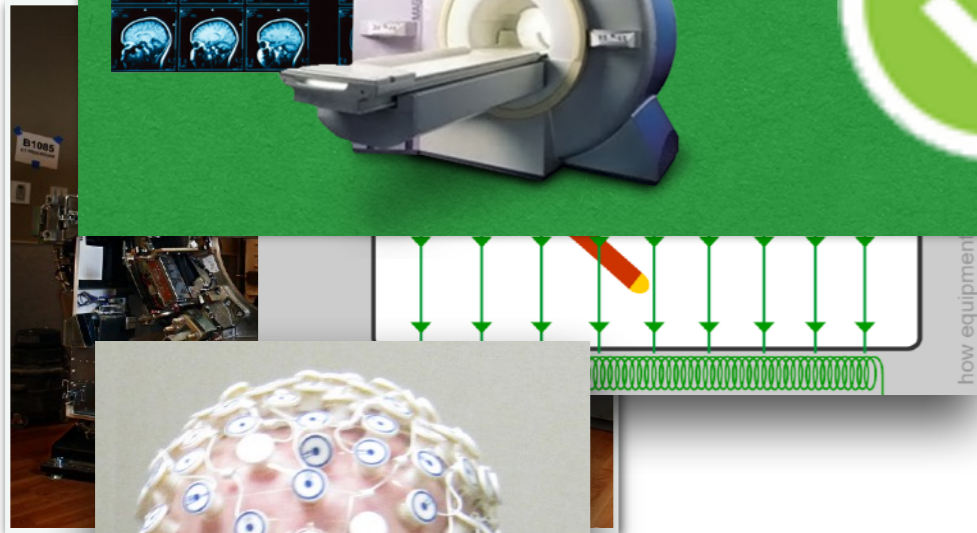
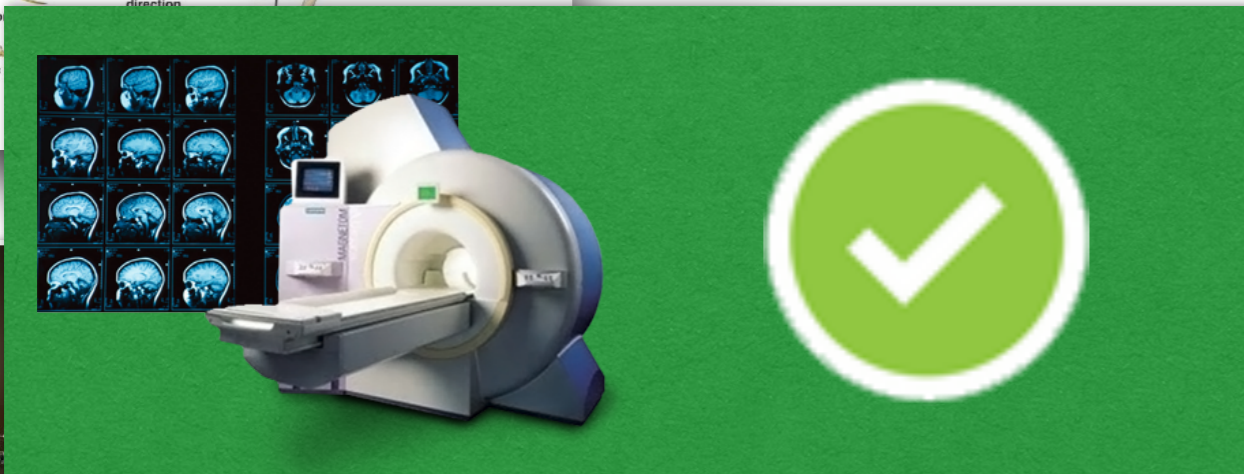
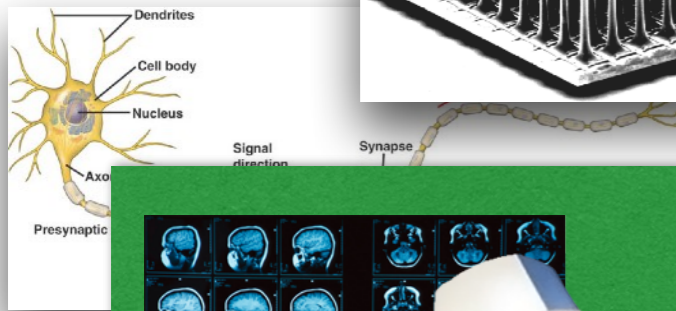
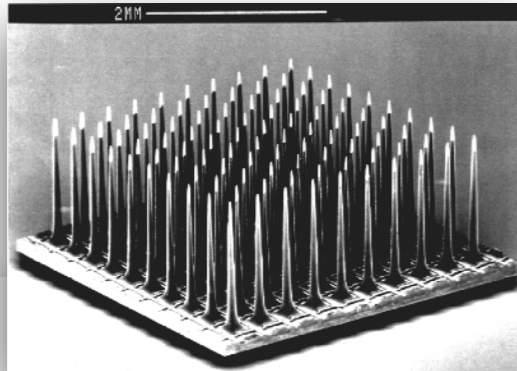
PREVIOUSLY ON SLIDES...



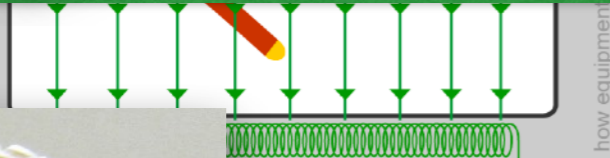
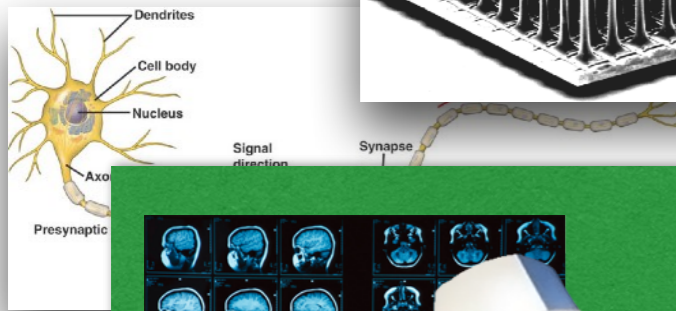
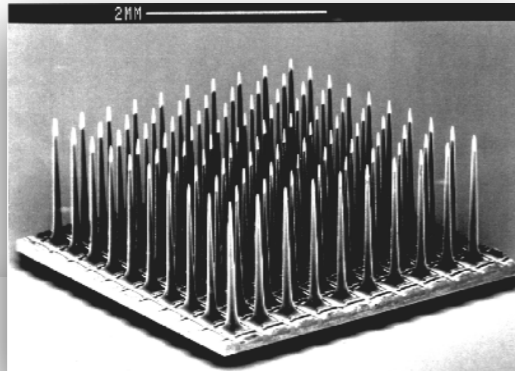
_____ data?
_____ plot?
_____ curve?



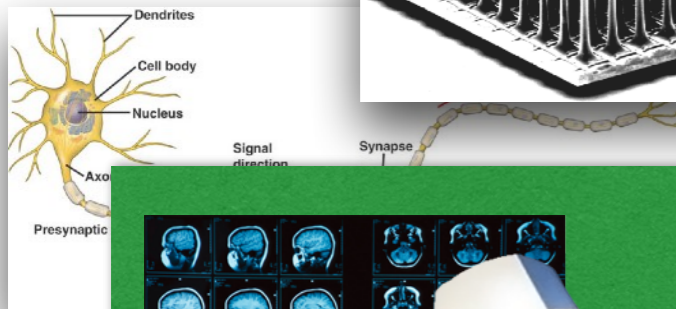
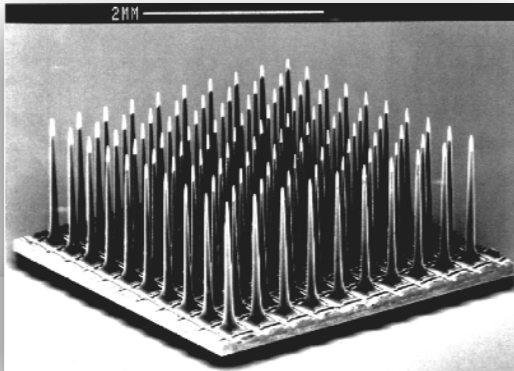
PREVIOUSLY ON SLIDES...



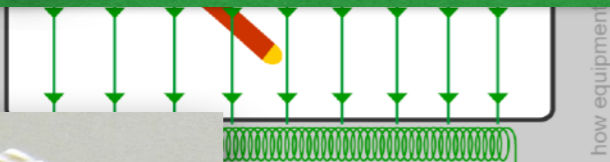
PREVIOUSLY ON SLIDES...



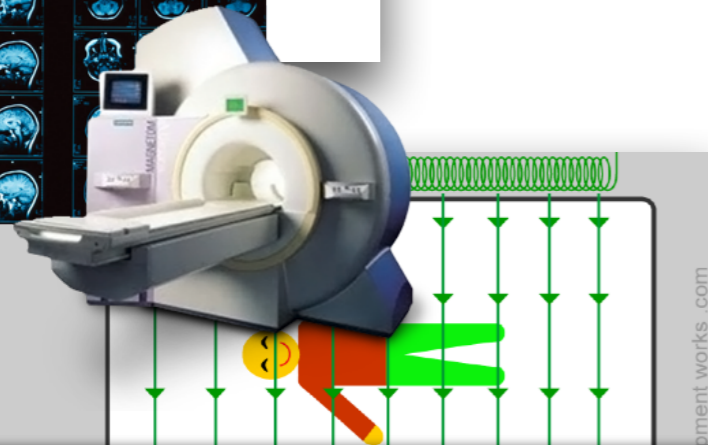
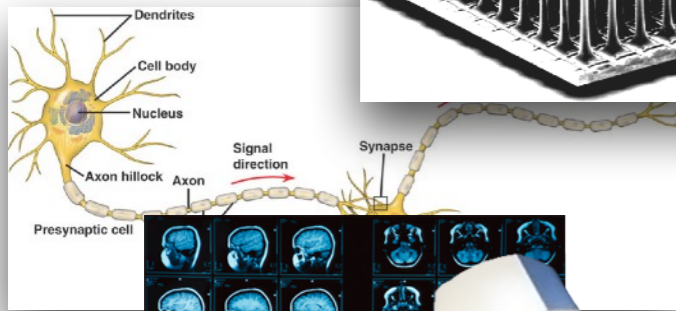
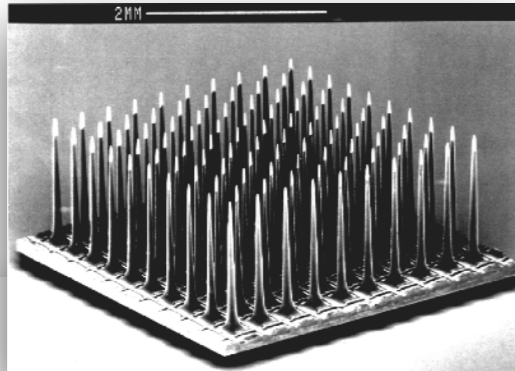
PREVIOUSLY ON SLIDES...



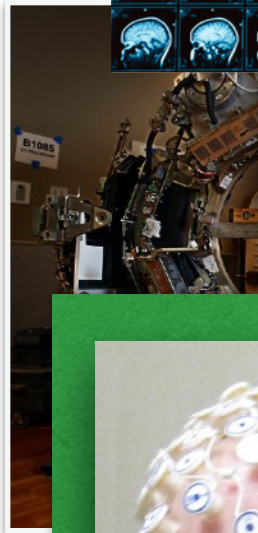
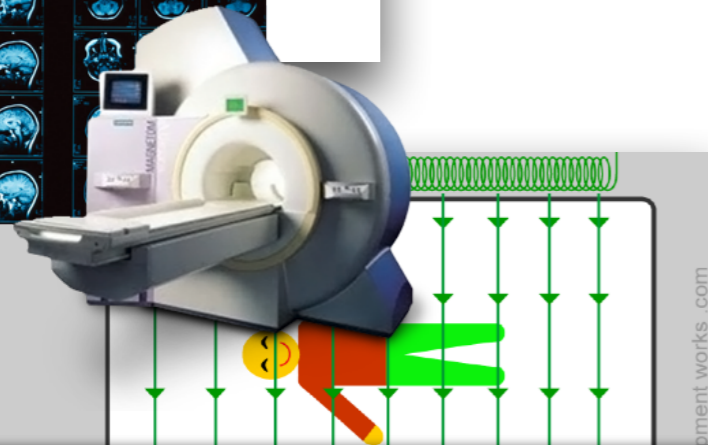
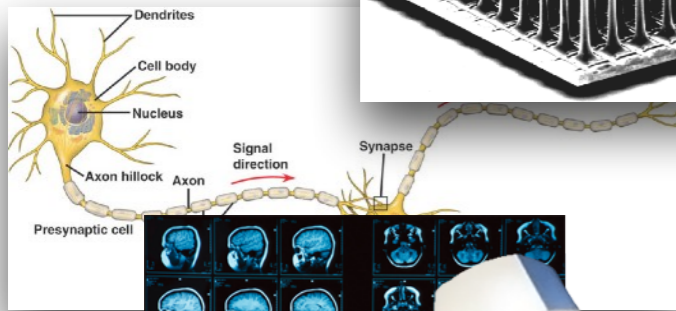
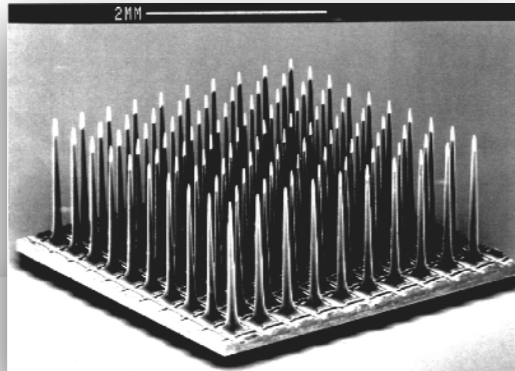
F? M? R? I?
BOLD?
~~pixels?~~
Data?



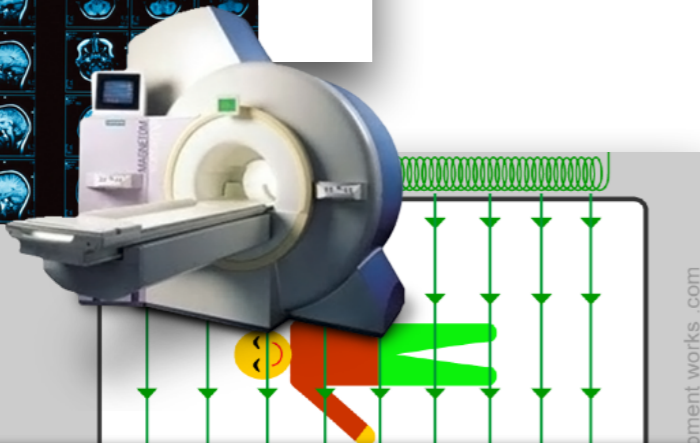
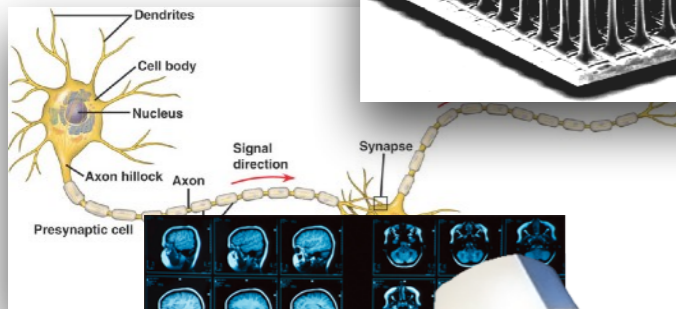
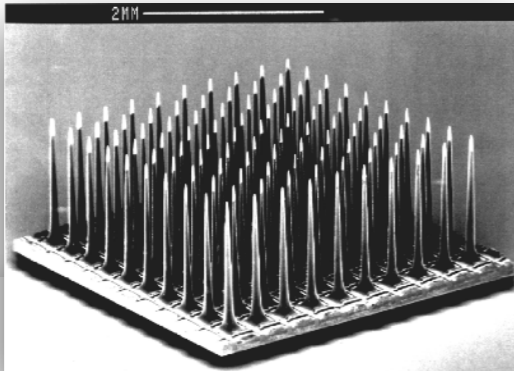
PREVIOUSLY ON SLIDES...



PREVIOUSLY ON SLIDES...

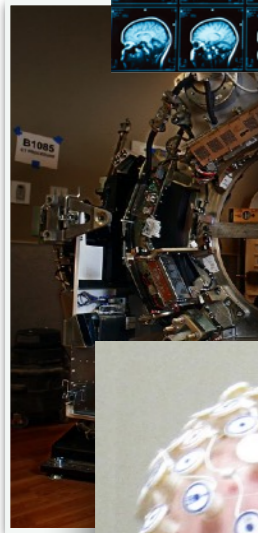
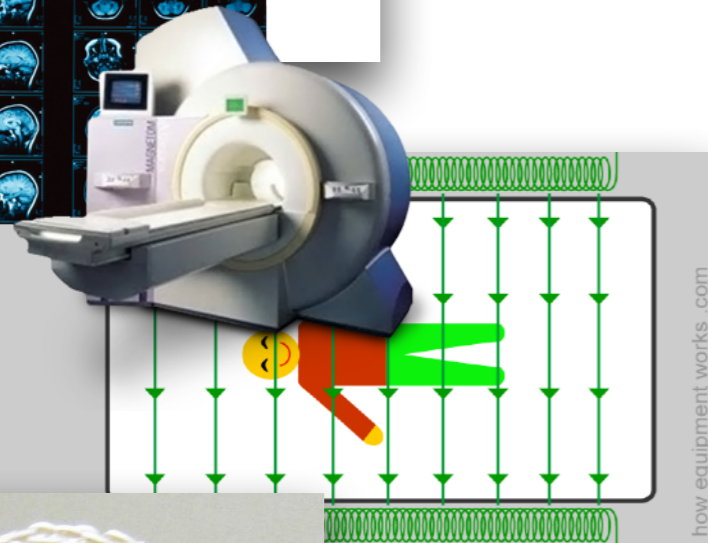
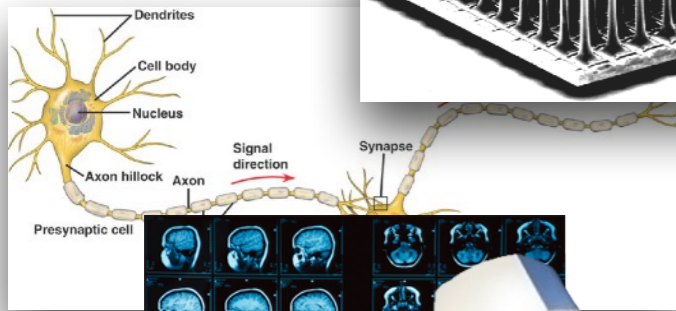
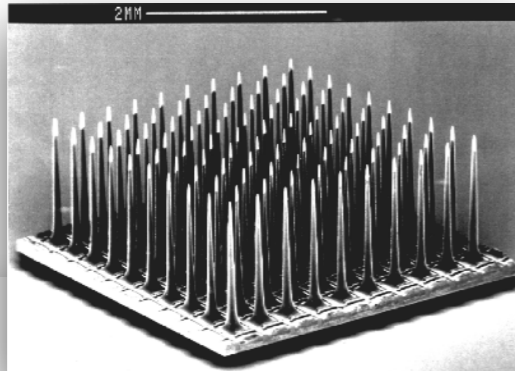


PREVIOUSLY ON SLIDES...



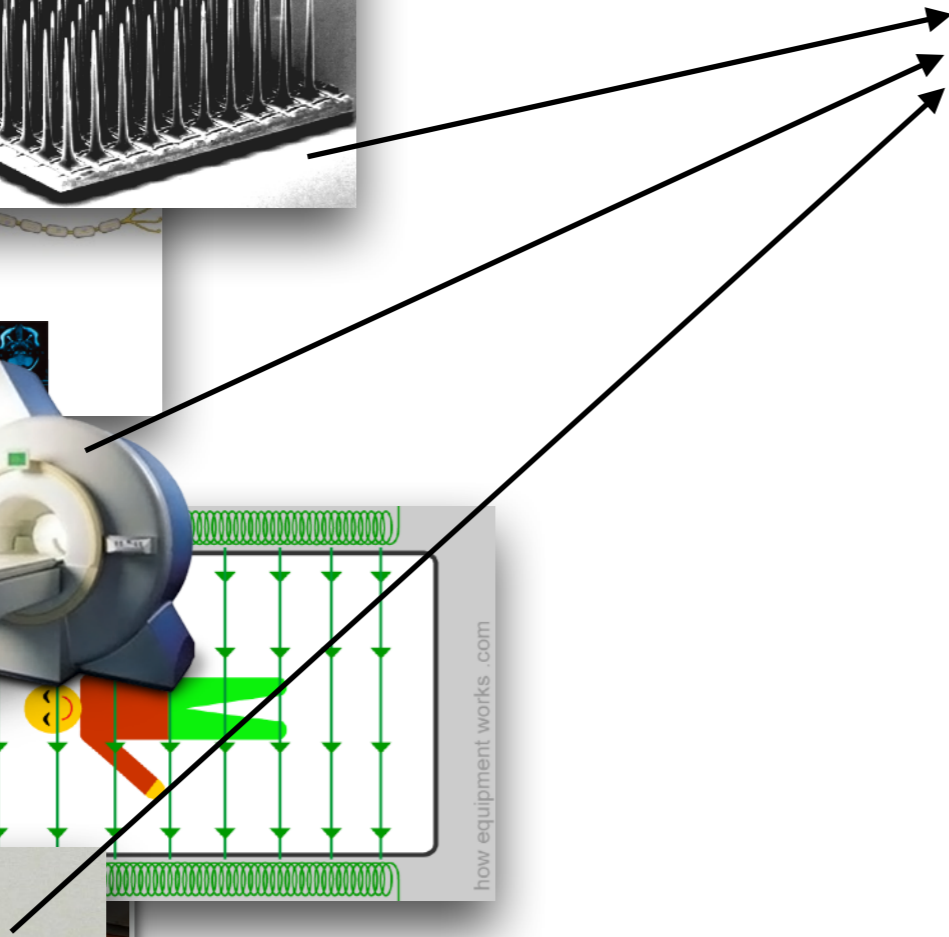
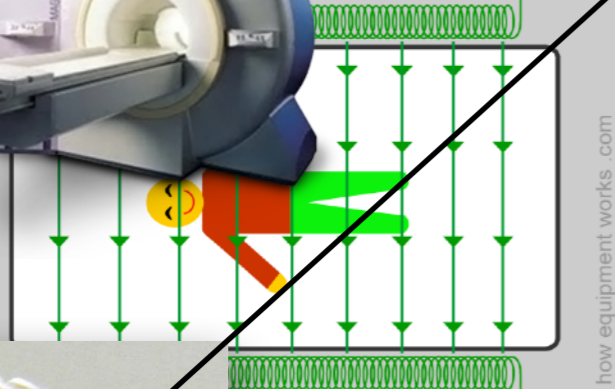
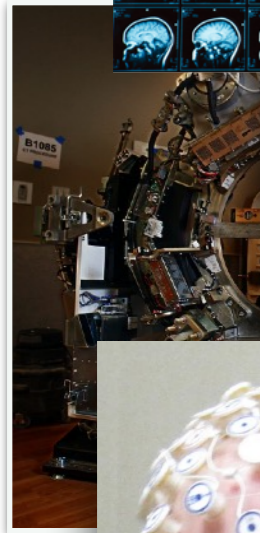
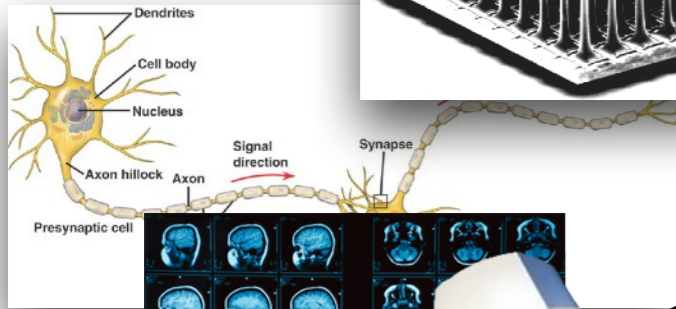
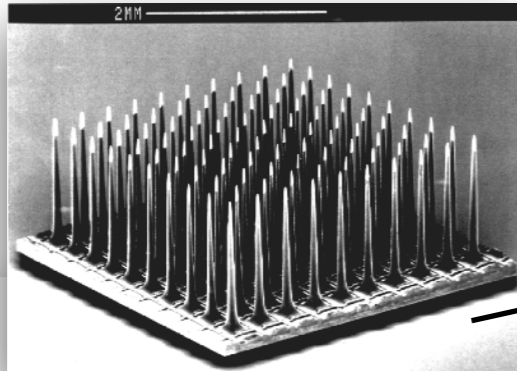
- What do we measure?
- Sampling rate?
- Waves?
- Fourier?

PREVIOUSLY ON SLIDES...



PREVIOUSLY ON SLIDES...

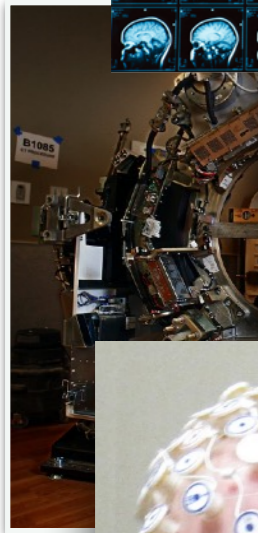
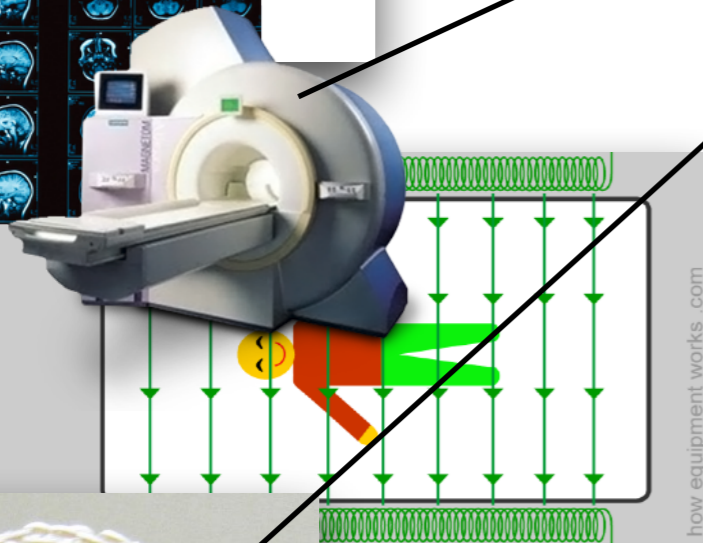
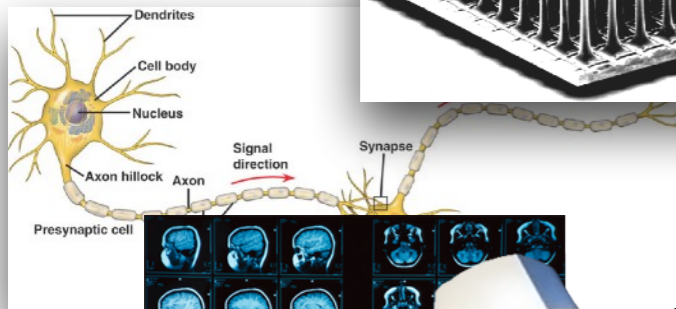
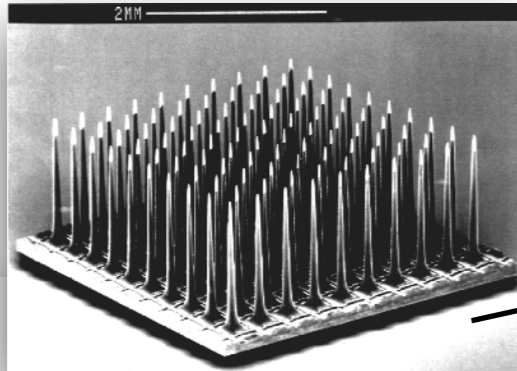
DATA



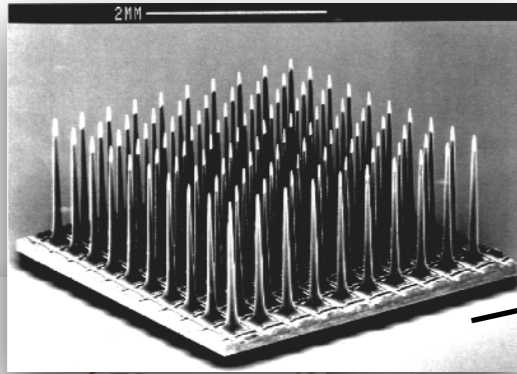
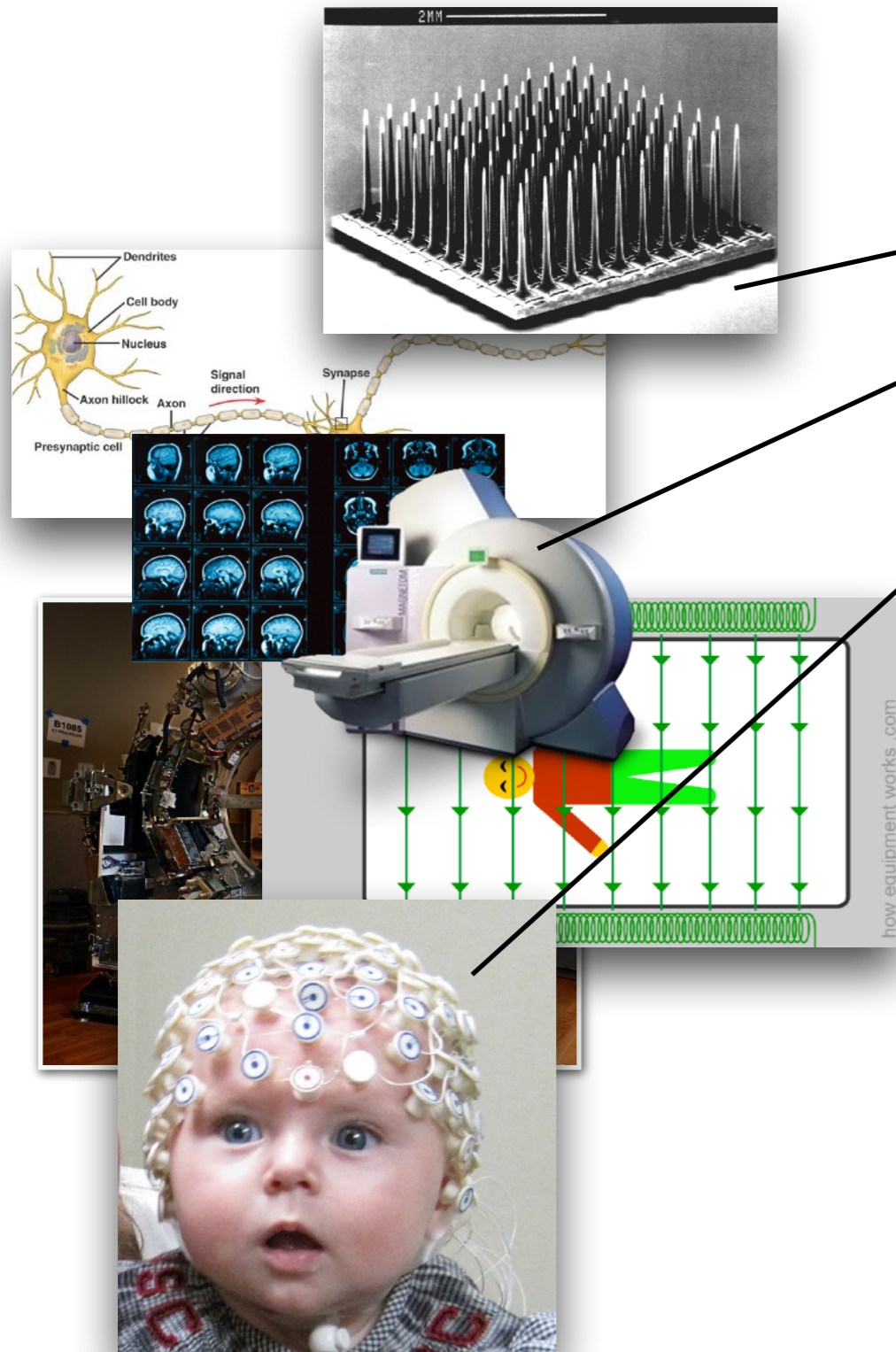
PREVIOUSLY ON SLIDES...

DATA

What's next?



PREVIOUSLY ON SLIDES...



DATA

What's next?

ANALYSIS

MANUAL ANALYSIS

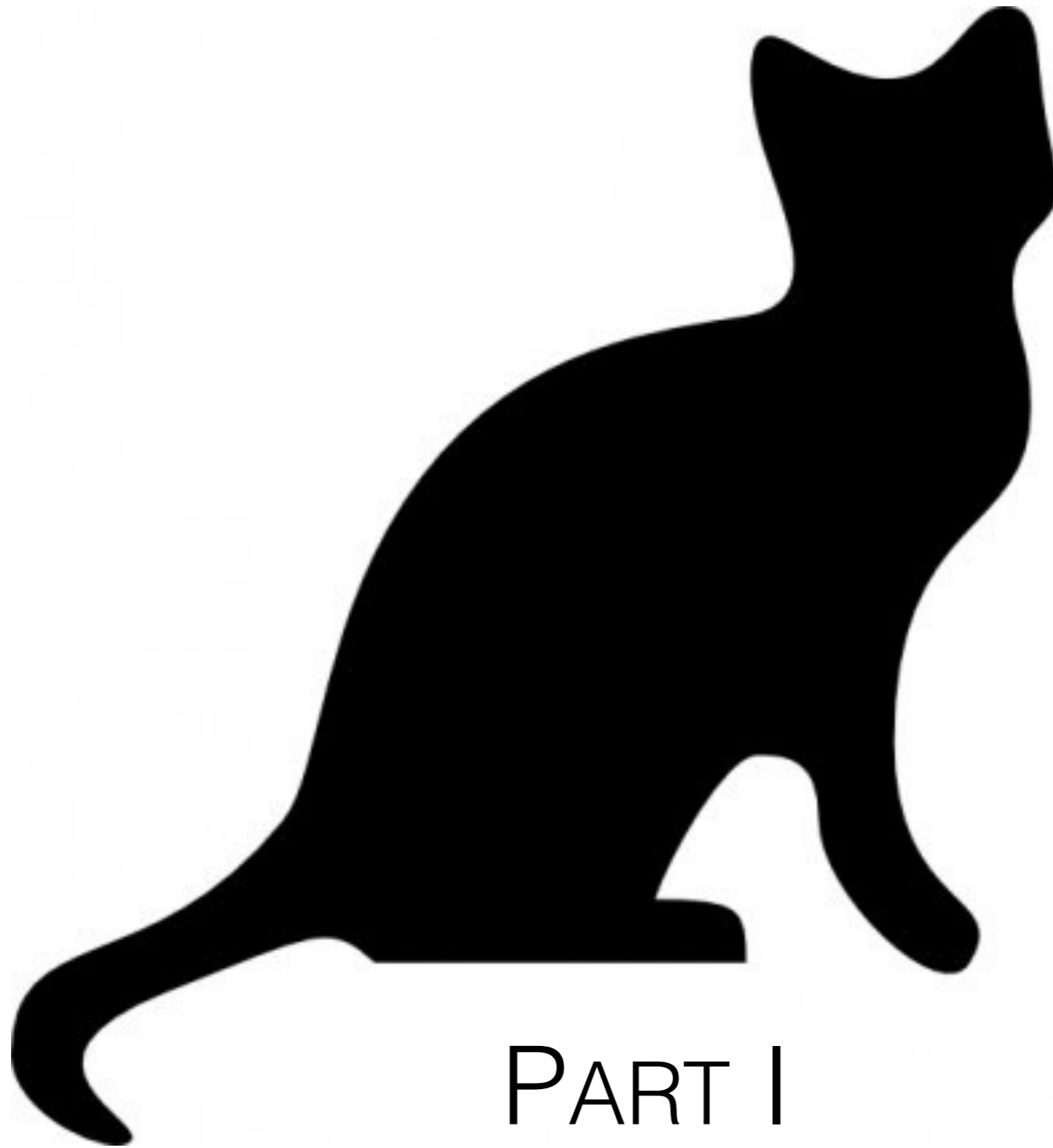
- - Very accurate
 - Easy to drop bad data
 - Human intuition
 - Human cognitive abilities to catch interesting stuff
 - As flexible as you want

- Takes time
- Takes manpower
- Boring*
- Infeasible on huge datasets

MACHINE LEARNING

-
- Fast*
- Calculates while you are free to do other things
- Automatic

- Makes errors
- Does not know which data is good and which is not
- Will try to find only what you asked for
- You need to learn about it



PART I

CONCEPTS WITH CATS



TRAINING SET

Machine learning algorithm **learns** from **examples**

TRAINING SET

Machine learning algorithm **learns** from **examples**



INSTANCE

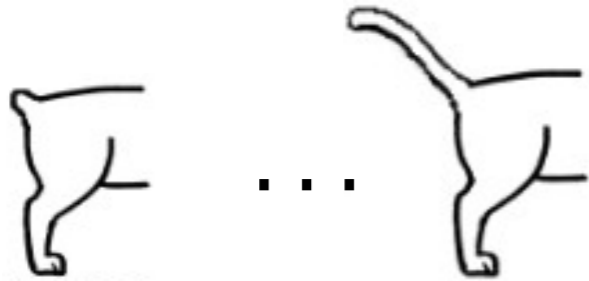
Each object (instance)



is described as a set set of parameters,
called **features**

FEATURES

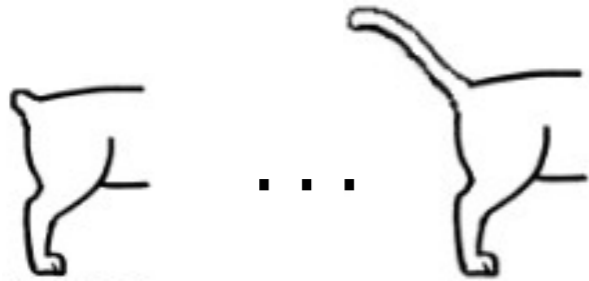
Length of the tail



f_1

FEATURES

Length of the tail



f_1

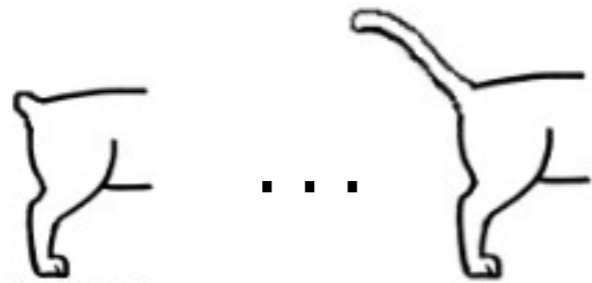
Amount of fur



f_2

FEATURES

Length of the tail



f_1

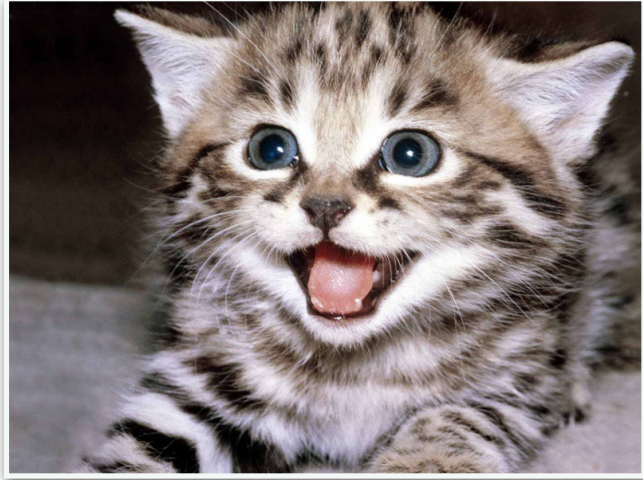
Amount of fur



f_2

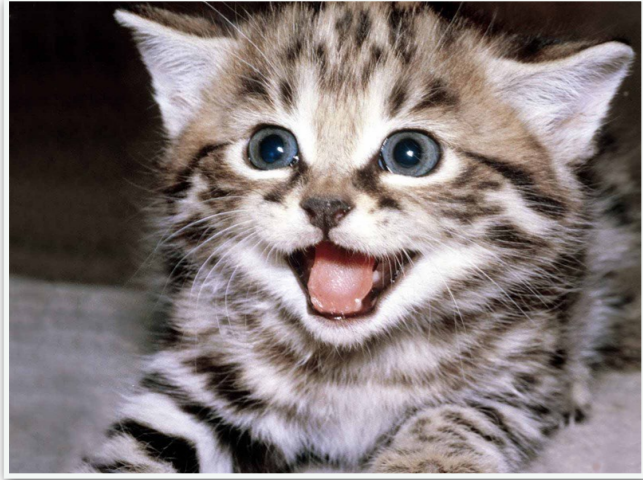
Feature vector $\mathbf{f} = (f_1, f_2)$

FEATURE VECTOR



= **f**

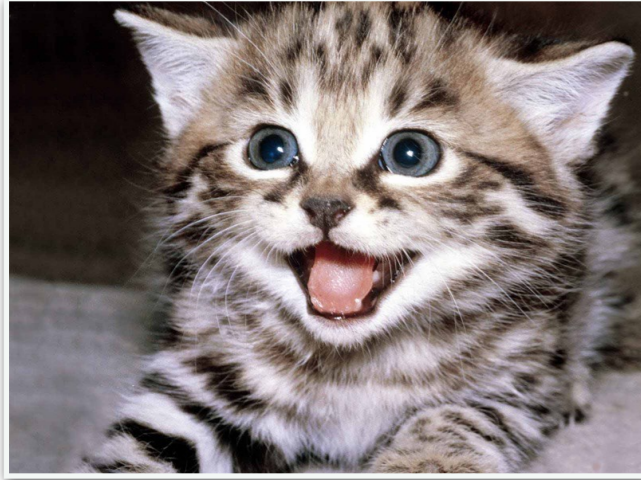
FEATURE VECTOR



$$= \mathbf{f} =$$

$$= (f_1, f_2)$$

FEATURE VECTOR

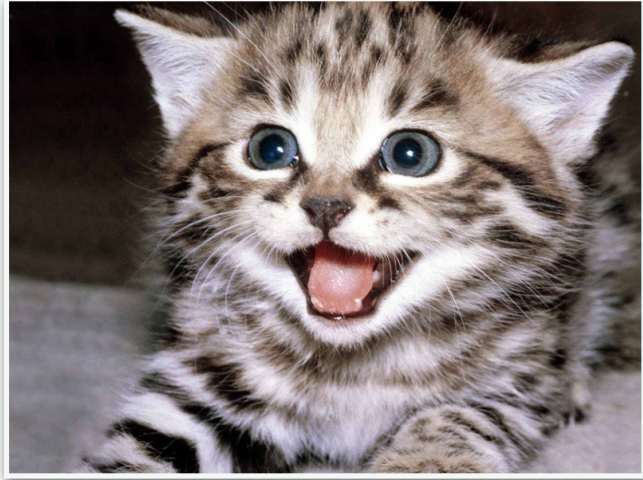


$$= \mathbf{f} =$$

$$= (f_1, f_2) =$$



$$= \left(\text{cat outline}, \text{cat silhouette} \right)$$

CLASS



= **f** =

= (*f*₁, *f*₂) =

= ( , ) →



DATASET



$(5, 143)$ →



$(12, 342)$ →



$(11, 234)$ →



$(7, 198)$ →



$(2, 210)$ →



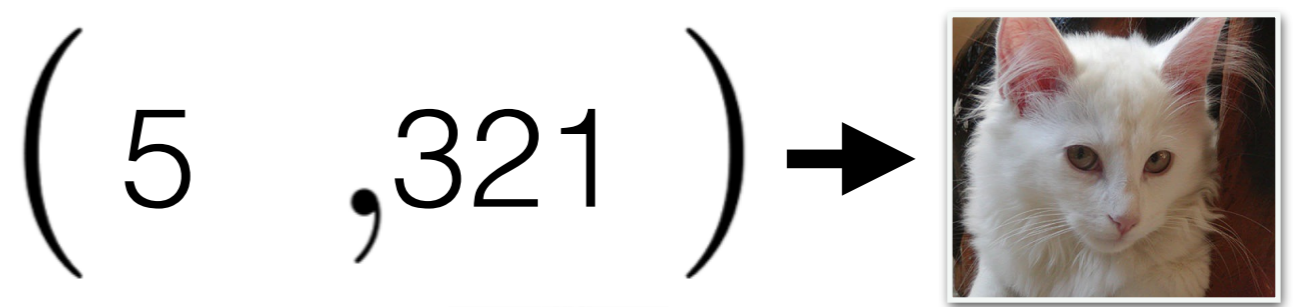
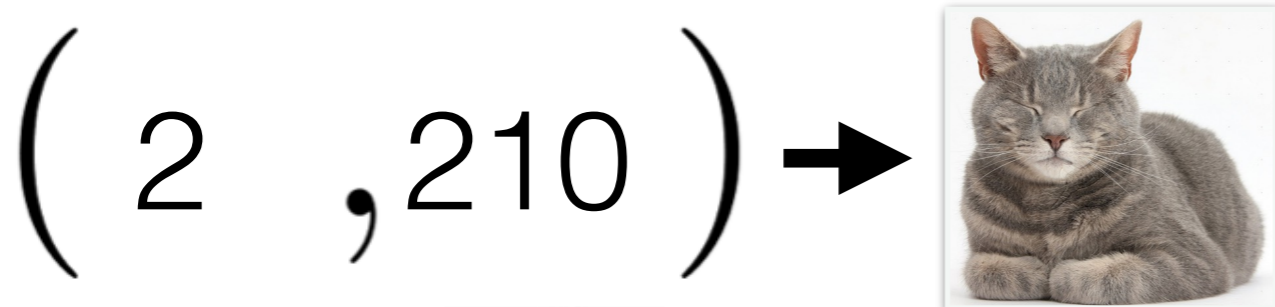
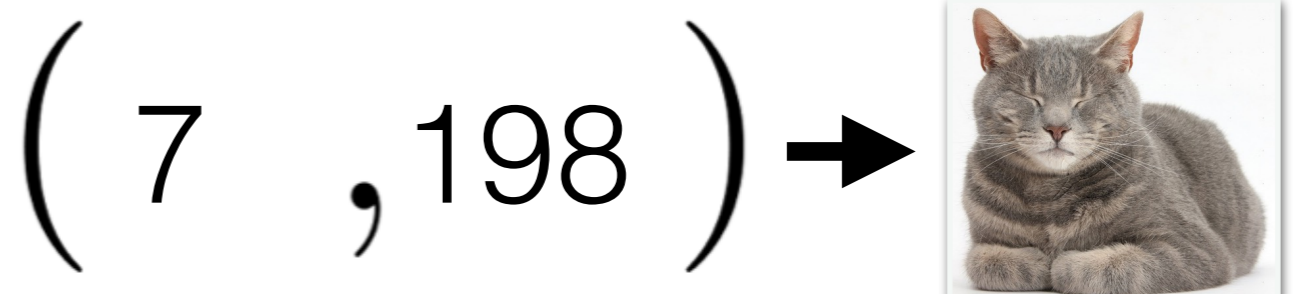
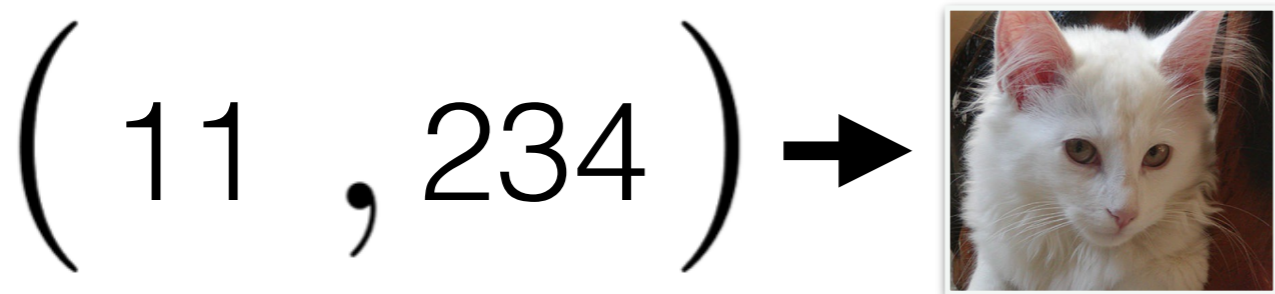
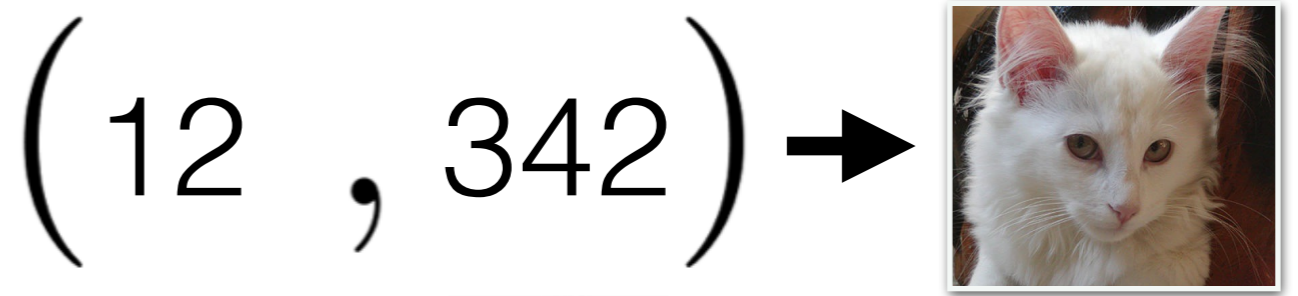
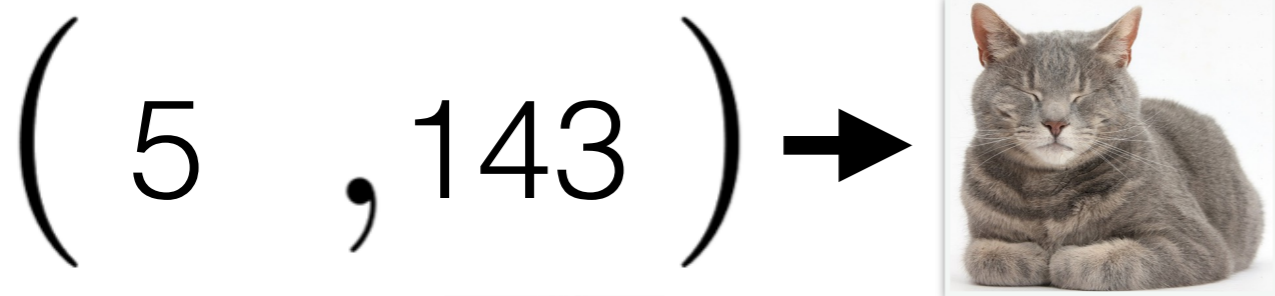
$(5, 321)$ →



DATASET



Infer a rule to classify these cats.



DATASET

Congratulations!
You have invented "OneR" algorithm*

oneR {FSelector}

R Documentation

OneR algorithm

Description

The algorithms find weights of discrete attributes basing on very simple association rules involving only one attribute in condition part.

Usage

```
oneR(formula, data)
```

Arguments

`formula` a symbolic description of a model

`data` data to process

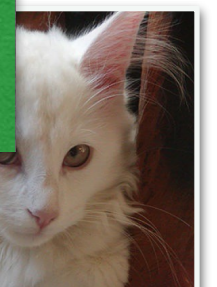
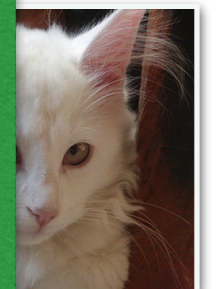
Details

The algorithm uses OneR classifier to find out the attributes' weights. For each attribute it creates a simple rule based only on that attribute and then calculates its error rate.

Value

a data.frame containing the worth of attributes in the first column and their names as row names

`Author(s)`



(5

(1

(2

, 210)

(3 , 321)

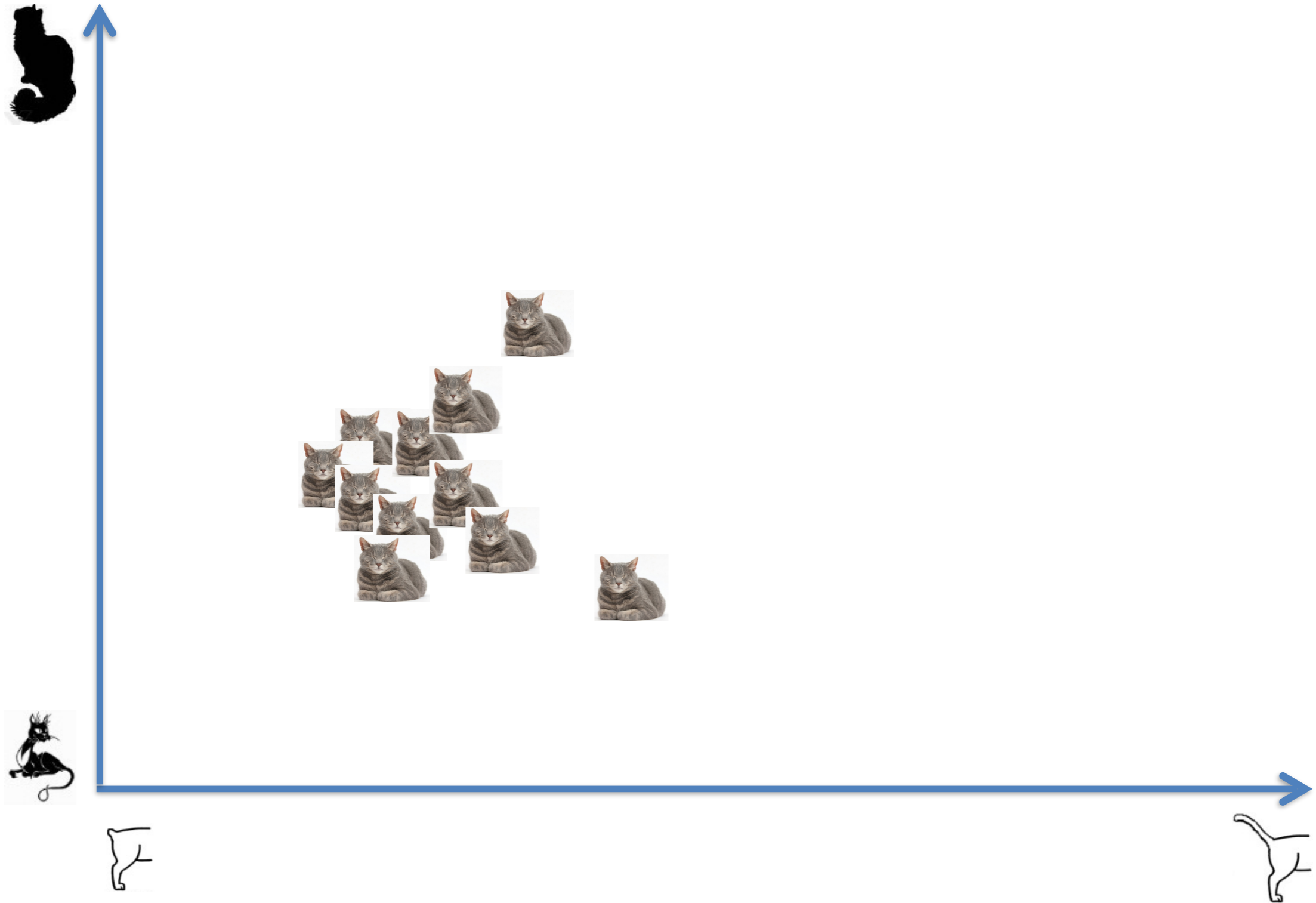
FEATURE SPACE



FEATURE SPACE



FEATURE SPACE



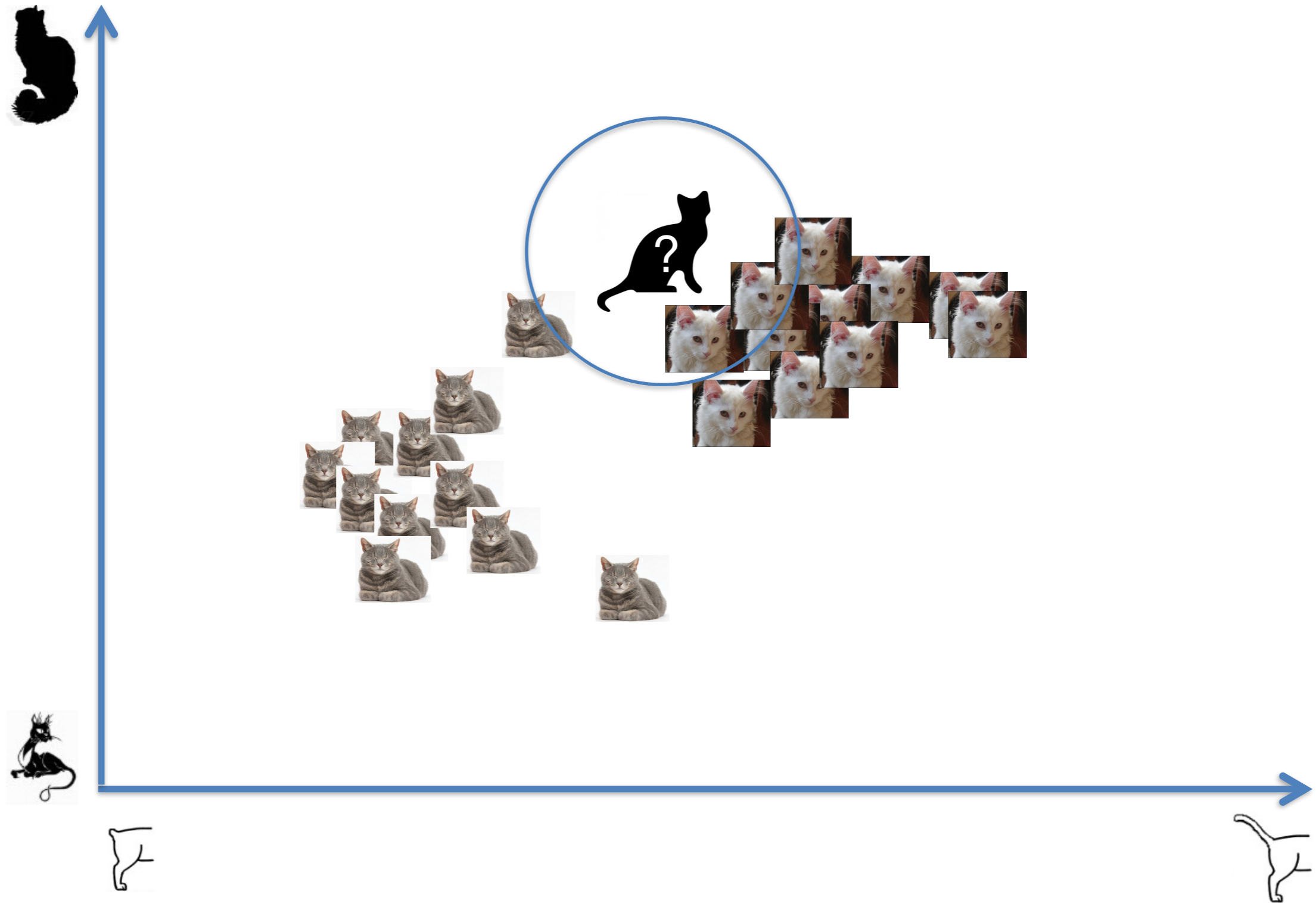
FEATURE SPACE



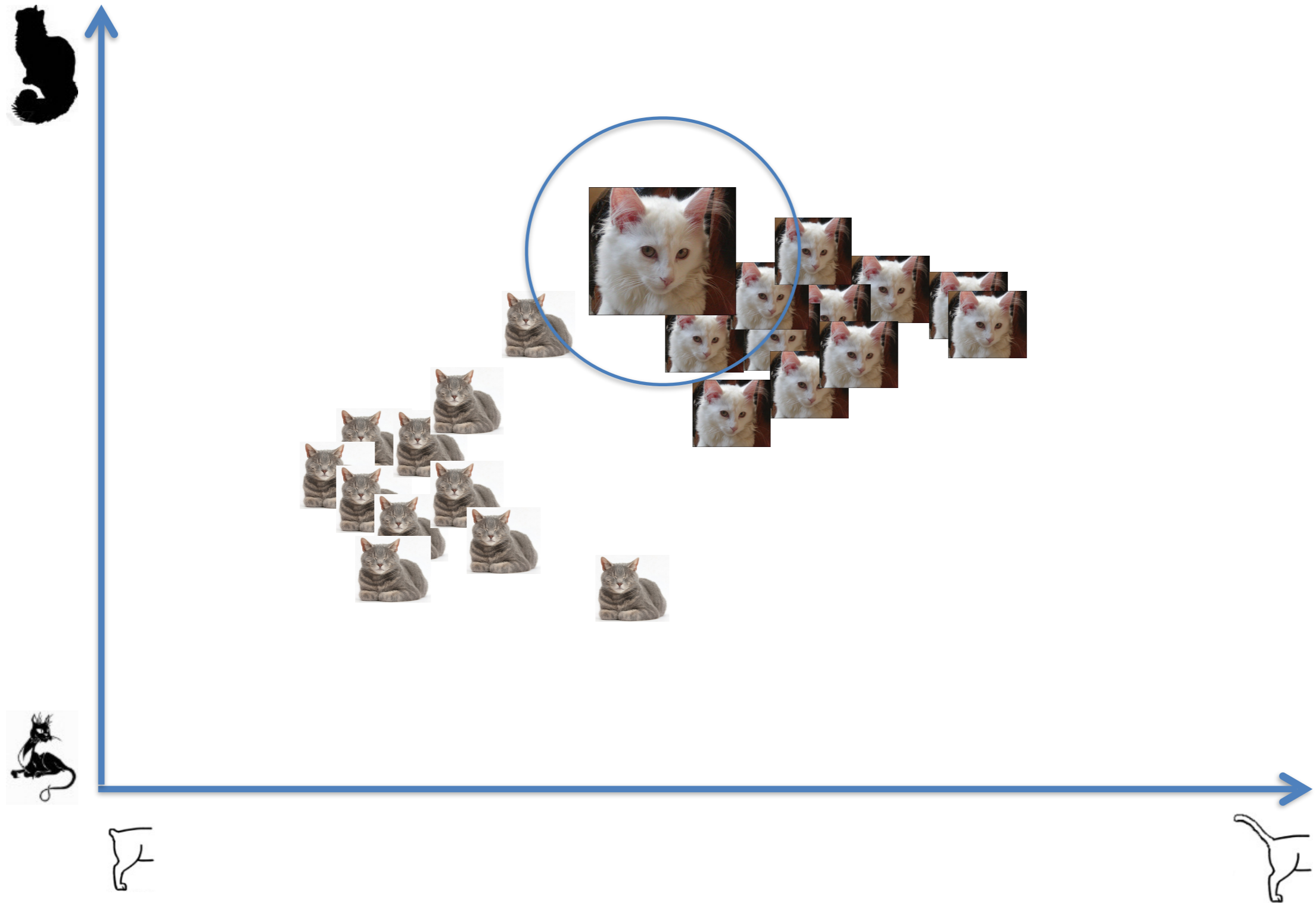
FEATURE SPACE



FEATURE SPACE



FEATURE SPACE



FEATURE SPACE

The image shows a screenshot of a Wikipedia article titled "k-nearest neighbors algorithm". The page layout includes a left sidebar with the Wikipedia logo and navigation links, a top navigation bar with "Article" and "Talk" tabs, and a search box. The main content area contains the article title, a subtitle "From Wikipedia, the free encyclopedia", and a paragraph explaining the algorithm in the context of pattern recognition and feature space. A bulleted list describes the classification process. To the right, there is a diagram titled "Machine learning and data mining" showing a scatter plot with a dashed decision boundary line separating two clusters of data points.

WIKIPEDIA
The Free Encyclopedia

Article [Talk](#) [Read](#) [Edit](#) [View history](#)

k-nearest neighbors algorithm

From Wikipedia, the free encyclopedia

In [pattern recognition](#), the **k-Nearest Neighbors algorithm** (or **k-NN** for short) is a [non-parametric](#) method used for [classification](#) and [regression](#).^[1] In both cases, the input consists of the *k* closest training examples in the [feature space](#). The output depends on whether *k*-NN is used for classification or regression:

- In *k*-NN *classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive [integer](#), typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.

Machine learning and data mining

Problems
[Classification](#) · [Clustering](#) · [Regression](#)



- Decision trees
- C4.5
- Random forests
- Bayesian networks
- Hidden Markov models
- Artificial neural network
- Data clustering
- Expectation-maximization algorithm
- Self-organizing map
- Radial basis function network
- Vector Quantization
- Generative topographic map
- Information bottleneck method
- IBSEAD
- Apriori algorithm
- Eclat algorithm
- FP-growth algorithm
- Single-linkage clustering
- Conceptual clustering
- K-means algorithm
- Fuzzy clustering
- Temporal difference learning
- Q-learning
- Learning Automata
- AODE
- Artificial neural network
- Backpropagation
- Naive Bayes classifier
- Bayesian network
- Bayesian knowledge base
- Case-based reasoning
- Decision trees
- Inductive logic programming
- Gaussian process regression
- Gene expression programming
- Group method of data handling (GMDH)
- Learning Automata
- Learning Vector Quantization
- Logistic Model Tree
- Decision trees
- Decision graphs
- Lazy learning
- Monte Carlo Method
- SARSA
- Instance-based learning
- Nearest Neighbor Algorithm
- Analogical modeling
- Probably approximately correct learning (PAC)
- Symbolic machine learning algorithms
- Subsymbolic machine learning algorithms
- Support vector machines
- Random Forests
- Ensembles of classifiers
- Bootstrap aggregating (bagging)
- Boosting (meta-algorithm)
- Ordinal classification
- Regression analysis
- Information fuzzy networks (IFN)
- Linear classifiers
- Fisher's linear discriminant
- Logistic regression
- Naive Bayes classifier
- Perceptron
- Support vector machines
- Quadratic classifiers
- k-nearest neighbor
- Boosting

DOES IT WORK?



ACCURACY

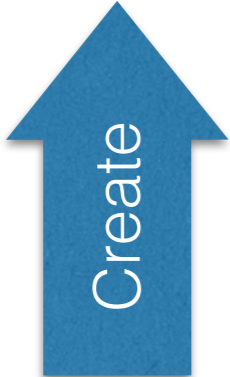


Training set

ACCURACY



Test set



Training set

ACCURACY



Test set

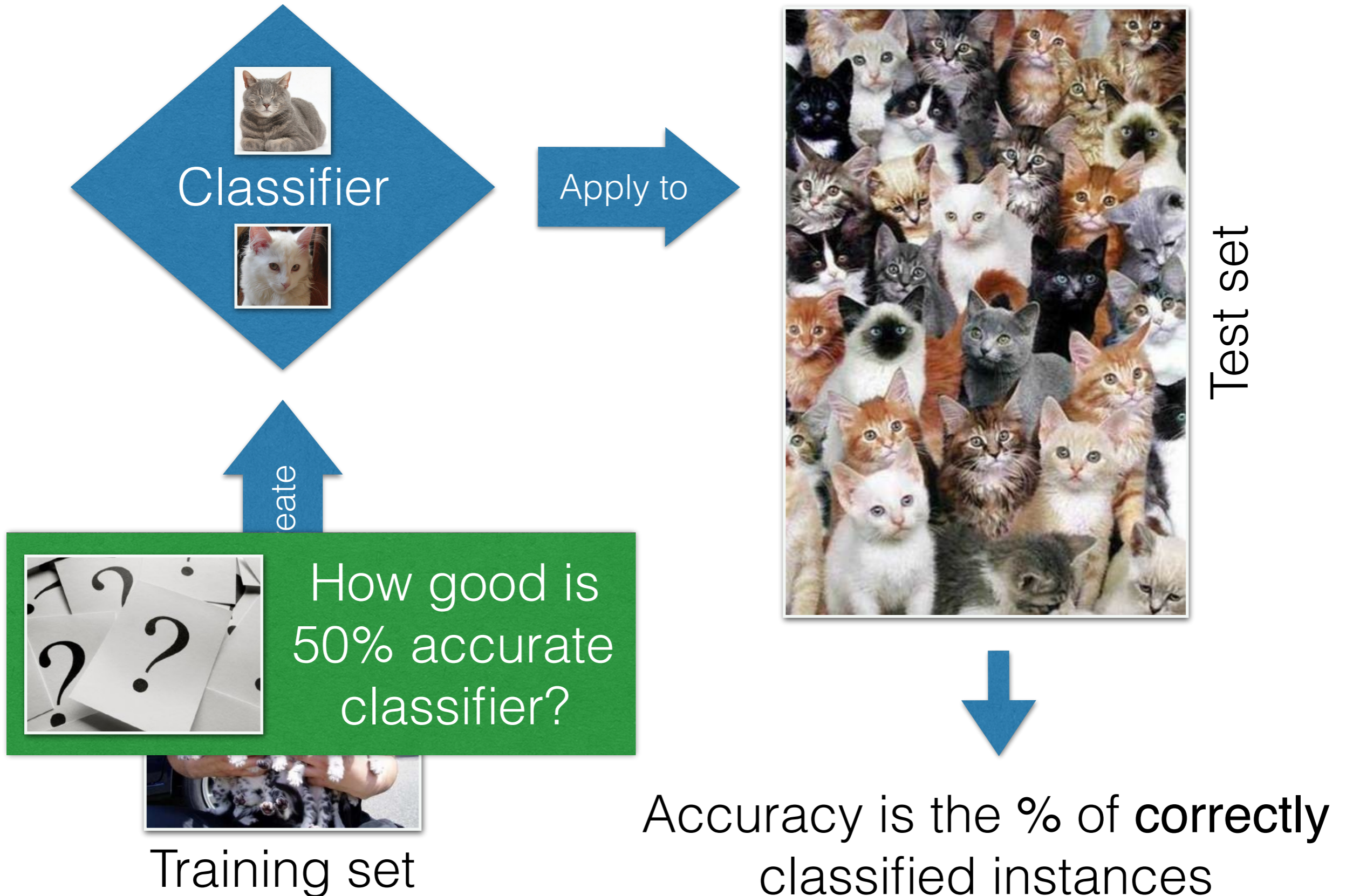


Training set

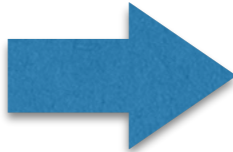


Accuracy is the % of **correctly** classified instances

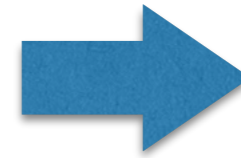
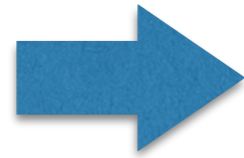
ACCURACY



UNBALANCED SET

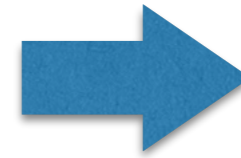
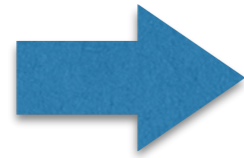


UNBALANCED SET



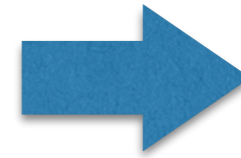
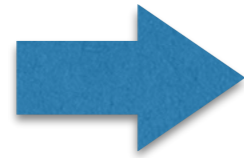
Accuracy is
?

UNBALANCED SET



Accuracy is
0.5

UNBALANCED SET

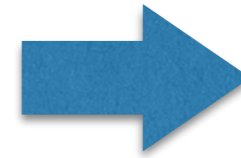
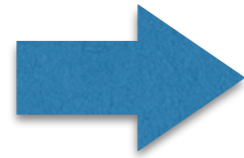


Accuracy is
0.5



Accuracy is
?

UNBALANCED SET



Accuracy is
0.5



Accuracy is
0.9

PRECISION

Correctly identified as



Everything identified as



PRECISION

Correctly identified as



Everything identified as



PRECISION

Correctly identified as



Everything identified as



Precision = ?



PRECISION

Correctly identified as



Everything identified as



Precision = 0.9



PRECISION

Correctly identified as



Everything identified as



Precision = 0.9



Precision = ?



PRECISION

Correctly identified as



Everything identified as



Precision = 0.9



Precision = 0



PRECISION

Correctly identified as



Everything identified as



Precision = 0.9



Precision = 0



Precision = 0.45

RECALL

Identified



All




in the test set

RECALL

Identified



All  in the test set



RECALL

Identified



All  in the test set




Recall = ?



RECALL

Identified



All  in the test set



Recall = 1



RECALL

Identified 

All  in the test set



Recall = 1



Recall = ?



RECALL

Identified 

All  in the test set



Recall = 1



Recall = 0



RECALL

Identified 

All  in the test set



Recall = 1



Recall = 0



Recall = 0.5

F1 SCORE

$$\text{F-score}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$$

F1 SCORE

$$F\text{-score}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$$



Precision = 0.9



Precision = 0



Recall = 1



Recall = 0

F1 SCORE

$$F\text{-score}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$$



Precision = 0.9



Precision = 0



Recall = 1



Recall = 0



F1 \approx 0.95



F1 = 0

F1 SCORE

$$F\text{-score}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$$



Precision = 0.9



Precision = 0



Recall = 1



Recall = 0



F1 \approx 0.95

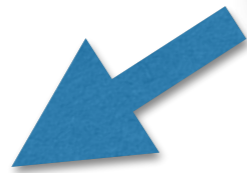


F1 = 0

Average F1 \approx 0.48

What is this **test set**
you are talking about?

TRAINING - VALIDATION - TEST



50%



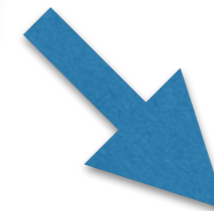
Fit model on a
training set



25%



Tune parameters
on a validation set



25%

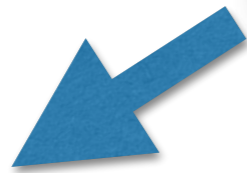


Final test on
a test set

TRAINING - VALIDATION - TEST



But why can't we do it all on one set?



50%



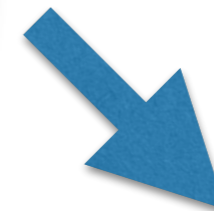
Fit model on a training set



25%



Tune parameters on a validation set

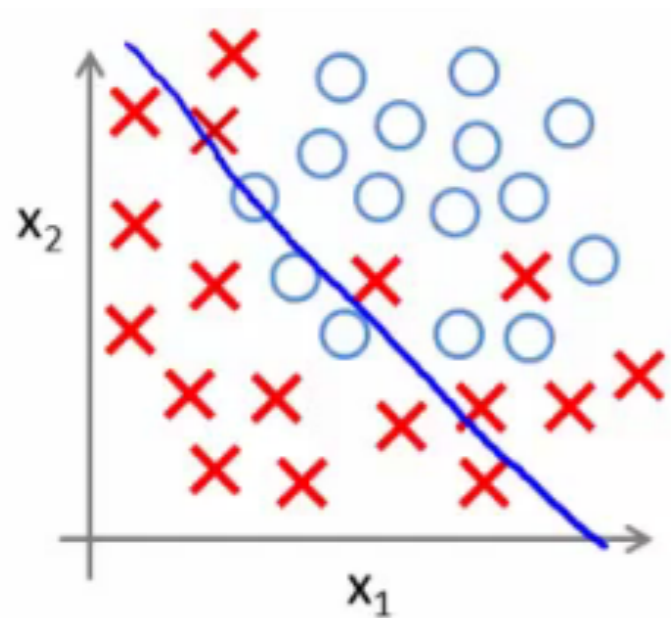


25%

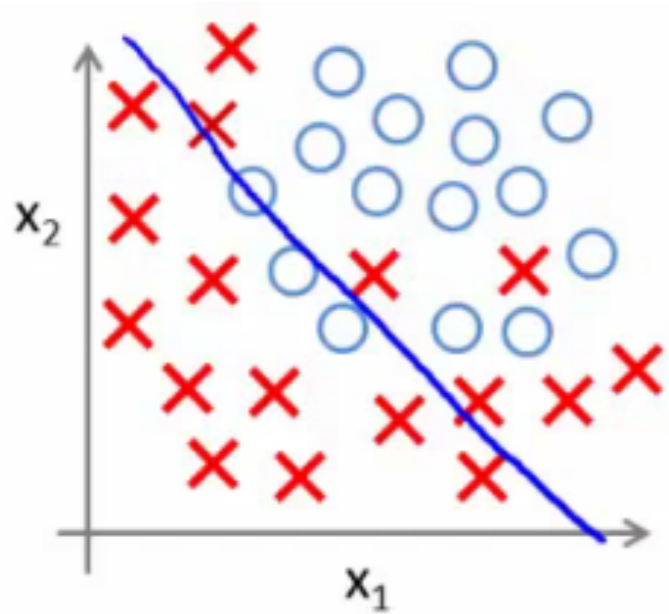


Final test on a test set

BIAS-VARIANCE TRADEOFF



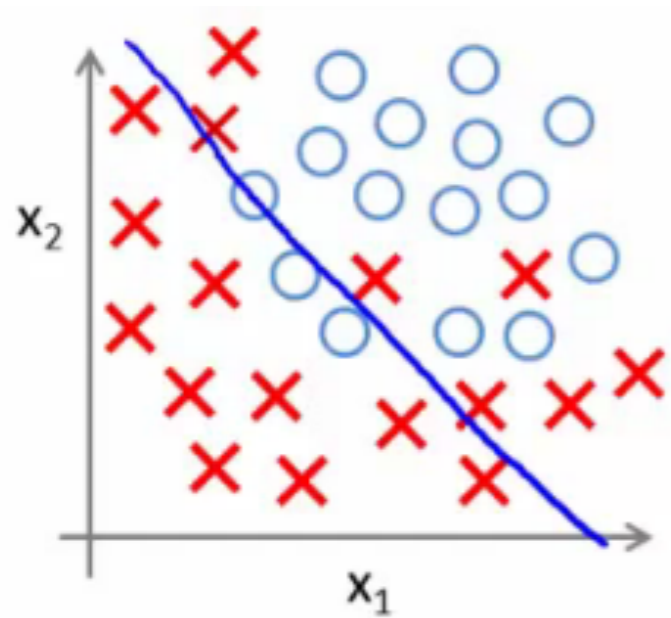
BIAS-VARIANCE TRADEOFF



- High bias
- Low variance
- Underfitting

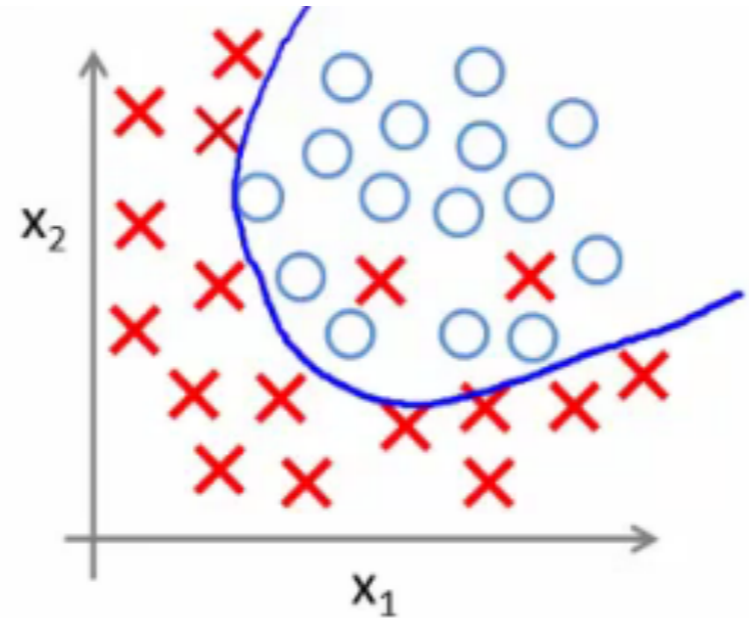
a.k.a
Too stupid

BIAS-VARIANCE TRADEOFF



- High bias
- Low variance
- Underfitting

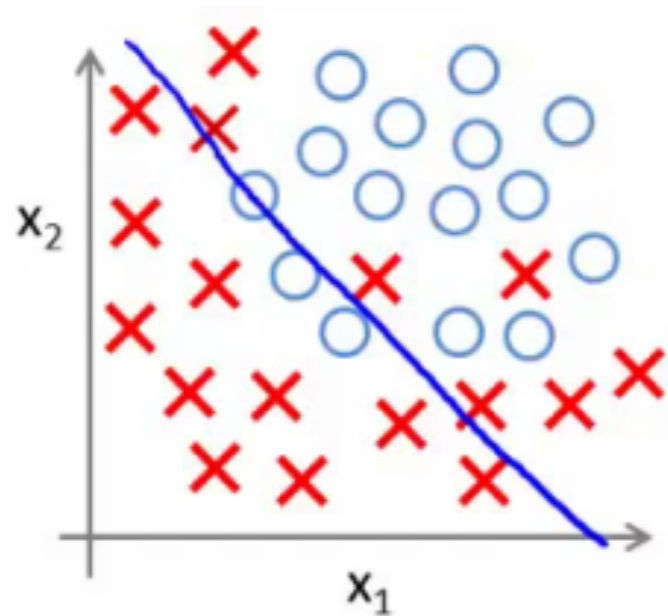
a.k.a
Too stupid



Balanced
bias-variance
tradeoff

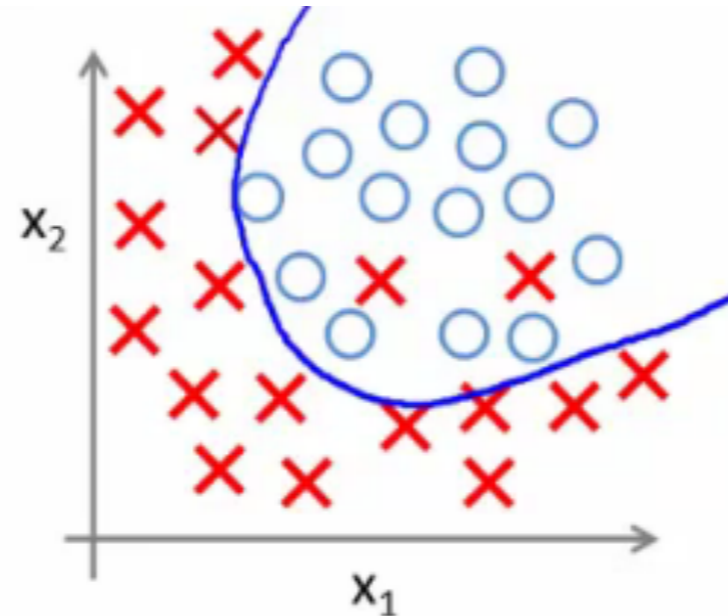
a.k.a
OK

BIAS-VARIANCE TRADEOFF



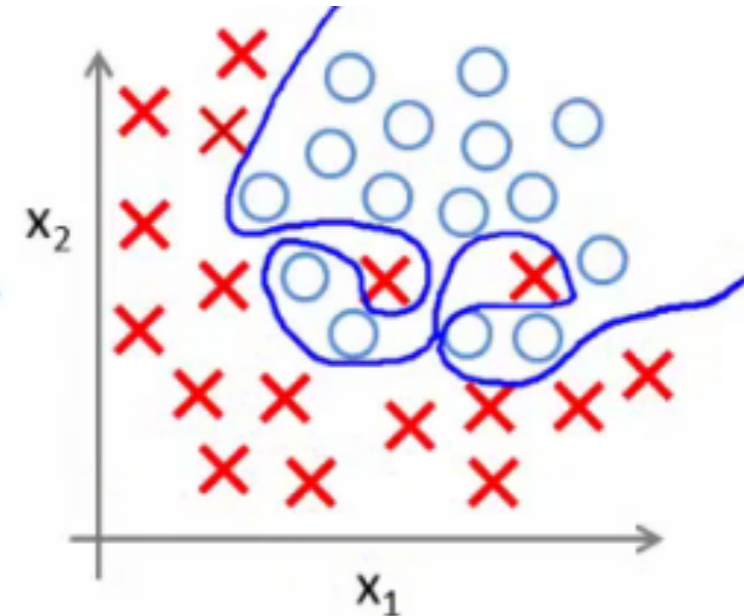
- High bias
- Low variance
- Underfitting

a.k.a
Too stupid



Balanced
bias-variance
tradeoff

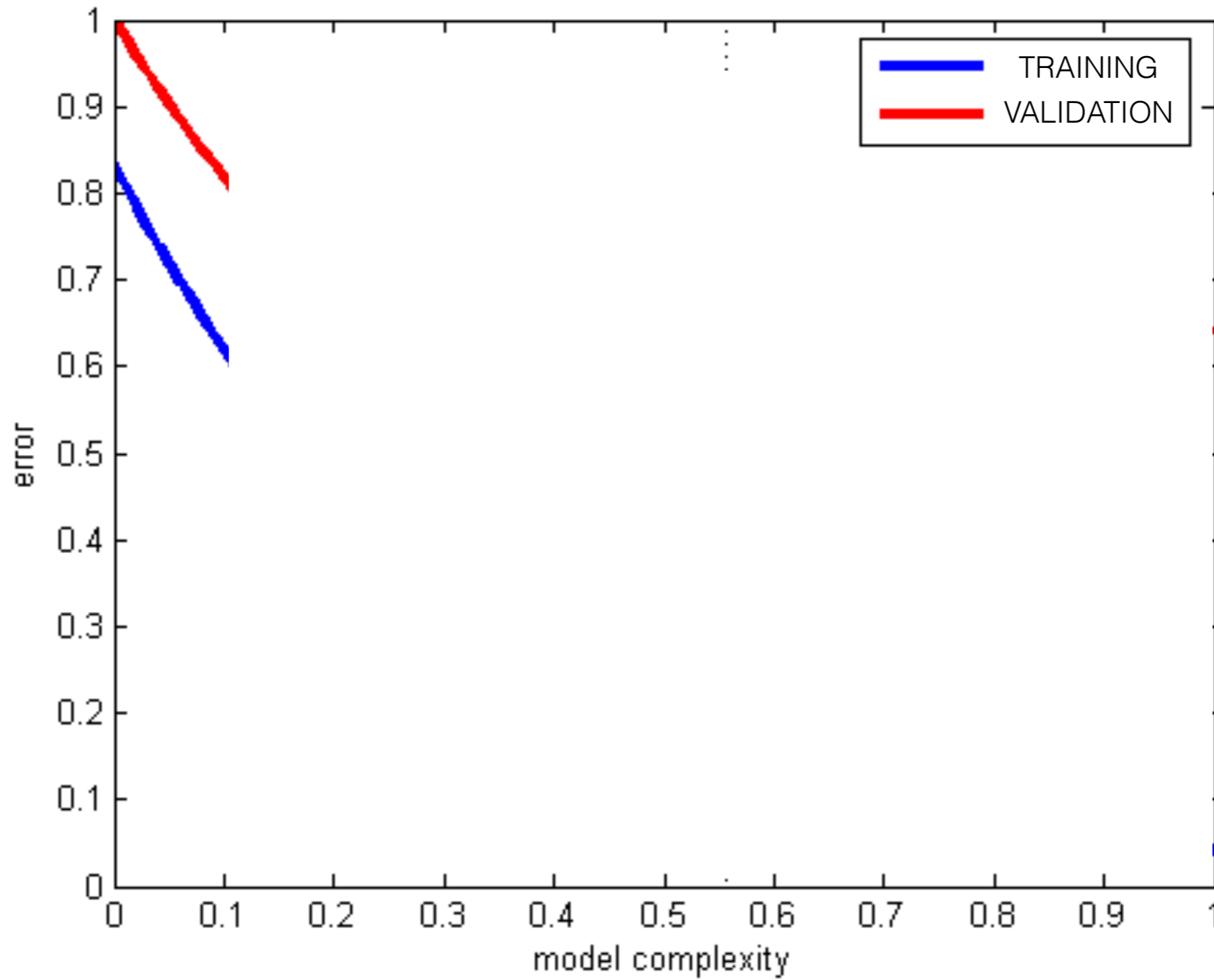
a.k.a
OK



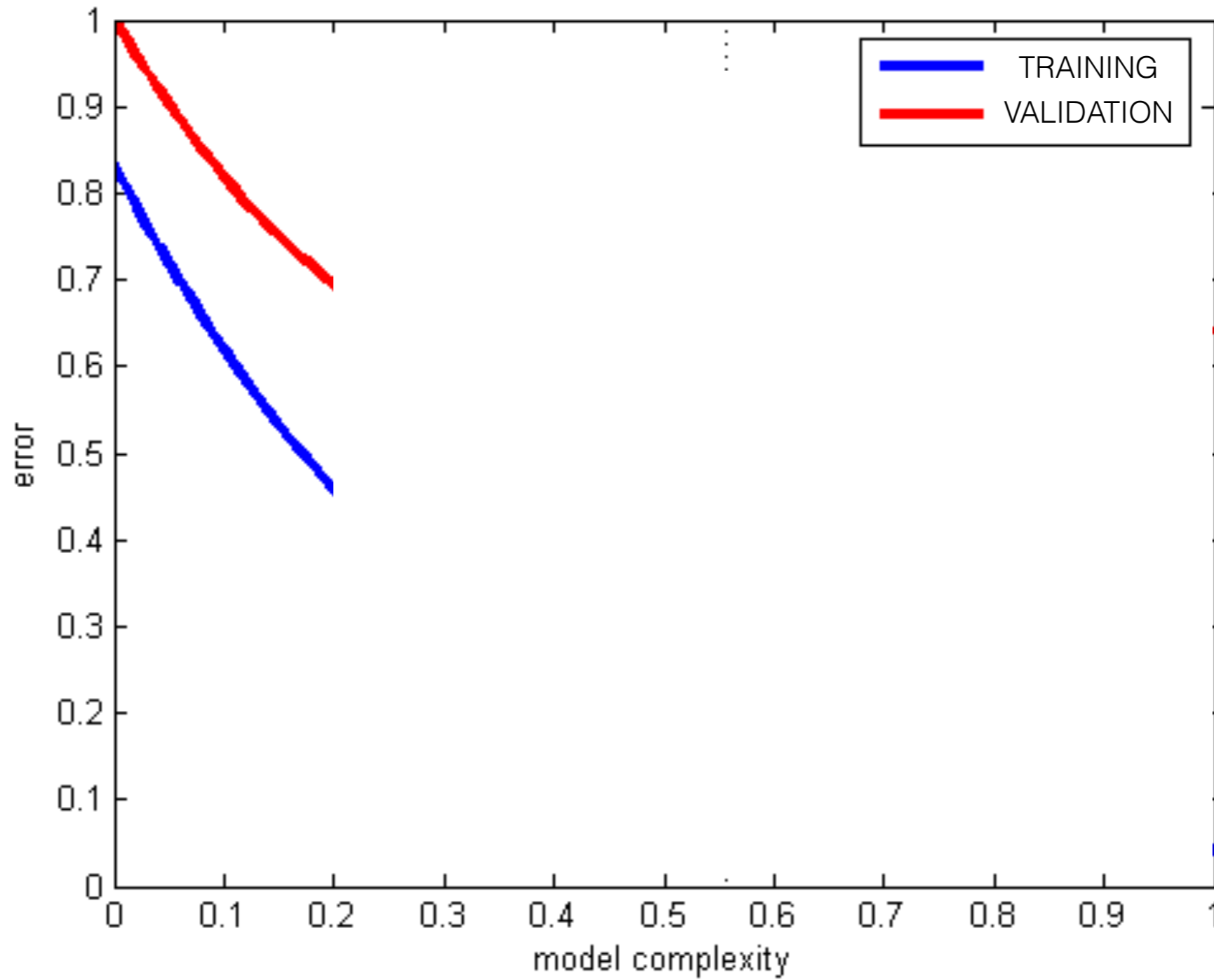
- Low bias
- High variance
- Overfitting

a.k.a
Too smart

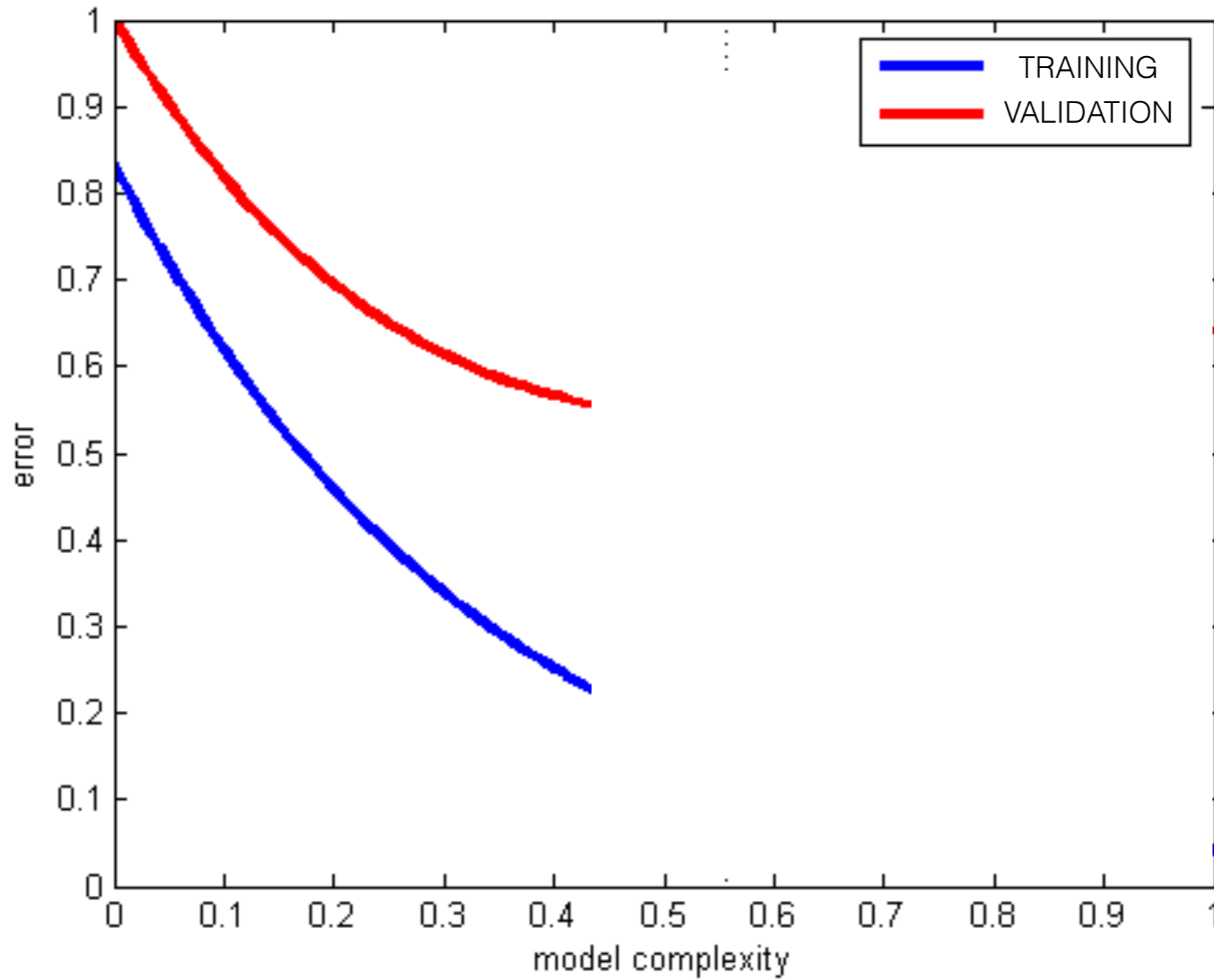
OVERFITTING



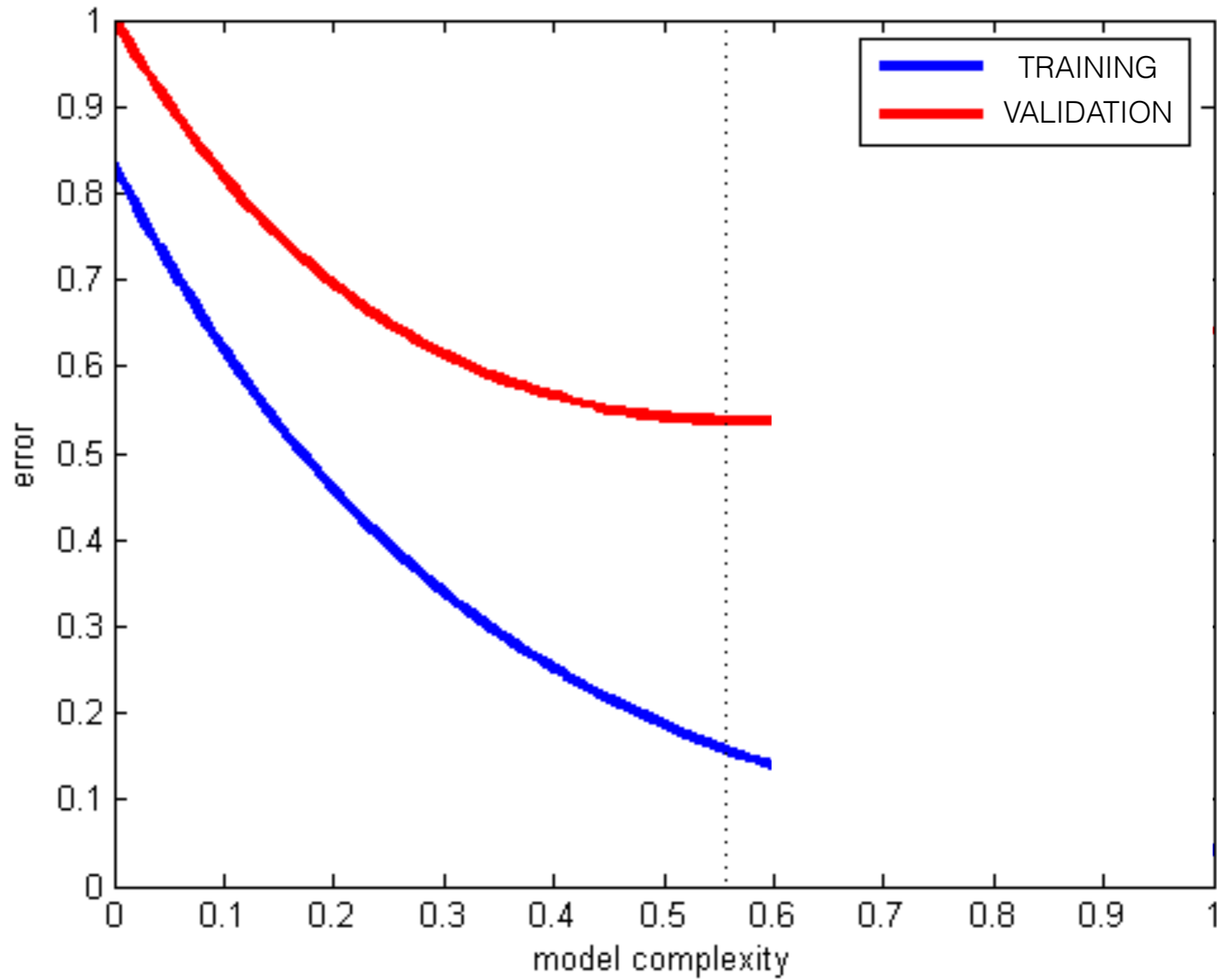
OVERFITTING



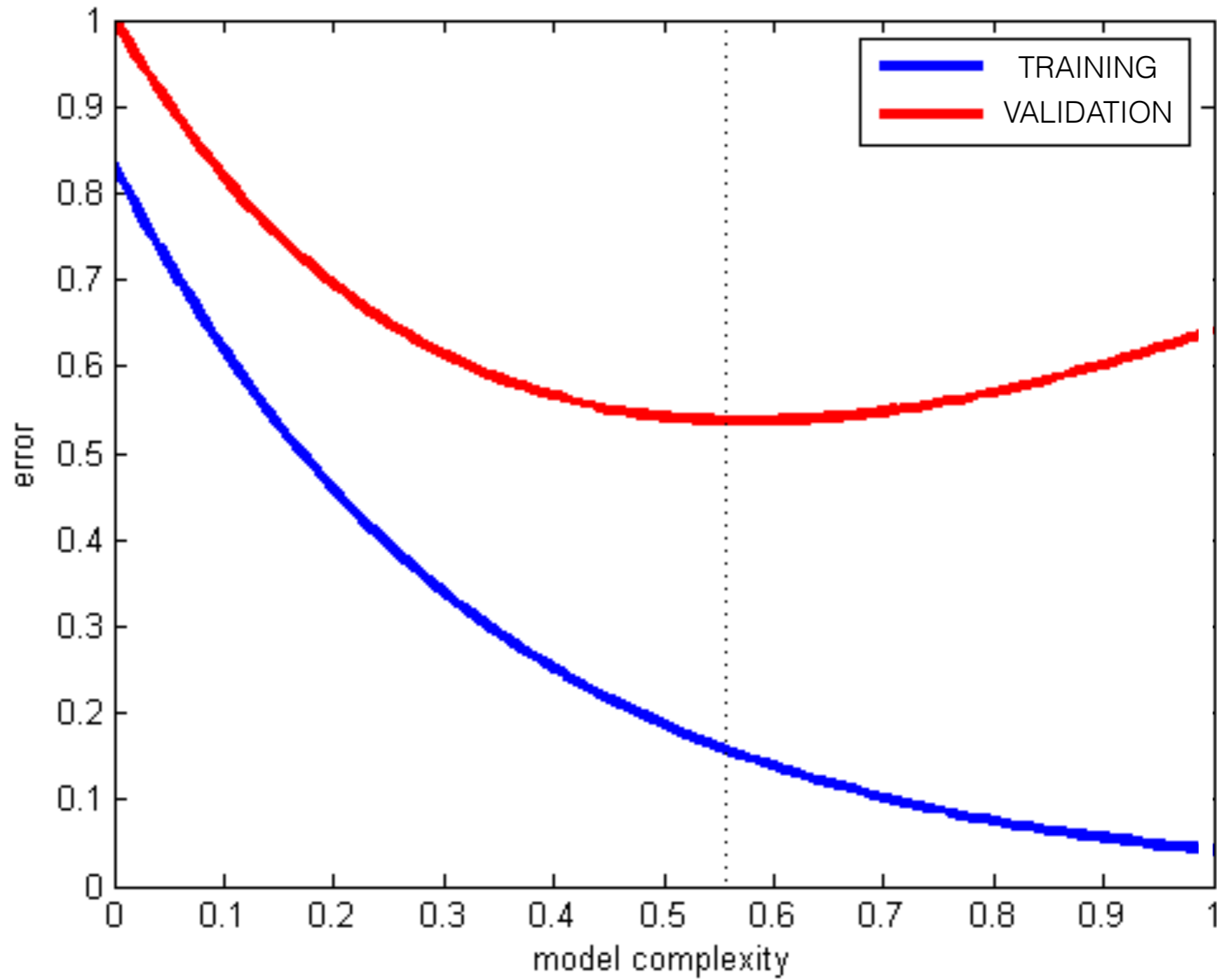
OVERFITTING



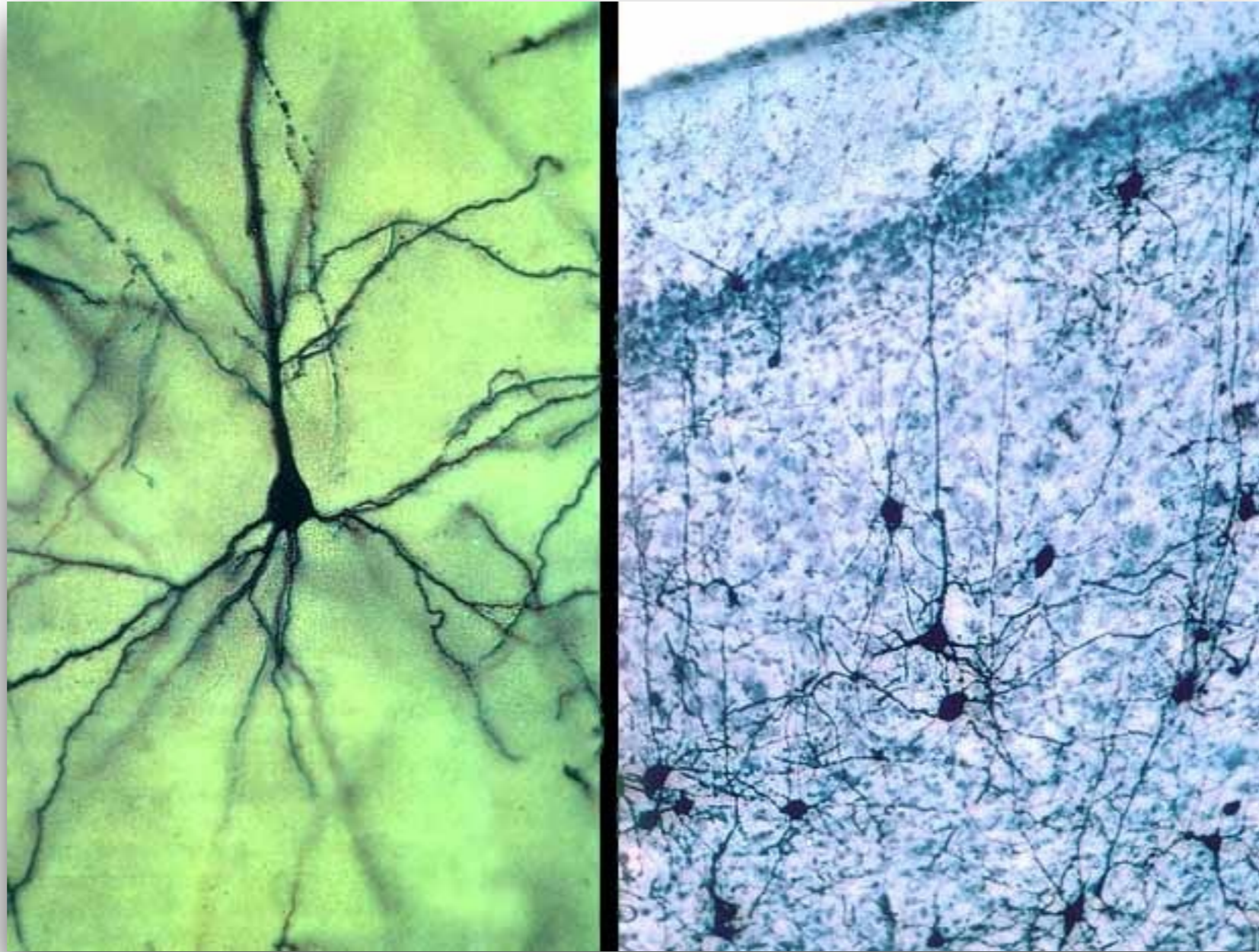
OVERFITTING



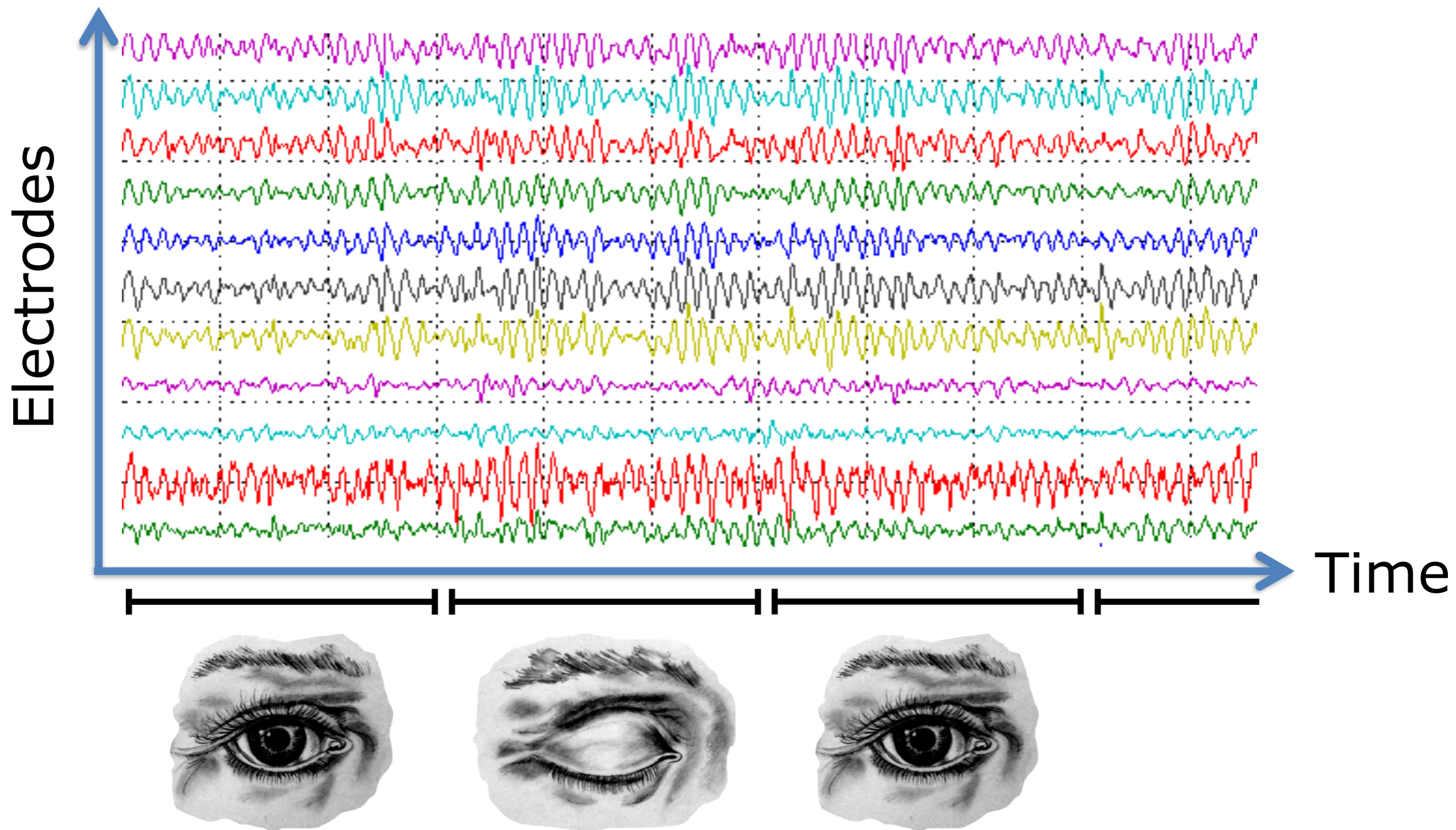
OVERFITTING

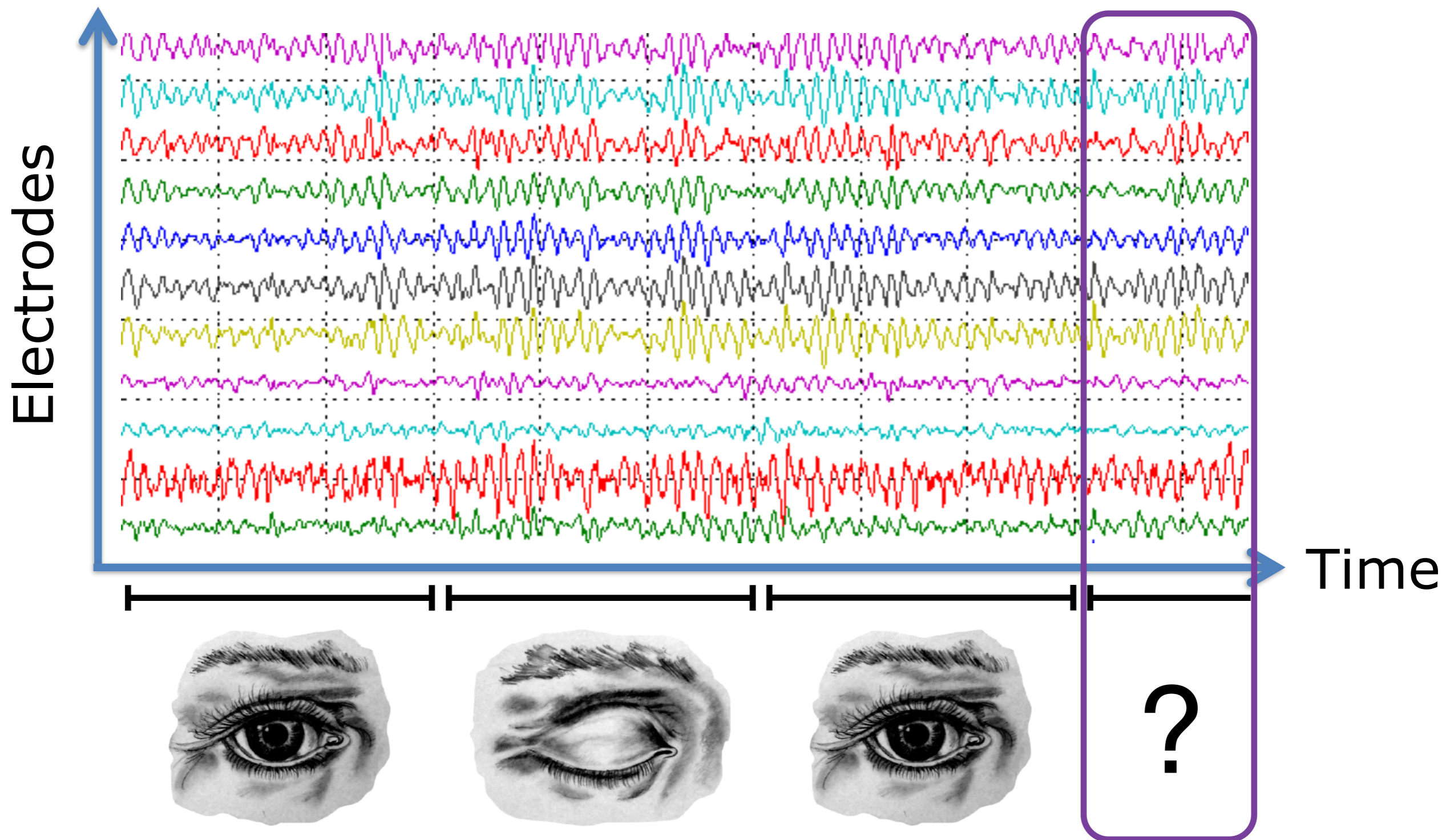


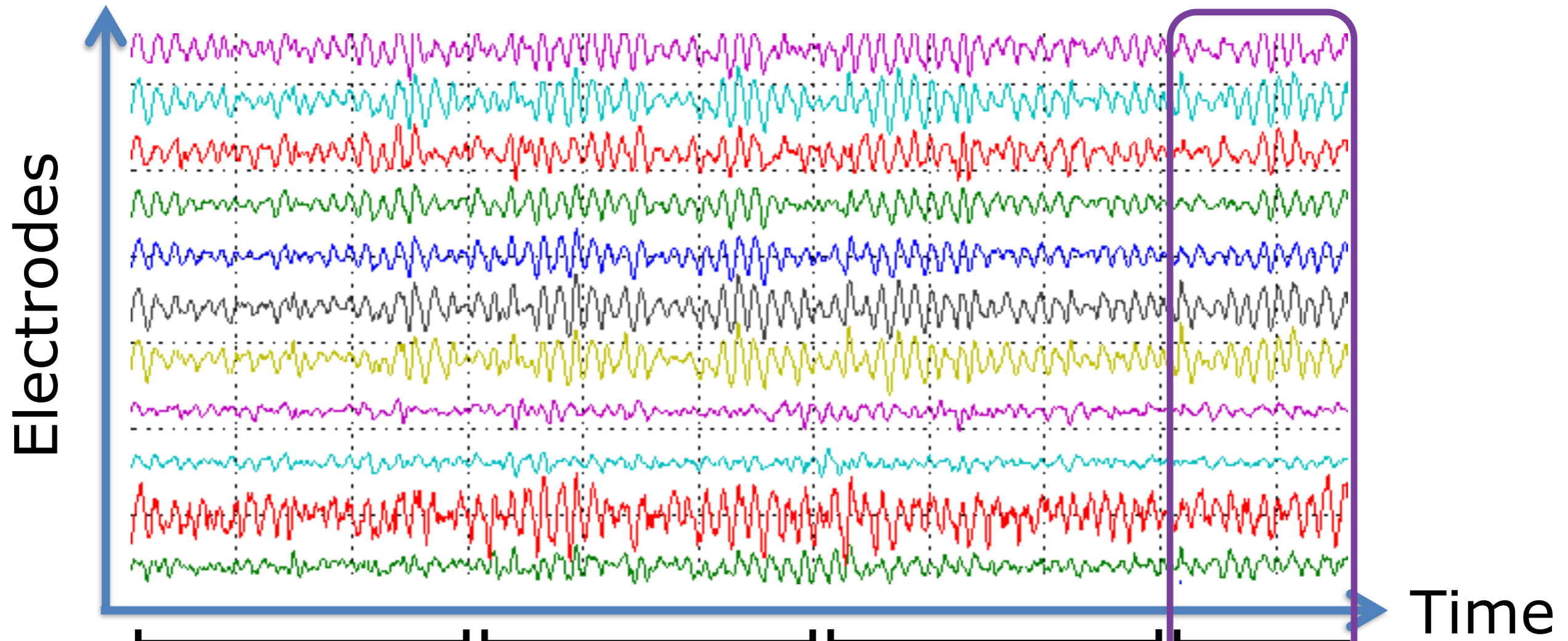
- ML for automatic analysis
- Data as features
- Feature space
- Algorithms
- Performance measures
- Training, Validation, Test
- Overfitting



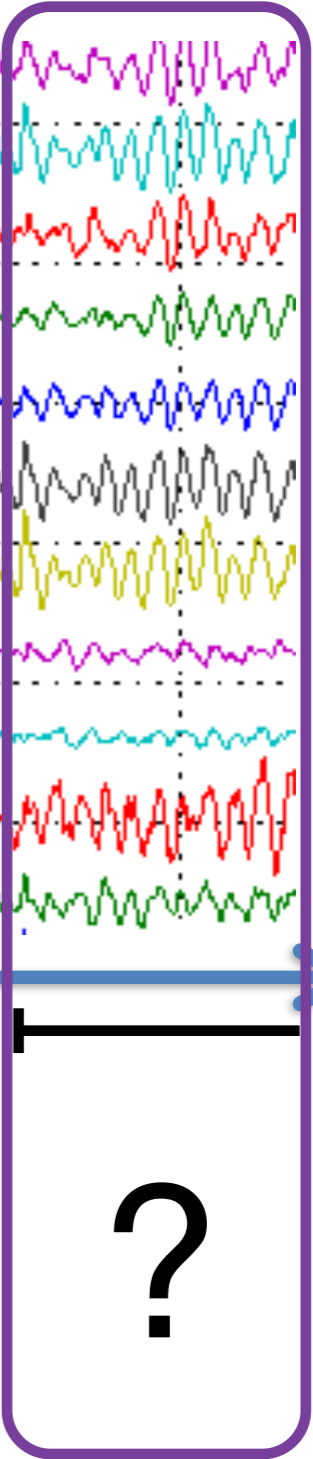
PART II
BACK TO BRAINS

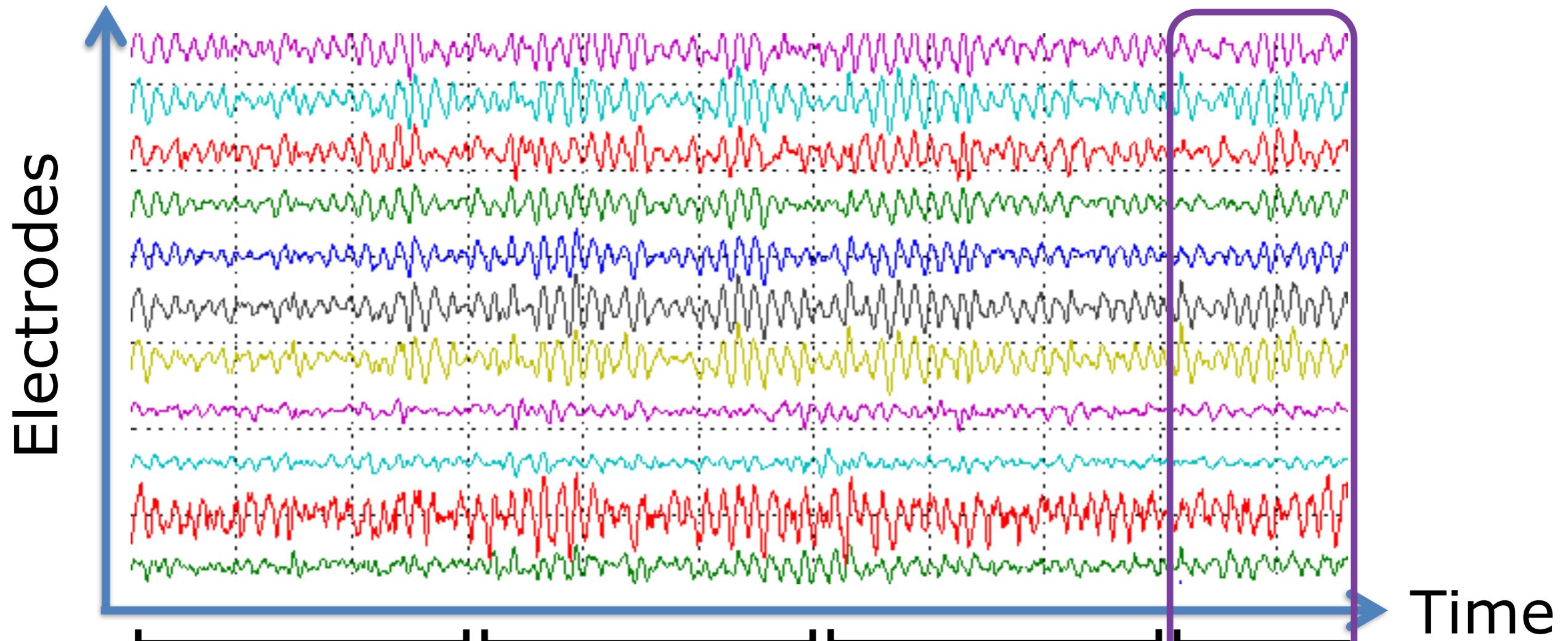






What is your next move?

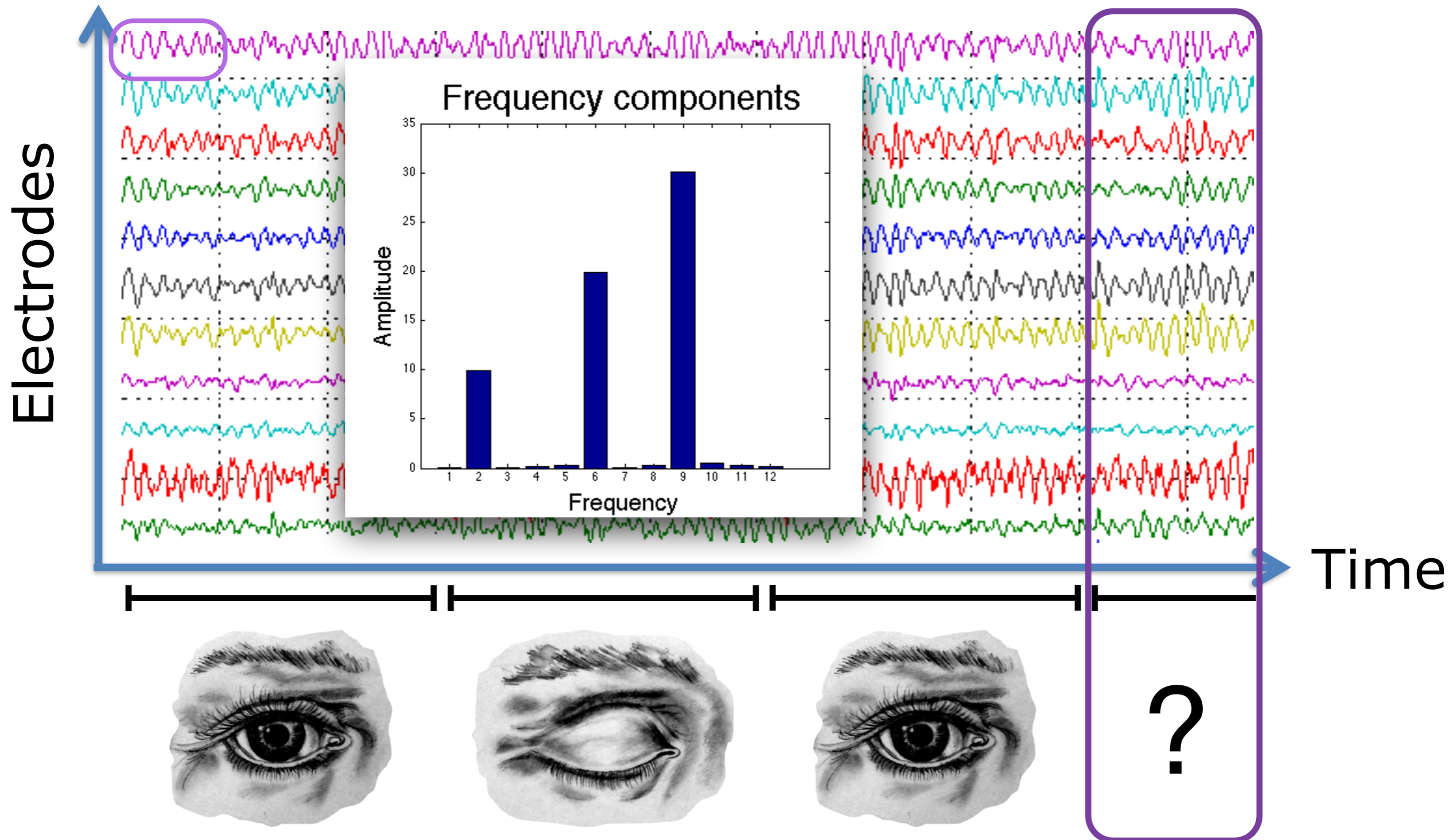




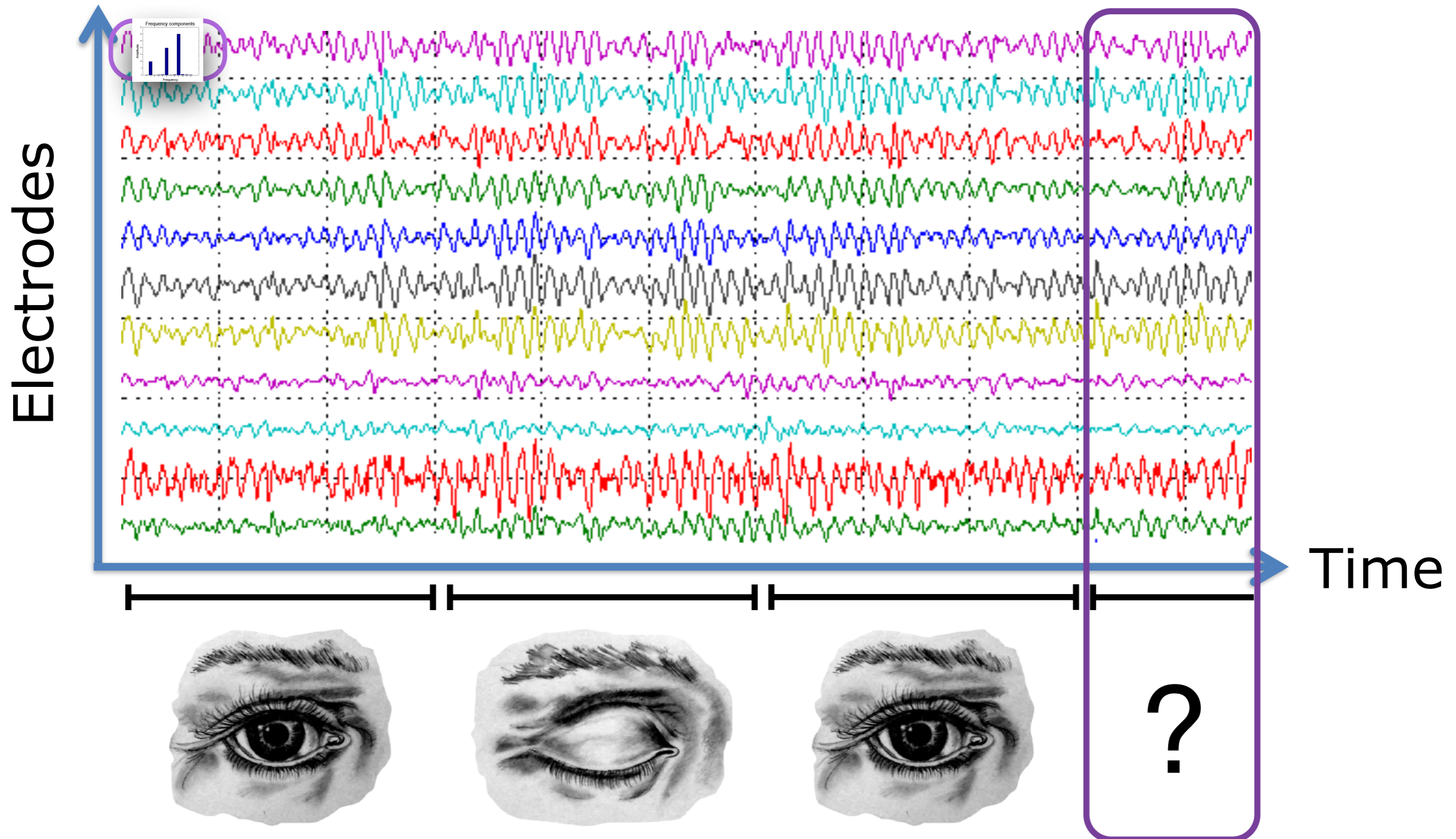
Fourier transform of what?

?

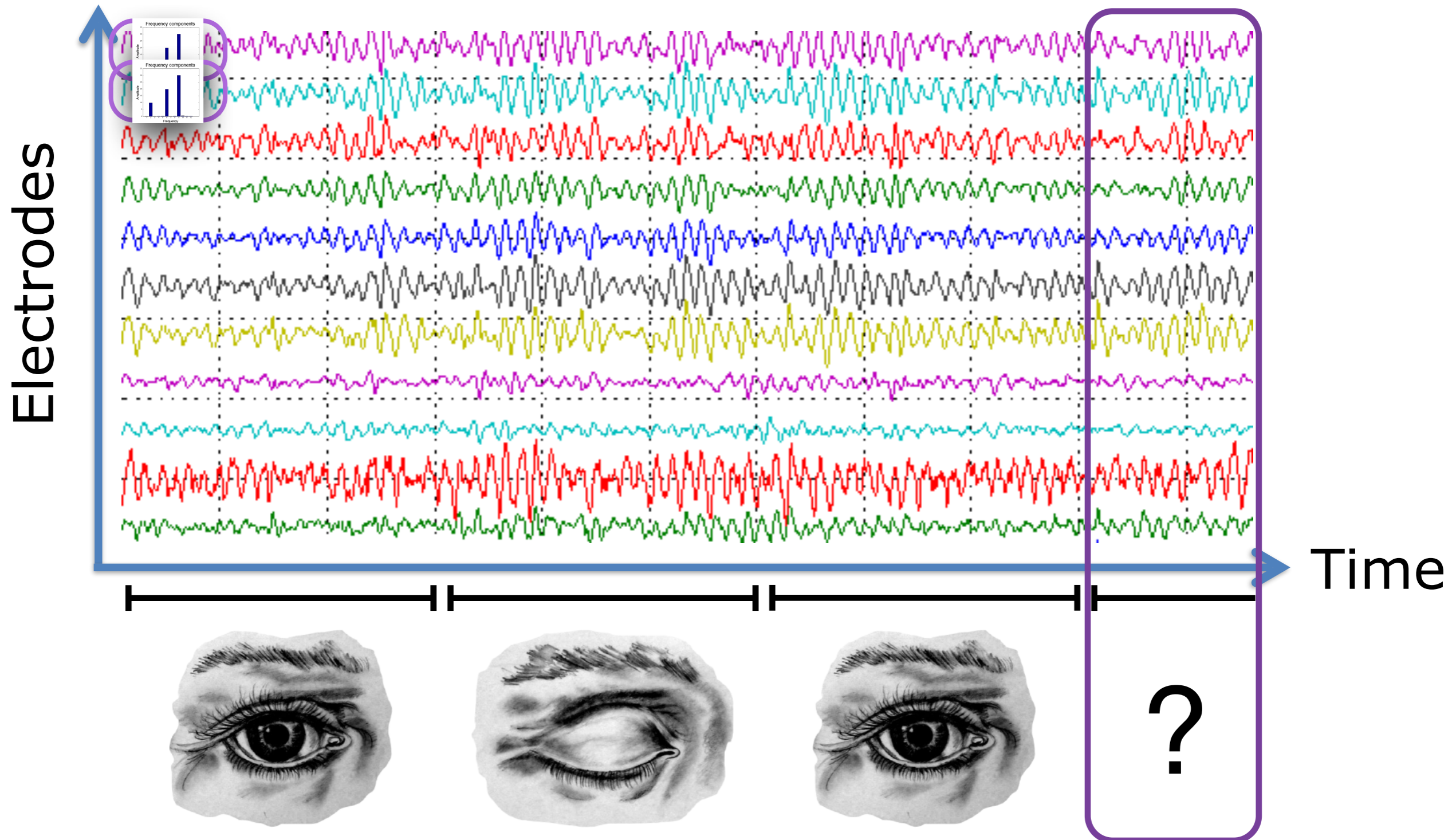
TIME-FREQUENCY DOMAIN



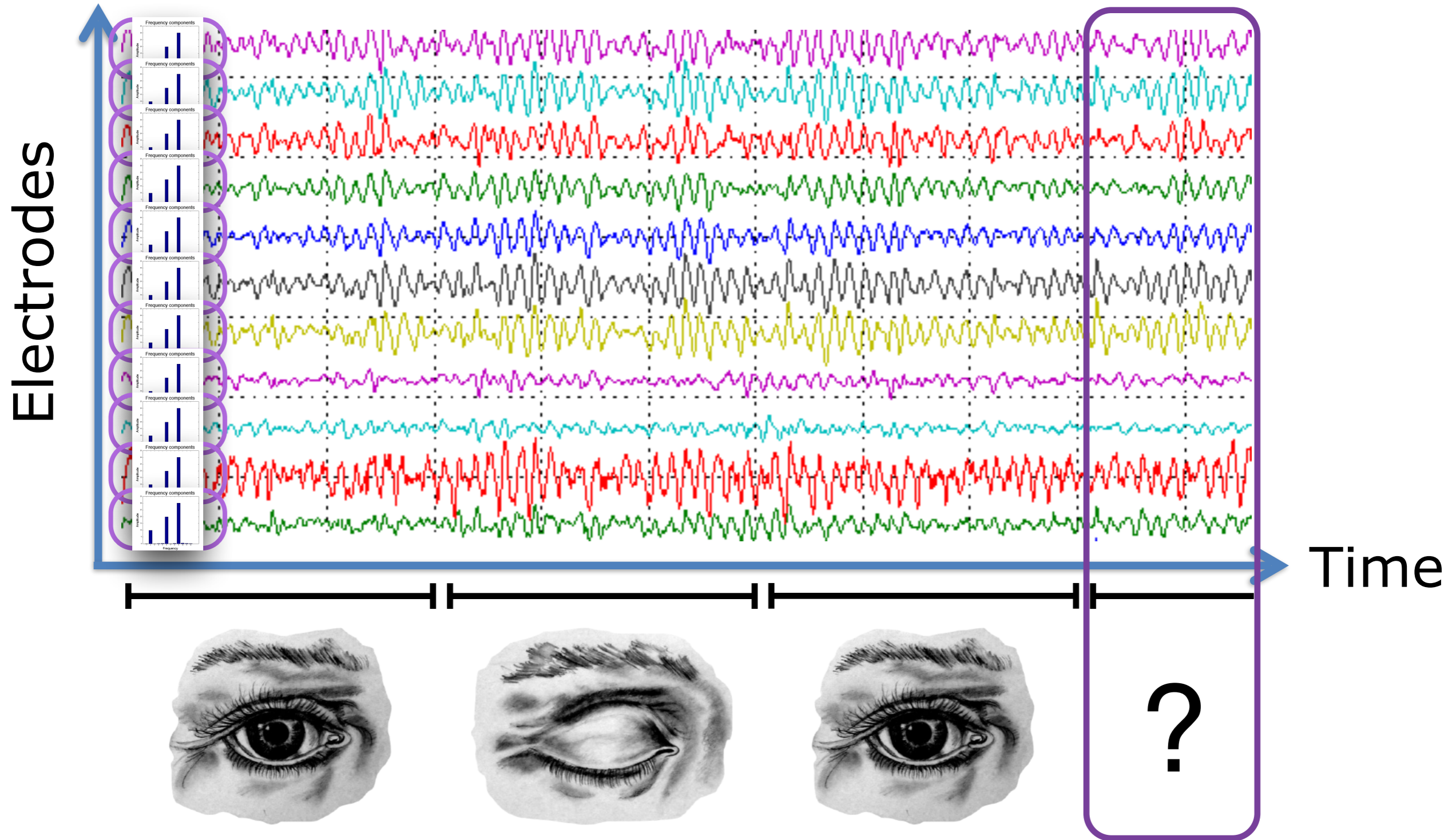
TIME-FREQUENCY DOMAIN



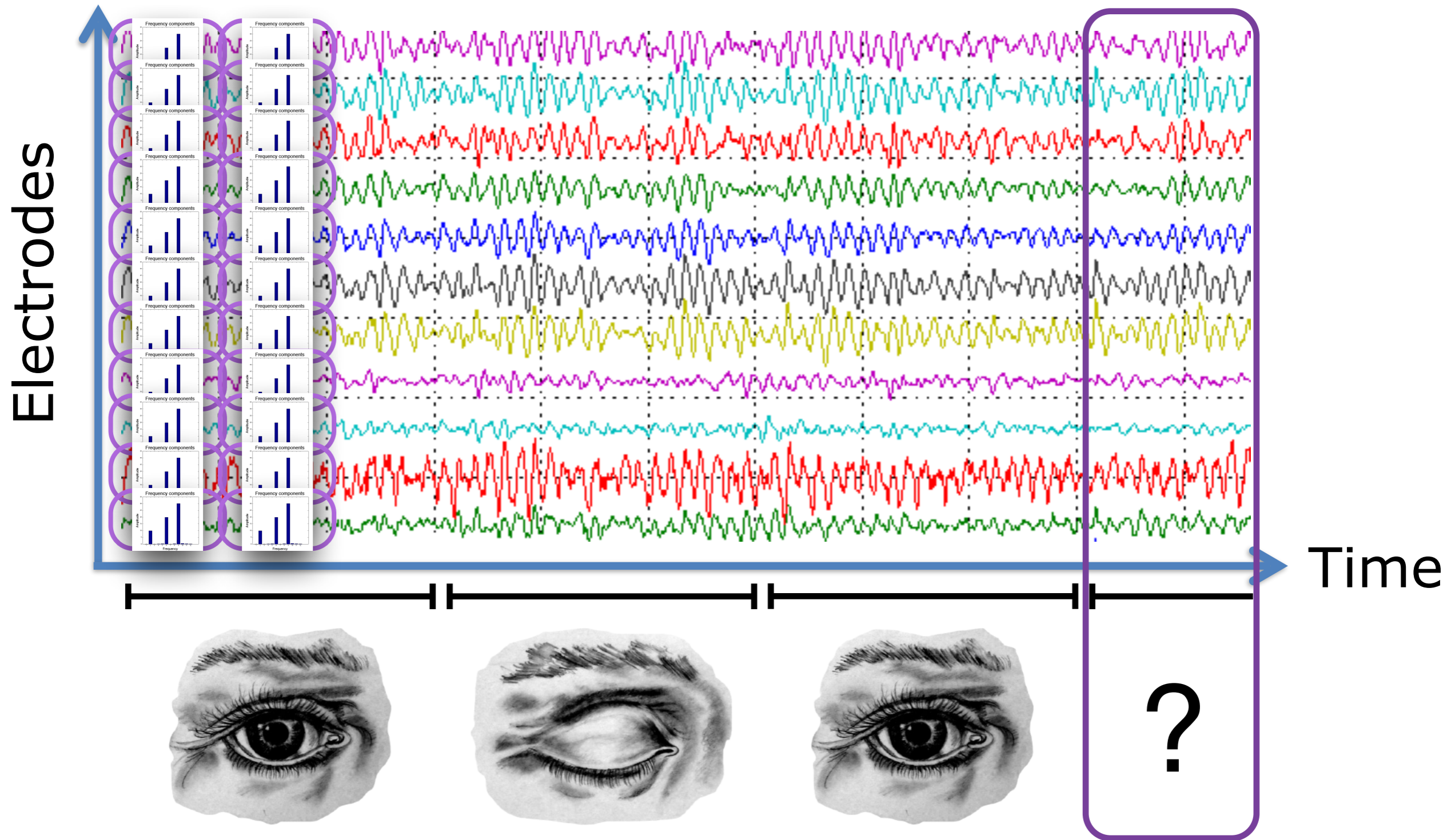
TIME-FREQUENCY DOMAIN



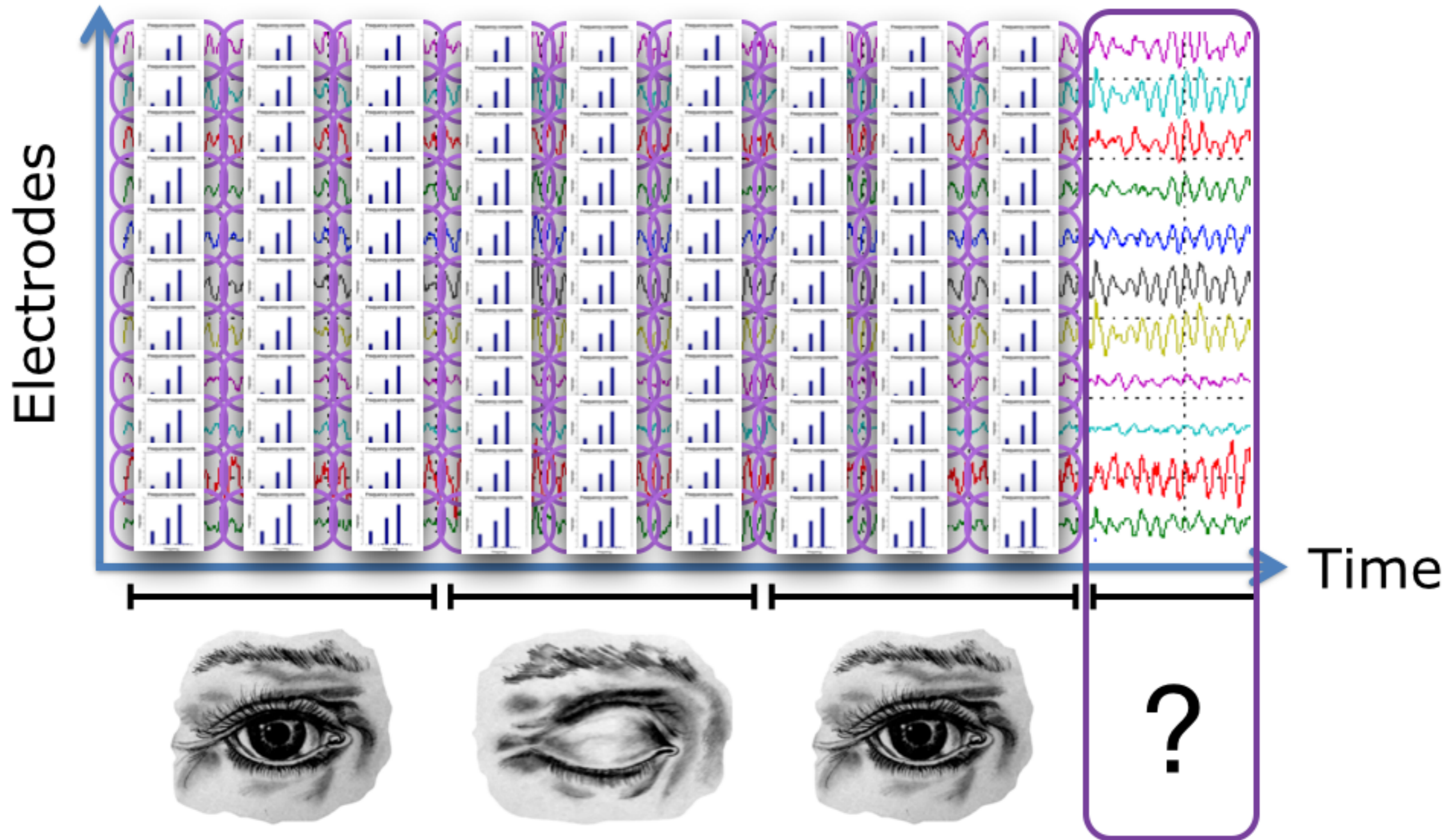
TIME-FREQUENCY DOMAIN



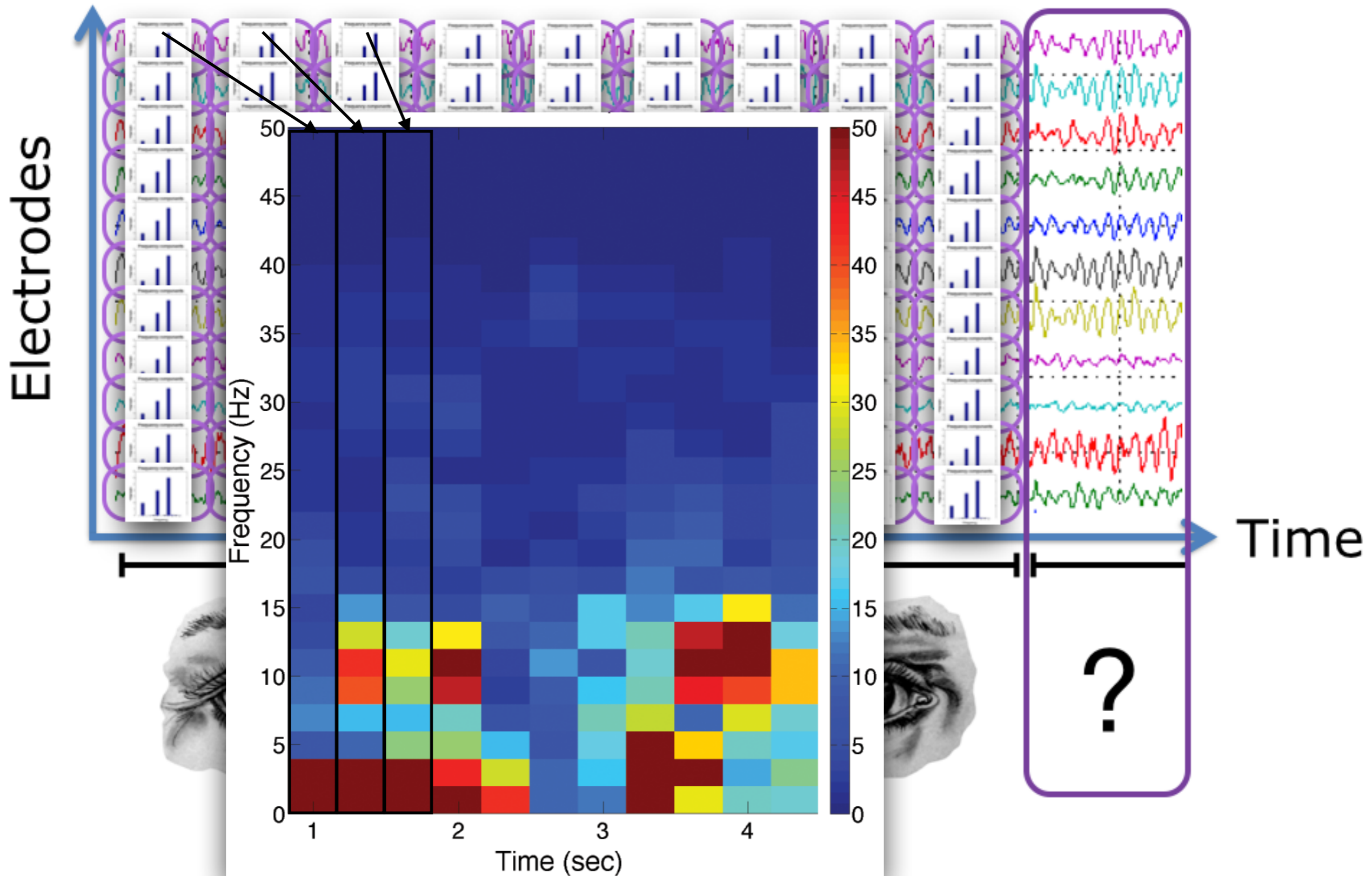
TIME-FREQUENCY DOMAIN



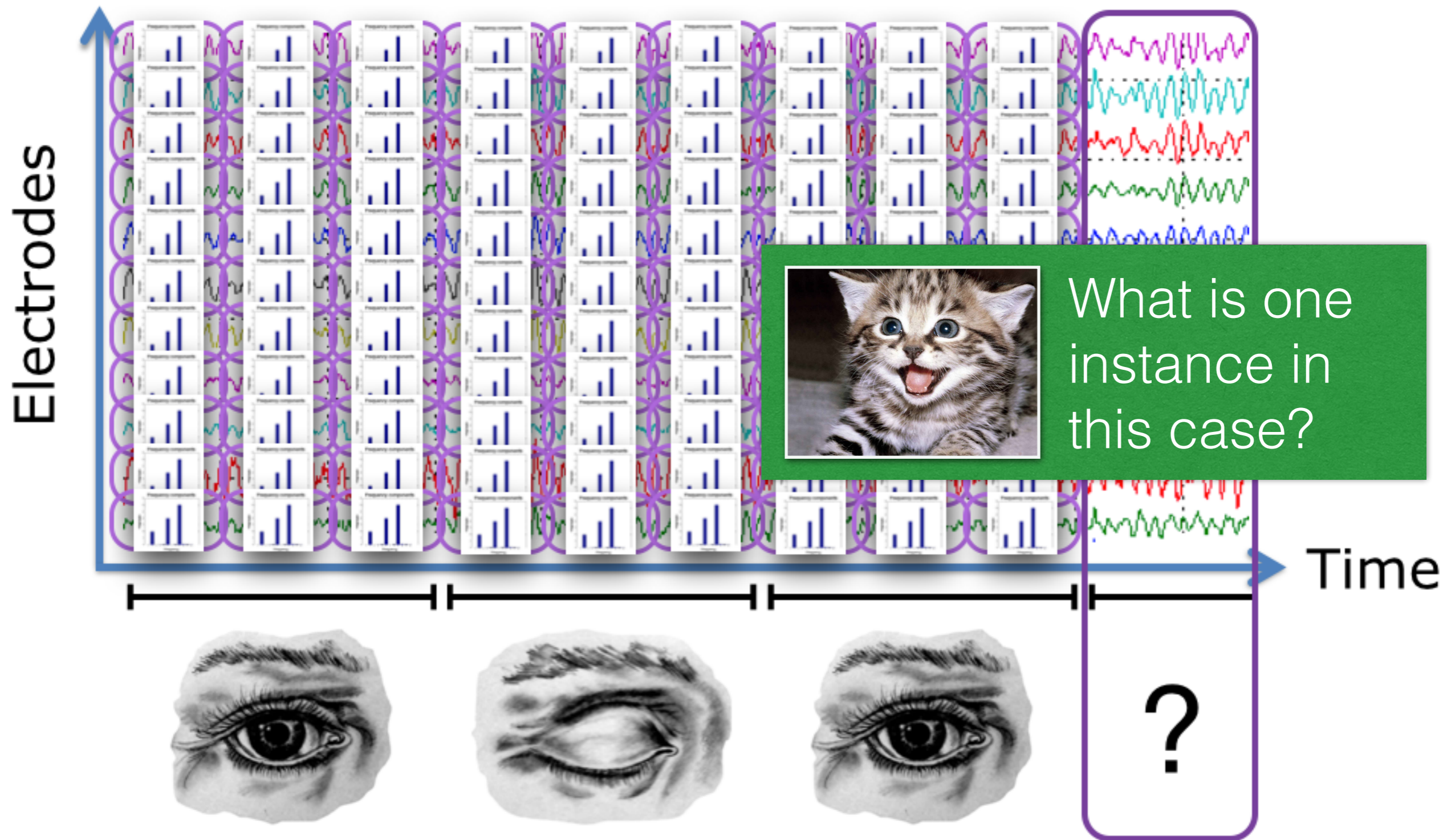
TIME-FREQUENCY DOMAIN



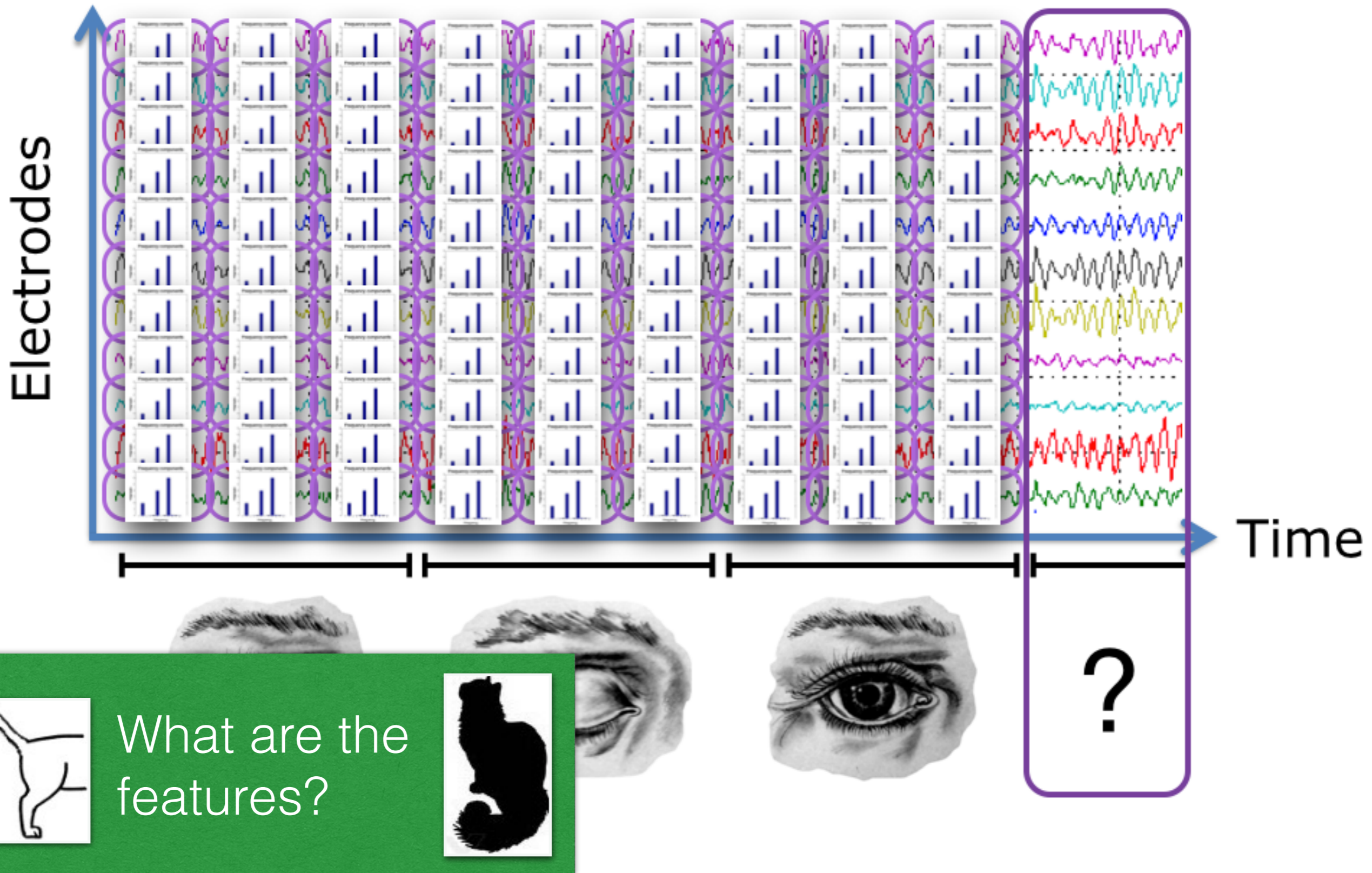
TIME-FREQUENCY DOMAIN



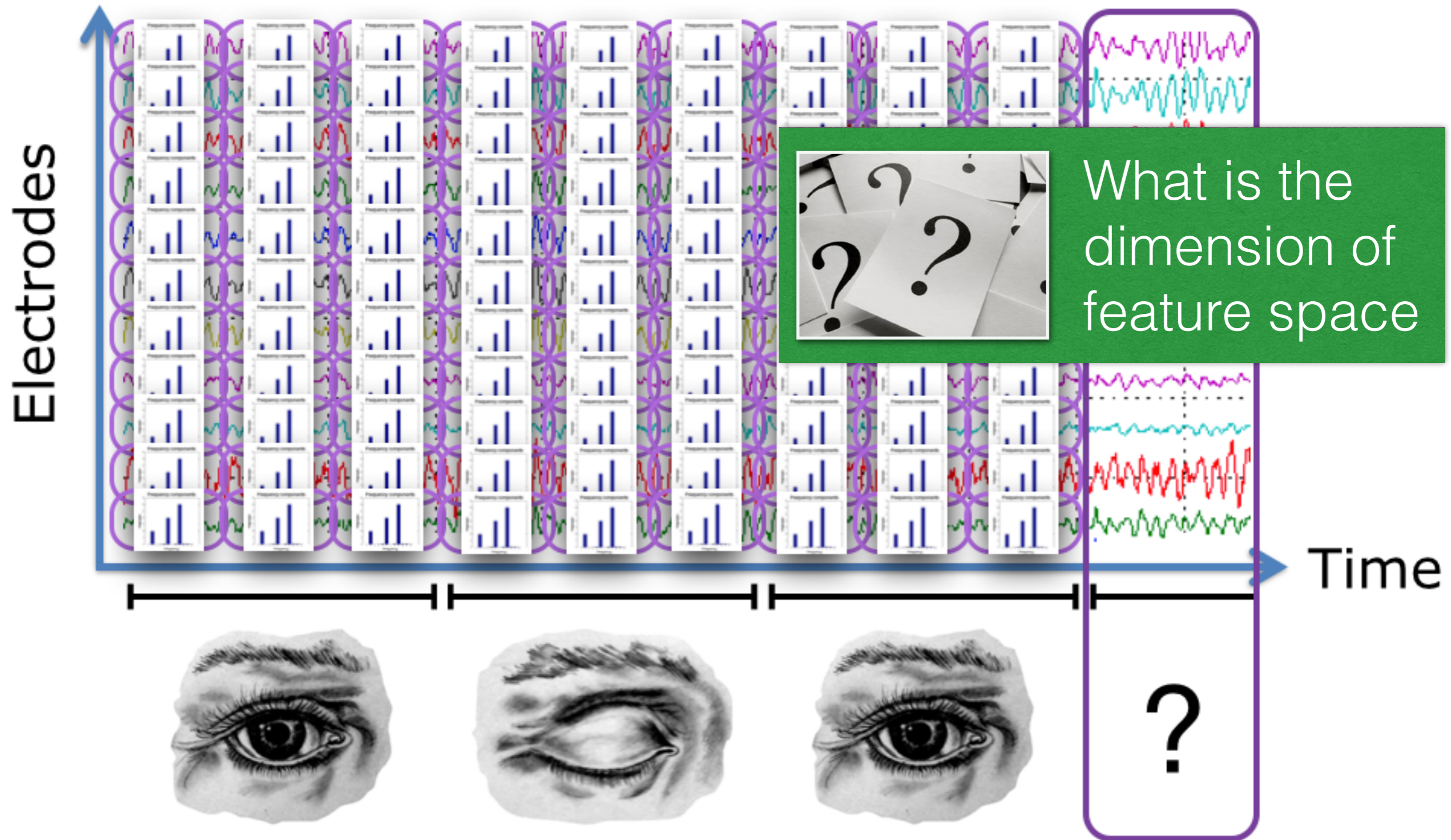
TIME-FREQUENCY DOMAIN



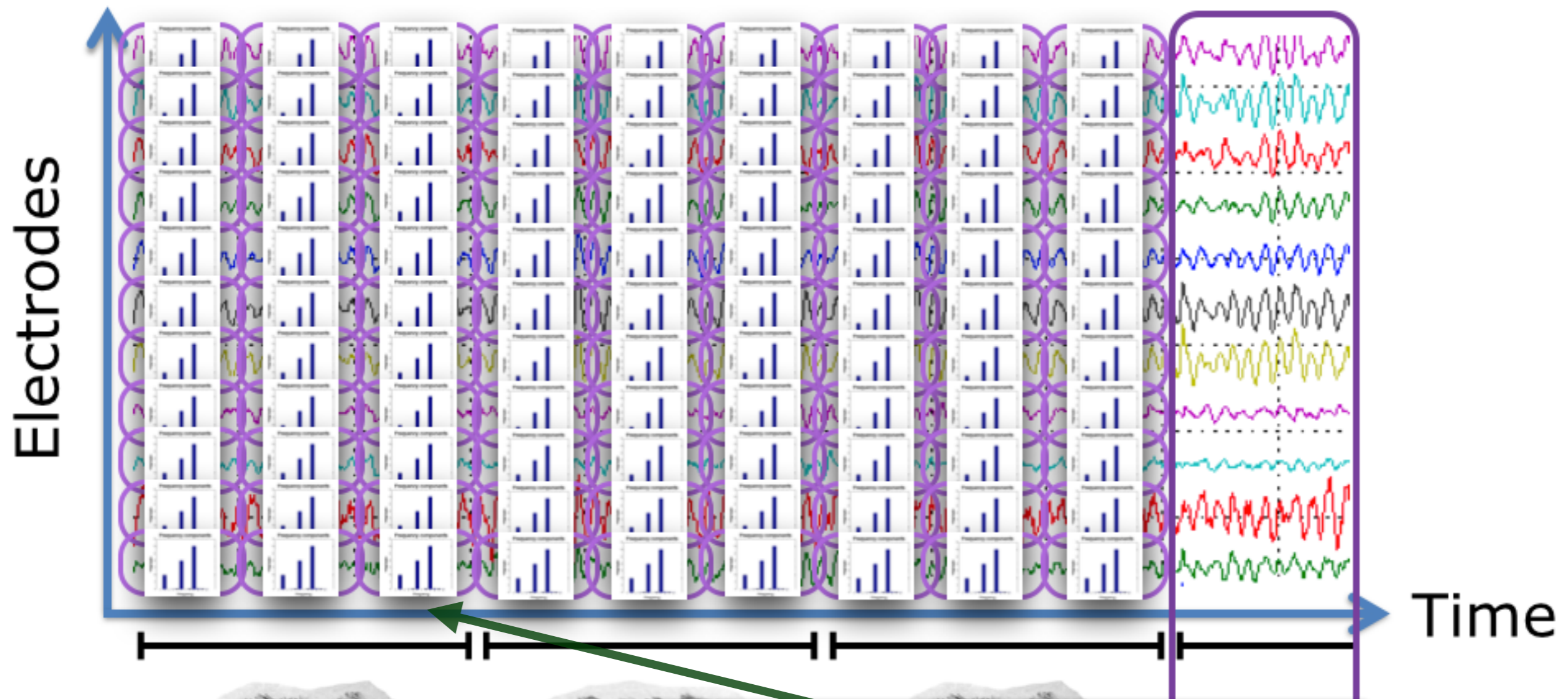
TIME-FREQUENCY DOMAIN



TIME-FREQUENCY DOMAIN

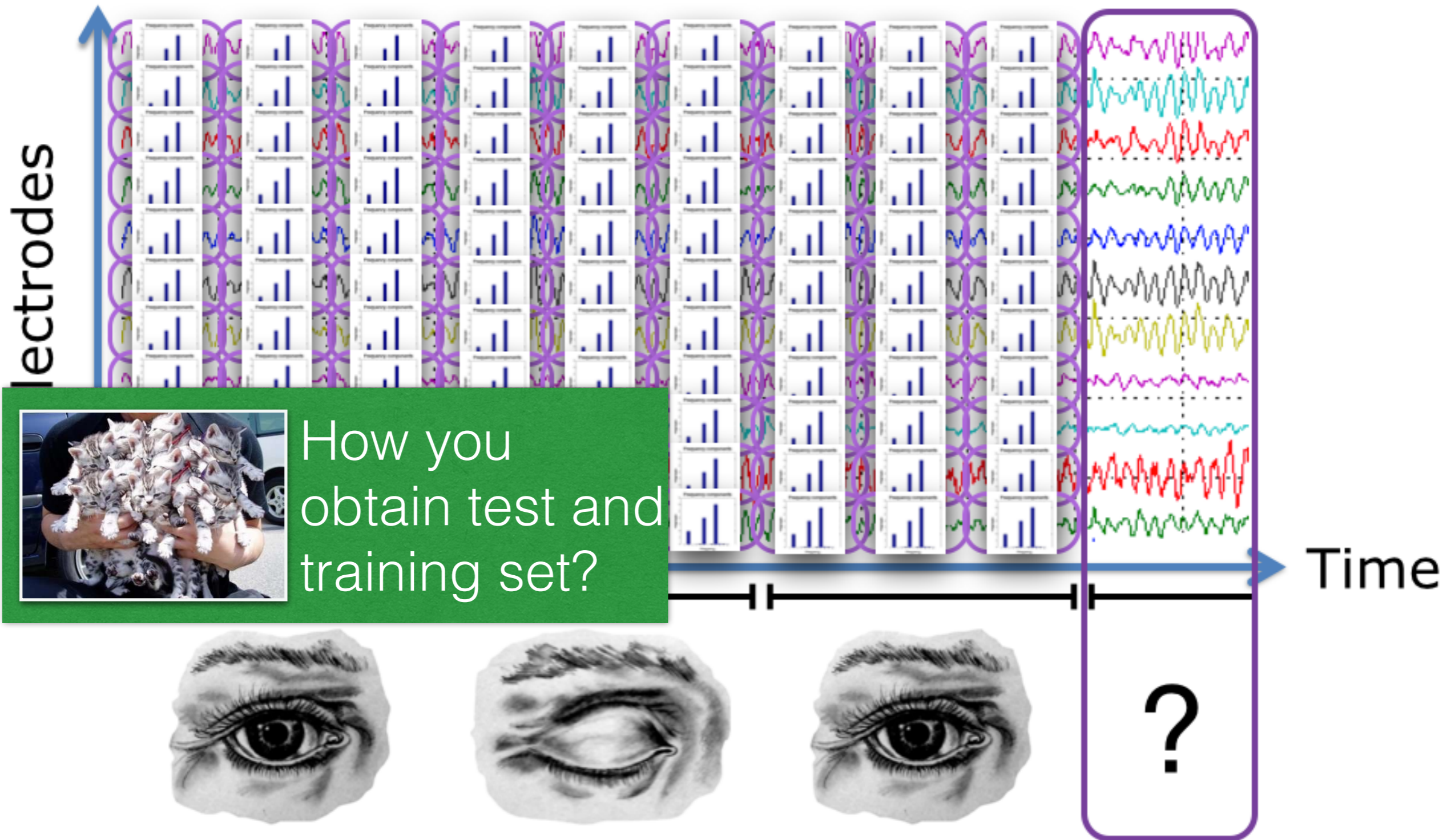


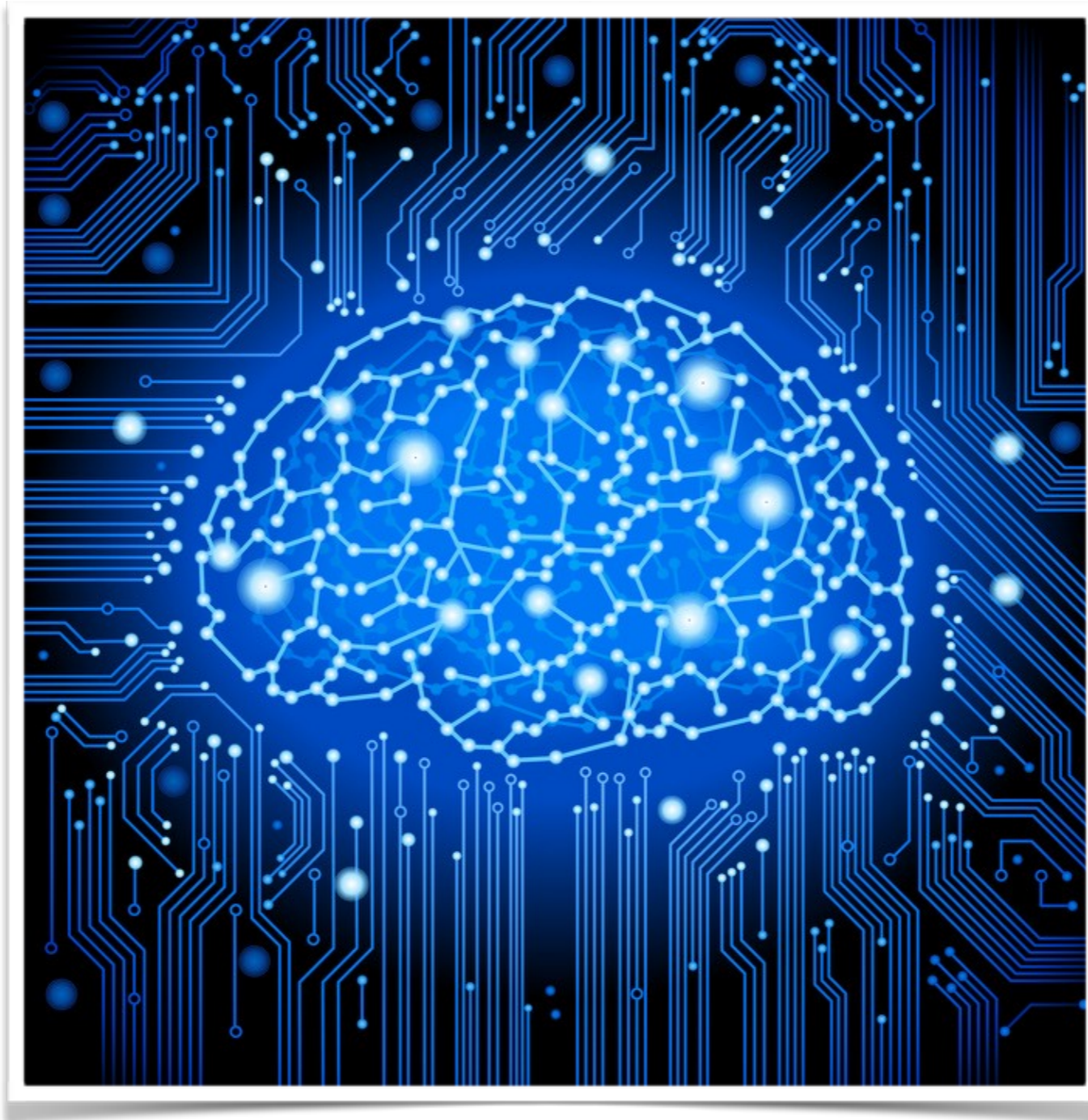
TIME-FREQUENCY DOMAIN



What is the class of this instance?

TIME-FREQUENCY DOMAIN

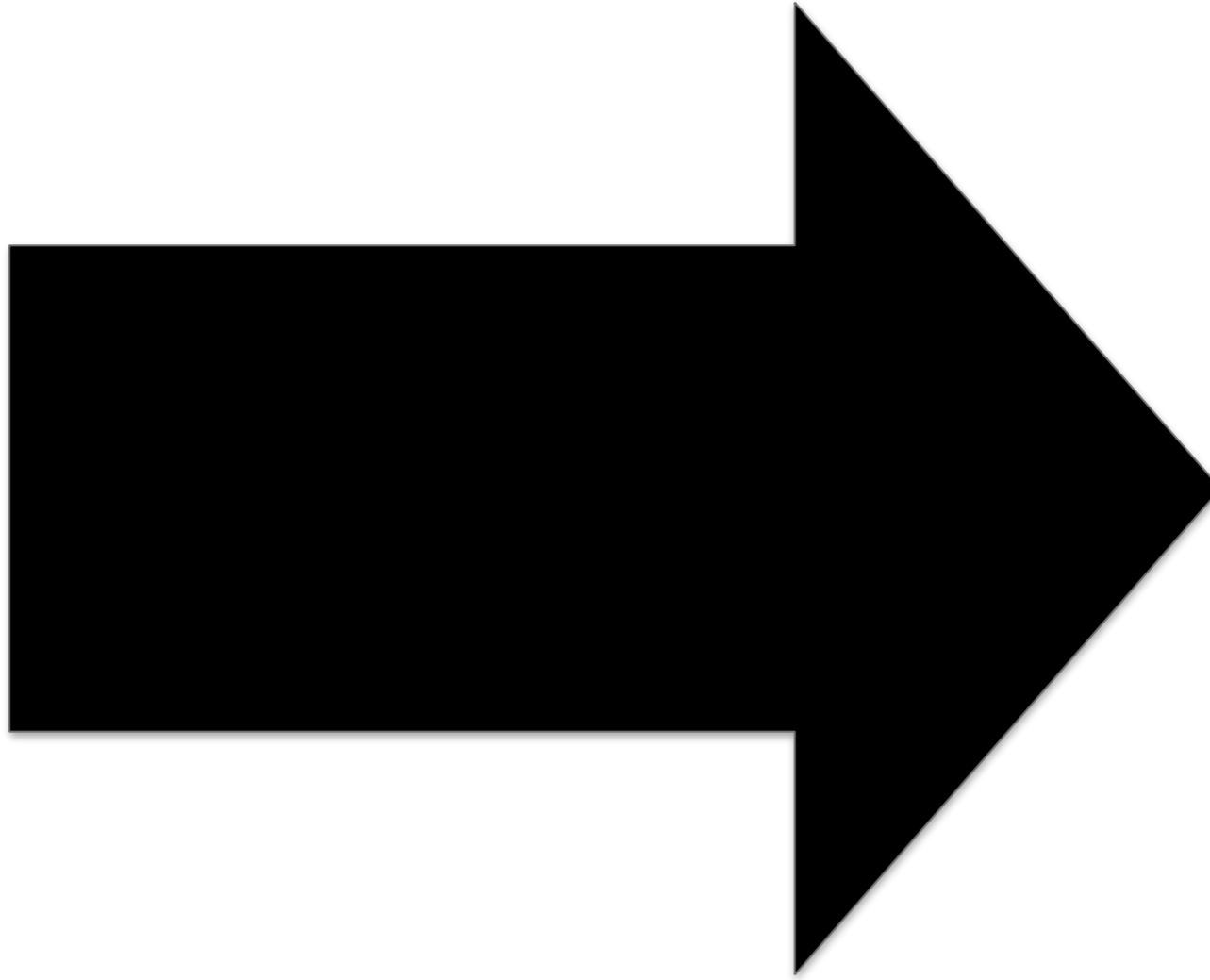


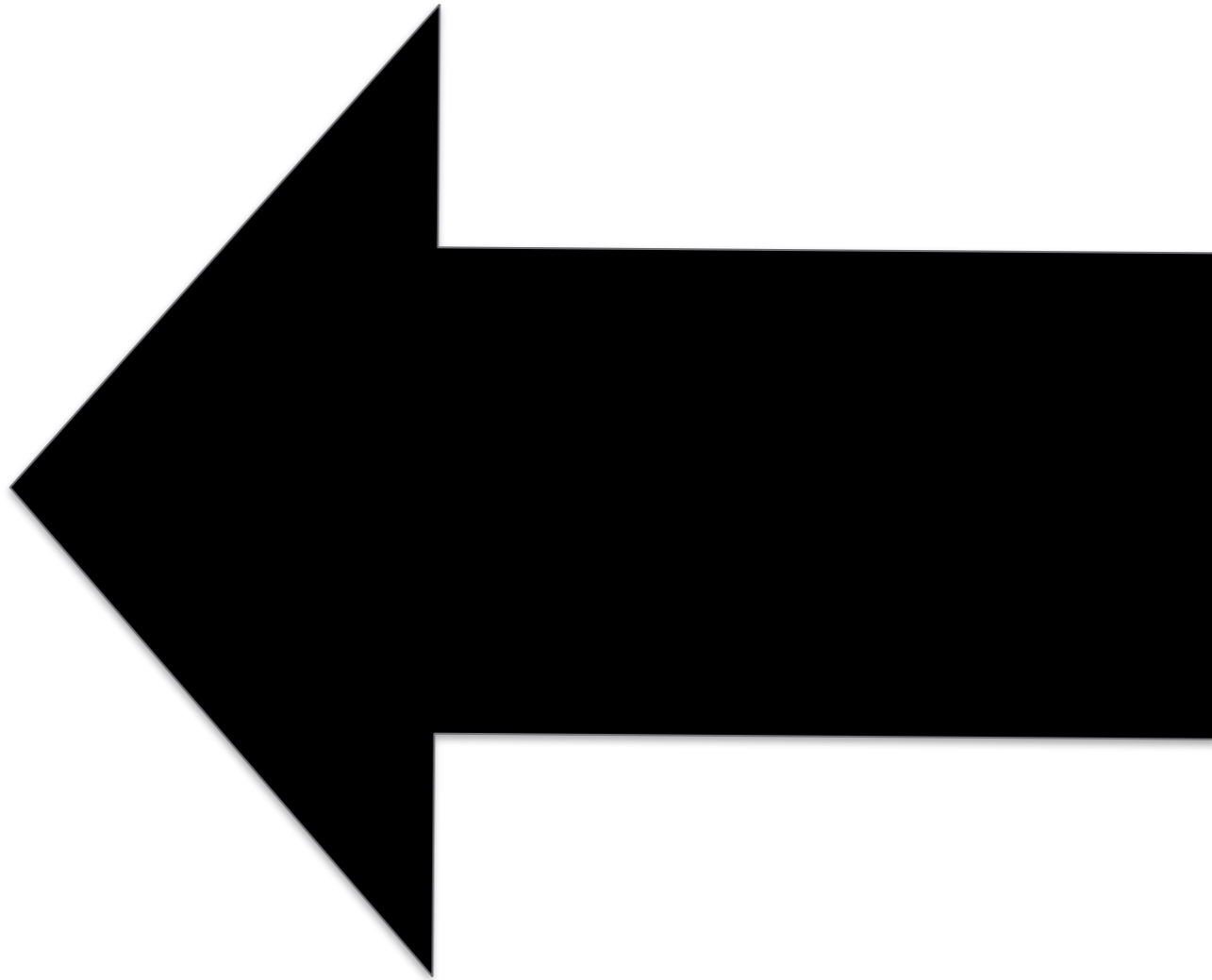


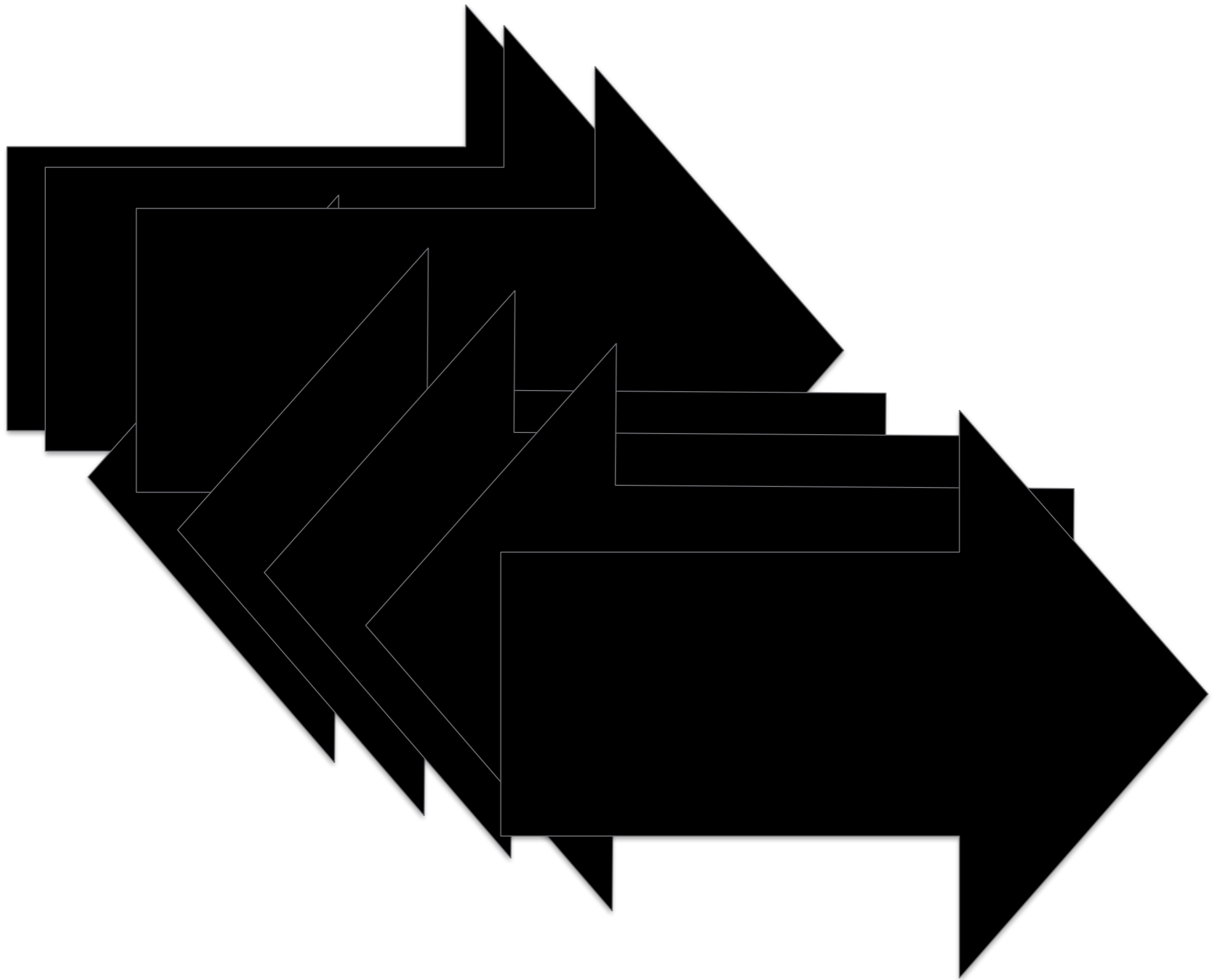
PART III

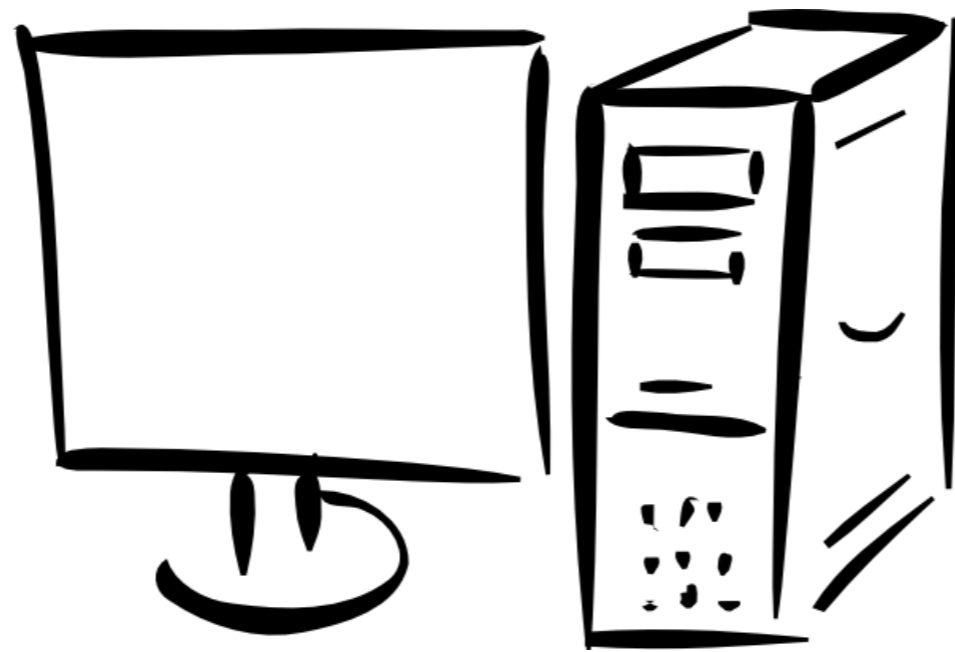
BRAIN-COMPUTER INTERFACE



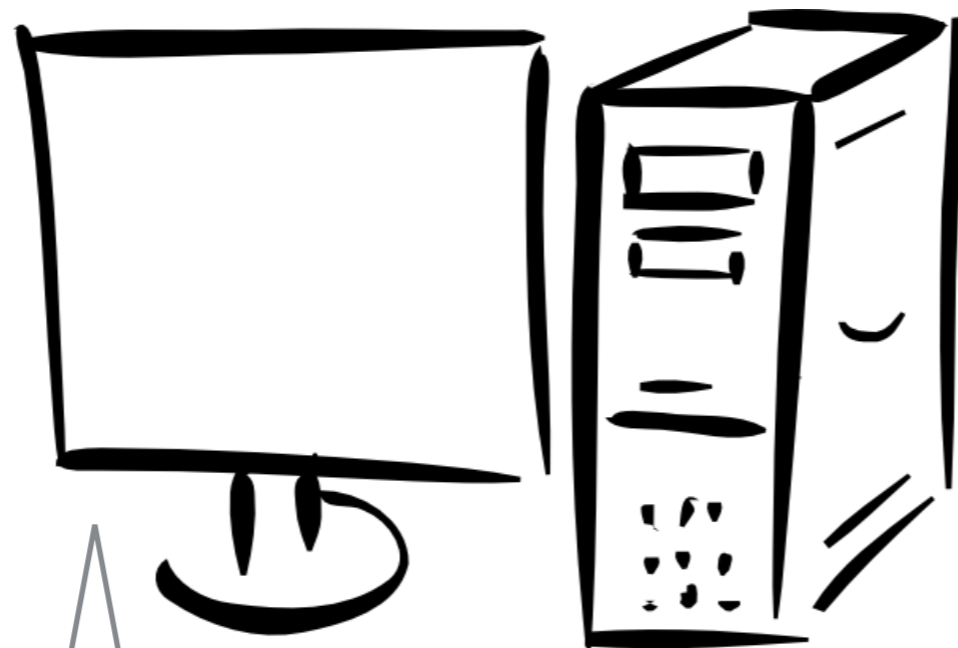






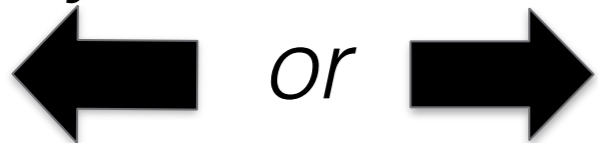


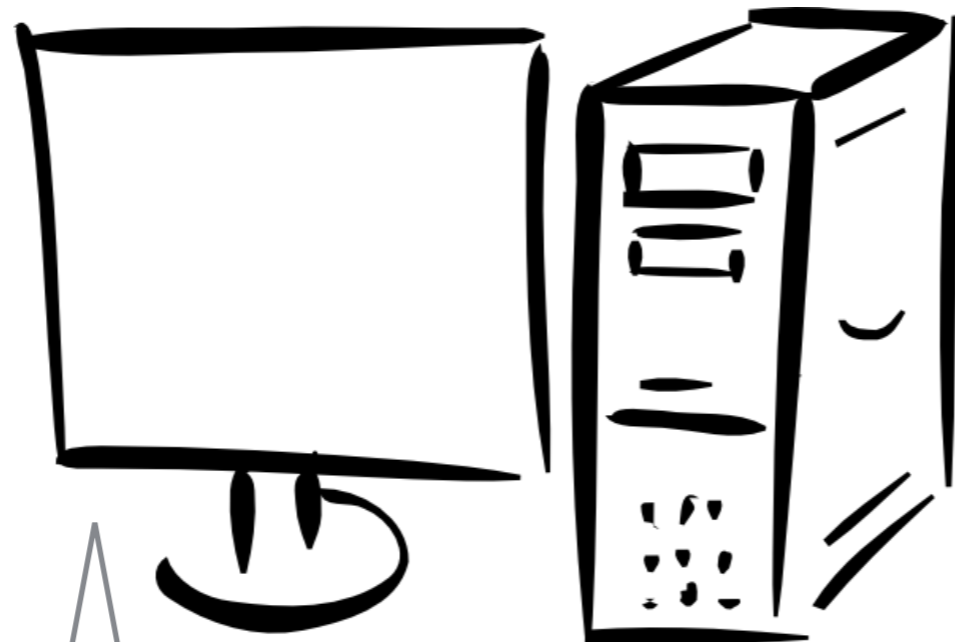
*Now I know how
your brain
signal looks like
when you think
“LEFT” and “RIGHT”*



*Now I know how
your brain
signal looks like
when you think
“LEFT” and “RIGHT”*

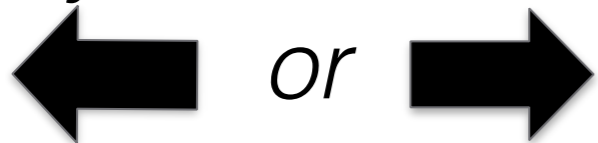
Try me — think



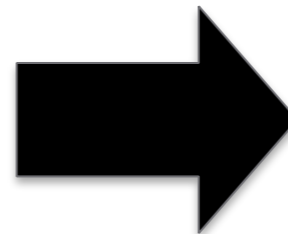


*Now I know how
your brain
signal looks like
when you think
“LEFT” and “RIGHT”*

Try me — think



It was



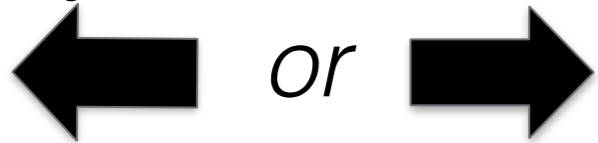
wasn't it?



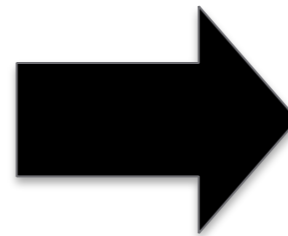
How would you use such technology?

Now I know how your brain signal looks like when you think "LEFT" and "RIGHT"

Try me — think

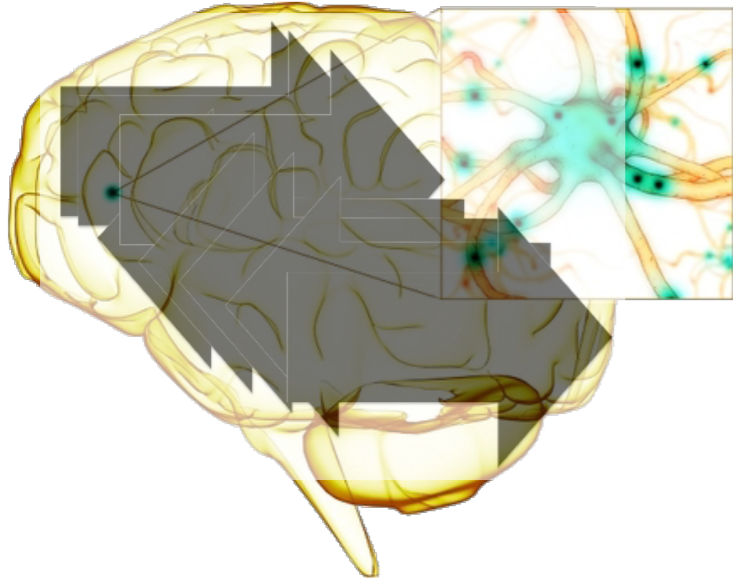


It was

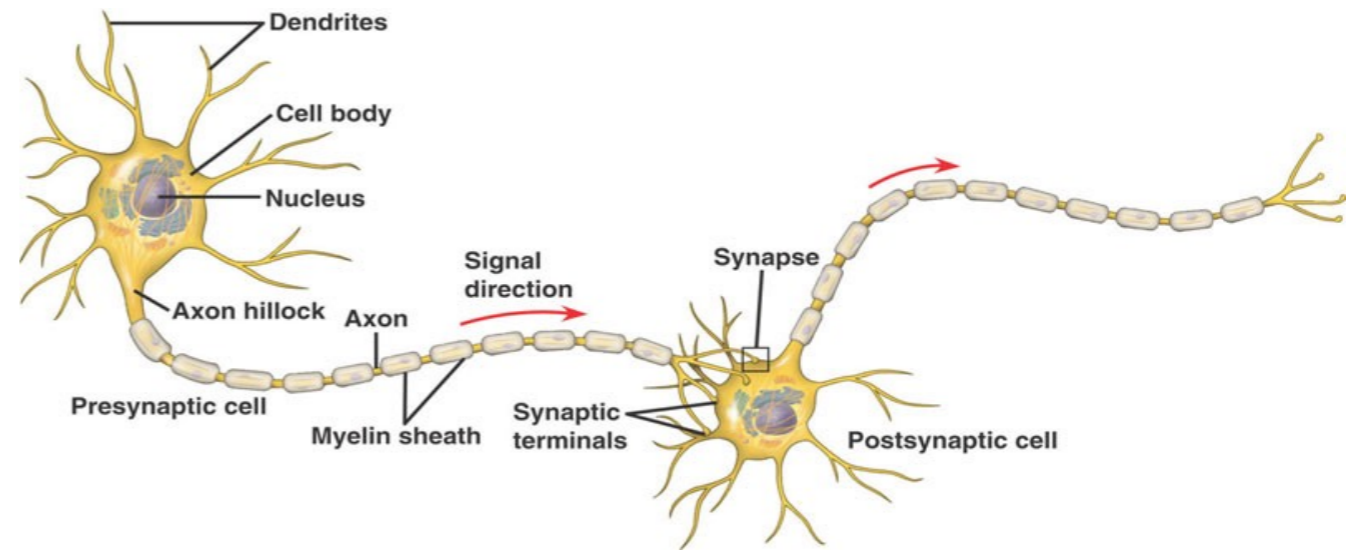
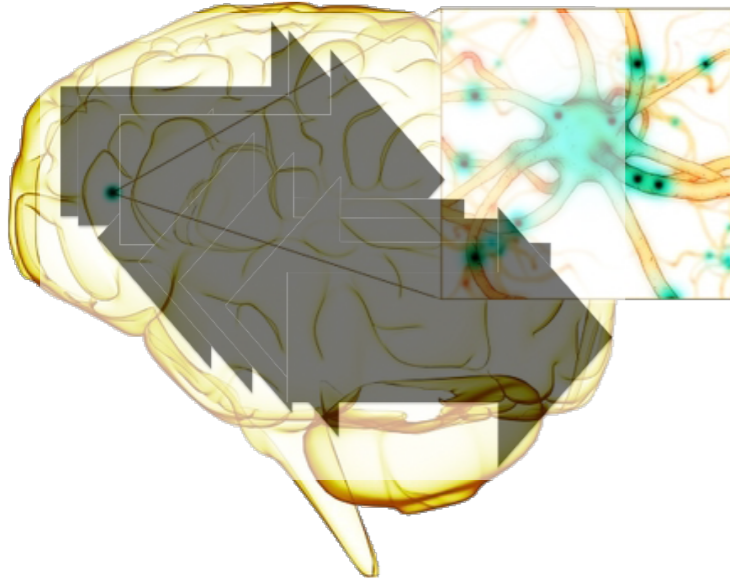


wasn't it?

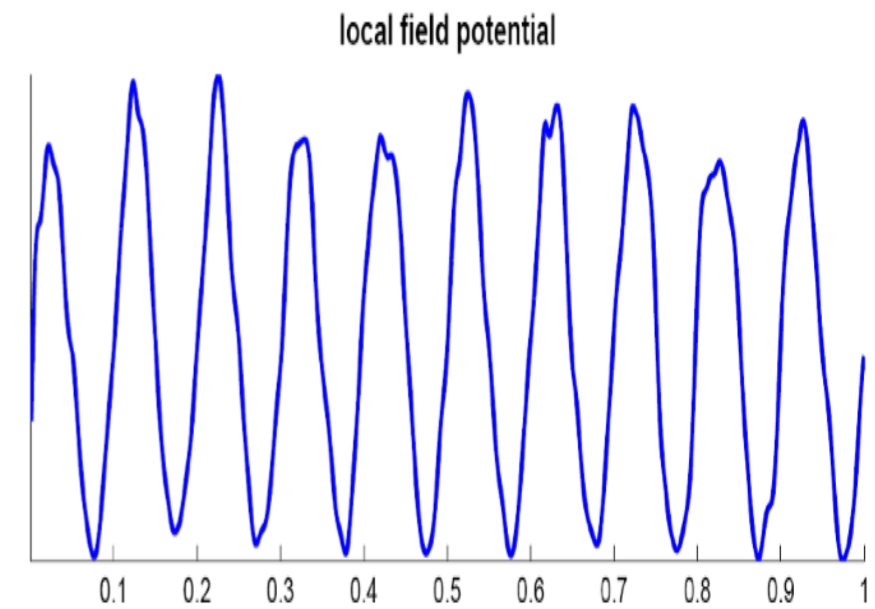
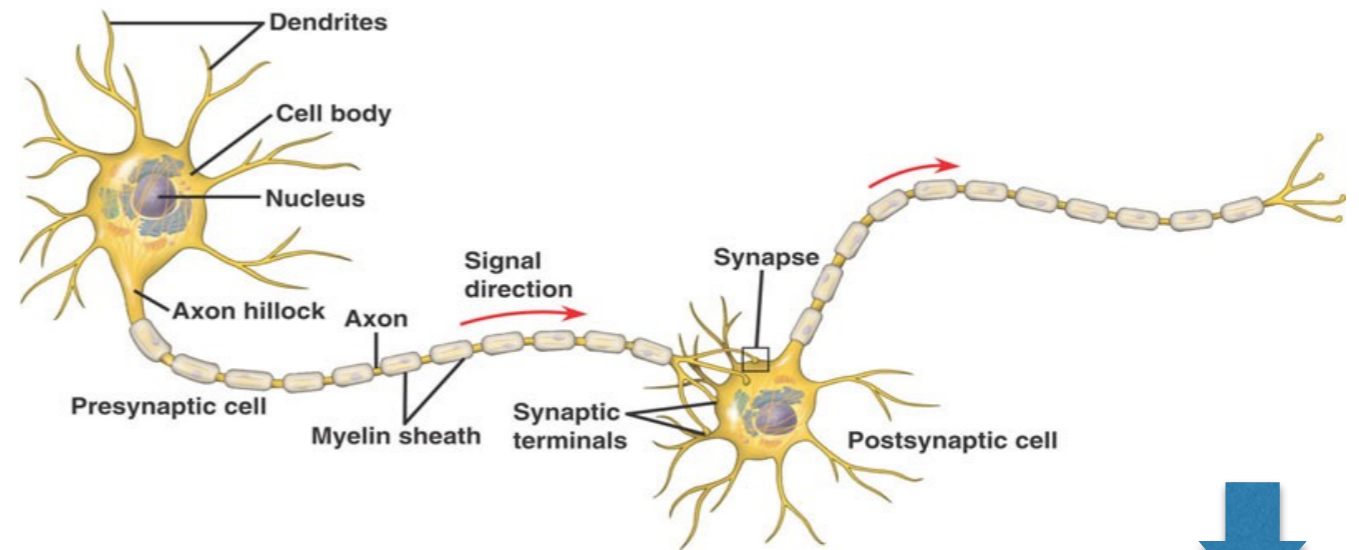
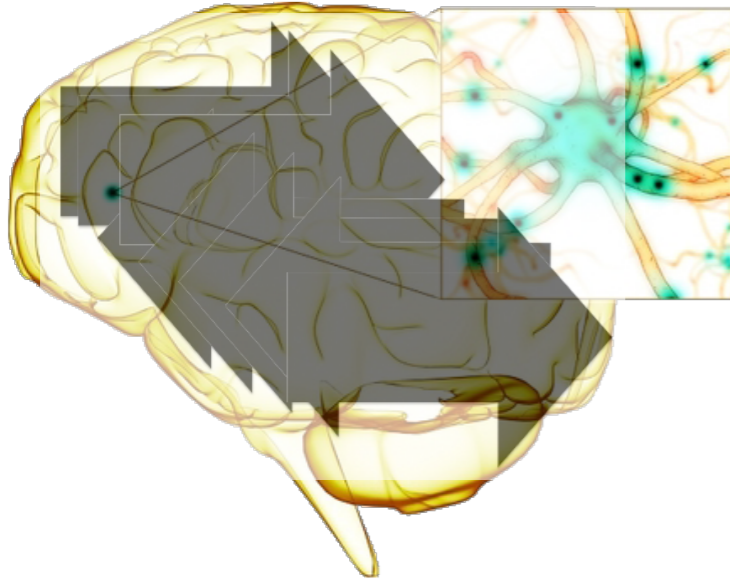
BRAIN-COMPUTER INTERFACE



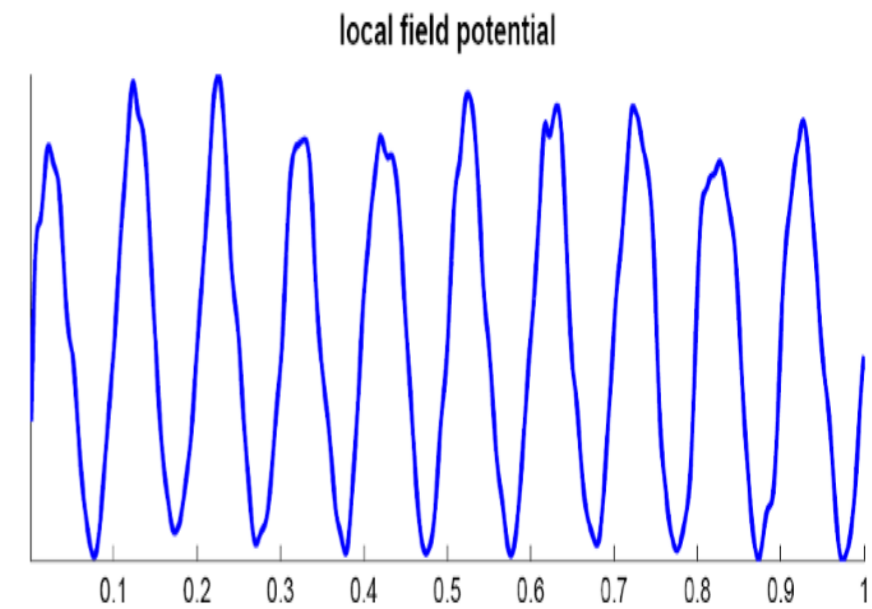
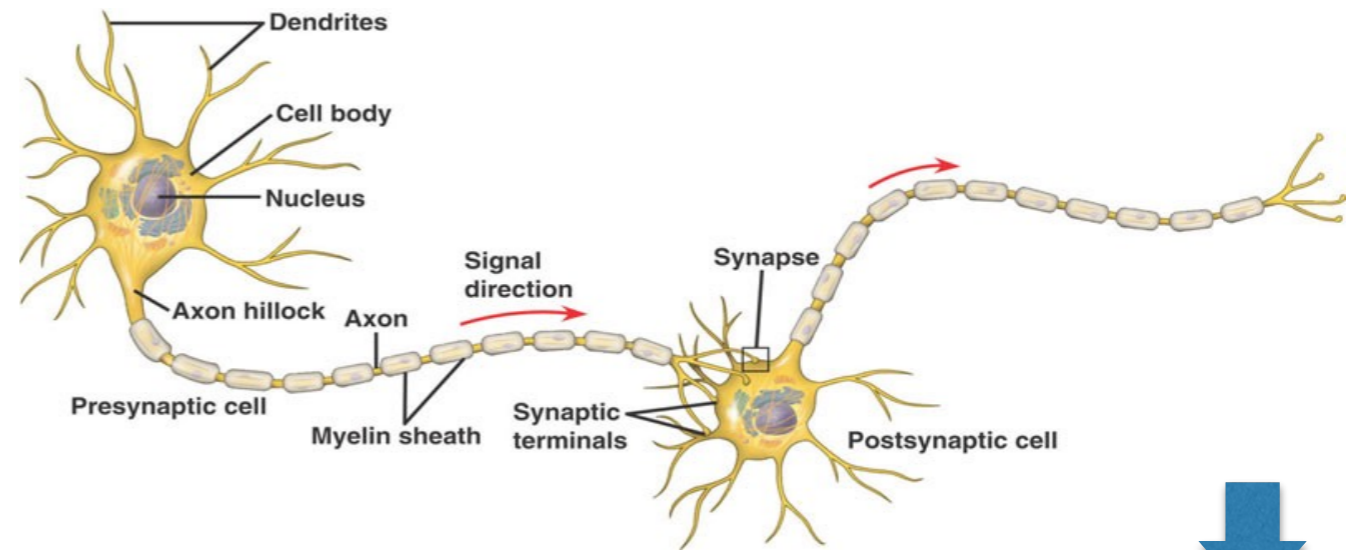
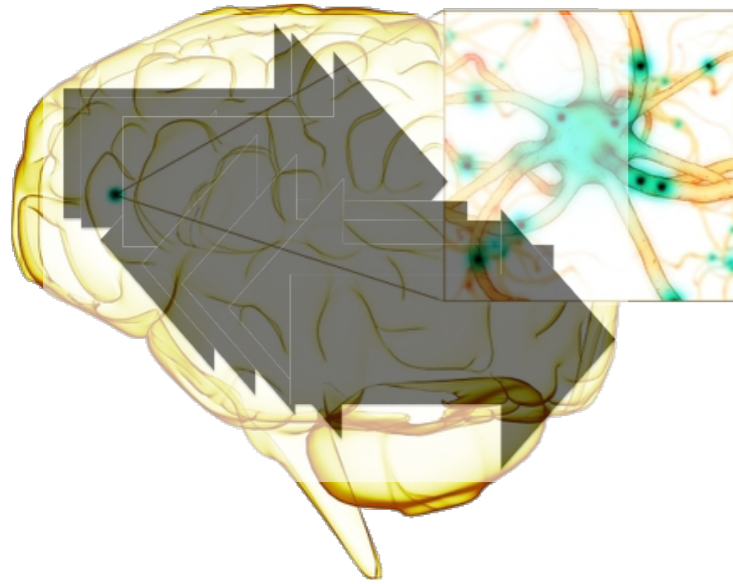
BRAIN-COMPUTER INTERFACE



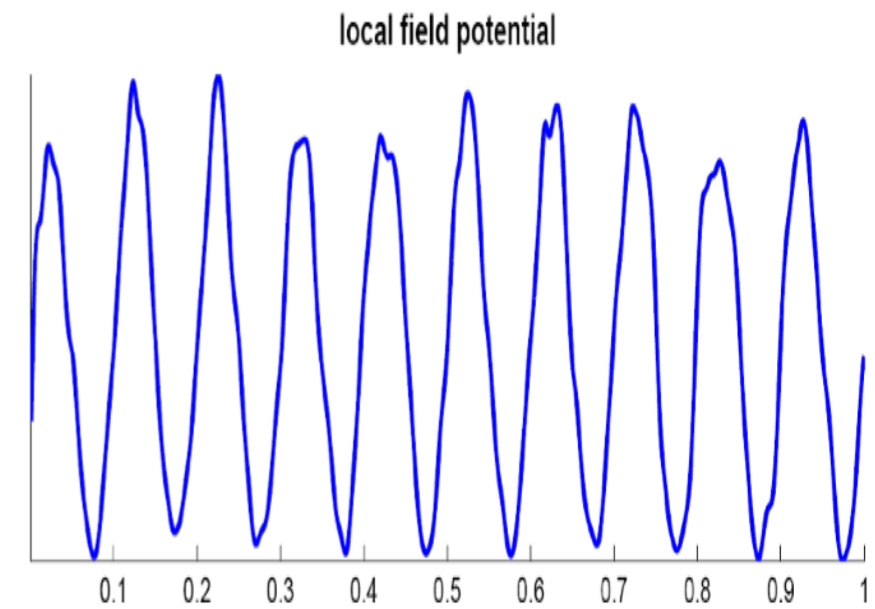
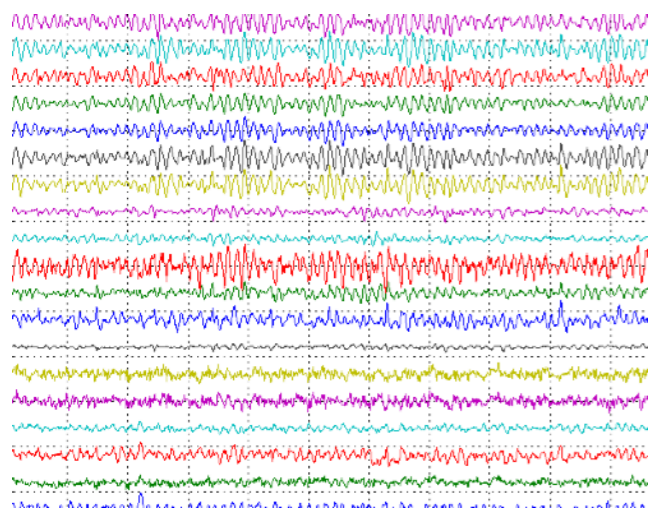
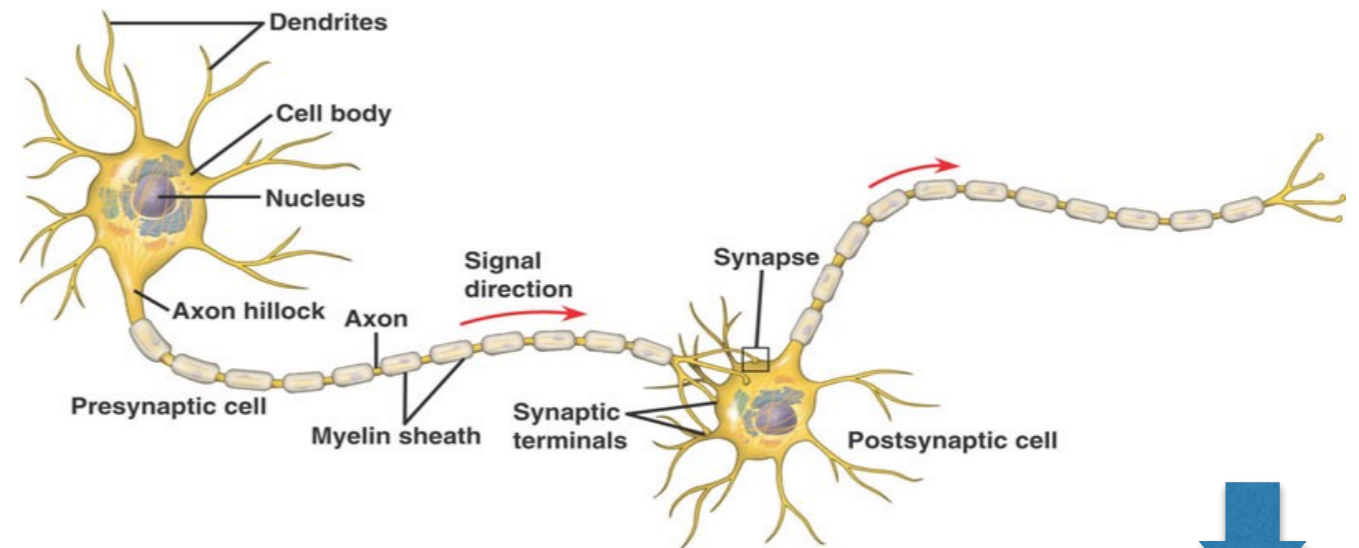
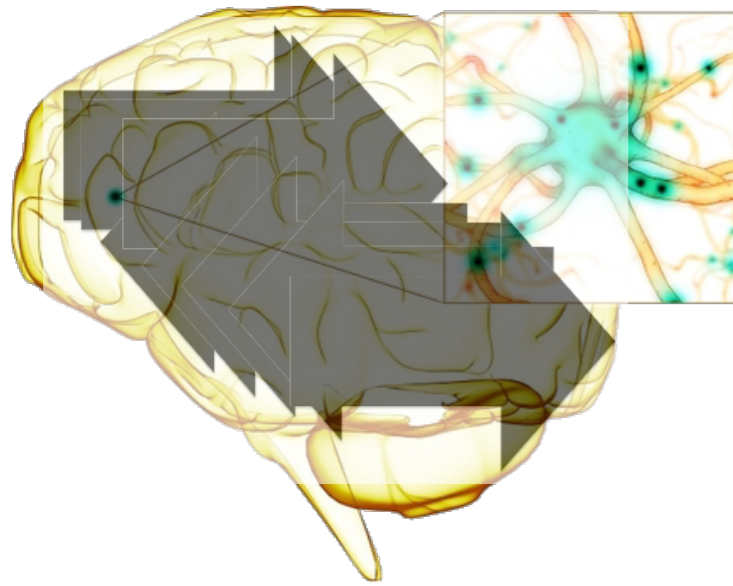
BRAIN-COMPUTER INTERFACE



BRAIN-COMPUTER INTERFACE

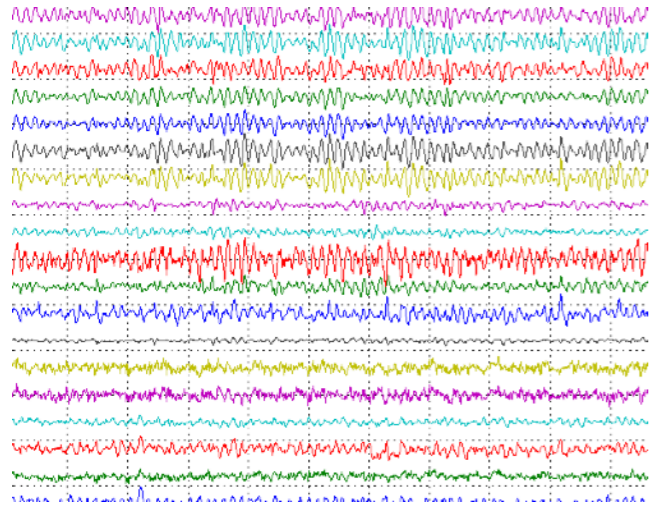


BRAIN-COMPUTER INTERFACE

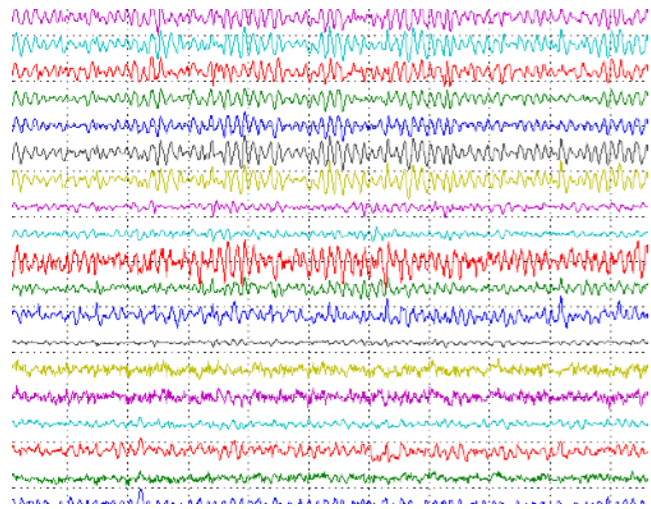




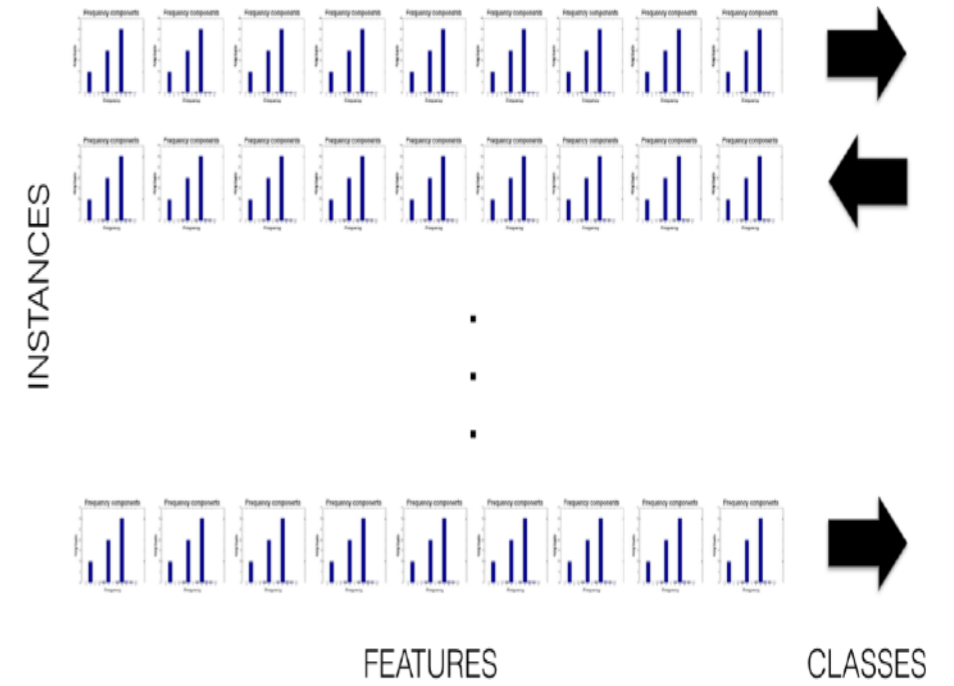
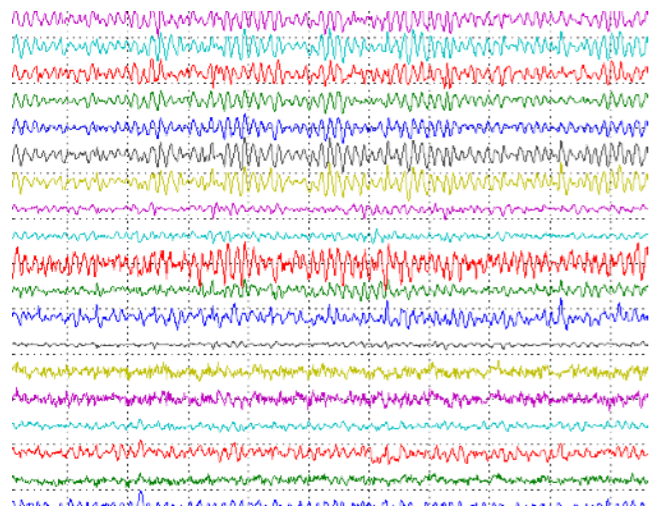
BRAIN-COMPUTER INTERFACE



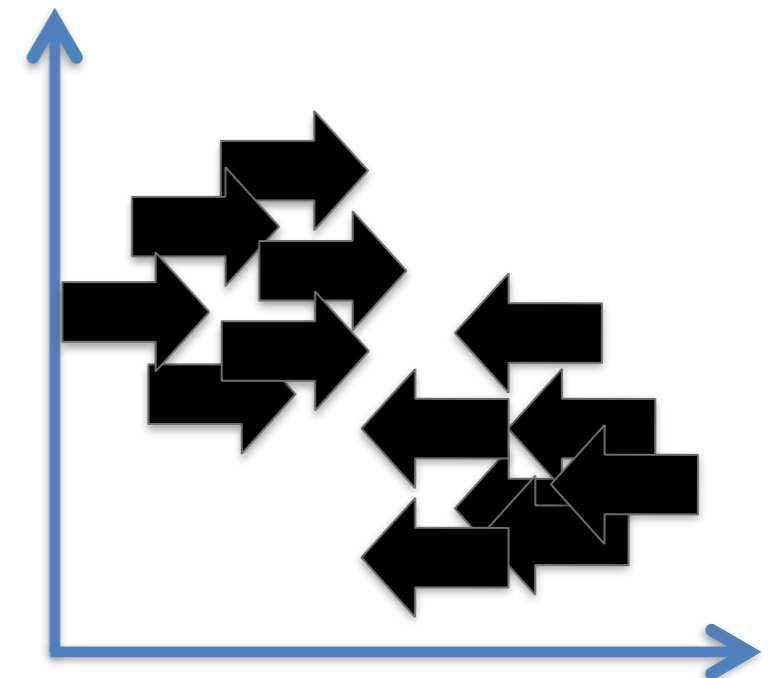
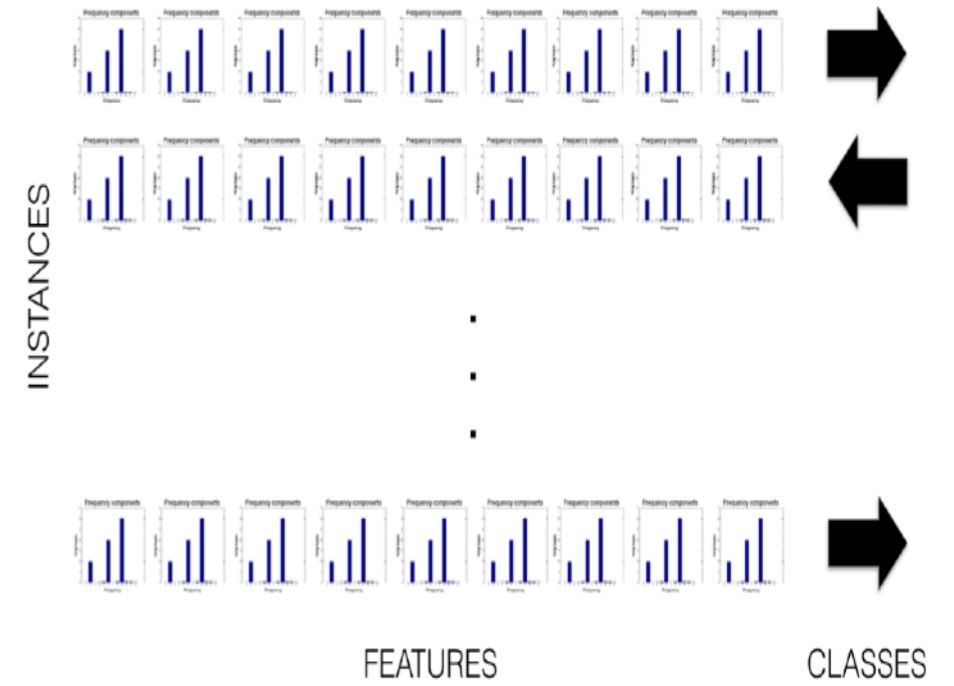
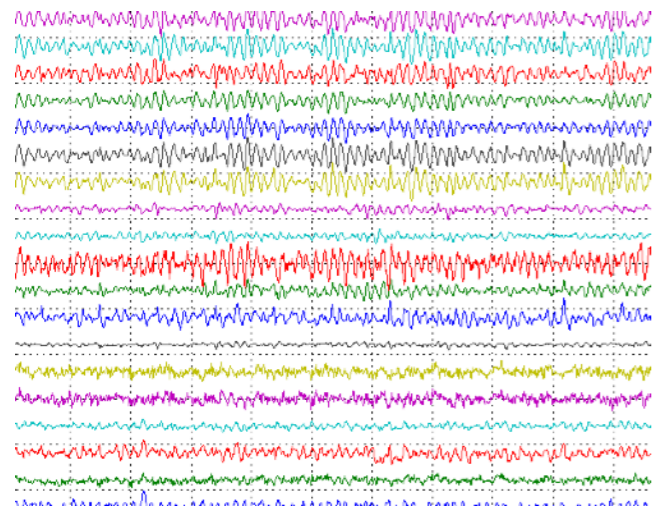
BRAIN-COMPUTER INTERFACE



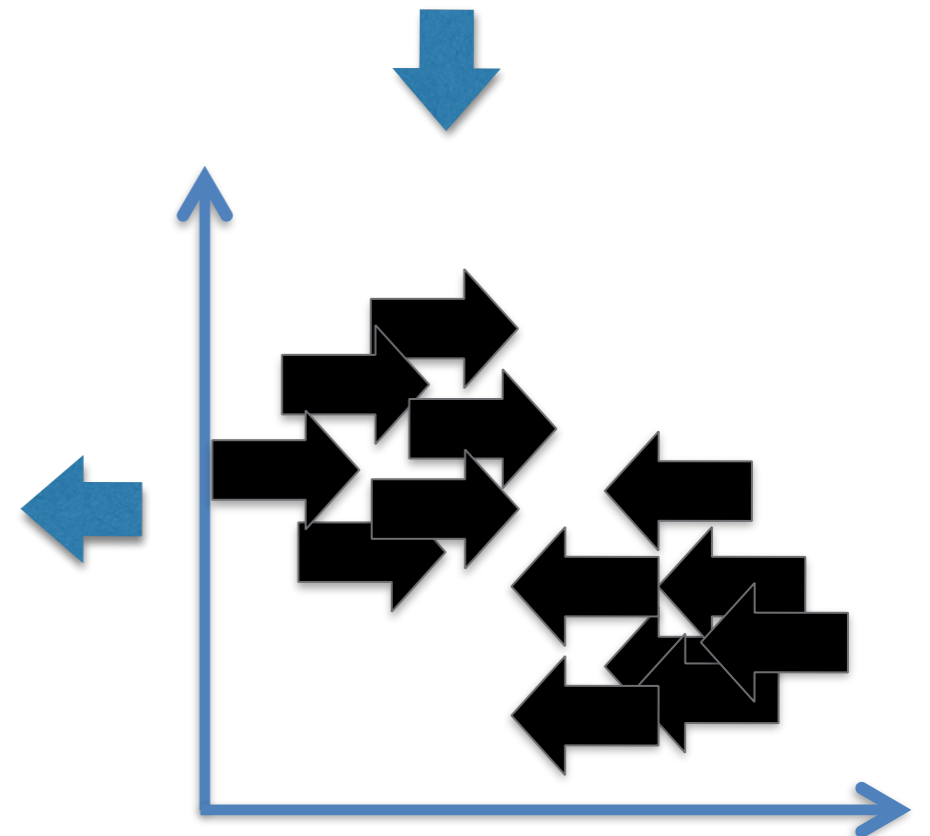
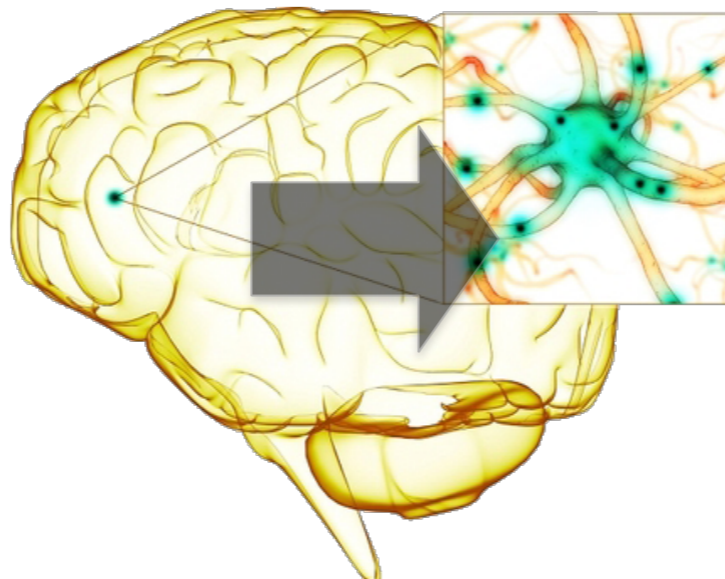
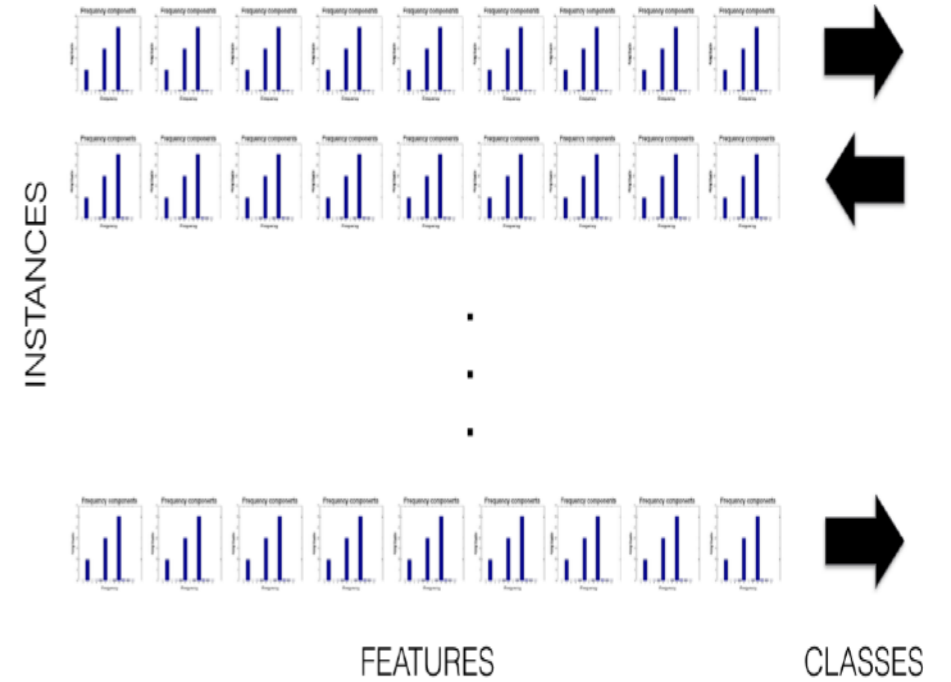
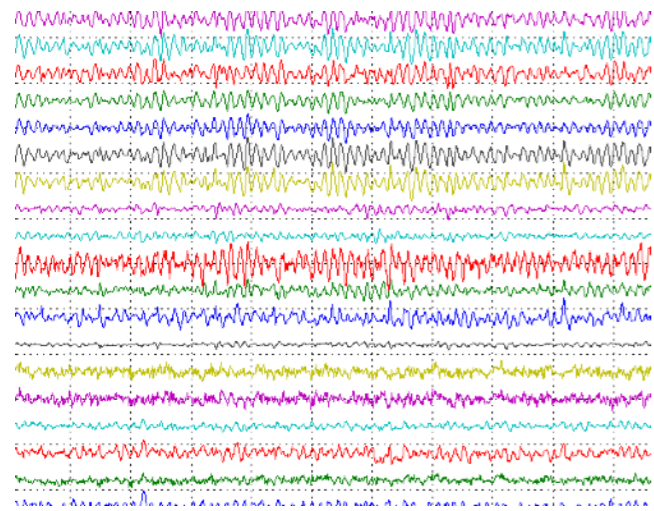
BRAIN-COMPUTER INTERFACE



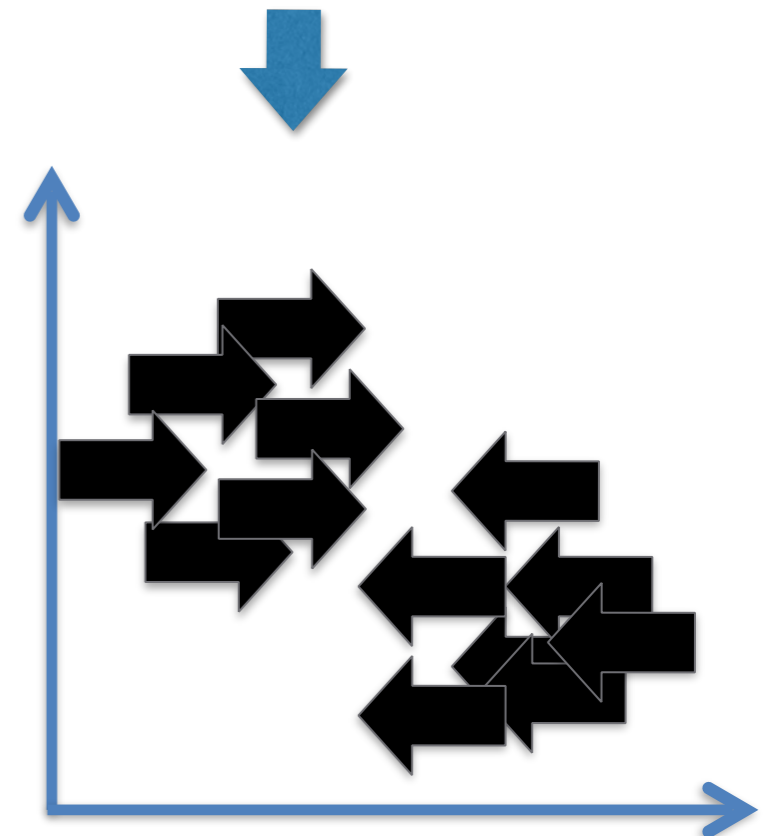
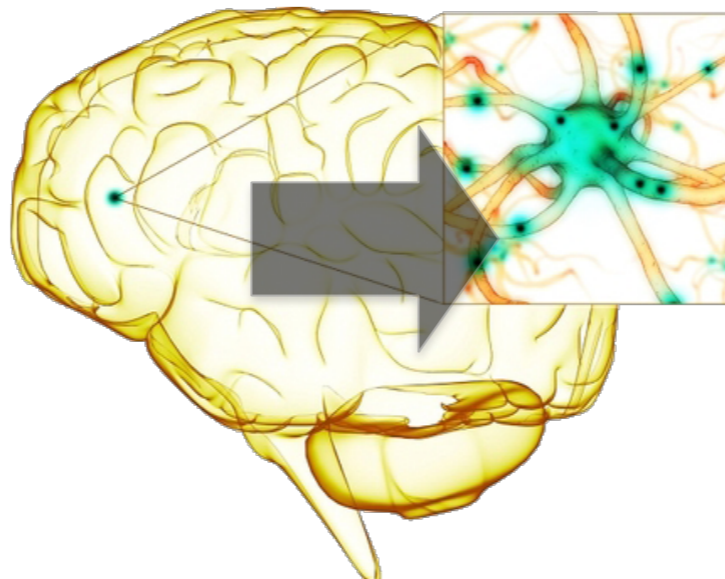
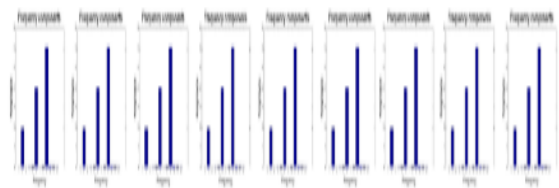
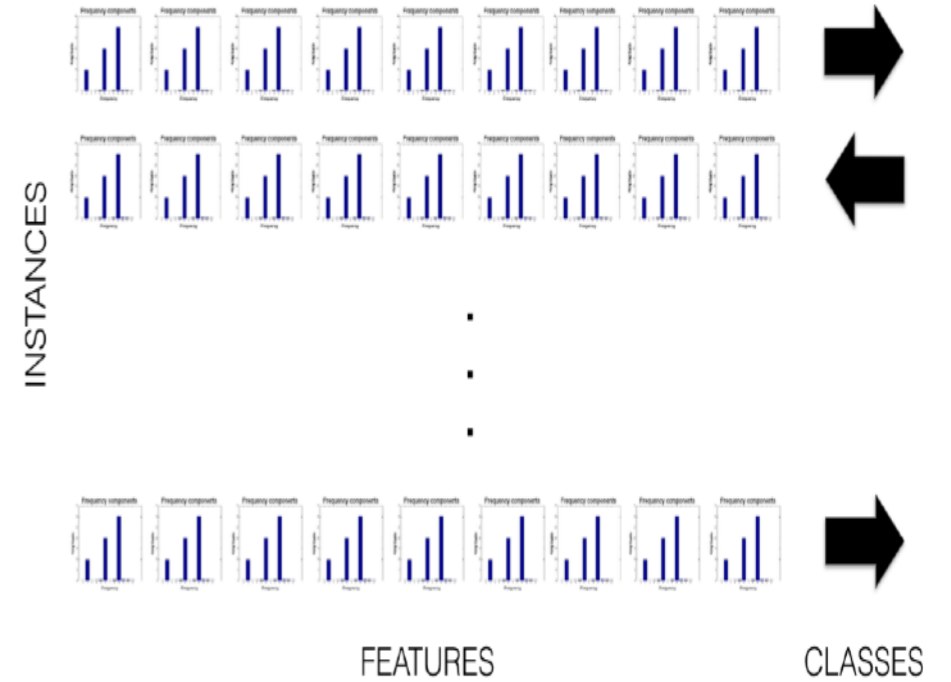
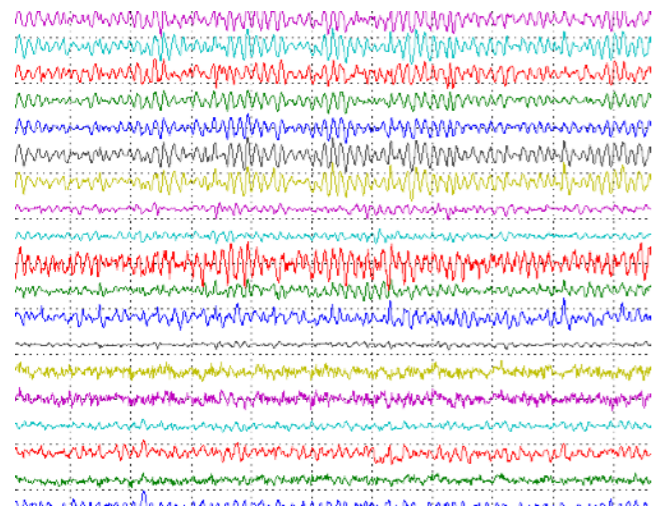
BRAIN-COMPUTER INTERFACE



BRAIN-COMPUTER INTERFACE

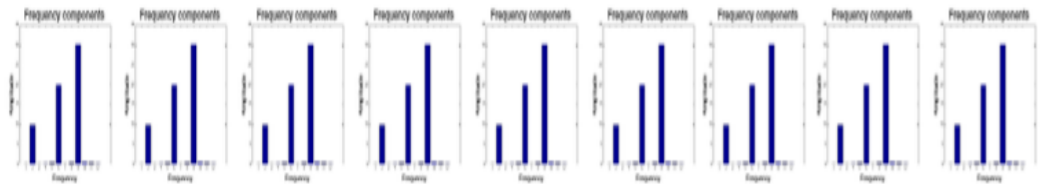


BRAIN-COMPUTER INTERFACE

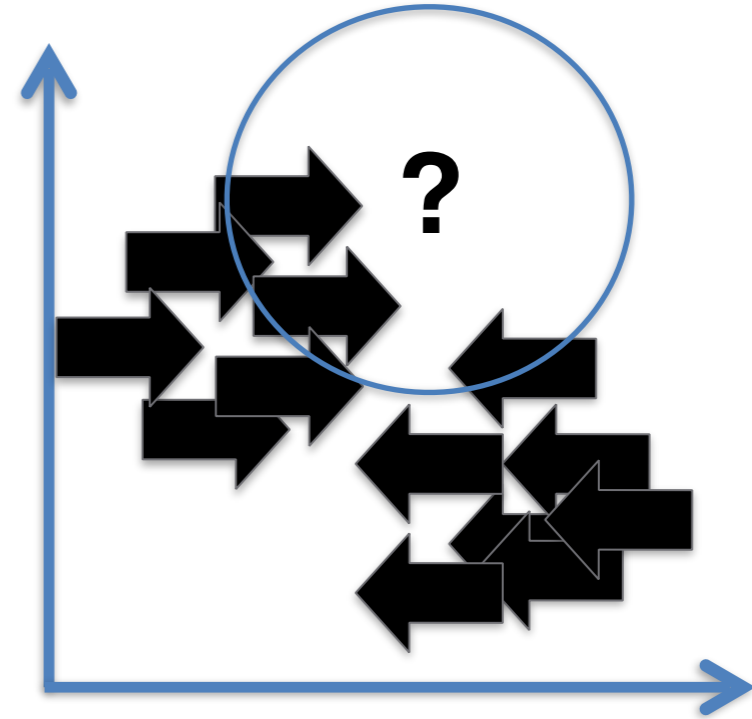
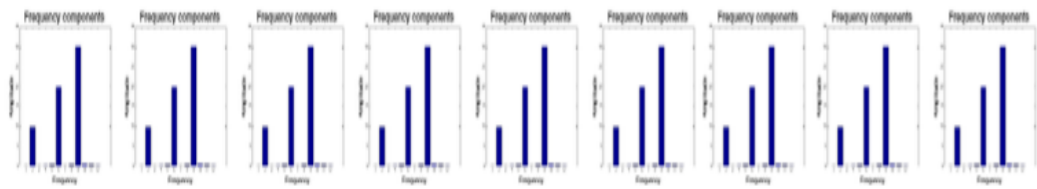




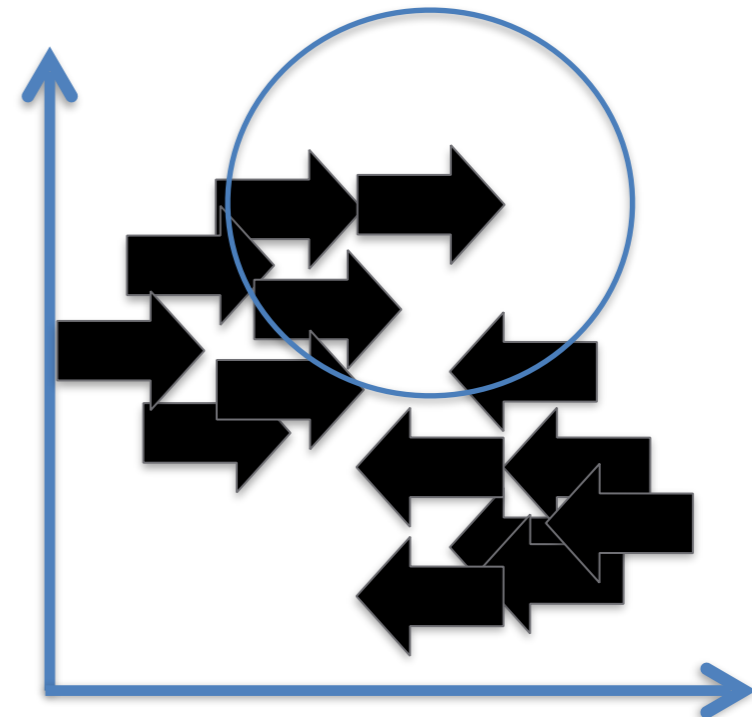
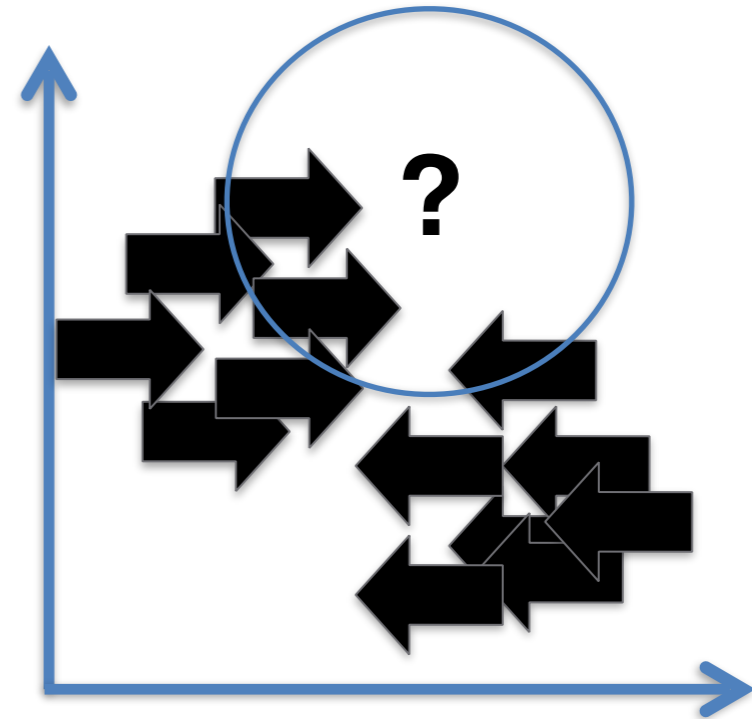
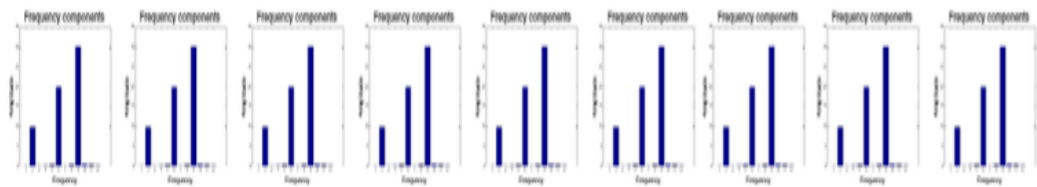
BRAIN-COMPUTER INTERFACE



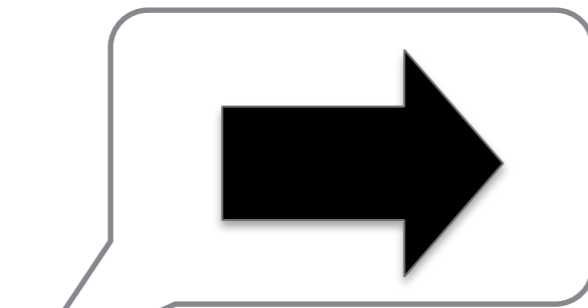
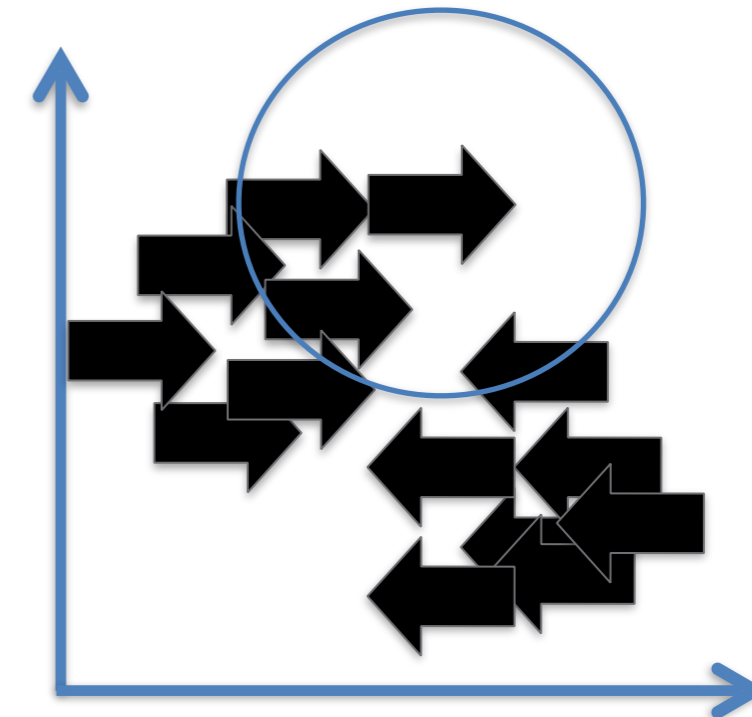
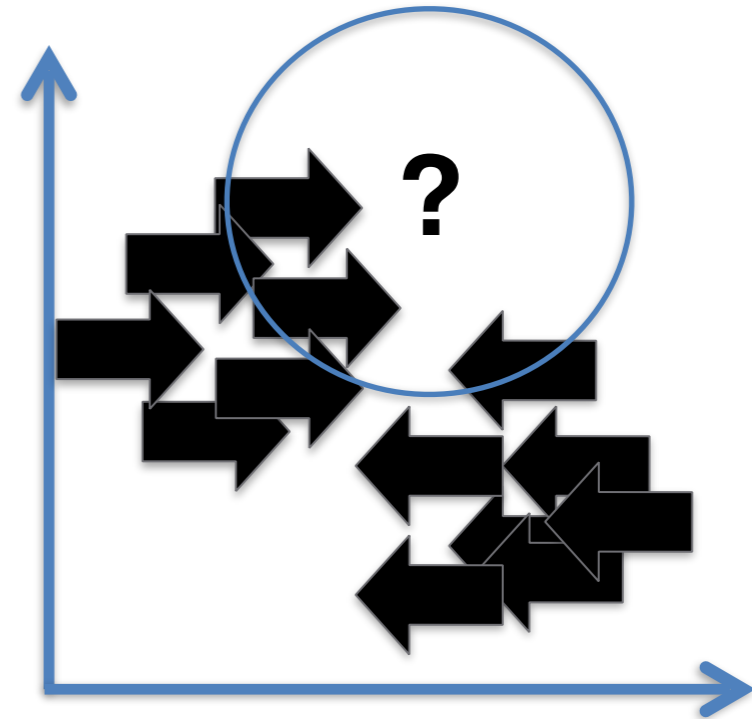
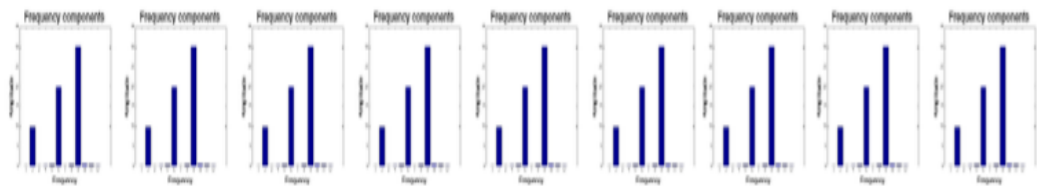
BRAIN-COMPUTER INTERFACE



BRAIN-COMPUTER INTERFACE



BRAIN-COMPUTER INTERFACE



Controlling An Avatar By Thought Using Real-Time fMRI

real-time mental imagery
free navigation in a tropical island
BCI using fMRI

(video speed 5x)

22/12/2011

Cohen O., Mendelsohn A., Malach R., Friedman D.

IDC Herzliya, Weizmann Institute, Bar Ilan
University, Israel.



C-VEP BCI

Brain-Computer Interface based on code-modulated visual evoked potentials



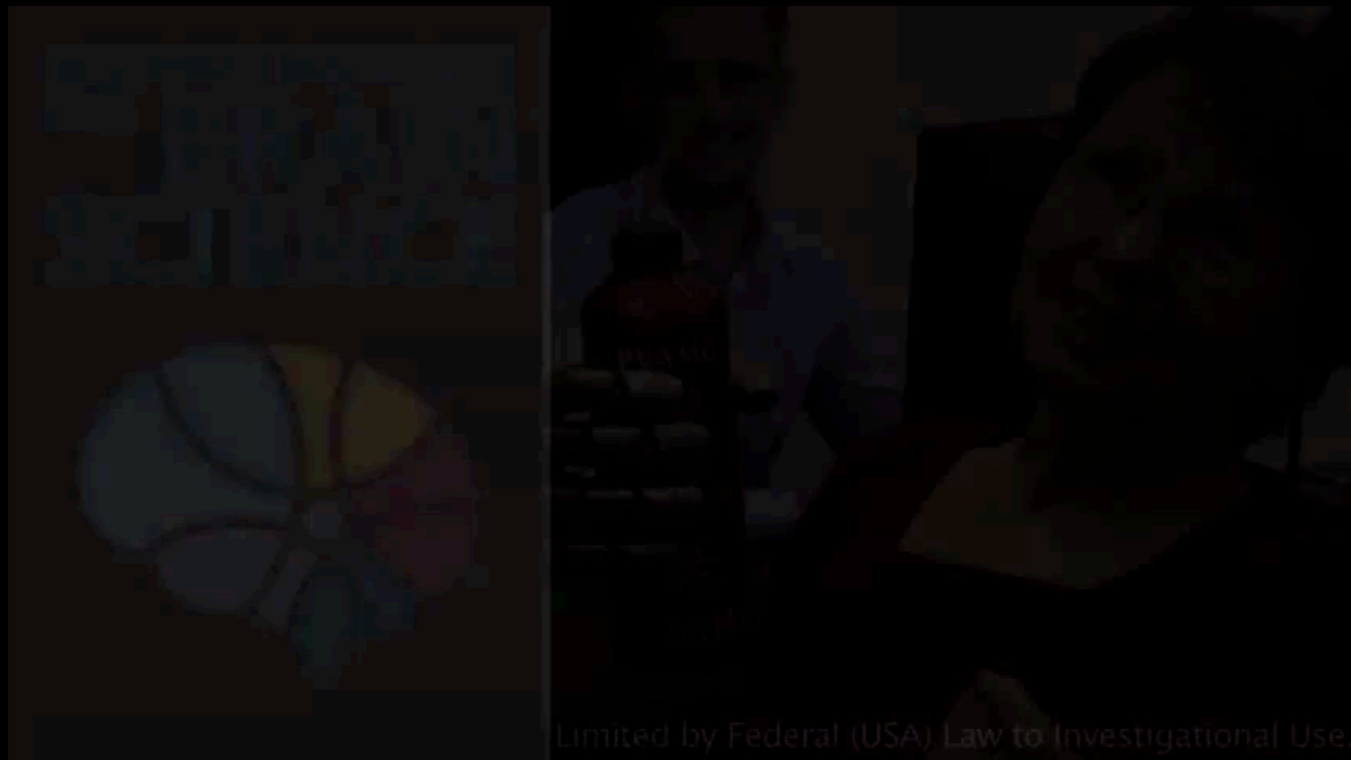
NEUROIMAGING TECHNIQUES & BCI

Technology	Electrical				Magnetic		Optical	
Name	EEG			ECoG	Intracortical	MEG	fMRI	fNIRS
Invasive	✗			✓	✓	✗	✗	✗
Portable	✓			✓	✓	✗	✗	✓
Cost	From \$100 to \$30,000+			\$1000 grid	\$2000 per array	\$1 mln	\$2-3 mln	\$200,000
Temporal resolution	50 ms			3 ms	3 ms	50ms	1-2 s	1 s
Spatial resolution	1+ cm			1 mm	0.5 mm - 0.05 mm	5 mm	1 mm voxels	5 mm
	Signal classification	VEP	P300					
Performance	2-class 90% 3-class 80% 4-class 60%	Large number of targets	Large number of targets	8-cl 90%	High*	~ same as EEG based	4-cl 90%	2-cl 90%



c-VEP BCI

Brain-Computer Interface based on code-modulated visual evoked potentials



Limited by Federal (USA) Law to Investigational Use.

Controlling An Avatar By
Thought Using Real-Time
fMRI