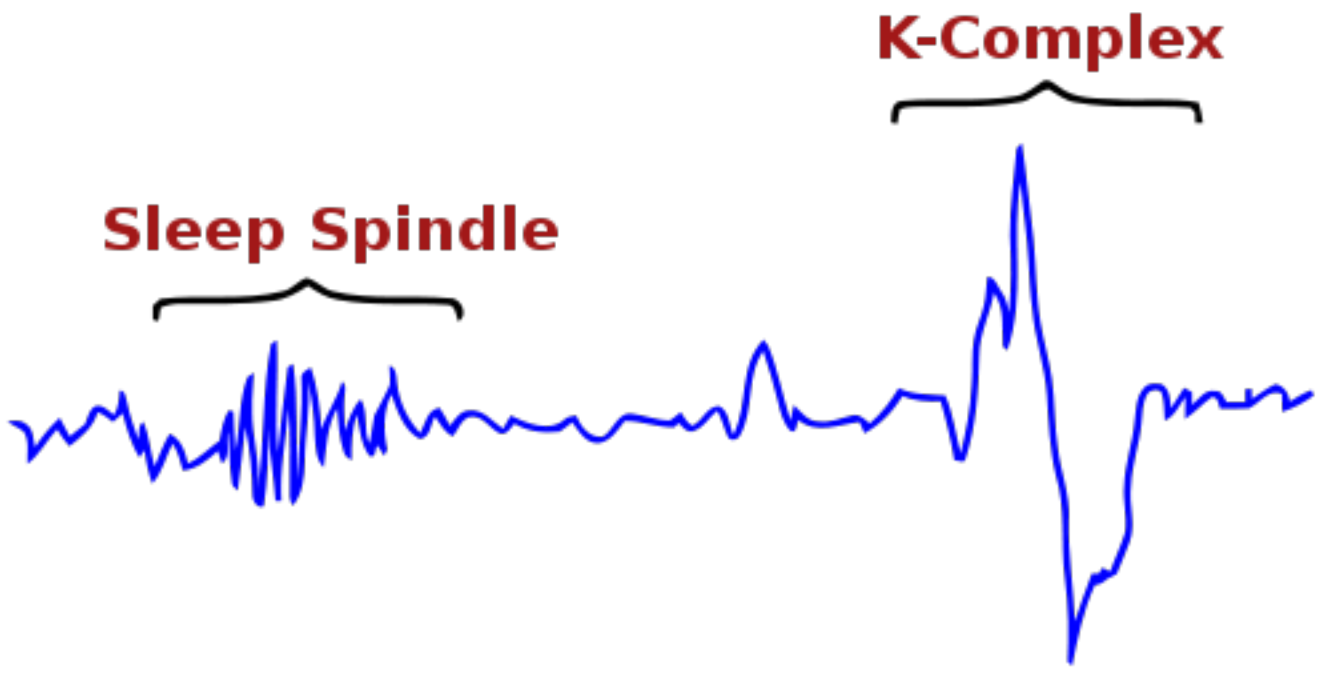
Machine learning the waveform shapes in highly sampled temporal dynamics

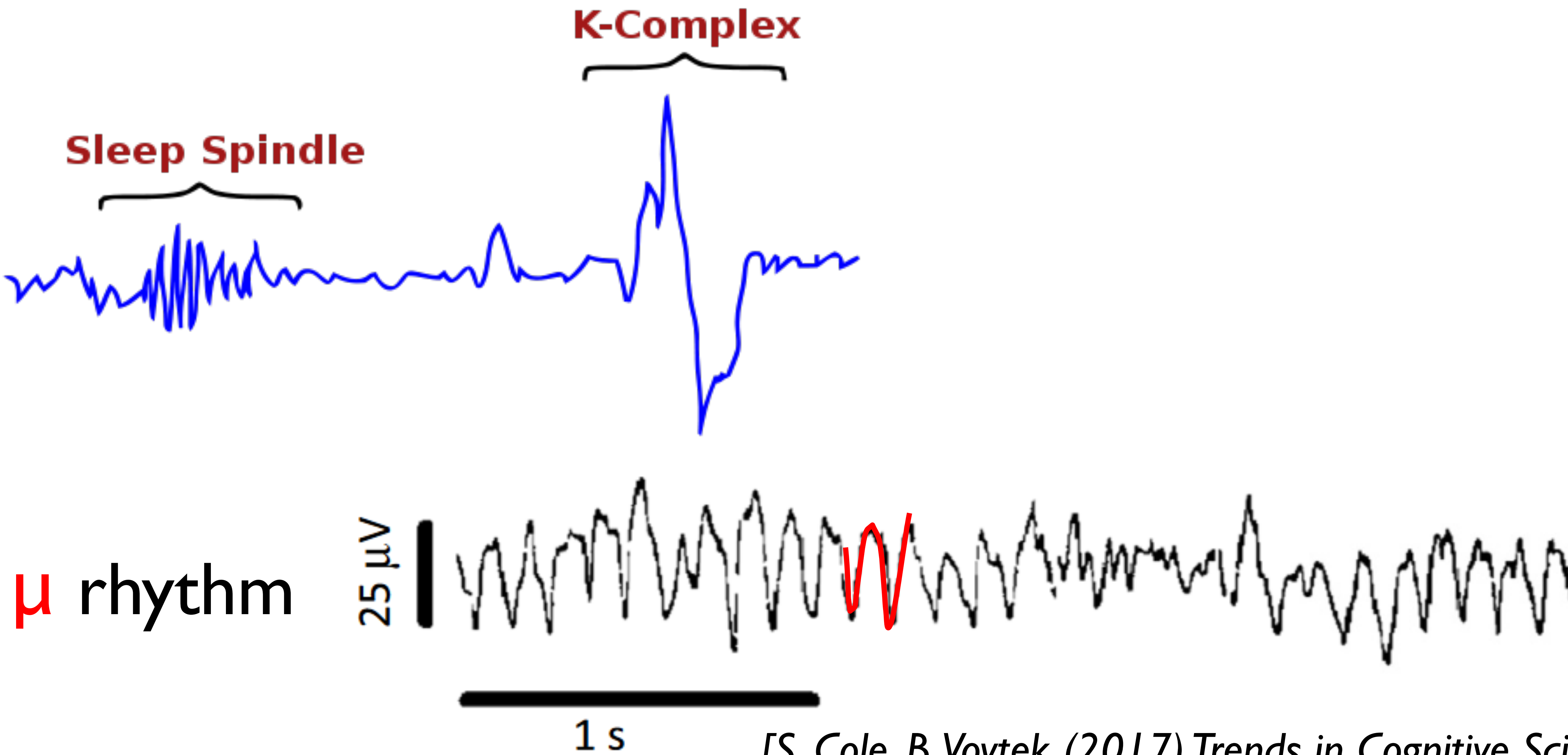
Alexandre Gramfort

http://alexandre.gramfort.net

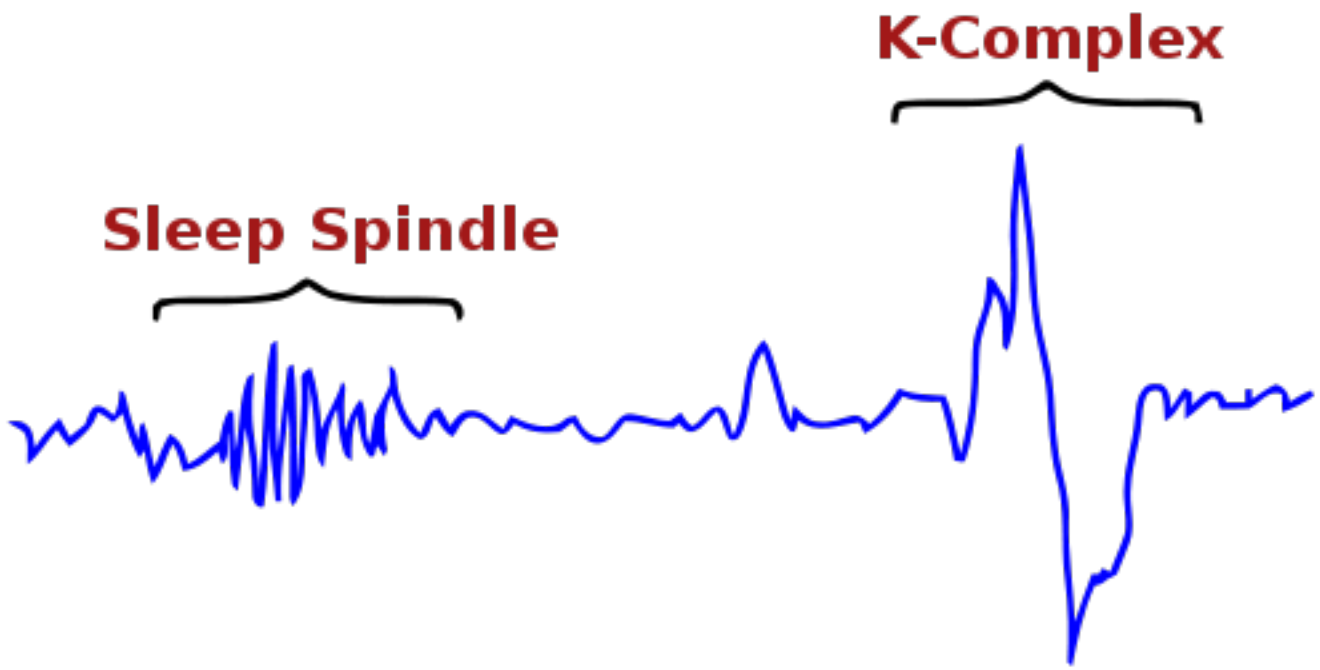
Inria, Parietal Team

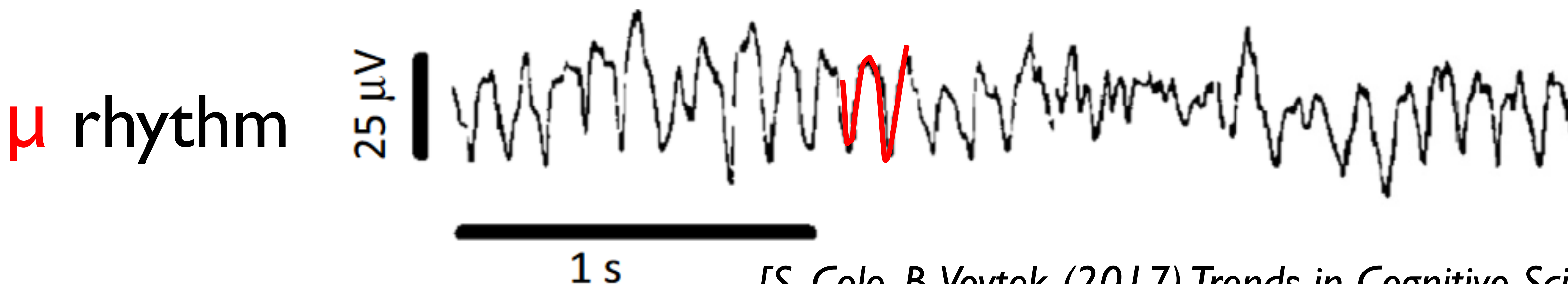
OHBM - June 2019


K-Complex
Sleep Spindle



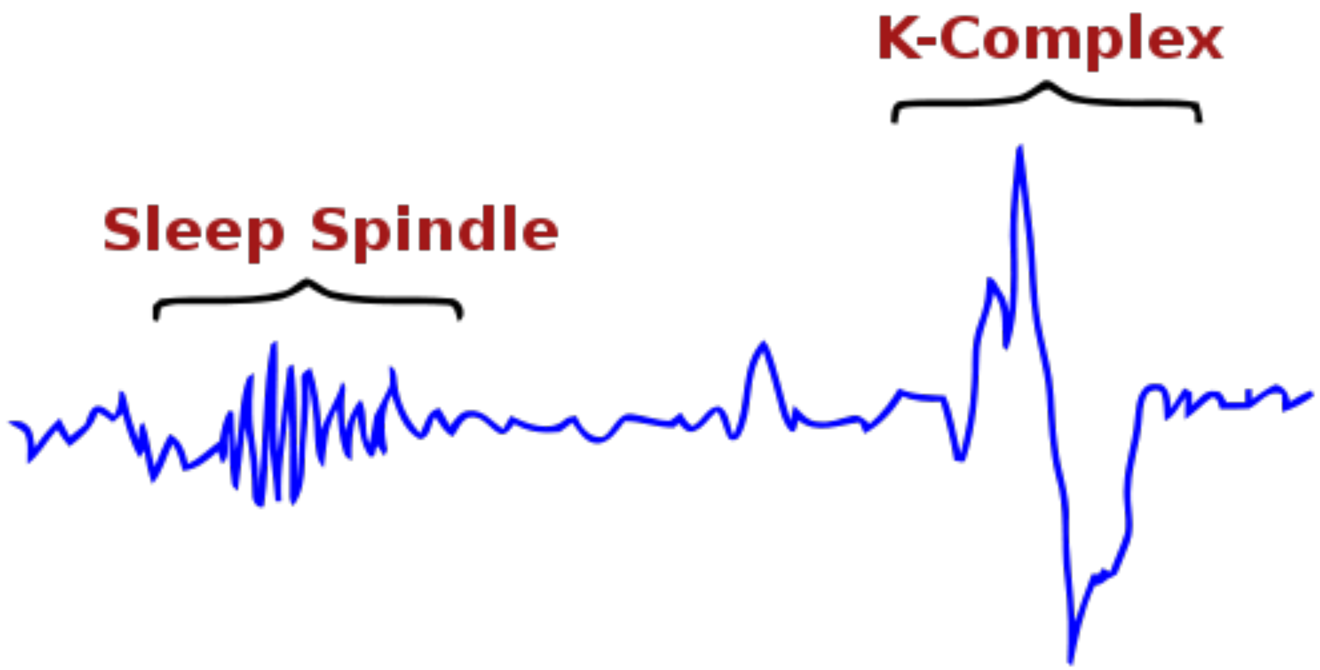
[S. Cole, B. Voytek (2017) Trends in Cognitive Sciences]



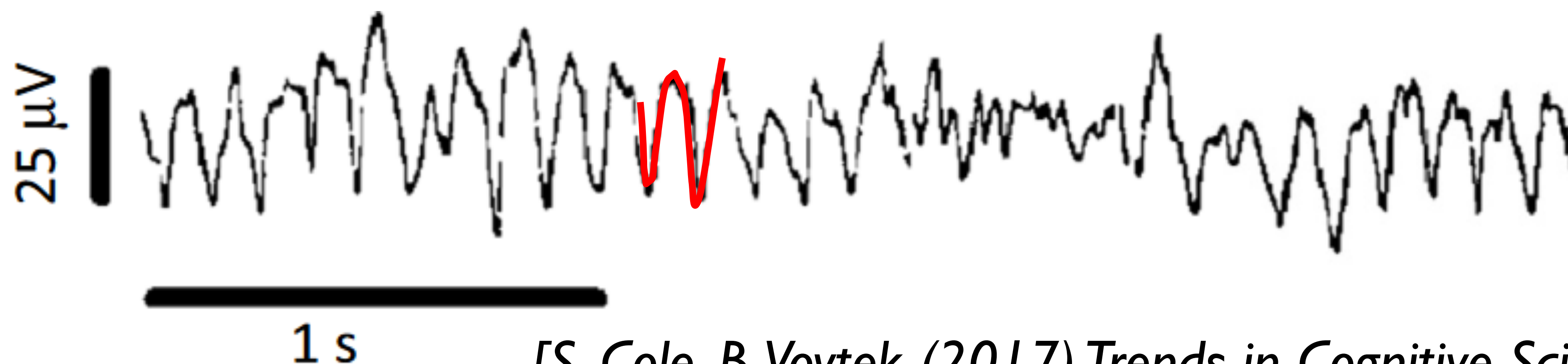


[S. Cole, B. Voytek (2017) Trends in Cognitive Sciences]

[T. Dupré la Tour, L. Tallot, L. Grabot, V. Doyère, V. van Wassenhove, Y. Grenier, A. Gramfort, (2017) PLOS Computational biology]



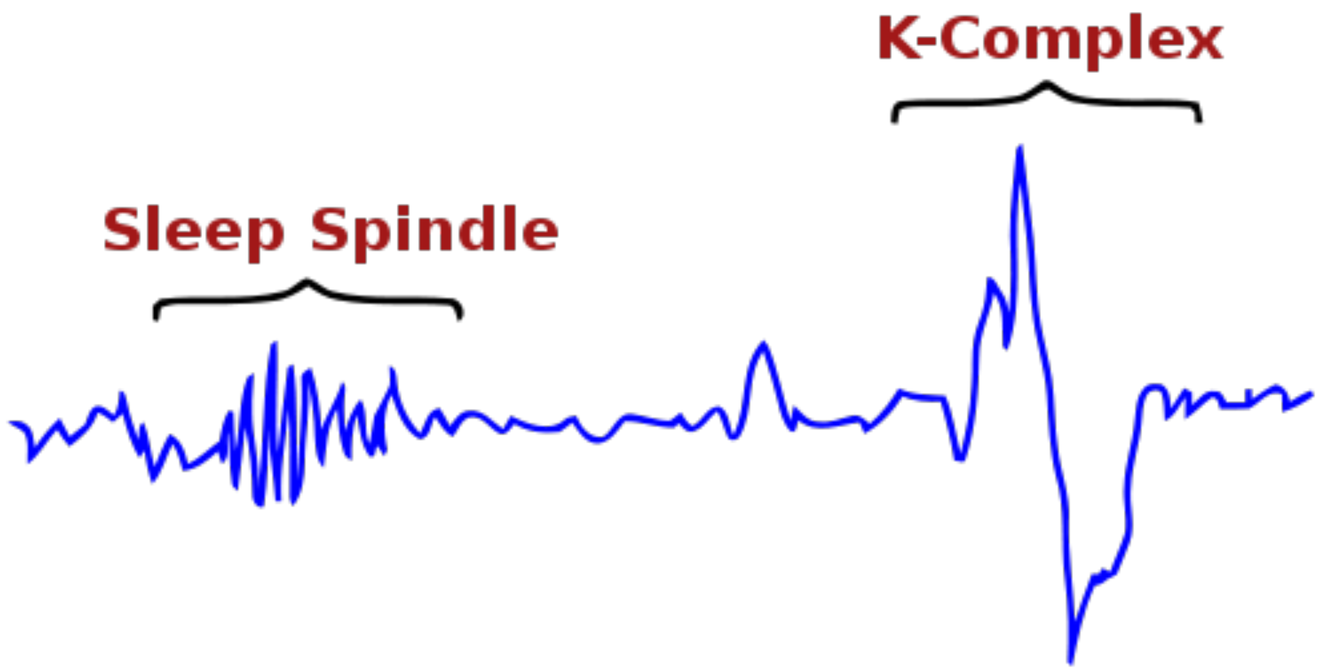
μ rhythm



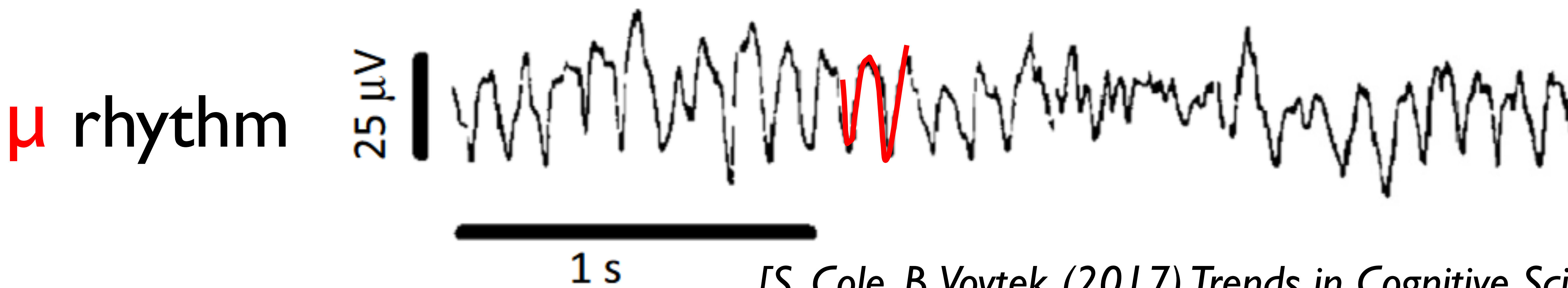
[S. Cole, B. Voytek (2017) Trends in Cognitive Sciences]

CFC: High frequency bursts coupled with slow waves

[T. Dupré la Tour, L. Tallot, L. Grabot, V. Doyère, V. van Wassenhove, Y. Grenier, A. Gramfort, (2017) PLOS Computational biology]



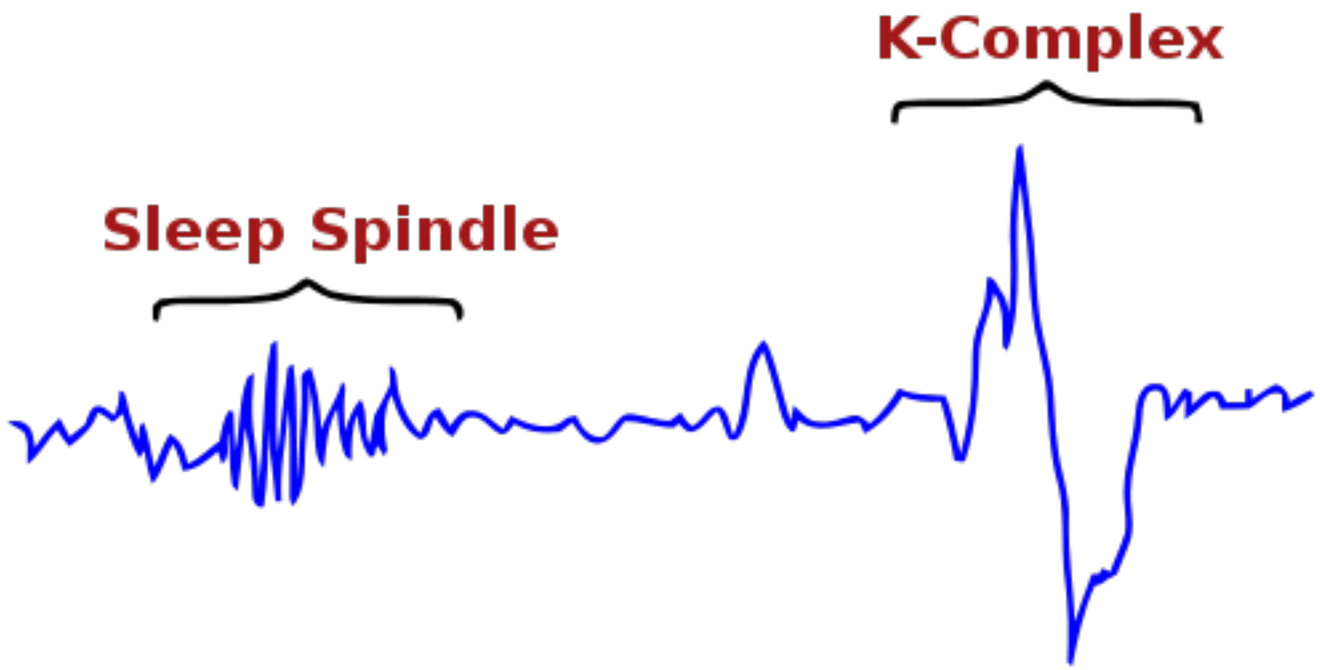
Neural signals exhibit diverse and complex morphologies



[S. Cole, B. Voytek (2017) Trends in Cognitive Sciences]

CFC: High frequency bursts coupled with slow waves

[T. Dupré la Tour, L. Tallot, L. Grabot, V. Doyère, V. van Wassenhove, Y. Grenier, A. Gramfort, (2017) PLOS Computational biology]



Neural signals exhibit diverse and complex morphologies

Waveform shape is related to disease:

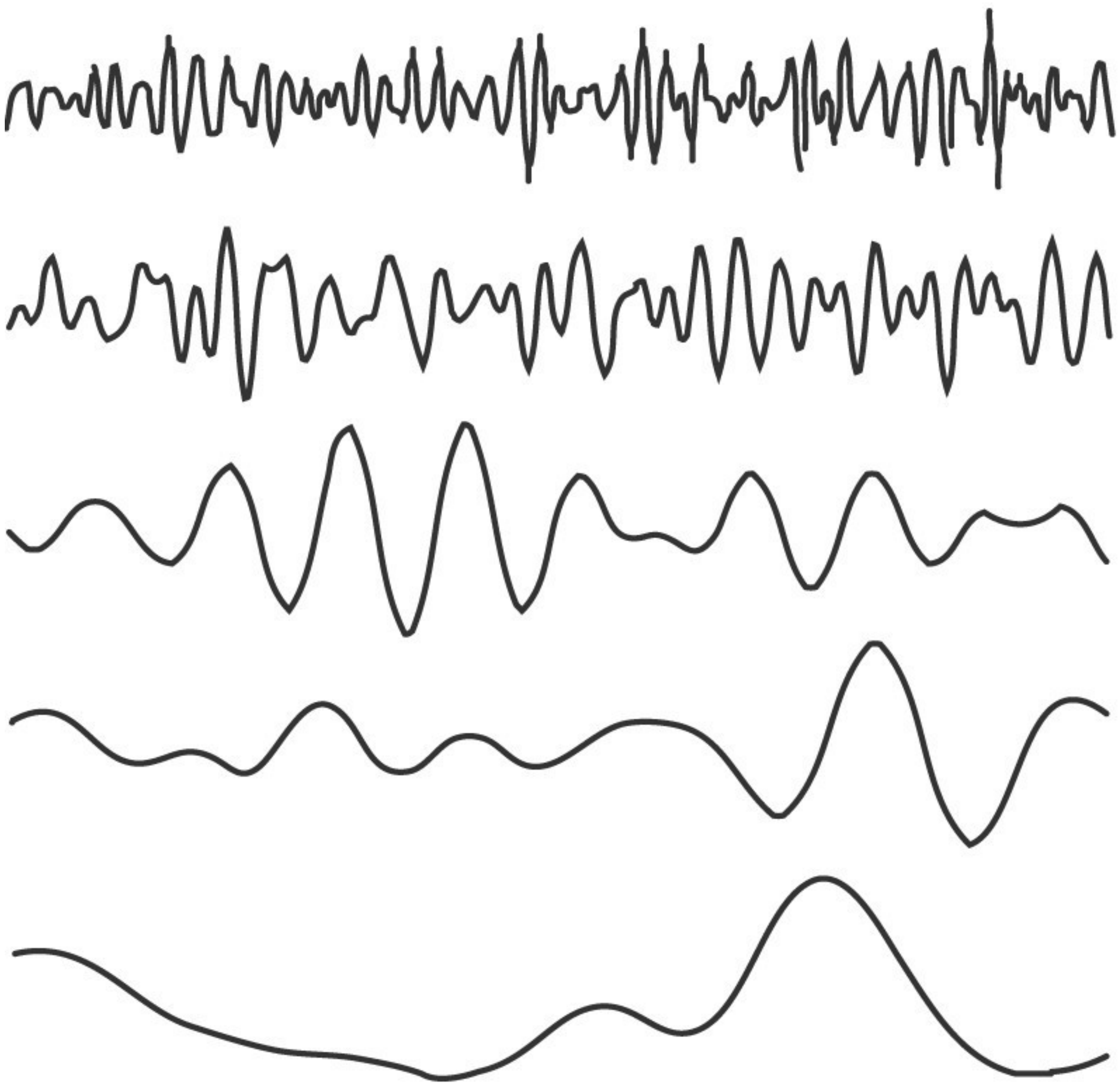
- Parkinson [Jackson et al. ENeuro 2019]

e Sciences]

CFC: High frequency bursts coupled with slow waves

[T. Dupré la Tour, L. Tallot, L. Grabot, V. Doyère, V. van Wassenhove, Y. Grenier, A. Gramfort, (2017) PLOS Computational biology]

“Textbook” brain rythms



Gamma
(> 25 Hz)

Beta
(12-25 Hz)

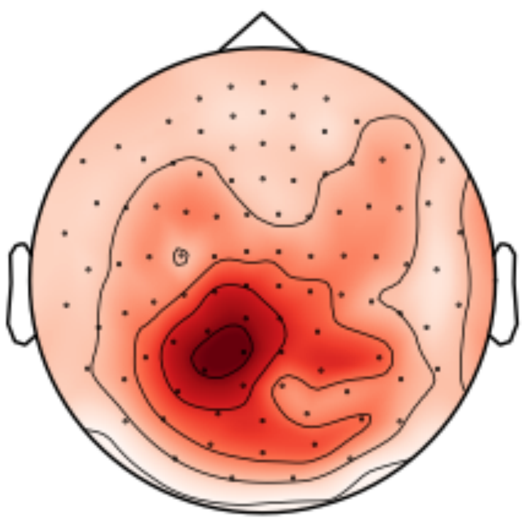
Alpha
(8-12 Hz)

Theta
(4-8 Hz)

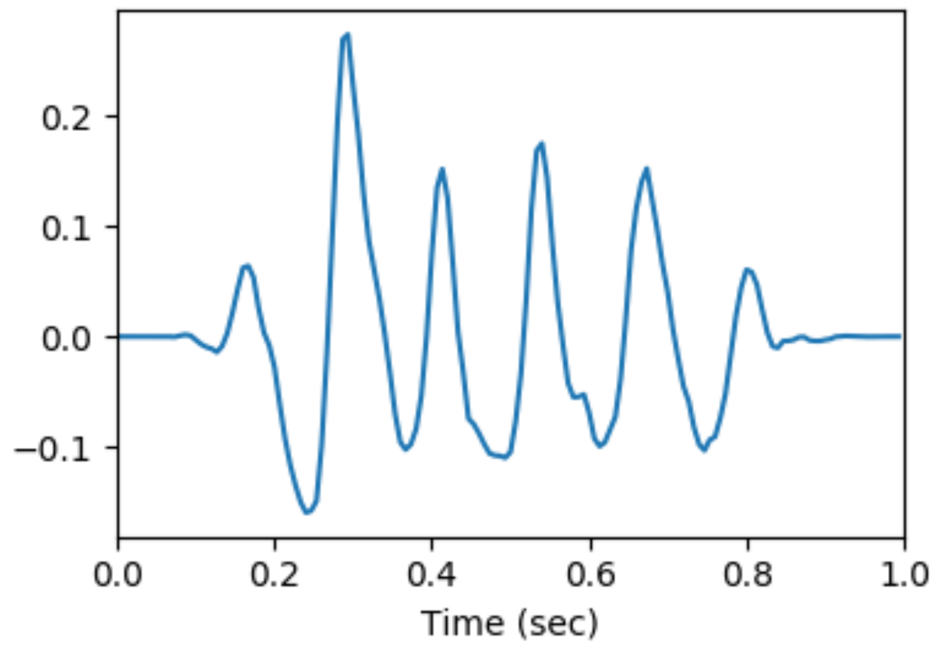
Delta
(1-4 Hz)

Fourier Fallacy [Jasper 48]

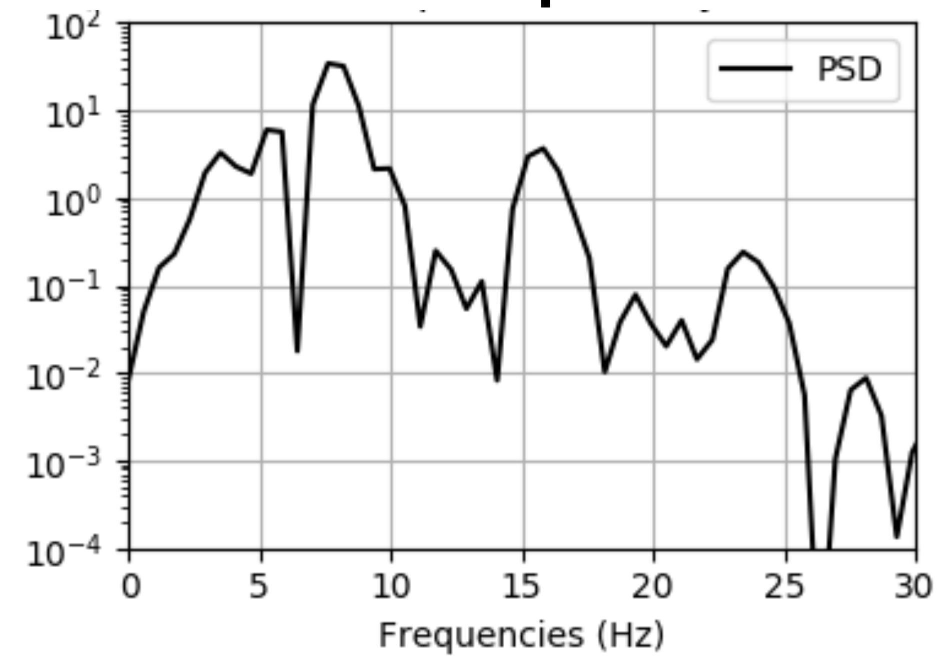
Topography:



Waveform:

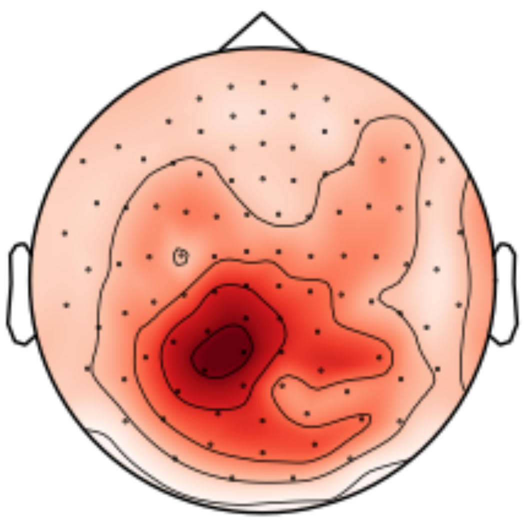


Power Spectra:

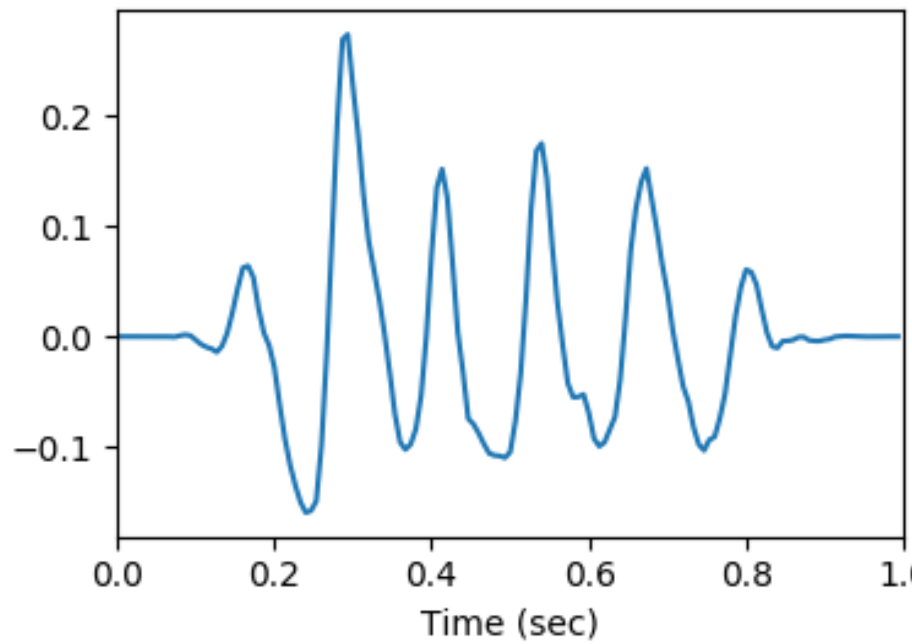


Fourier Fallacy [Jasper 48]

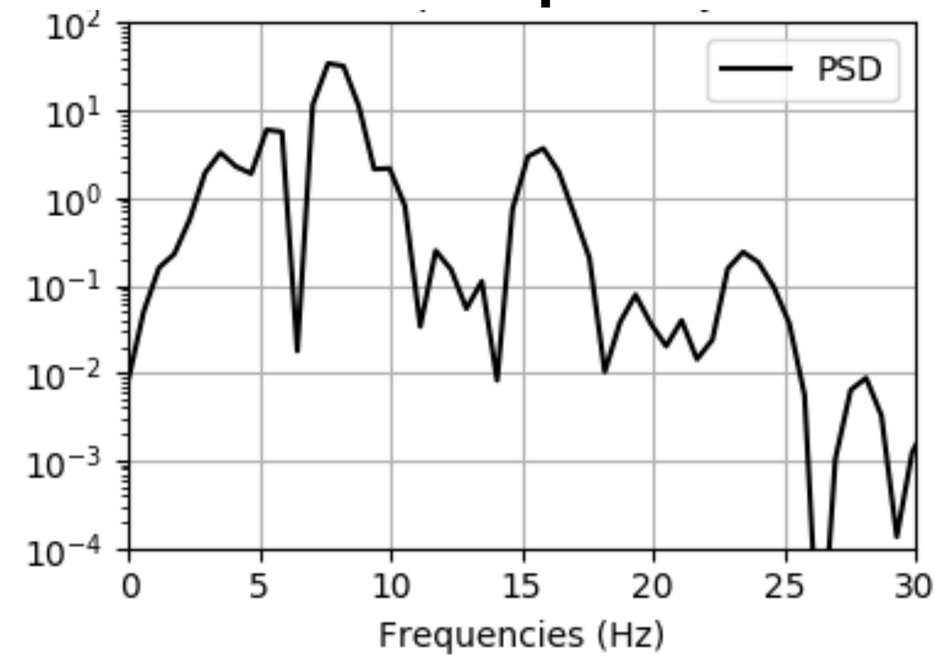
Topography:



Waveform:



Power Spectra:

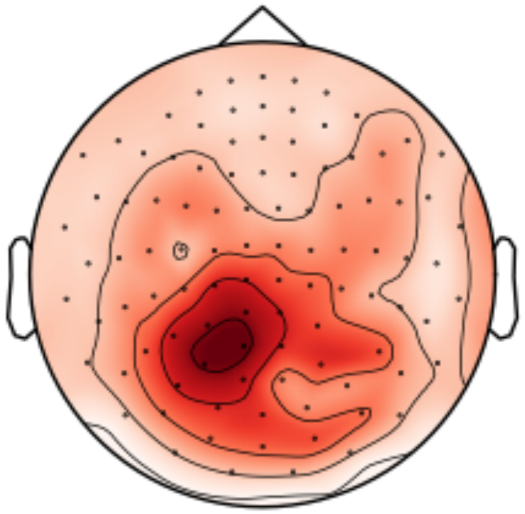


«Even though it may be possible to analyze the complex forms of brain waves into **a number of different sine-wave** frequencies, this may lead only to what might be termed a **"Fourier fallacy"**, if one assumes ***ad hoc*** that all of the necessary frequencies actually occur as periodic phenomena in **cell groups** within the brain.»

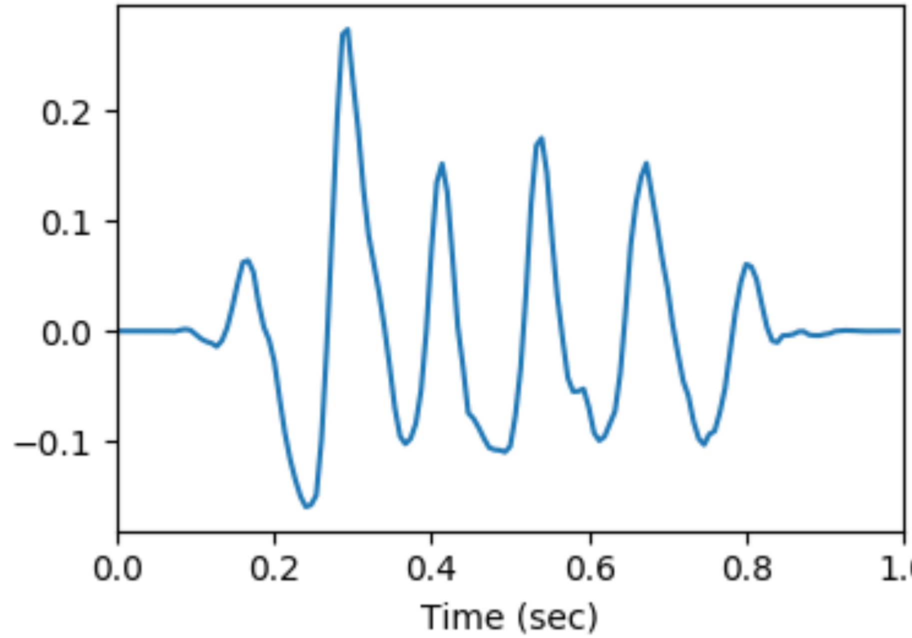
[Jasper, 1948]

Fourier Fallacy [Jasper 48]

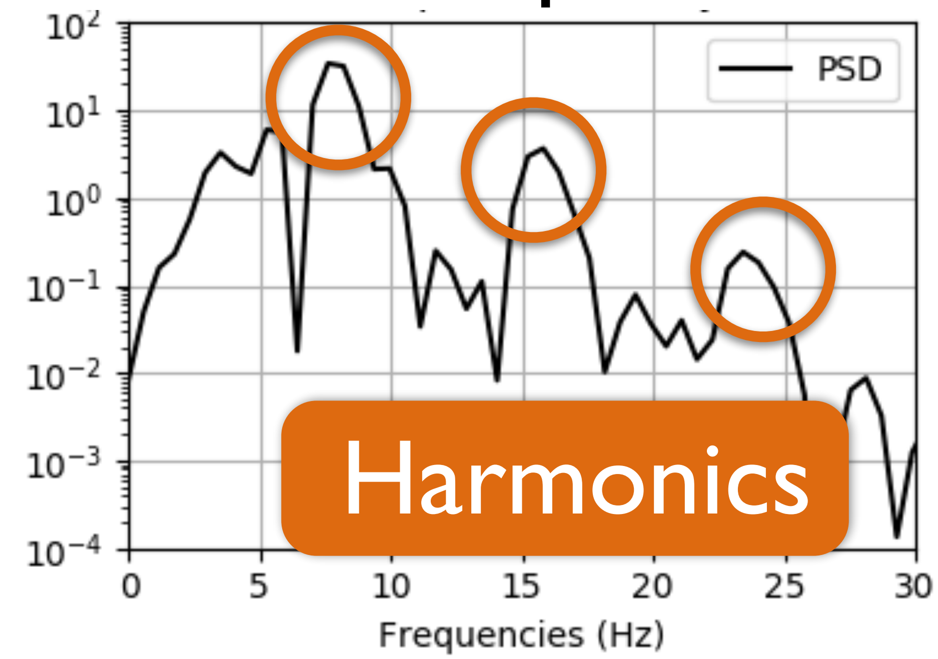
Topography:



Waveform:



Power Spectra:

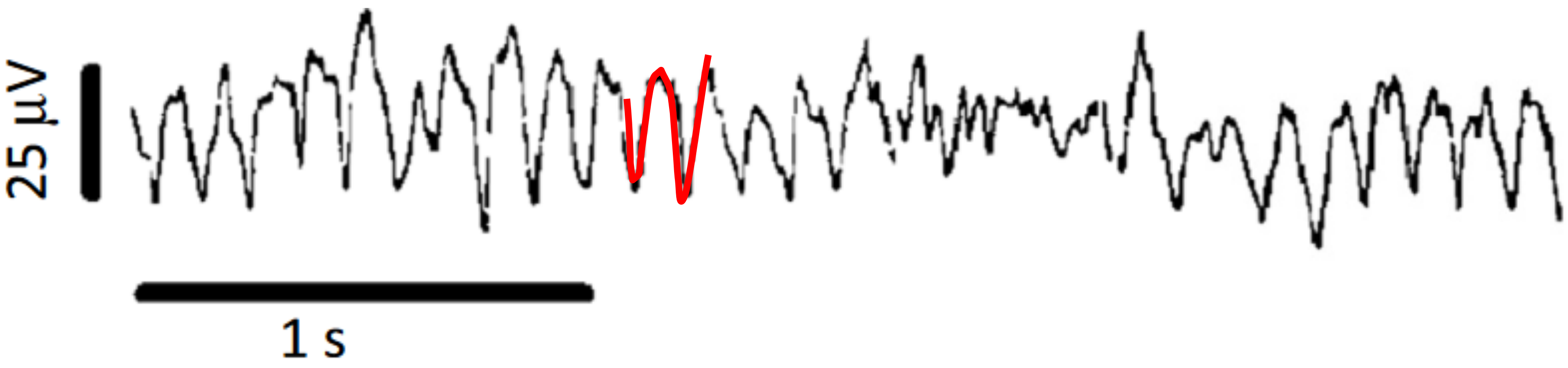


«Even though it may be possible to analyze the complex forms of brain waves into **a number of different sine-wave** frequencies, this may lead only to what might be termed a **"Fourier fallacy"**, if one assumes ***ad hoc*** that all of the necessary frequencies actually occur as periodic phenomena in **cell groups** within the brain.»

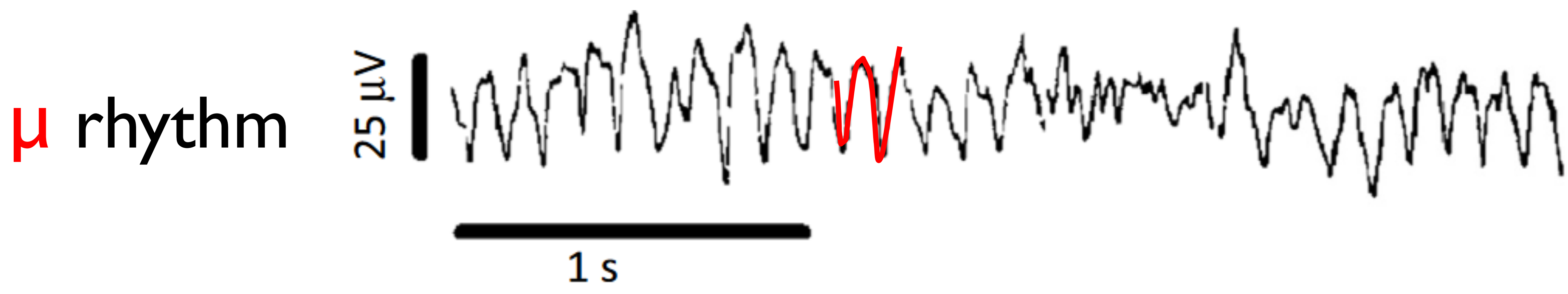
[Jasper, 1948]

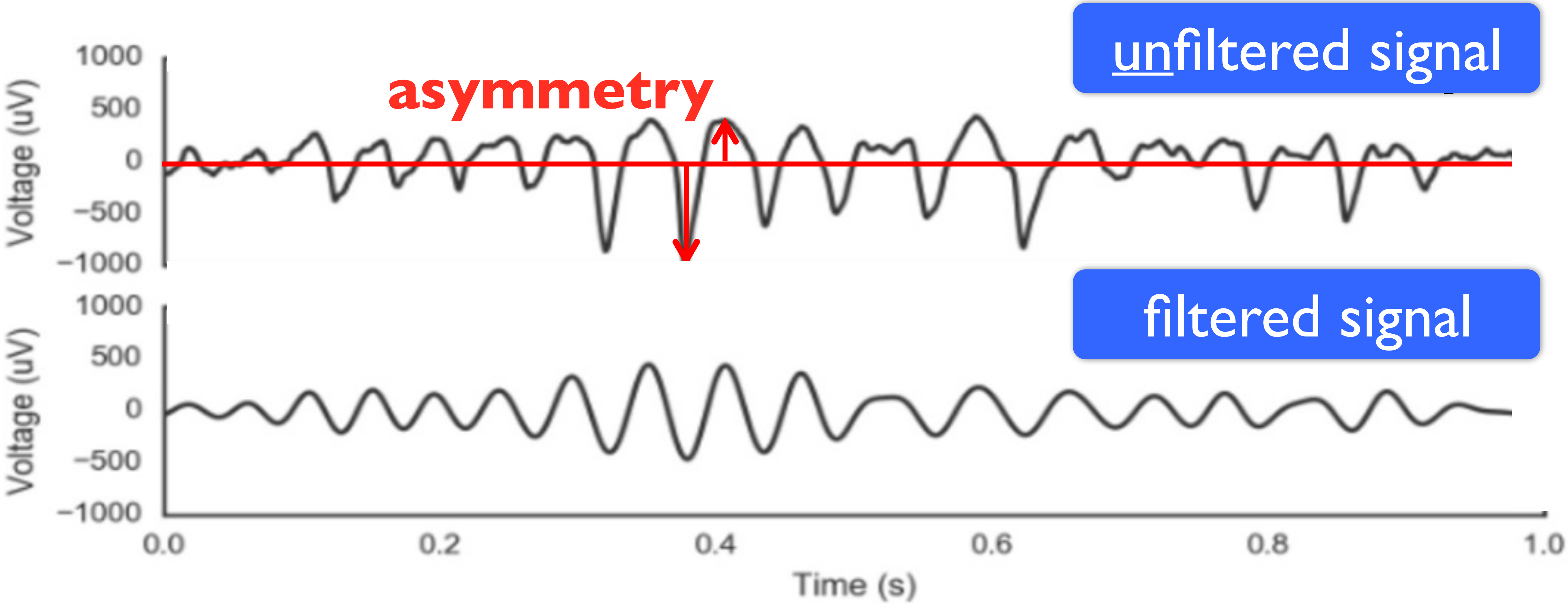
Problem of linear filtering

μ rhythm

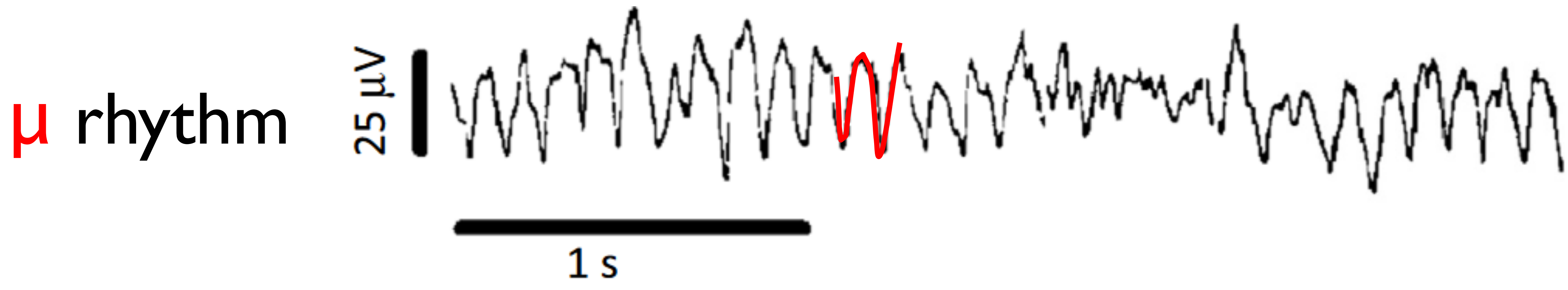


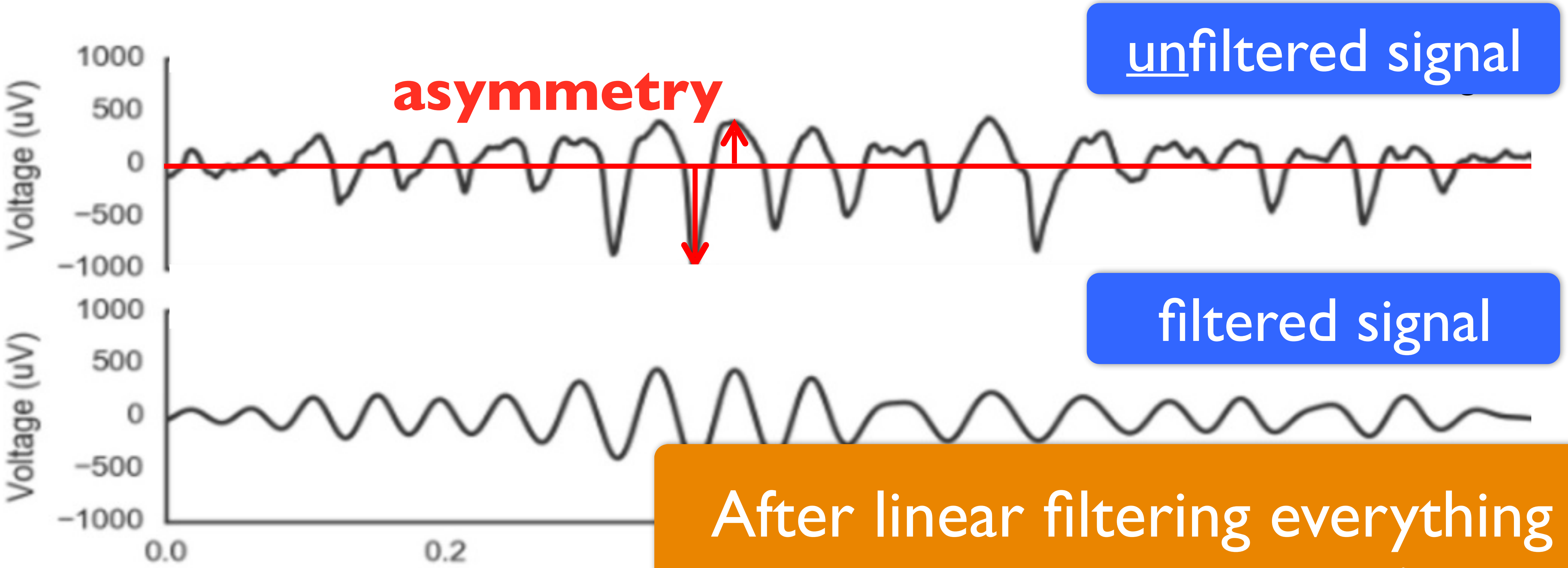
Problem of linear filtering





Problem of linear filtering





After linear filtering everything looks like a sinusoid!


So can we learn the waveform shapes?

Convolutional Sparse Coding (CSC) for learning the morphology of neural signals

Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.

Learning the Morphology of Brain Signals Using Alpha-Stable Convolutional Sparse Coding, (2017), M. Jas, T. Dupré la Tour, U. Simsekli, A. Gramfort, Proc. NeurIPS Conf.

Code: https://alphacsc.github.io

Signal representations

- Sparse representations: wavelet basis

  [Morlet 70', Meyer 80', Mallat 90' etc.]

- Sparse coding / dictionary learning

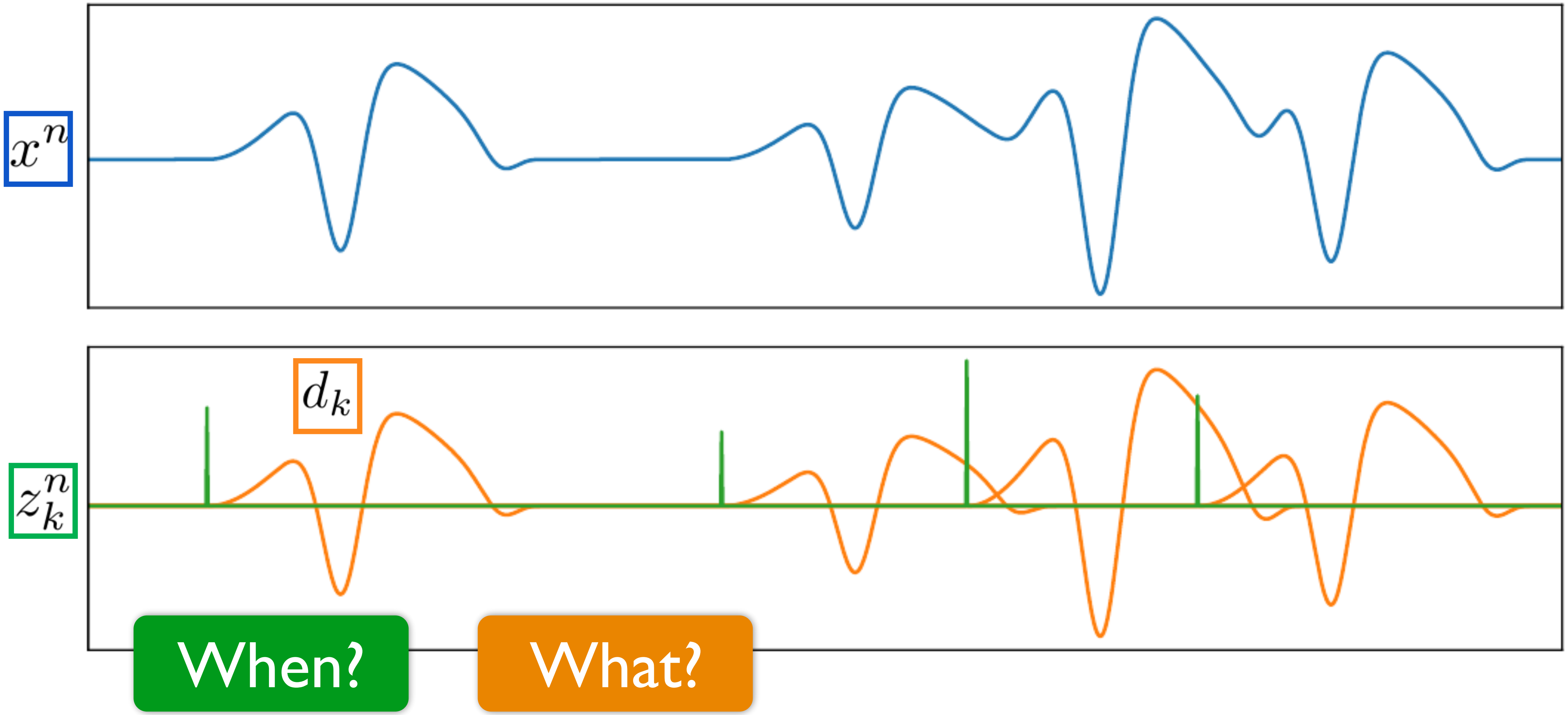
  [Olshausen and Field, 1996, Elad and Aharon, 2006]

- Shift-invariant representations

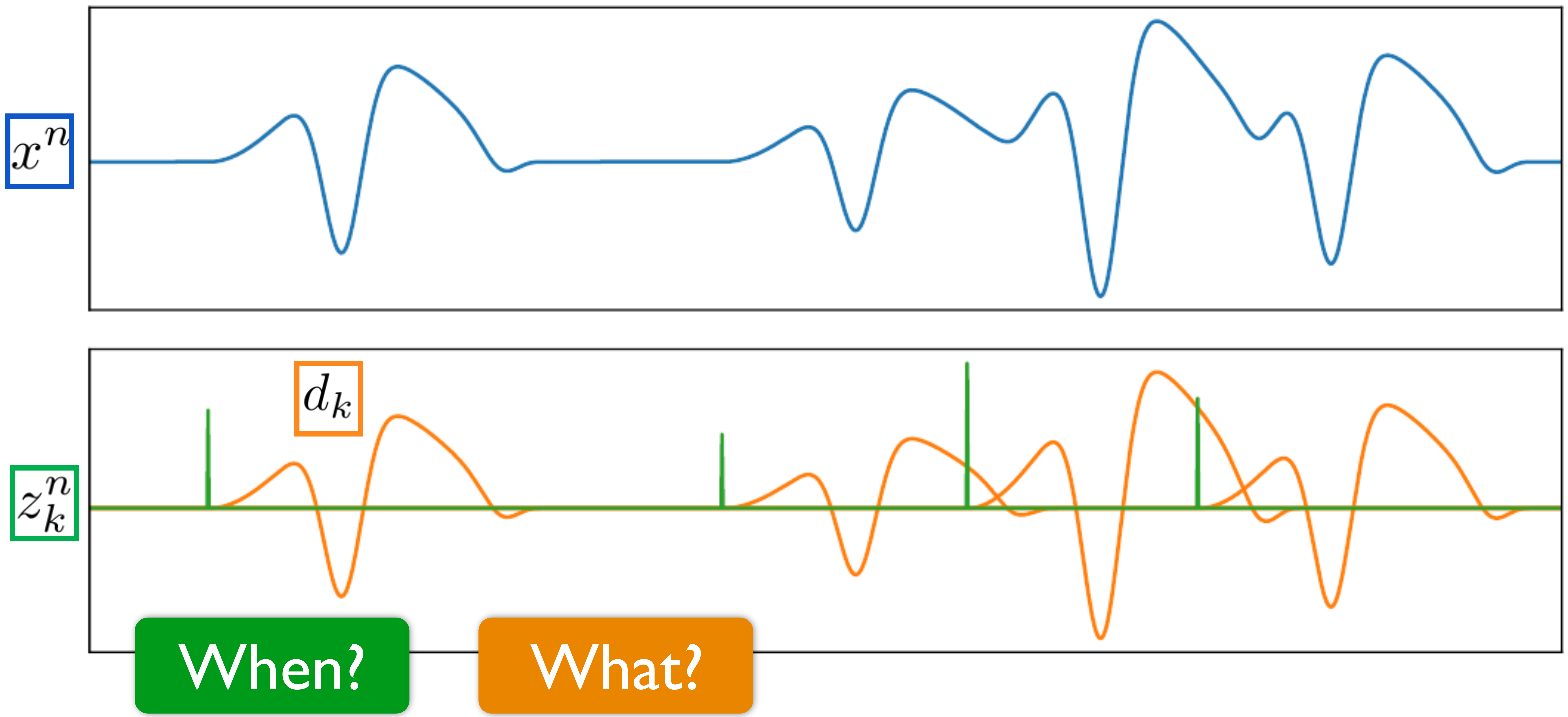
  [Lewicki and Sejnowski, 1999, Grosse et al, 2007]

- In neurophysiology:
  - Matching of time-invariant filters (MOTIF) [Jost et al, 2006]
  - Multivariate orthogonal matching pursuit [Barthélemy et al, 2012]
  - Matching pursuit and heuristics [Brokmeier and Principe, 2016]
  - Sliding window machine [Gips et al, 2017]
  - Adaptive waveform learning [Hitziger et al, 2017]

Convolutional sparse coding



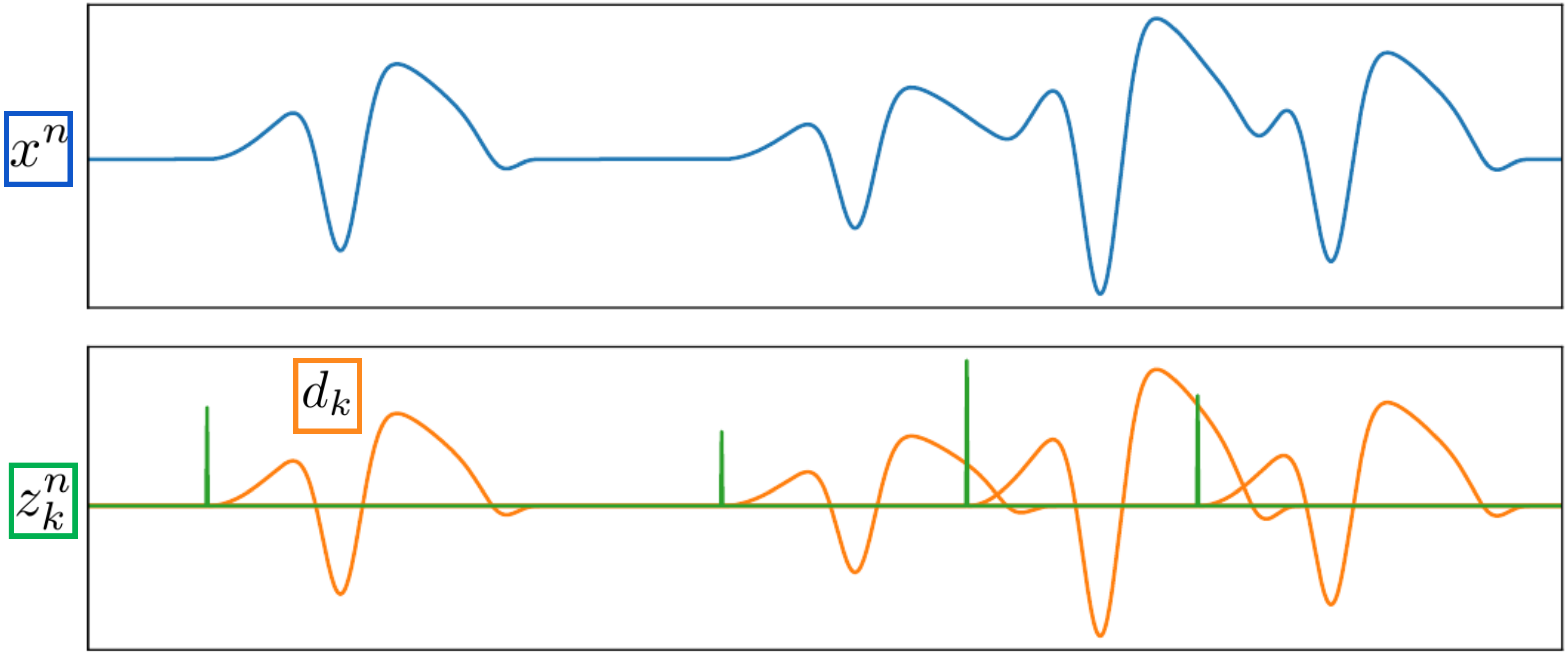
Convolutional sparse coding



$$\boxed{x^n}[t] = \sum_{k=1}^{K} (\boxed{z_k^n} * \boxed{d_k})[t] + \varepsilon[t]$$

[Grosse et al, 2007]

Convolutional sparse coding



$$\min_{d,z} \sum_{n=1}^{N} \frac{1}{2} \left\| \boxed{x^n} - \sum_{k=1}^{K} \boxed{z_k^n} * \boxed{d_k} \right\|_2^2 + \lambda \sum_{k=1}^{K} \|\boxed{z_k^n}\|_1,$$

$$\text{s.t.} \quad \|\boxed{d_k}\|_2^2 \leq 1 \text{ and } \boxed{z_k^n} \geq 0.$$

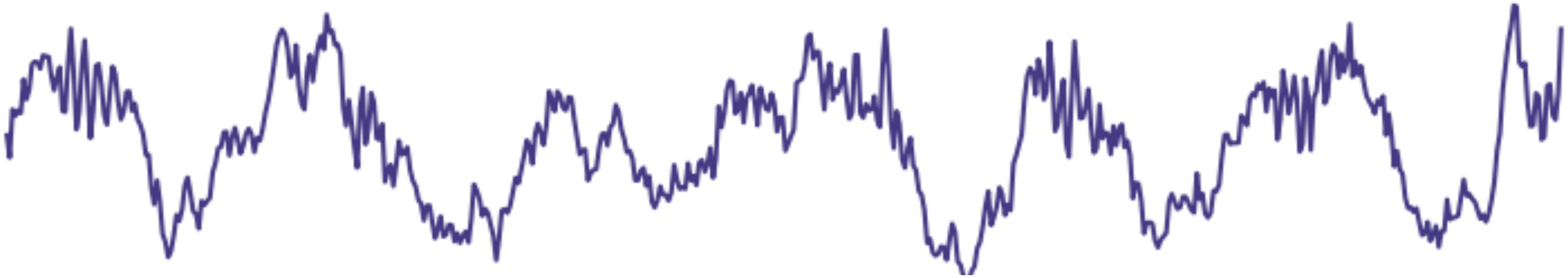
When?

What?

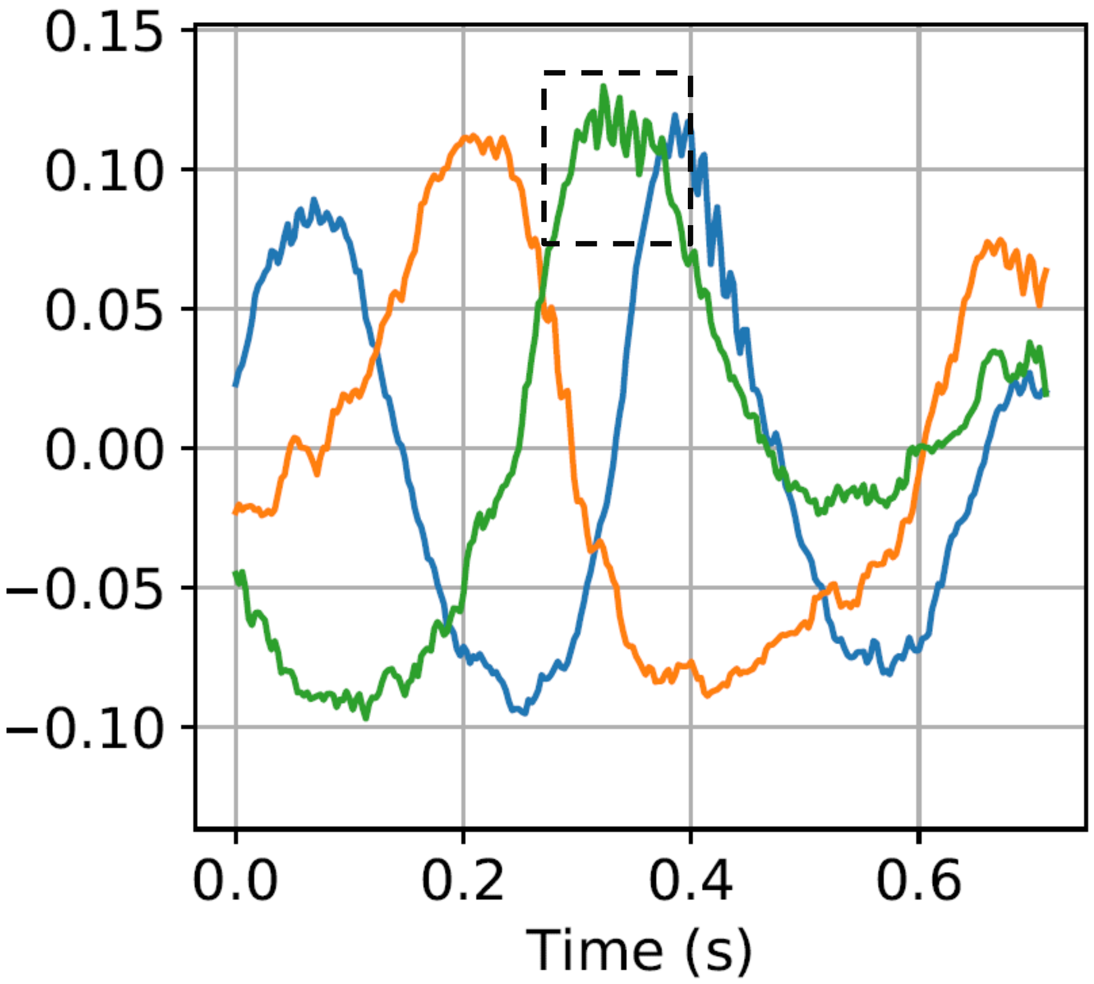
[Grosse et al, 2007]

Learned atoms

Data:

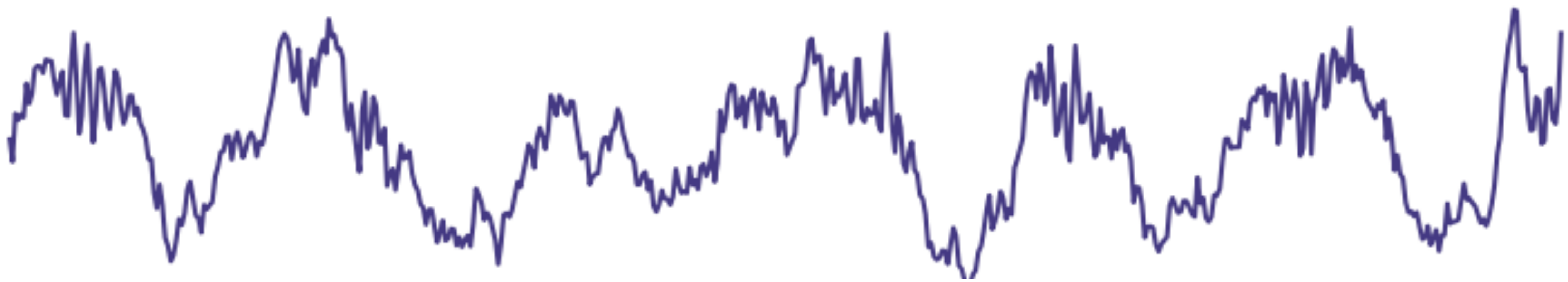


~80 Hz

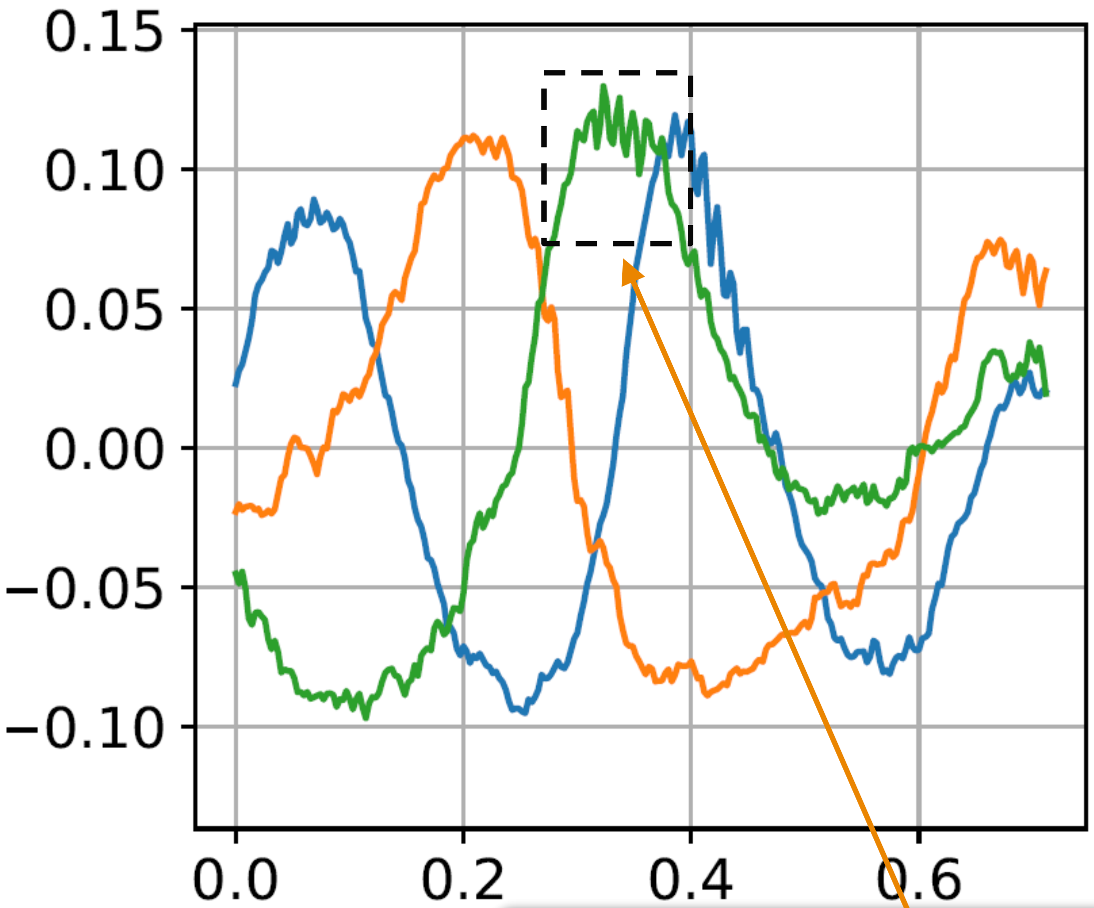


Learned atoms

Data:



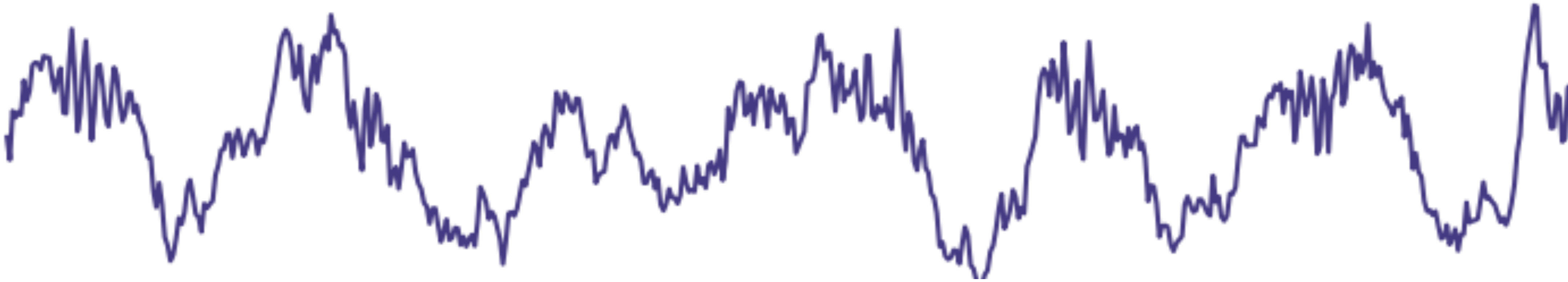
~80 Hz



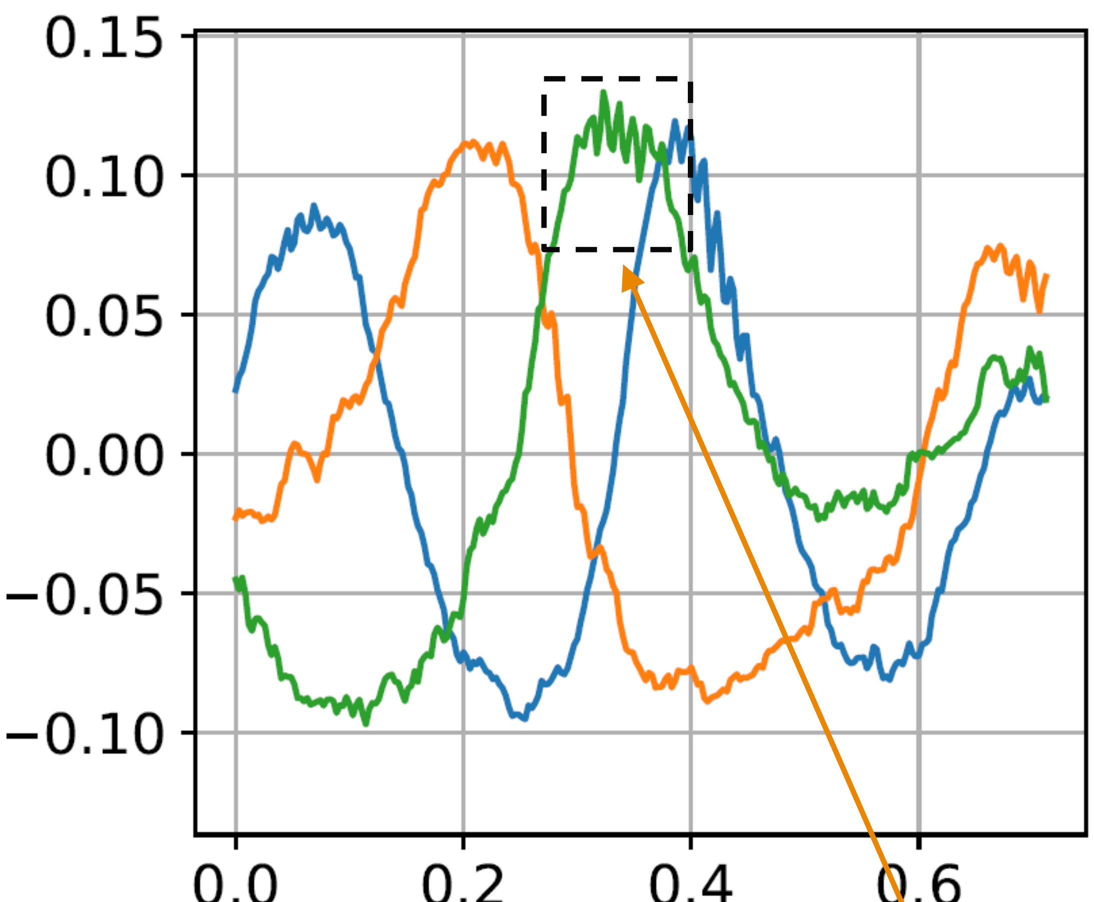
CSC reveals CFC

Learned atoms

Data:



~80 Hz

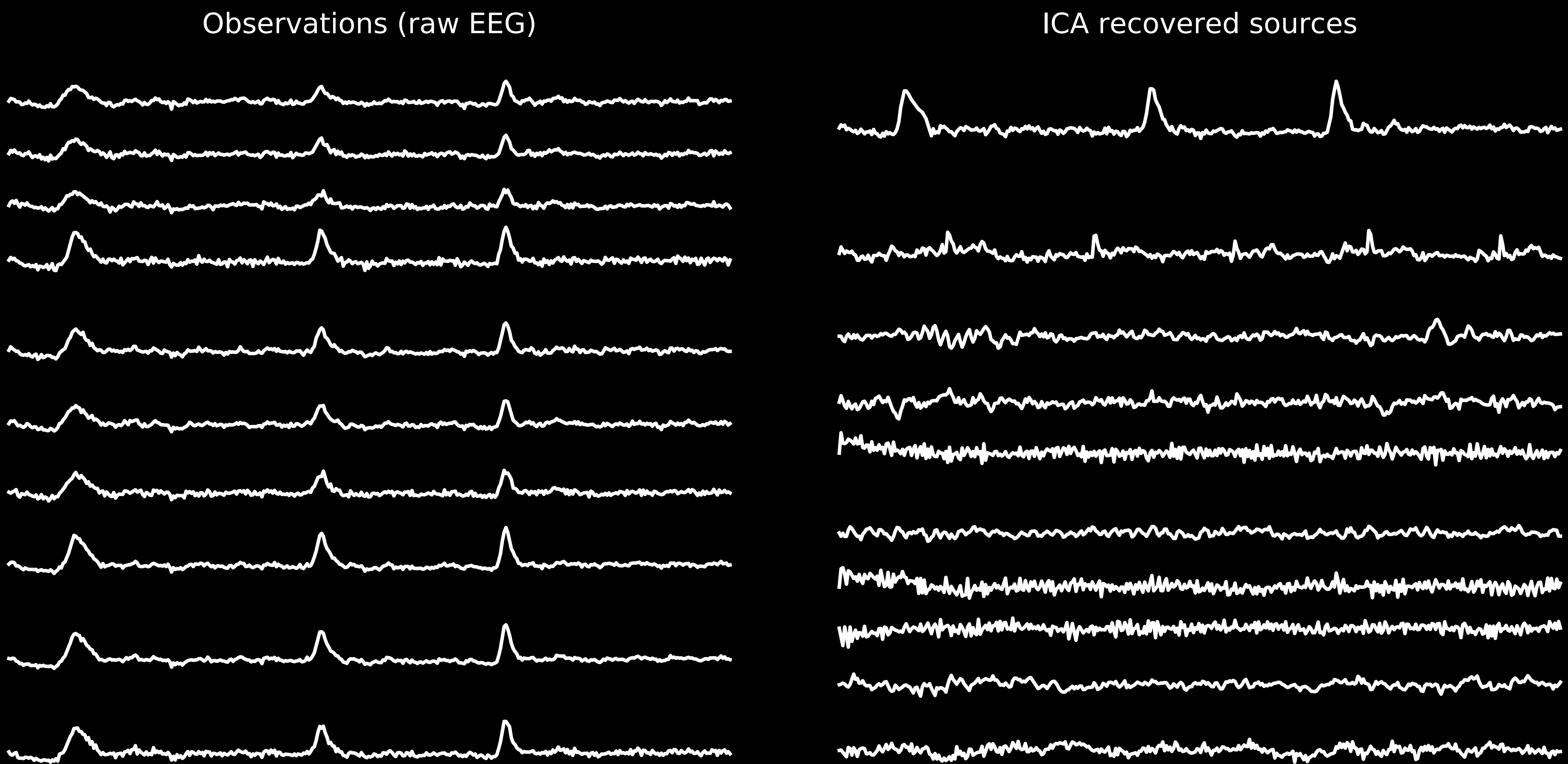


How about if I have many channels?

CSC reveals CFC

From ICA to CSC

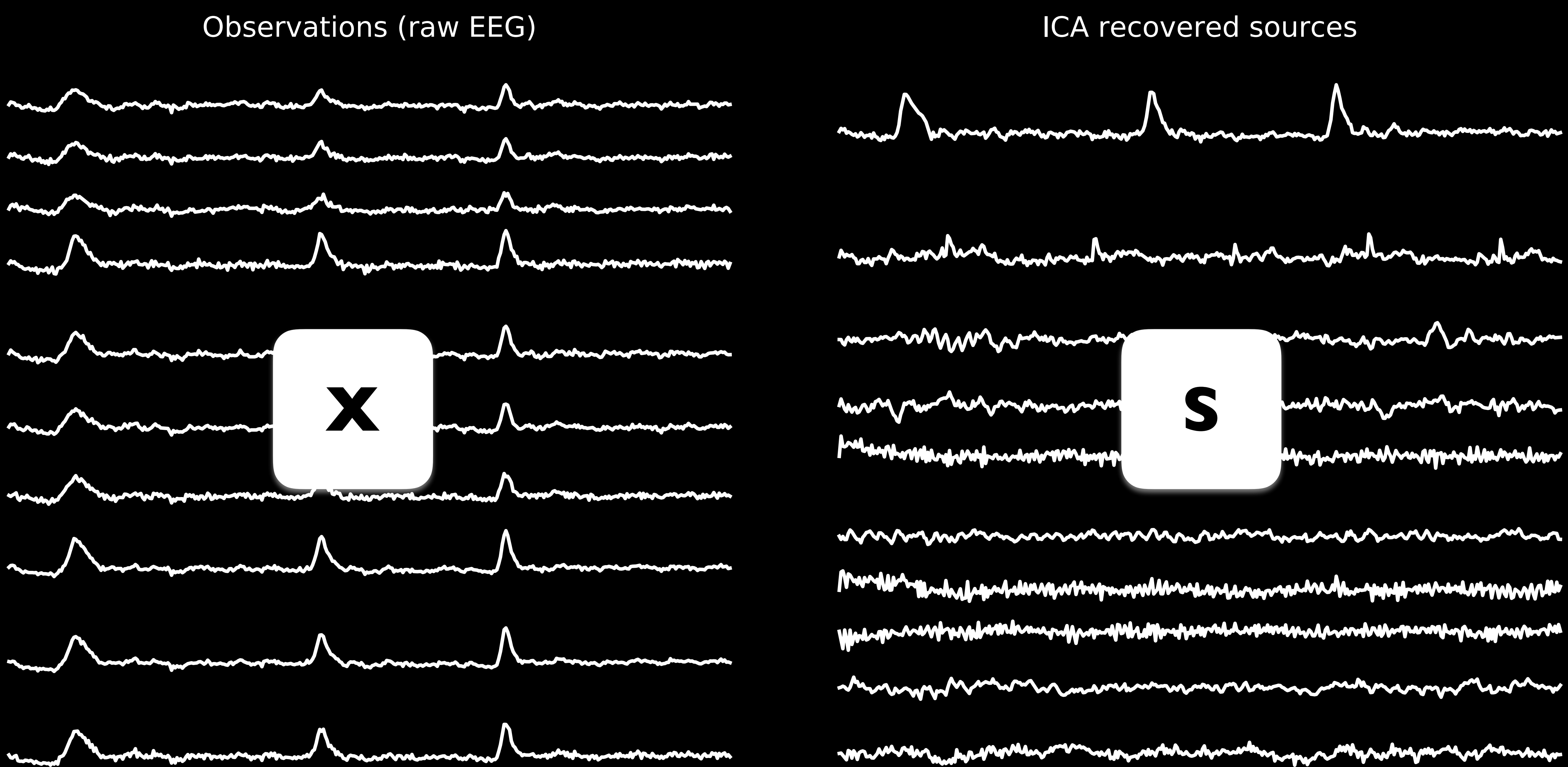
Independent Component Analysis (ICA)



https://pypi.python.org/pypi/python-picard/0.1

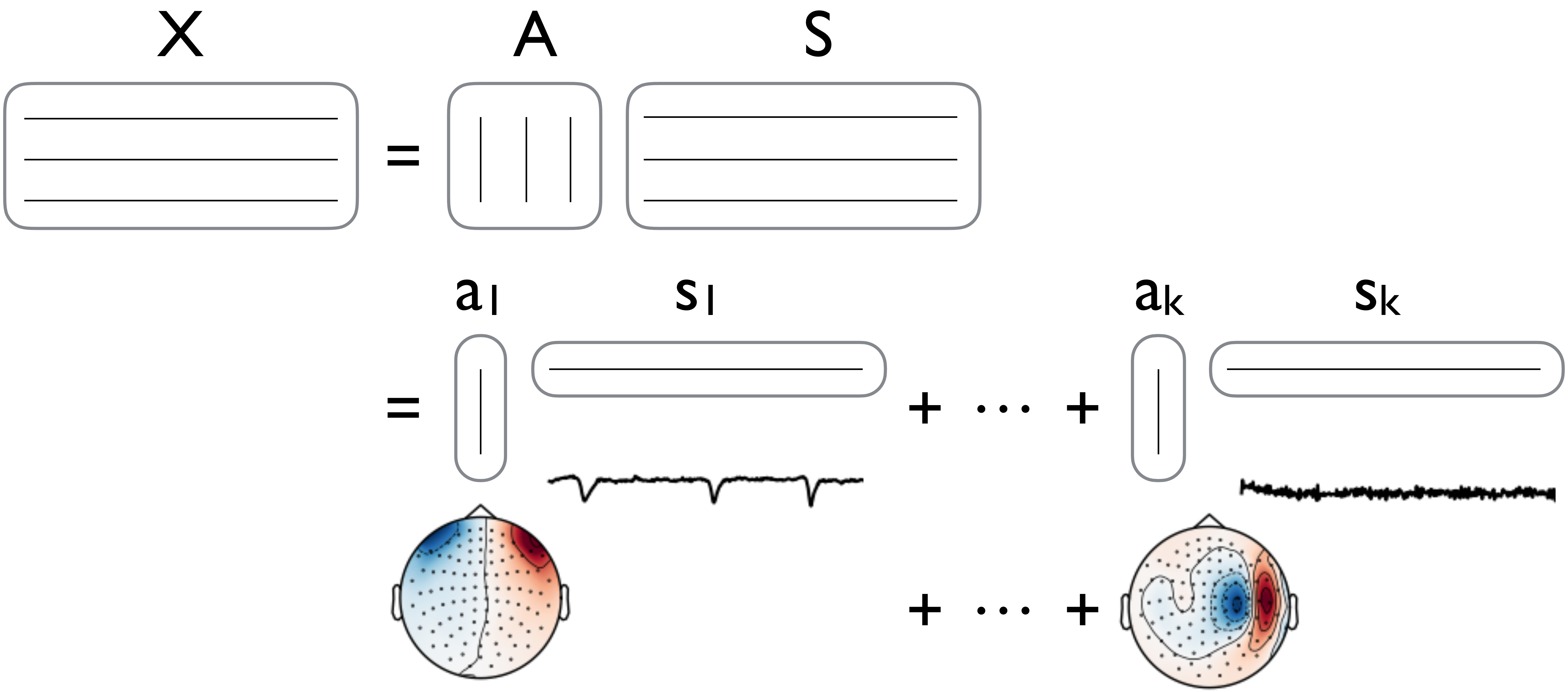
From ICA to CSC

Independent Component Analysis (ICA)



https://pypi.python.org/pypi/python-picard/0.1

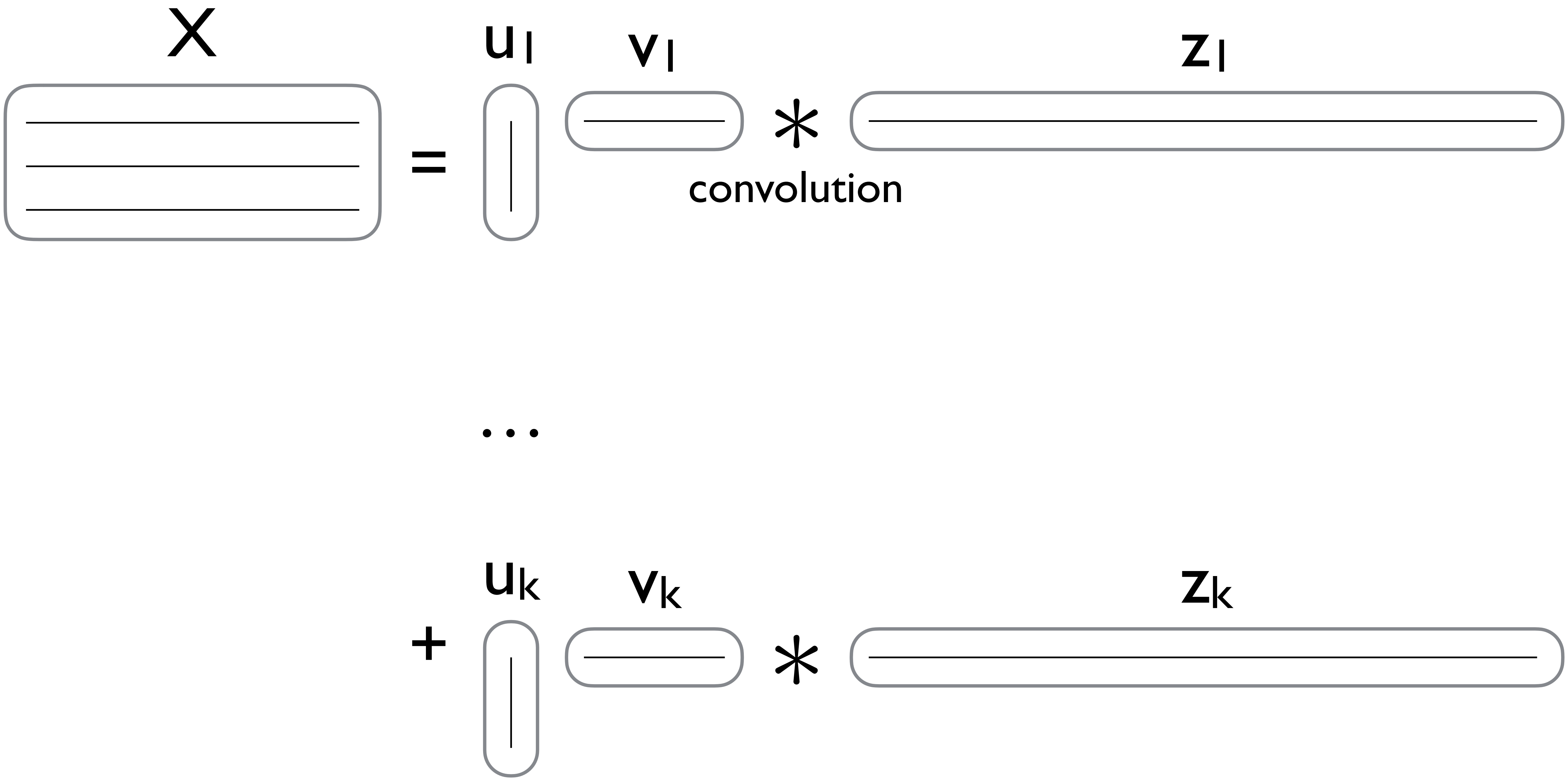
From ICA…



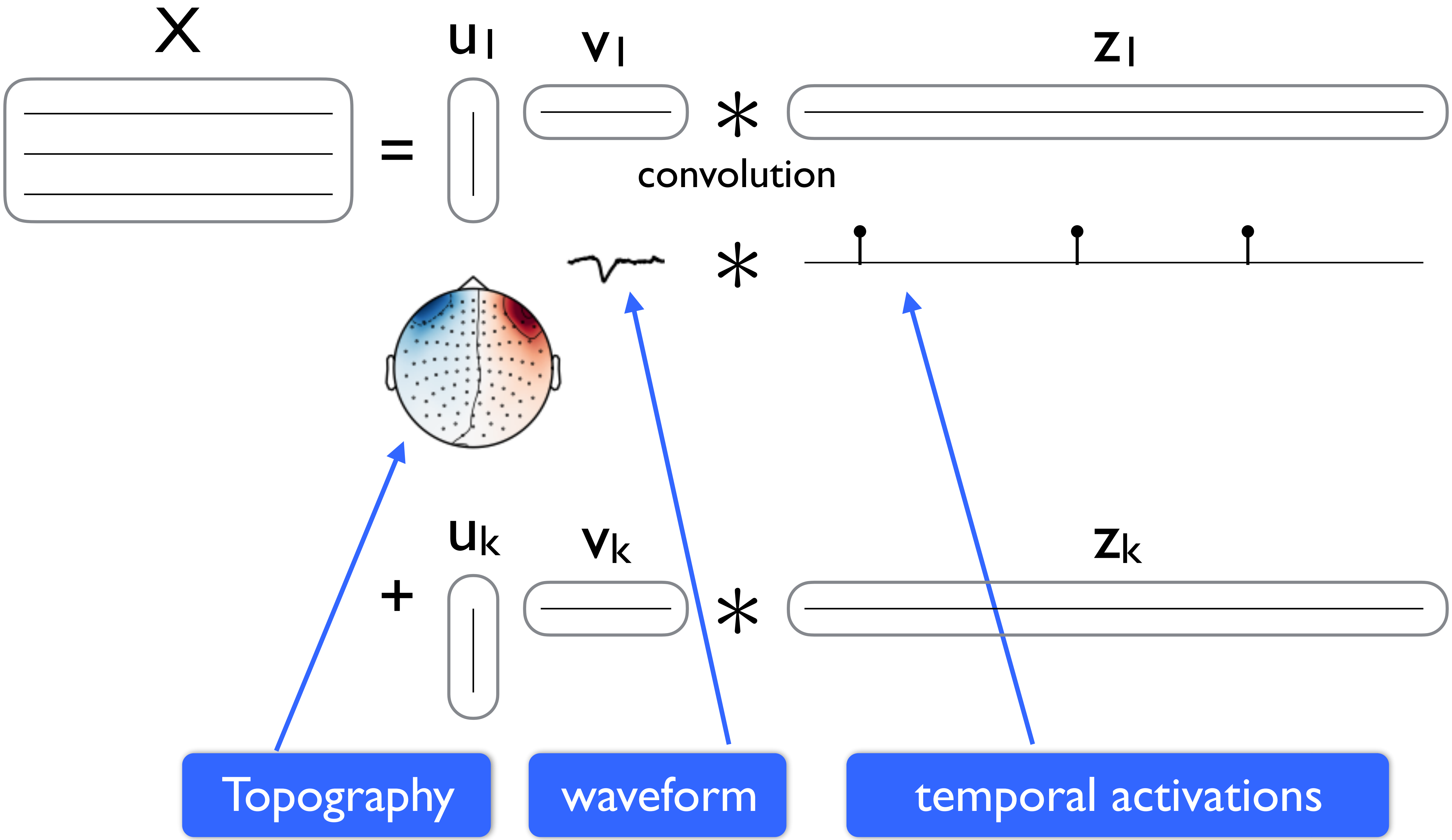
https://www.martinos.org/mne/stable/auto_tutorials/plot_artifacts_correction_ica.html

https://pierreablin.github.io/picard/auto_examples/plot_ica_eeg.html

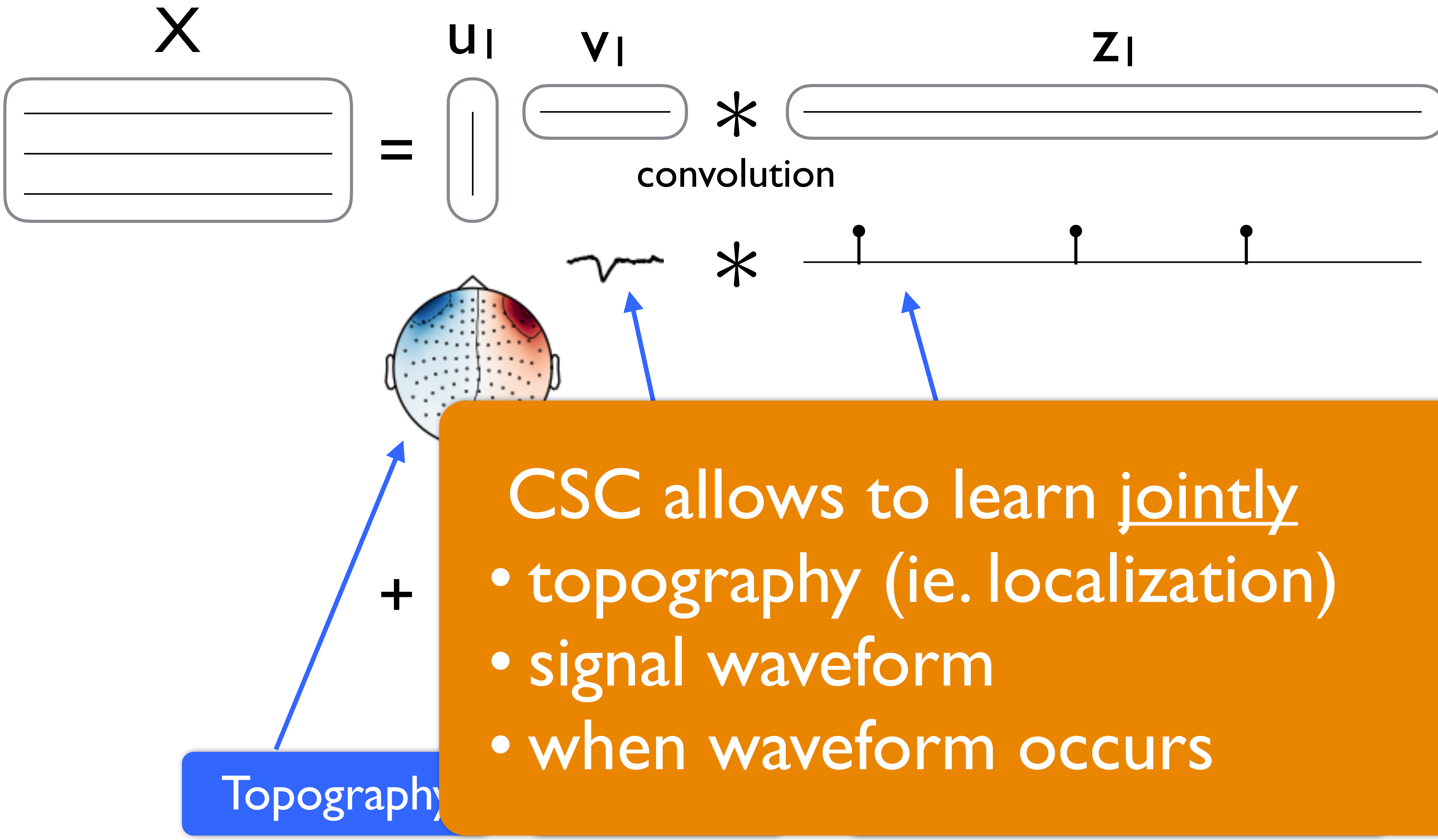
. . . to CSC



... to CSC


X
=
u_1
v_1
z_1
convolution
+
u_k
v_k
z_k
Topography
waveform
temporal activations

... to CSC



CSC allows to learn <u>jointly</u>

- topography (ie. localization)
- signal waveform
- when waveform occurs

Multivariate CSC

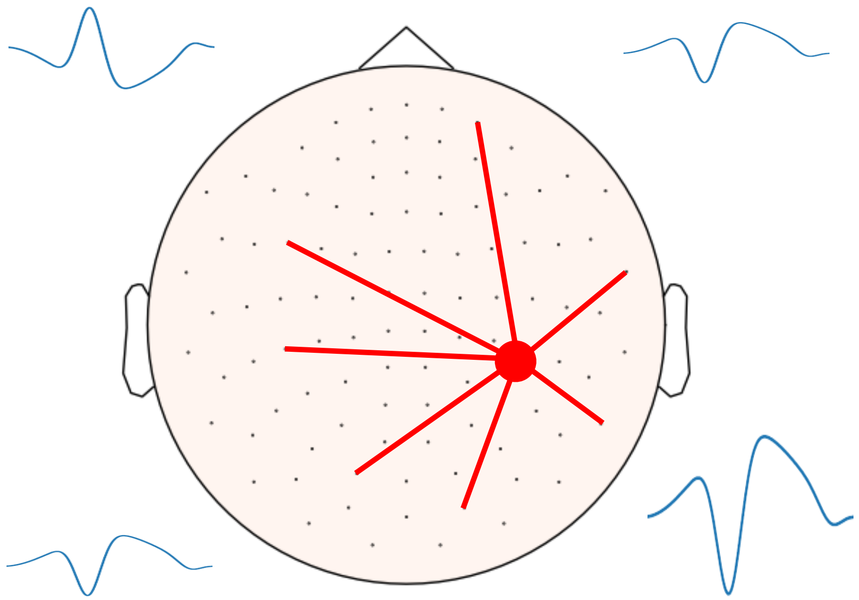
$$\min_{D,z} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 ,$$
$$\text{s.t.} \quad \|D_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0.$$

[Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.]

Multivariate CSC

$$\min_{D,z} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1,$$

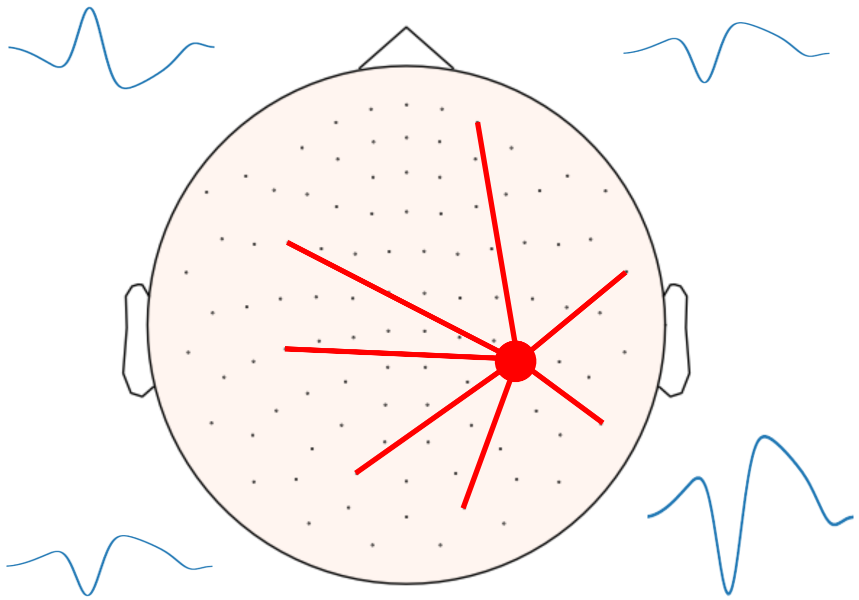
$$\text{s.t.} \quad \|D_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0.$$



[Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.]

Multivariate CSC

$$\min_{D,z} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1,$$
$$\text{s.t.} \quad \|D_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0.$$

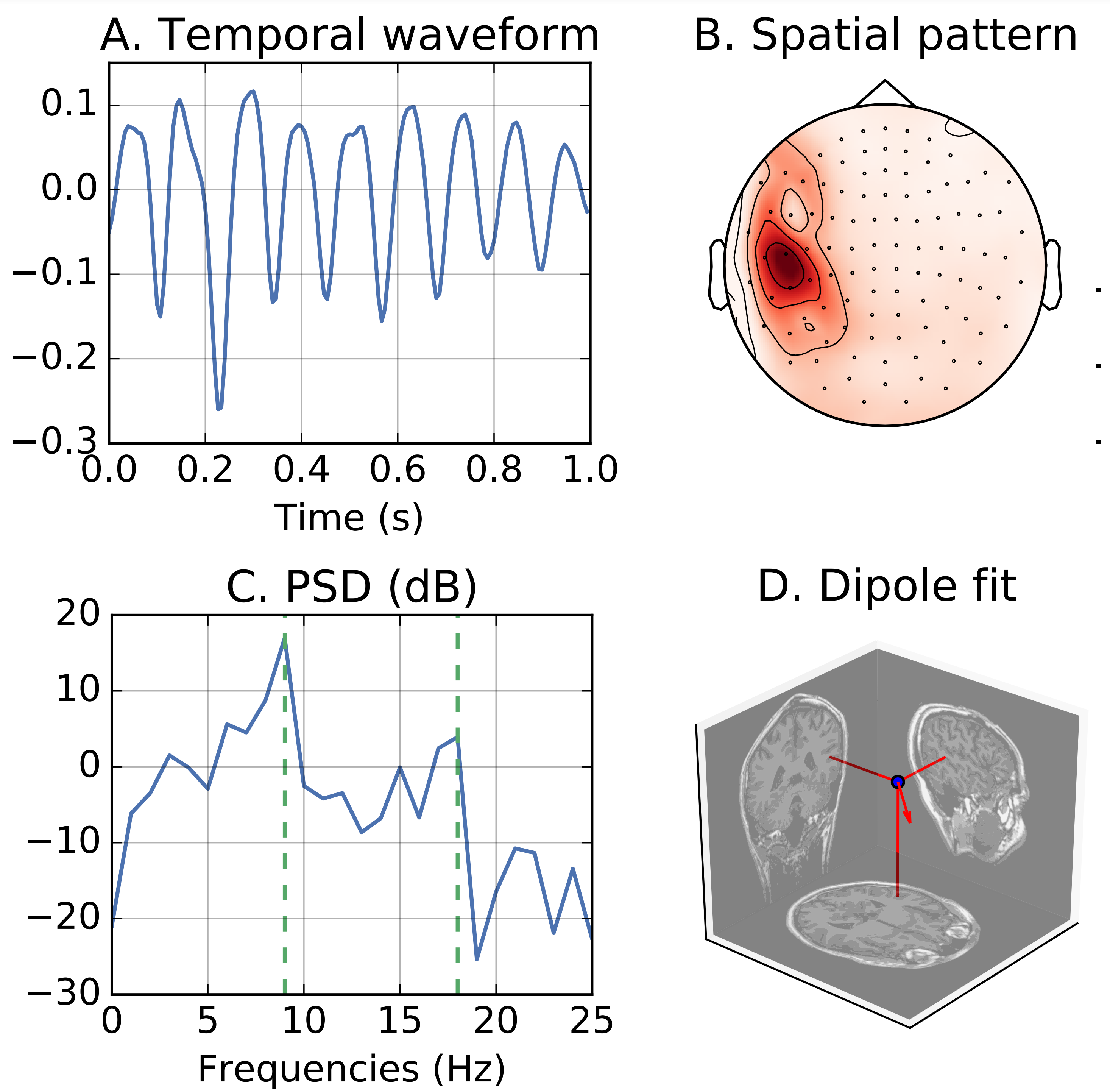


Rank 1 constraint:

$$\min_{u,v,z} \sum_{n=1}^{N} \frac{1}{2} \left\| X^n - \sum_{k=1}^{K} z_k^n * (u_k v_k^\top) \right\|_2^2 + \lambda \sum_{k=1}^{K} \|z_k^n\|_1 ,$$
$$\text{s.t.} \quad \|u_k\|_2^2 \leq 1 \;,\; \|v_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0.$$

[Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.]

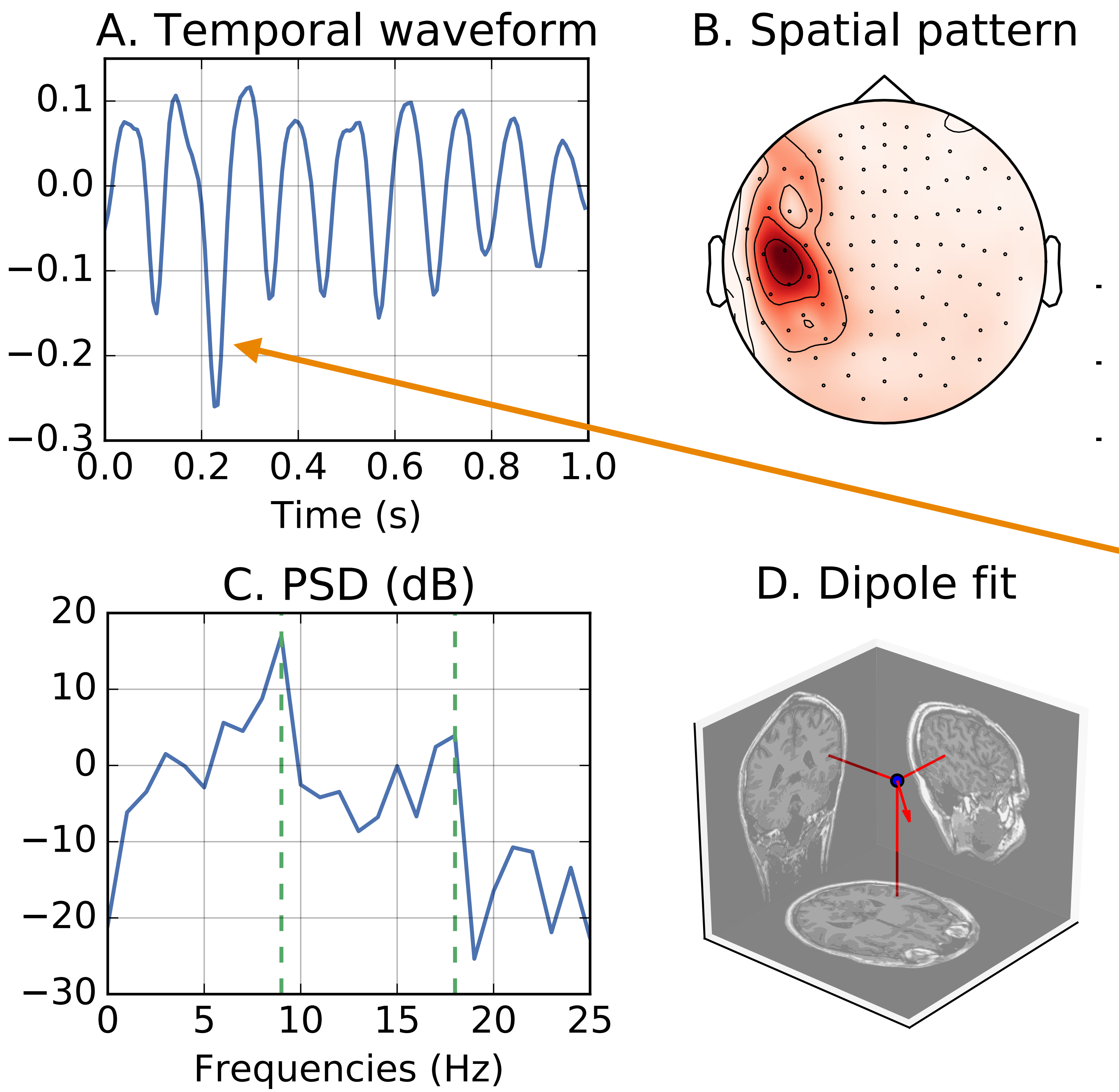
CSC on MEG



- MEG vectorview
- Median nerve stim.

[Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.]

CSC on MEG

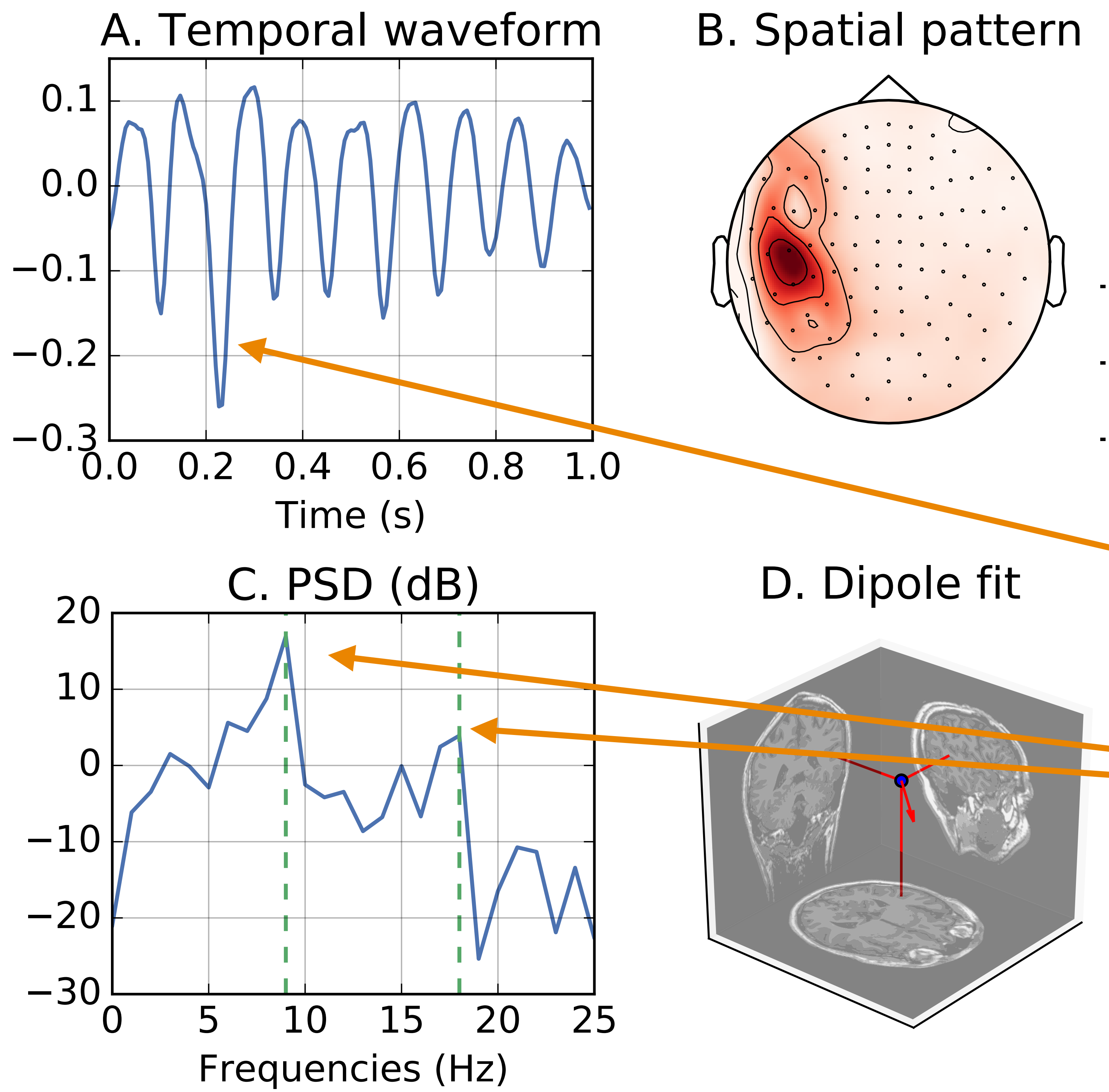


- MEG vectorview
- Median nerve stim.

CSC reveals mu-shaped waveforms

[Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.]

CSC on MEG



- MEG vectorview
- Median nerve stim.

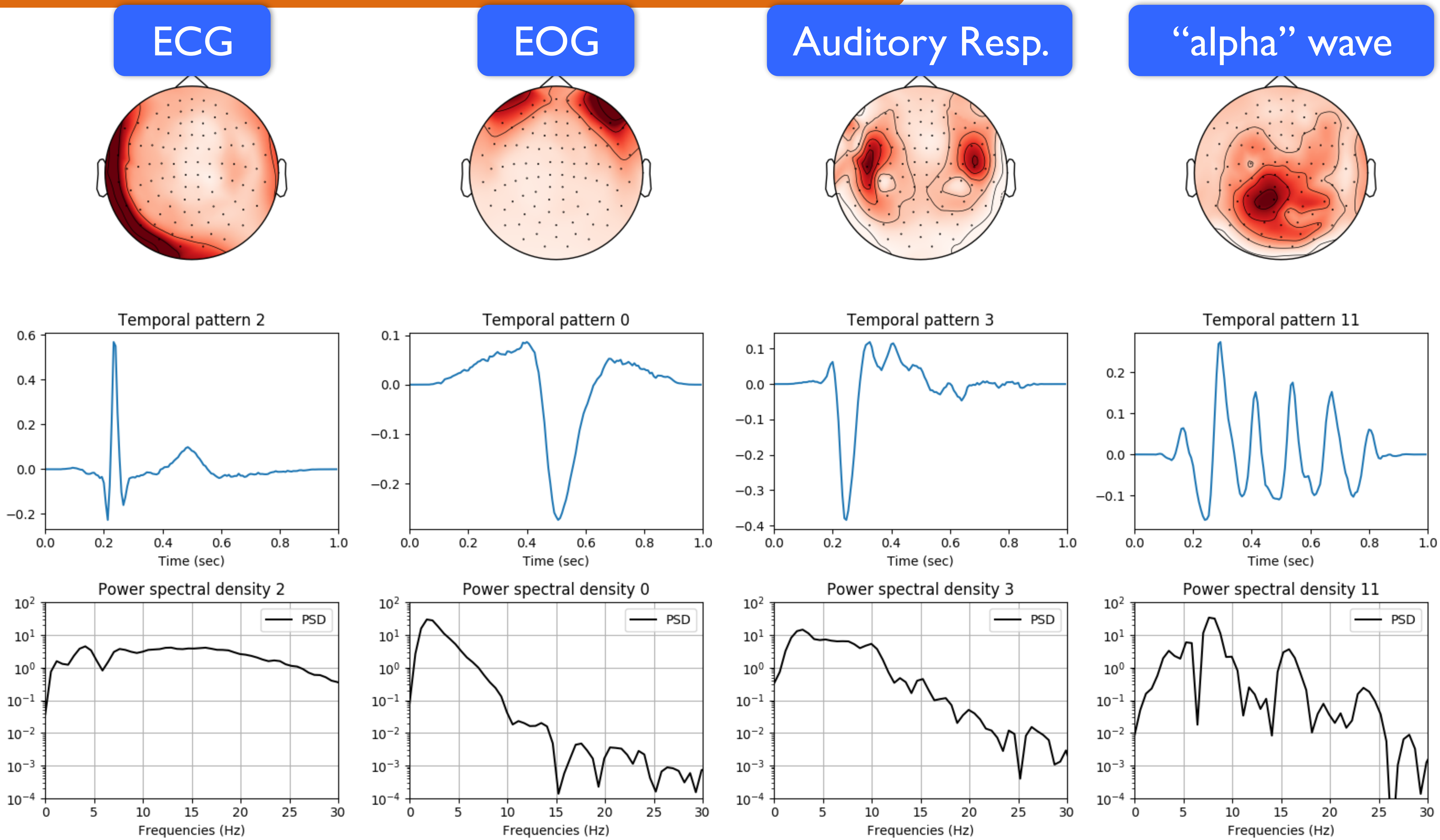
CSC reveals mu-shaped waveforms

See the frequency harmonics

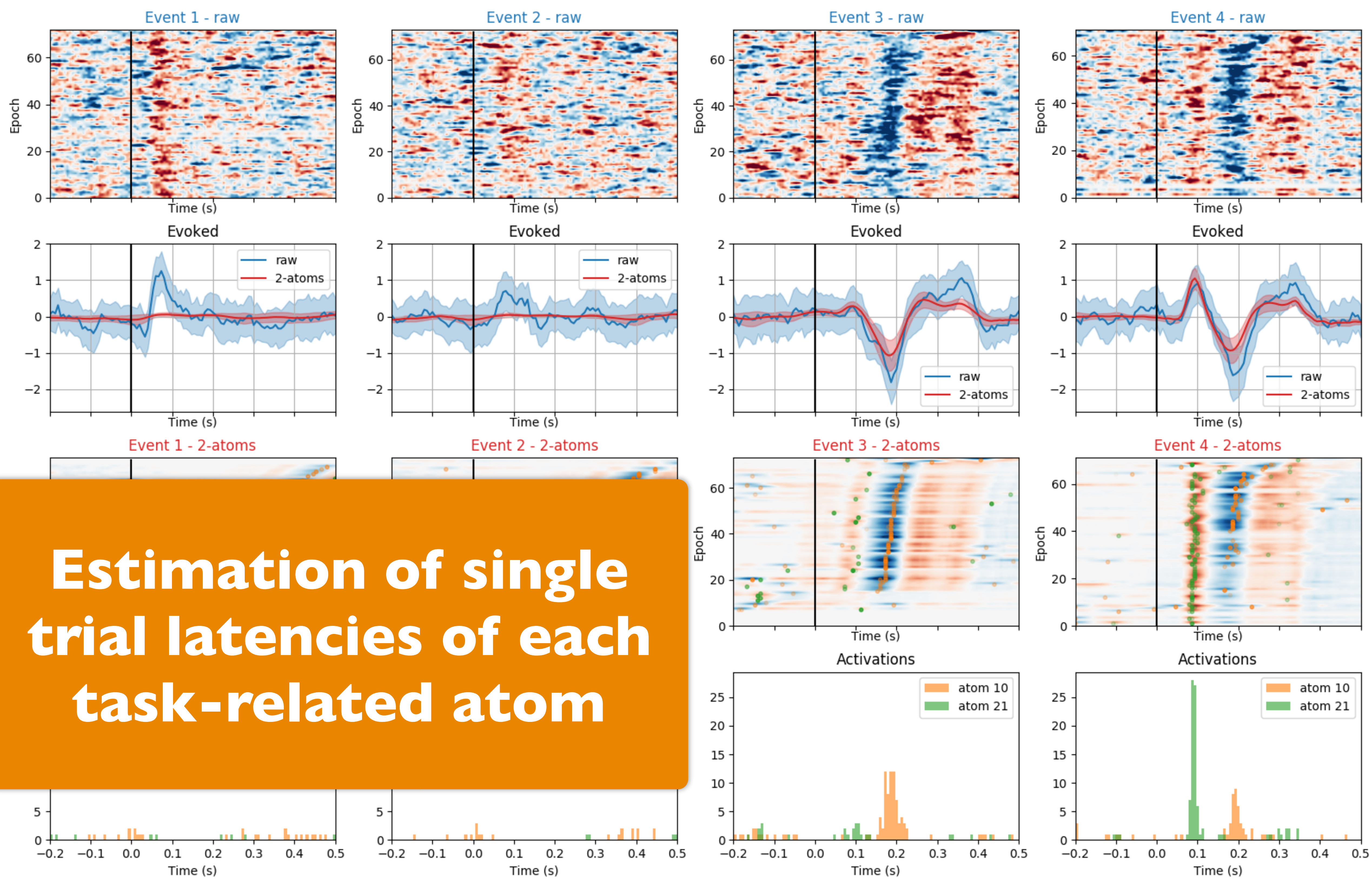
[Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.]

CSC on MEG

Results on auditory task:



Work in progress…



Estimation of single trial latencies of each task-related atom

https://github.com/alphacsc/alphacsc/pull/19

OPEN SCIENCE

alphaCSC: Convolution sparse coding for time-series

build passing codecov 81%

This is a library to perform shift-invariant sparse dictionary learning, also known as convolutional sparse coding (CSC), on time-series data. It includes a number of different models:

1. univariate CSC
2. multivariate CSC
3. multivariate CSC with a rank-1 constraint [1]
4. univariate CSC with an alpha-stable distribution [2]

A mathematical descriptions of these models is available in the documentation.

Installation

To install this package, the easiest way is using `pip`. It will install this package and its dependencies. The `setup.py` depends on `numpy` and `cython` for the installation so it is advised to install them beforehand. To install this package, please run

```
pip install numpy cython
pip install git+https://github.com/alphacsc/alphacsc.git#egg=alphacsc
```

If you do not have admin privileges on the computer, use the `--user` flag with `pip`. To upgrade, use the `--upgrade` flag provided by `pip`.

To check if everything worked fine, you can run:

```
python -c 'import alphacsc'
```

and it should not give any error messages.

https://alphacsc.github.io

Quickstart

Here is an example to present briefly the API:

```
import numpy as np
```

Extracting artifact and evoked response atoms from the sample dataset

This example illustrates how to learn rank1 atoms on the sample dataset from `mne`. We display a selection of atoms, featuring heartbeat and eyeblink artifacts, three atoms of evoked responses, and a non-sinusoidal oscillation.

```
# Authors: Thomas Moreau <thomas.moreau@inria.fr>
#          Mainak Jas <mainak.jas@telecom-paristech.fr>
#          Tom Dupre La Tour <tom.duprelatour@telecom-paristech.fr>
#          Alexandre Gramfort <alexandre.gramfort@telecom-paristech.fr>
#
# License: BSD (3-clause)
```

Let us first define the parameters of our model.

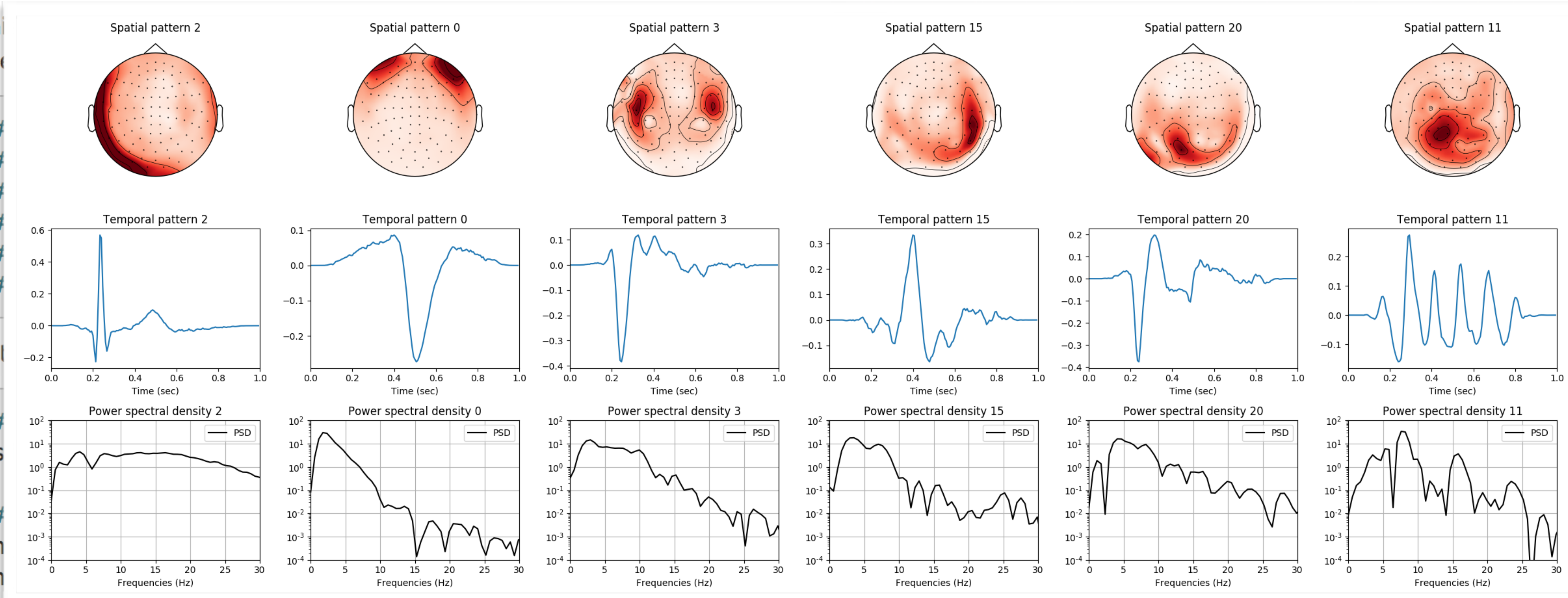
```
# sample frequency
sfreq = 150.

# Define the shape of the dictionary
n_atoms = 40
n_times_atom = int(round(sfreq * 1.0))  # 1000. ms

# Regularization parameter which control sparsity
reg = 0.1

# number of processors for parallel computing
n_jobs = 5
```

Next, we define the parameters for multivariate CSC

https://alphacsc.github.io/auto_examples/multicsc/plot_sample_evoked_response.html

Extracting artifact and evoked response atoms from the sample dataset

Th
ey

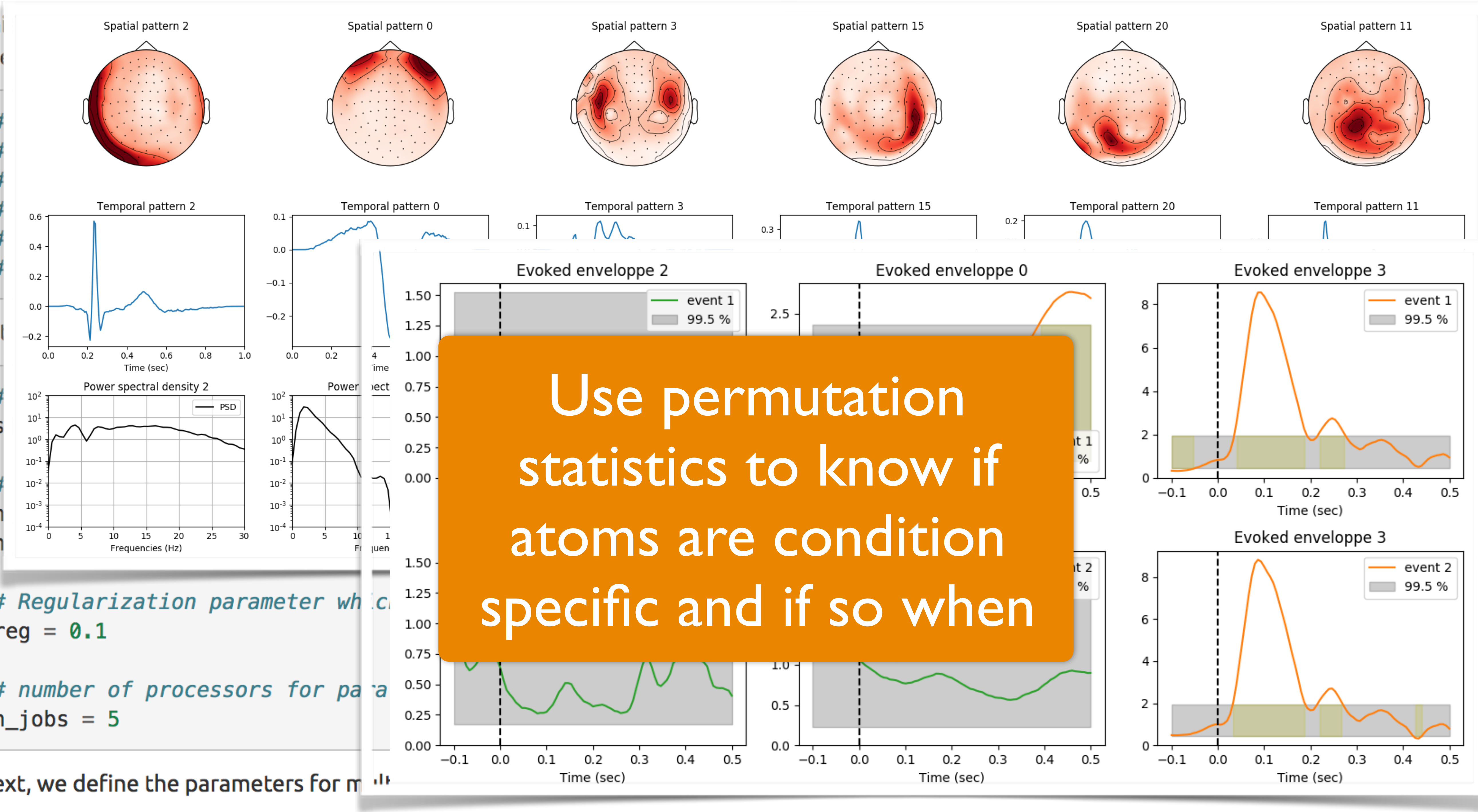



```
# Regularization parameter which control sparsity
reg = 0.1

# number of processors for parallel computing
n_jobs = 5
```

Next, we define the parameters for multivariate CSC

https://alphacsc.github.io/auto_examples/multicsc/plot_sample_evoked_response.html

Extracting artifact and evoked response atoms from the sample dataset



Use permutation statistics to know if atoms are condition specific and if so when

```
# Regularization parameter whic
reg = 0.1

# number of processors for para
n_jobs = 5
```

Next, we define the parameters for m

https://alphacsc.github.io/auto_examples/multicsc/plot_sample_evoked_response.html

https://pypi.org/project/alphacsc/

Search projects

Help

alphacsc 0.3

```
pip install alphacsc
```

Convolutional dictionary learning for noisy signals

Navigation

- Project description
- Release history
- Download files

Project description

build passing | codecov 81%

This is a library to perform shift-invariant sparse dictionary learning, also kno
coding (CSC), on time-series data. It includes a number of different models:

1. univariate CSC
2. multivariate CSC

Picard

This is a library to run the Preconditioned ICA for Real Data (PICARD) algorithm [1] and its orthogonal version (PICARD-O) [2]. These algorithms show fast convergence even on real data for which sources independence do not perfectly hold.

Installation

We recommend the Anaconda Python distribution. Otherwise, to install `picard`, you first need to install its dependencies:

```
$ pip install numpy matplotlib numexpr scipy
```

Then install Picard:

```
$ pip install python-picard
```

If you do not have admin privileges on the computer, use the `--user` flag with *pip*. To upgrade, use the `--upgrade` flag provided by *pip*.

To check if everything worked fine, you can do:

```
$ python -c 'import picard'
```

and it should not give any error message.

https://pierreablin.github.io/picard/

picard 0.4.dev0 Examples API GitHub Site Page Source Search

Picard

This is a library to run the Precon
fast convergence even on real da

Installation

We recommend the Anaconda Py

```
$ pip install numpy matpl
```

Then install Picard:

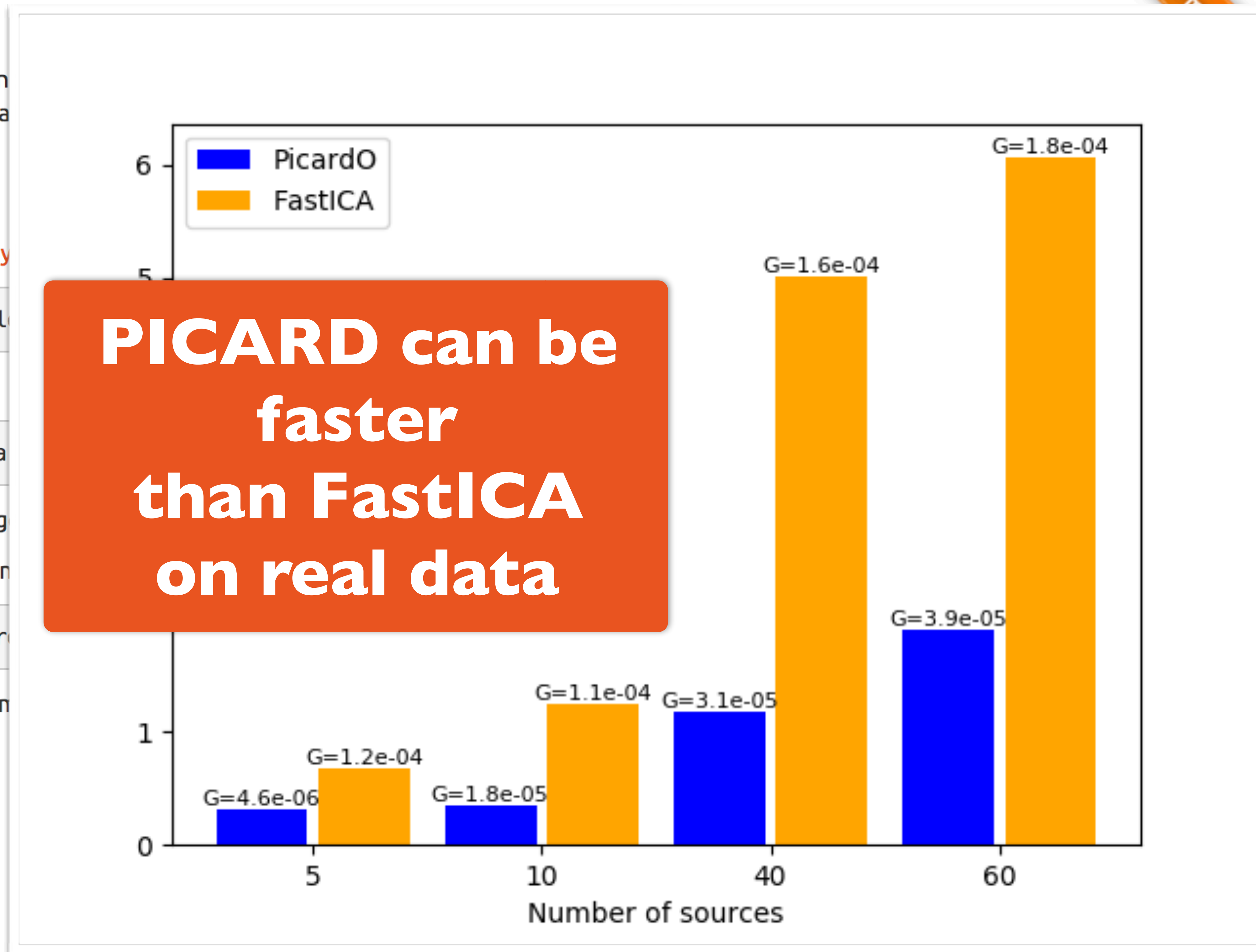
```
$ pip install python-pica
```

If you do not have admin privileg

To check if everything worked fin

```
$ python -c 'import picar
```

and it should not give any error n



PICARD can be faster than FastICA on real data

https://pierreablin.github.io/picard/

Conclusion

- Maybe frequency “bands” are too simplistic for clinical or cognitive neuroscience?
- CSC could be used for localizing epileptic spikes?
- Try it, break it and let us know how to fix it!

Joint work with:

Tom Dupré la Tour *Thomas Moreau*
Mainak Jas *Umut Simsekli*
Pierre Ablin *J-F Cardoso*

Thanks !

T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), Proc. NeurIPS Conf.

M. Jas, T. Dupré la Tour, U. Simsekli, A. Gramfort, Learning the Morphology of Brain Signals Using Alpha-Stable Convolutional Sparse Coding, (2017), Proc. NeurIPS Conf.

P. Ablin, J.-F. Cardoso & A. Gramfort Faster independent component analysis by preconditioning with Hessian approximations, (2017) IEEE Trans. Signal Processing

Come see poster W828

Contact

http://alexandre.gramfort.net

GitHub : @agramfort *Twitter : @agramfort*

Support

ERC SLAB, ANR THALAMEEG ANR-14-NEUC-0002-01
NIH R01 MH106174, DFG HA 2899/21-1.