

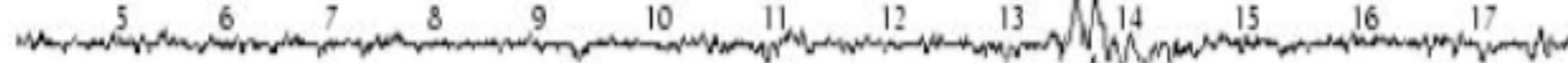
Optimization meets machine learning for neuroimaging

Alexandre Gramfort
<http://alexandre.gramfort.net>

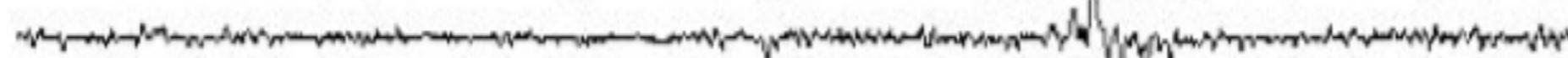


Electroencephalography (EEG))

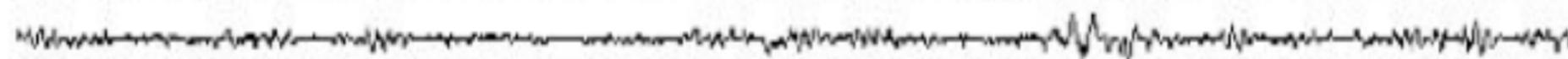
FP1-Ref



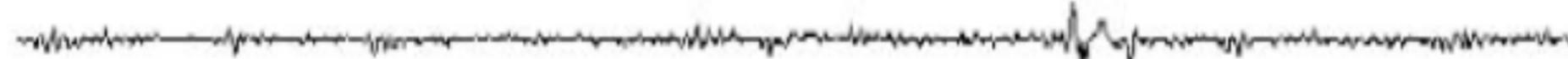
F3-Ref



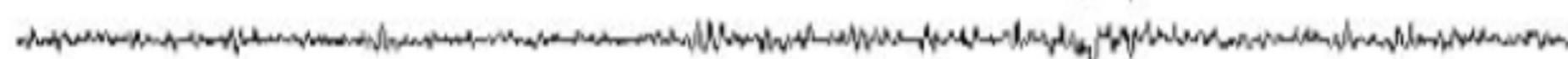
C3-Ref



P3-Ref



O1-Ref



F7-Ref



T3-Ref



T5-Ref



F9-Ref



TP9-Ref



FP2-Ref



F4-Ref



C4-Ref



P4-Ref



O2-Ref



F8-Ref



T4-Ref



T6-Ref



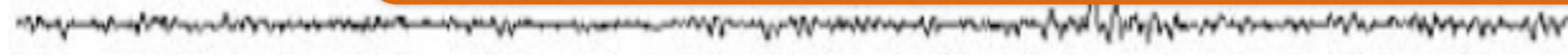
F10-Ref



TP10-Ref



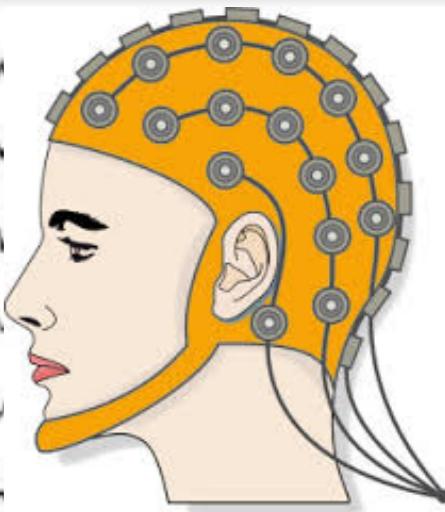
Fz-Ref



Data are multivariate
time series

Physics of acquisition well
understood thanks to Maxwell

Data are a linear instantaneous
mixture of electric neural sources



Question 1: What can unsupervised machine learning tell us about such data (the brain)?

Question 1: What can unsupervised machine learning tell us about such data (the brain)?

Question 2: Can we use the known physics to find adequate models?

Question 1: What can unsupervised machine learning tell us about such data (the brain)?

Question 2: Can we use the known physics to find adequate models?

Question 3: Can we come up with efficient learning/optimization algorithms?

Linear ICA model

- **Assumption:** Observed signals are a linear mix of independent identically distributed signals.

$$\begin{array}{c} \boxed{\text{---}} \\ \boxed{\text{---}} \\ \boxed{\text{---}} \end{array} = \begin{array}{c} | \\ | \\ | \end{array} \begin{array}{c} \boxed{\text{---}} \\ \boxed{\text{---}} \\ \boxed{\text{---}} \end{array}$$

\mathbf{X} \mathbf{A} \mathbf{S}

- N : Number of signals
- T : Number of samples
- \mathbf{X} : Observed signals. Size $N \times T$
- \mathbf{S} : Independent **sources** signals. Size $N \times T$

Both \mathbf{A} and \mathbf{S} are
unknown

[Jutten & Herault 91]

ICA vs Dictionary Learning

- **ICA:**

$$X = AS$$

Uses independence
assumption

- **Dictionary learning:**

$$\min_{A,S} \|X - AS\|_F^2 + \lambda \|S\|_p \quad p \leq 1$$

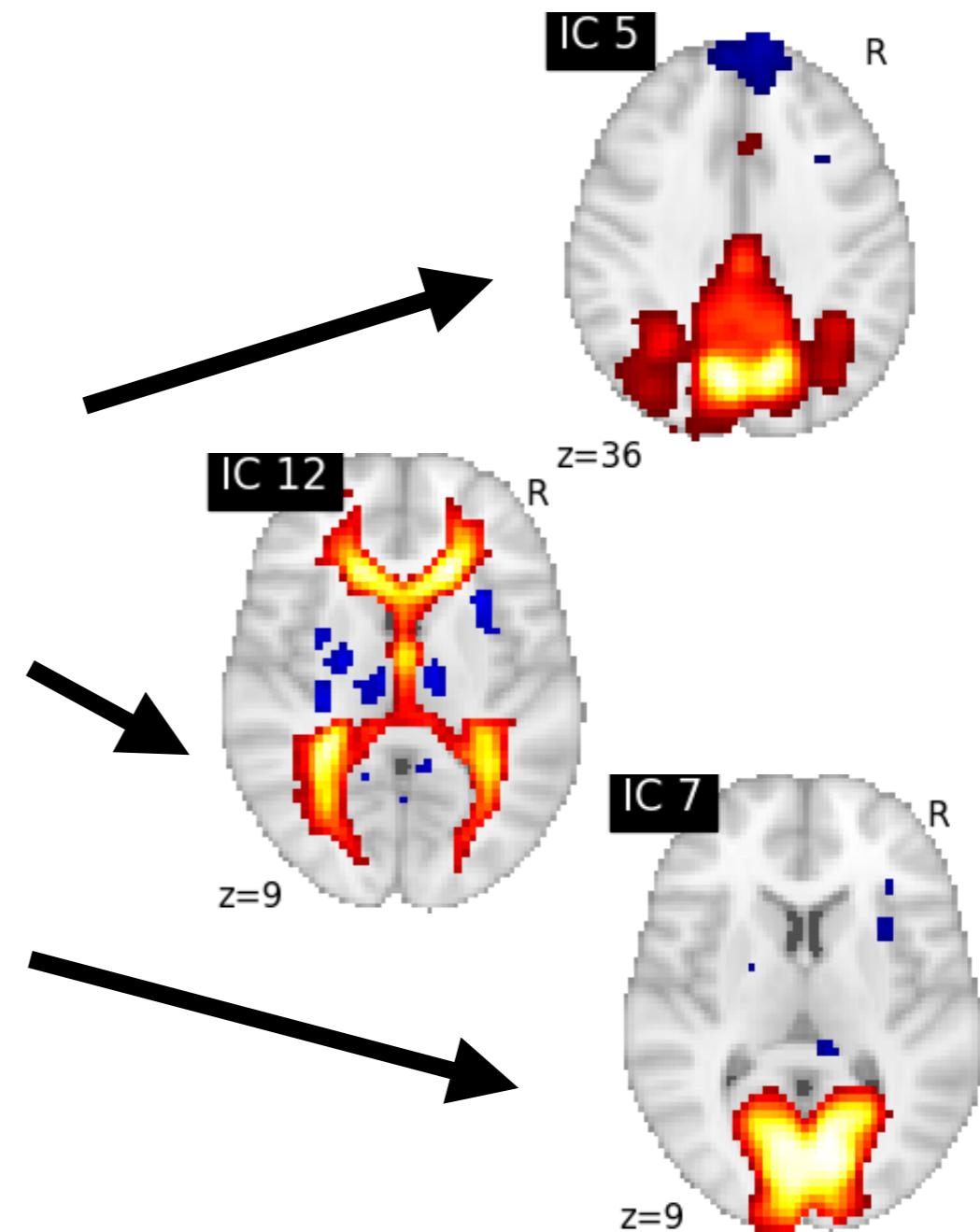
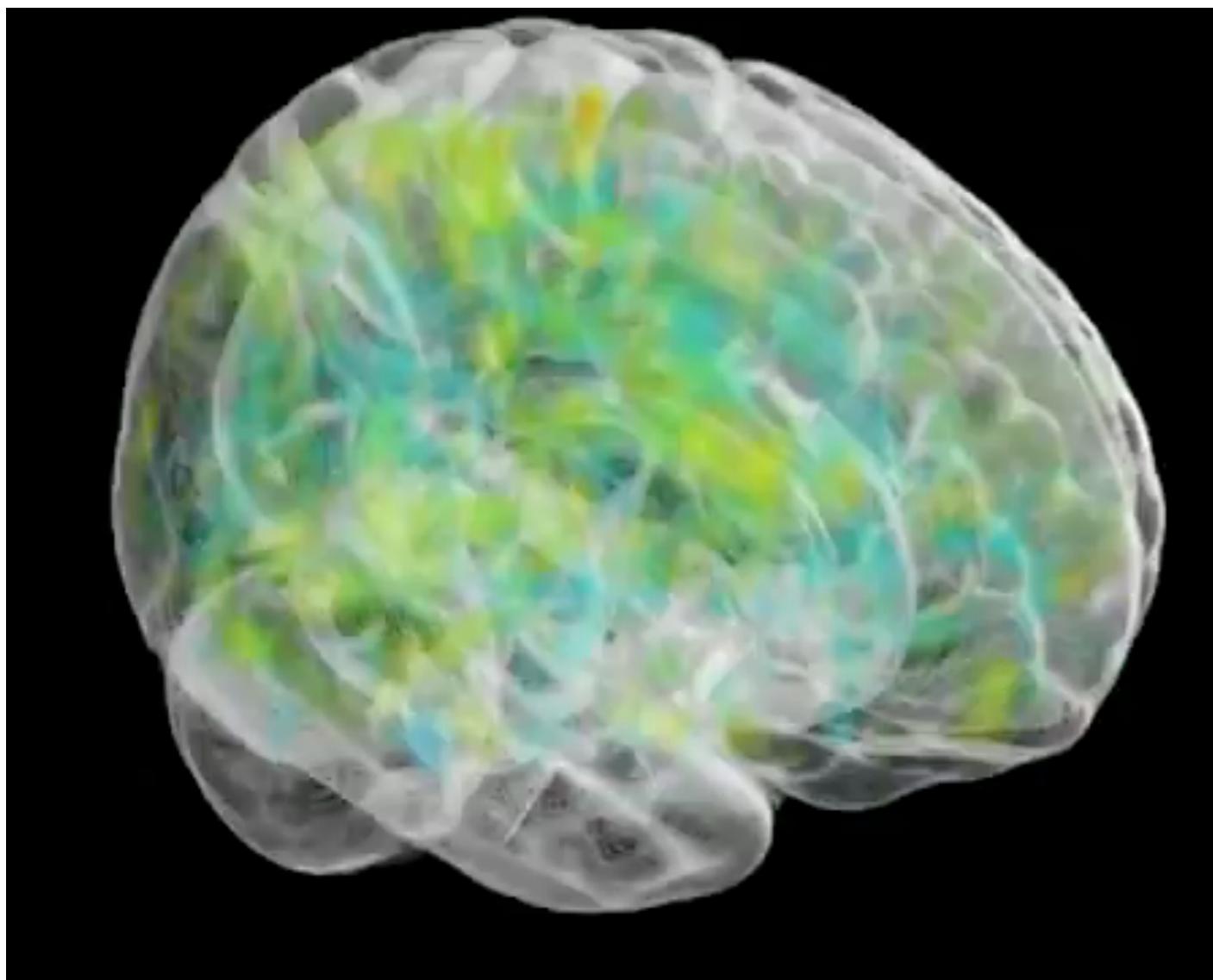
$$\text{s.t. } \|a_j\| \leq 1$$

Uses sparsity
assumption

[Olshausen et al., Elad et al., Mairal et al. etc...]

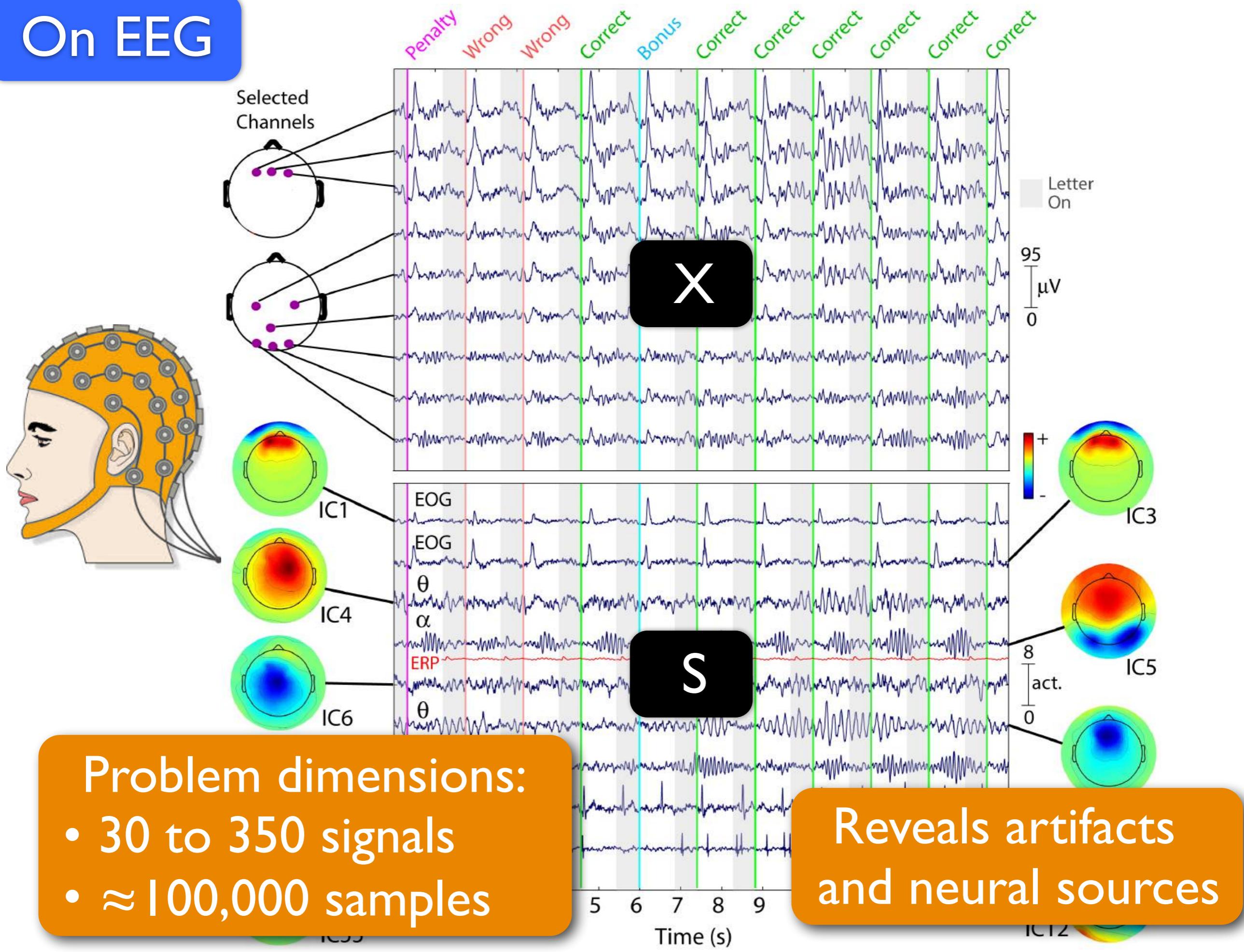
ICA on fMRI data

- Used in neuroimaging to find “brain networks” (not neural networks...) in functional MRI data



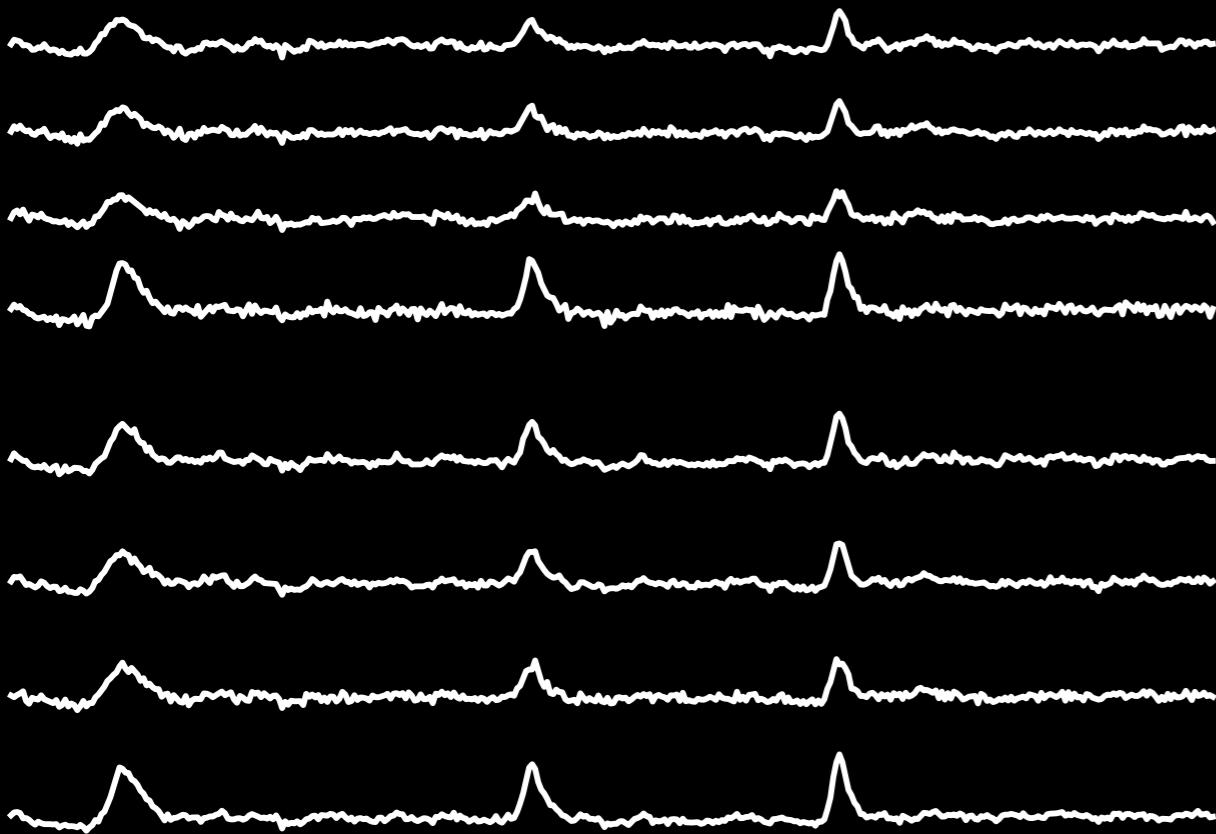
Source: http://nilearn.github.io/auto_examples/03_connectivity/plot_cica_resting_state.html

On EEG

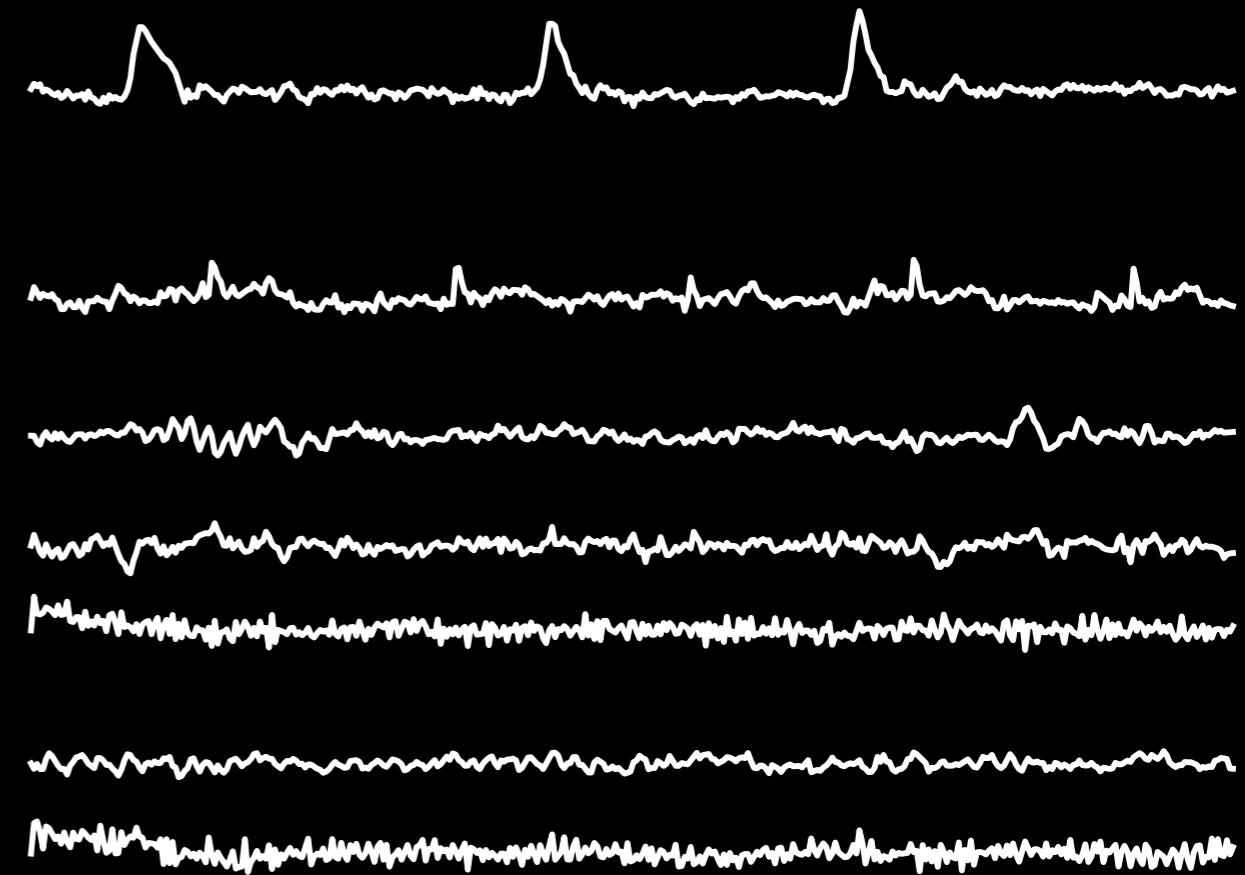


How to accelerate Independent Component Analysis (ICA) solvers?

Observations (raw EEG)



ICA recovered sources



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

[EM algorithms for ICA, P.Ablin, A. Gramfort, J.F. Cardoso and F. Bach, AISTATS 2019]

Code: <https://pierreablin.github.io/picard>

Statistical Principles of ICA

$$X = AS$$

Objective:

Find W s.t. $Y = WX$ has maximally independent rows.

Note:

If more than 2 Gaussian sources, ICA problem is not possible.

[Comon 94]

Remark:

Different ways to quantify independence lead to different methods.

Maximum Likelihood Formulation

- **Model:** $Y = WX$ has independent rows

$$Y = [y_1, \dots, y_N]^\top \quad p_i \text{ p.d.f. of } y_i$$

- Independence:

$$p(Y(t)) = p_1(y_1(t))p_2(y_2(t)) \dots p_N(y_N(t))$$

- Likelihood of a time sample of X :

$$p(X(t)) = |\det(W)|p_1(y_1(t))p_2(y_2(t)) \dots p_N(y_N(t))$$

Maximum Likelihood Formulation

ICA = minimizing the averaged negative log-likelihood:

$$\mathcal{L}(W) = -\log|\det(W)| - \sum_{i=1}^N E_x[\log(p_i([Wx]_i))]$$

[Pham & Garat 1997]

Maximum Likelihood Formulation

ICA = minimizing the averaged negative log-likelihood:

$$\mathcal{L}(W) = -\log|\det(W)| - \sum_{i=1}^N E_x[\log(p_i([Wx]_i))]$$

[Pham & Garat 1997]

ICA boils down to minimizing an objective function

Maximum Likelihood Formulation

ICA = minimizing the averaged negative log-likelihood:

$$\mathcal{L}(W) = -\log|\det(W)| - \sum_{i=1}^N E_x[\log(p_i([Wx]_i))]$$

[Pham & Garat 1997]

ICA boils down to minimizing an objective function

- Infomax model [Bell & Sejnowski 1995]

$$\psi_i(\cdot) = -\log(p_i(\cdot))' = \tanh(\cdot/2)$$

Maximum Likelihood Formulation

ICA = minimizing the averaged negative log-likelihood:

$$\mathcal{L}(W) = -\log|\det(W)| - \sum_{i=1}^N E_x[\log(p_i([Wx]_i))]$$

[Pham & Garat 1997]

ICA boils down to minimizing an objective function

- Infomax model [Bell & Sejnowski 1995]

$$\psi_i(\cdot) = -\log(p_i(\cdot))' = \tanh(\cdot/2)$$

Most widely used ICA technique in neuroscience

Geometry of the problem

- non-convex (multiple minima)
- Optimization on the invertible matrices manifold
- Use of **relative changes** in solver iterations:

Absolute

$$W_{n+1} = W_n + dW$$

Relative

$$W_{n+1} = (I + \mathcal{E})W_n$$

n: iteration number

Relative gradient / Hessian

Relative matrix form Taylor expansion:

$$\mathcal{L}((I + \mathcal{E})W) = \mathcal{L}(W) + \langle G|\mathcal{E} \rangle + \frac{1}{2}\langle \mathcal{E}|H|\mathcal{E} \rangle + \mathcal{O}(\|\mathcal{E}\|^3)$$

- Gradient is a $N \times N$ matrix:

$$G_{ij} = E[\psi_i(y_i)y_j] - \delta_{ij}$$

closed form
solutions

- Hessian is a $N \times N \times N \times N$ 4th order tensor.

$$H_{ijkl} = \delta_{il}\delta_{jk} + \delta_{ik}E[\psi'_i(y_i)y_jy_l]$$

[Cardoso & Laheld 1996]

(relative) Newton method

$$H_{ijkl} = \delta_{il}\delta_{jk} + \delta_{ik}E[\psi'_i(y_i)y_jy_l]$$

One iteration:

$$W_{n+1} = (I - H^{-1}G)W_n$$

Problem:

Large linear system / not positive (regularization needed)

Newton method is possible but not practical

(relative) Newton method

$$H_{ijkl} = \delta_{il}\delta_{jk} + \delta_{ik}E[\psi'_i(y_i)y_jy_l]$$

One iteration:

$$W_{n+1} = (I - H^{-1}G)W_n$$

Problem:

Large linear system / not positive (regularization needed)

Newton method is possible but not practical

but if model holds:

$$\delta_{ik}E[\psi'_i(y_i)y_jy_l] = \delta_{ik}\delta_{jl}E[\psi'_i(y_i)]E[y_j^2]$$

Hessian approximations

True hessian:

$$H_{ijkl} = \delta_{il}\delta_{jk} + \delta_{ik}E[\psi'_i(y_i)y_jy_l]$$

- If we **suppose that the signals are independent**
- Two Hessian approximations :

$$H_{ijkl}^2 = \delta_{il}\delta_{jk} + \delta_{ik}\delta_{jl}E[\psi'_i(y_i)y_j^2]$$

Complexity:

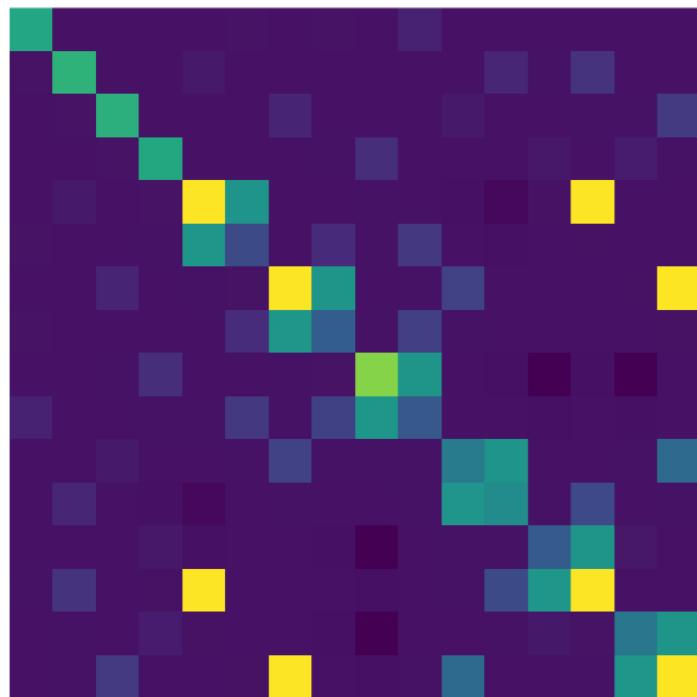
$O(T N^2)$

$$H_{ijkl}^1 = \delta_{il}\delta_{jk} + \delta_{ik}\delta_{jl}E[\psi'_i(y_i)]E[y_j^2]$$

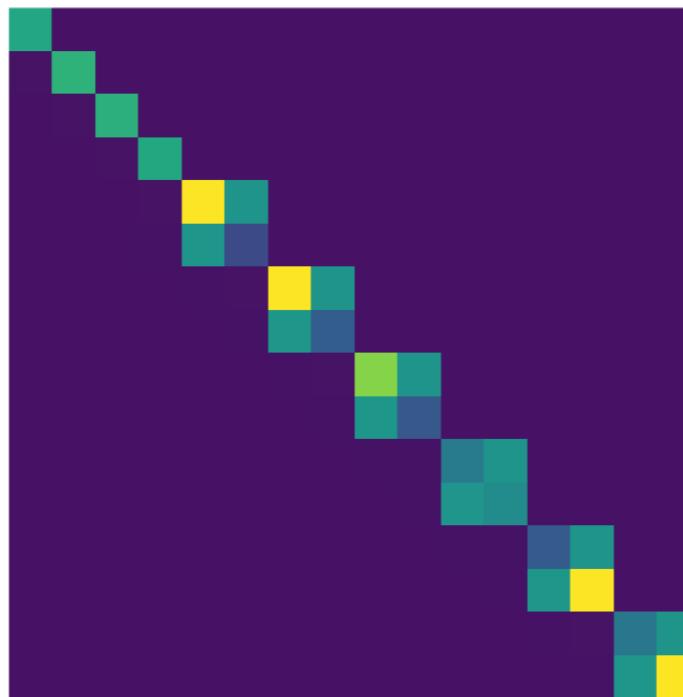
$O(T N)$

Hessian approximations (cont.)

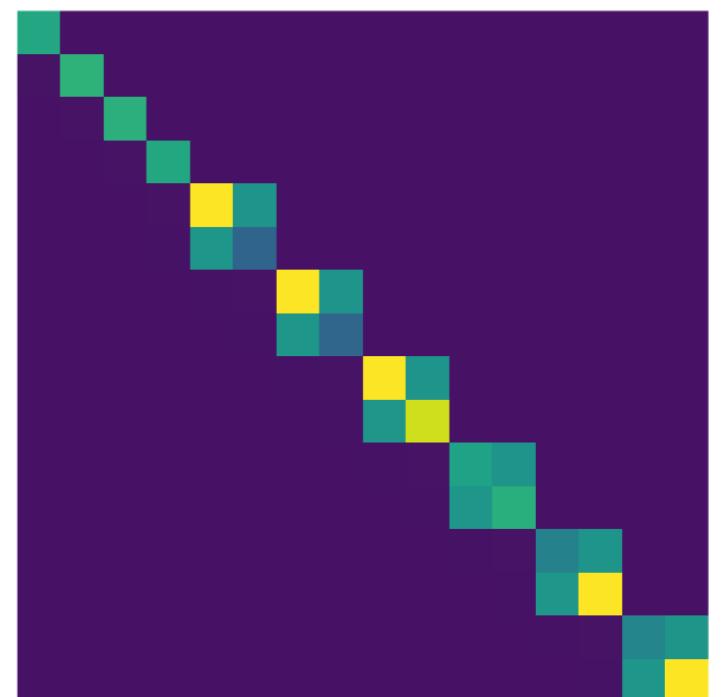
H



H^2



H^1



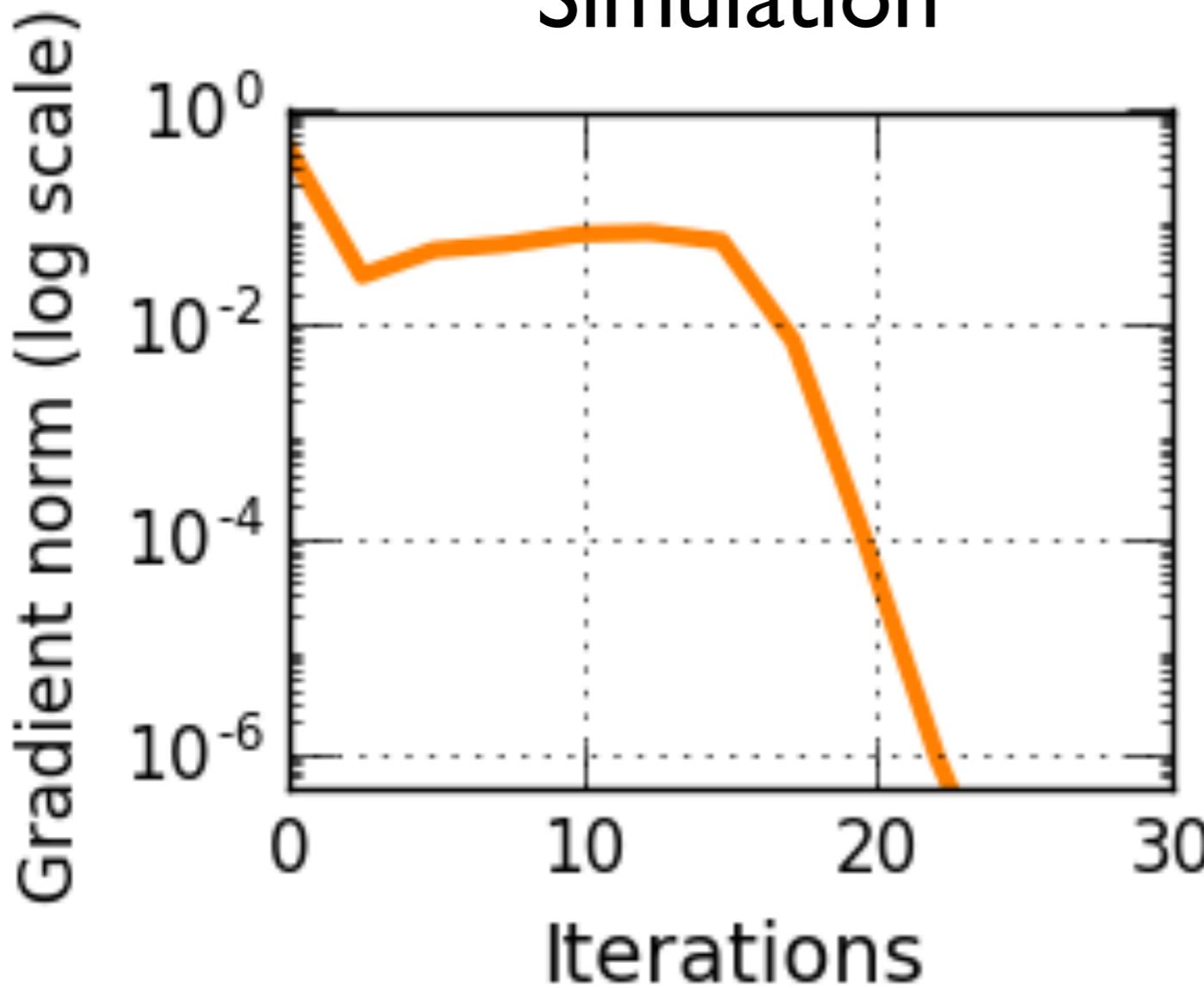
- Hessians approximations are **block-diagonal** hence **much easier to invert**
- **Idea:**

Do Newton with the approximated Hessian !

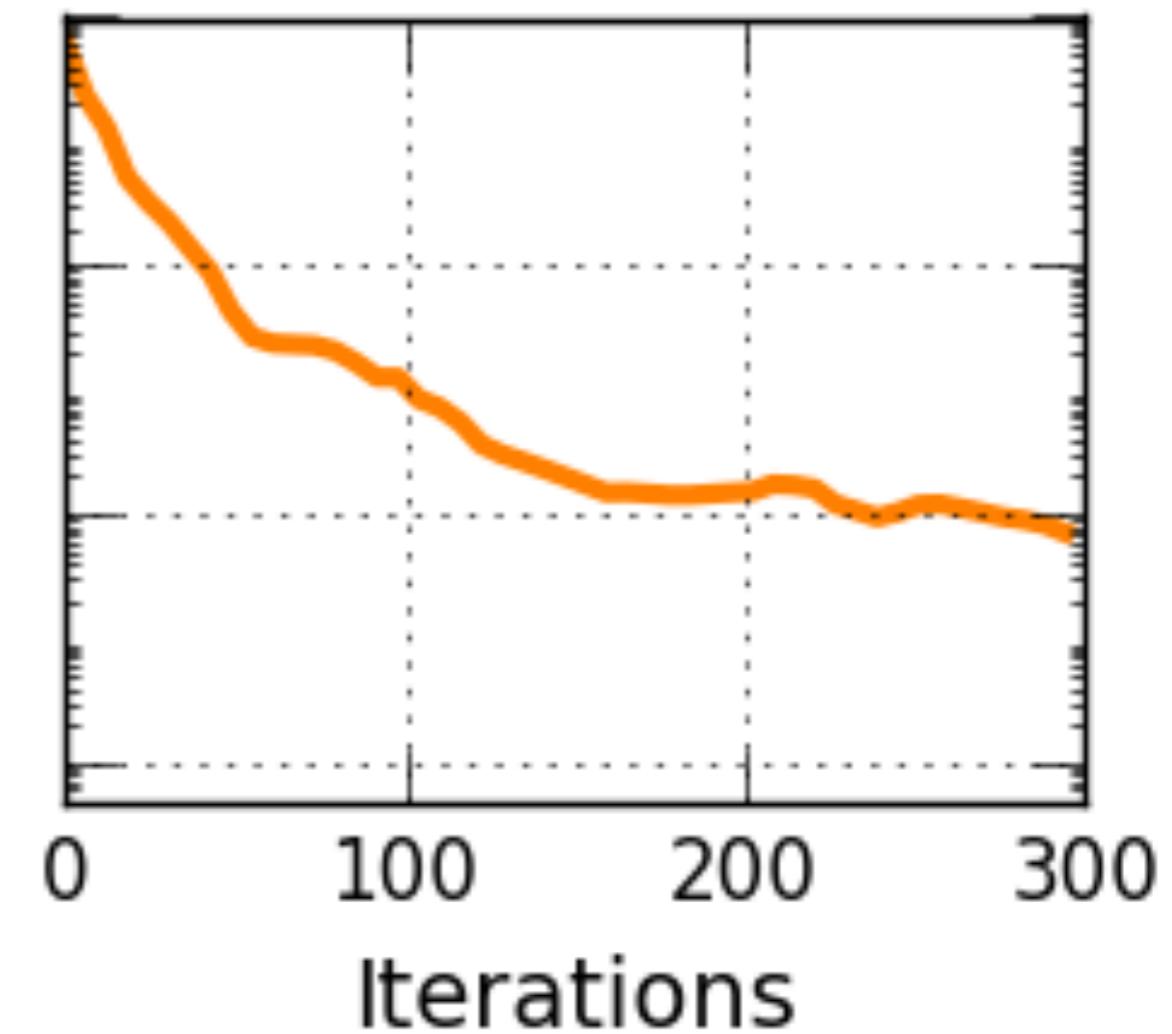


But...

Simulation

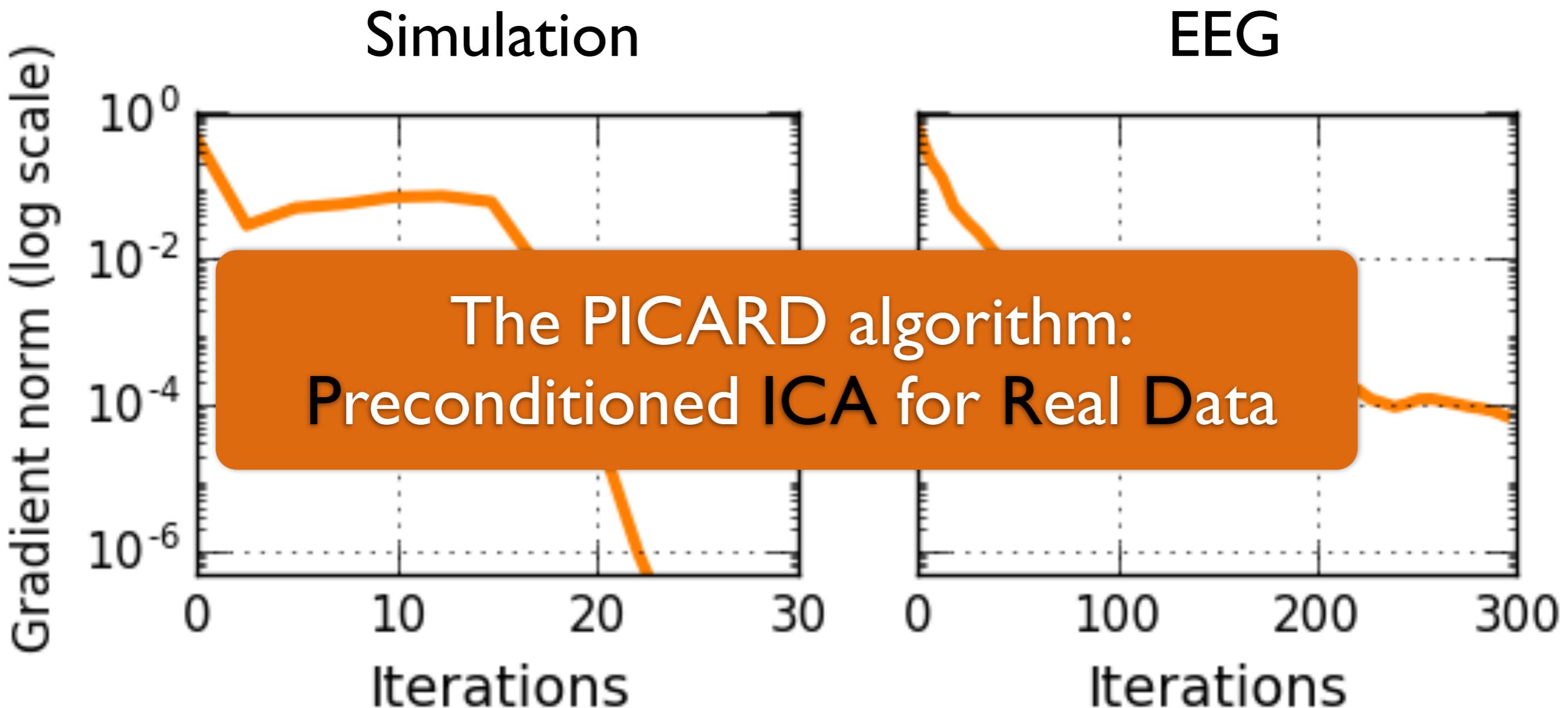


EEG



It does not work with real data !

But...



It does not work with real data !

Picard relies on L-BFGS algorithm

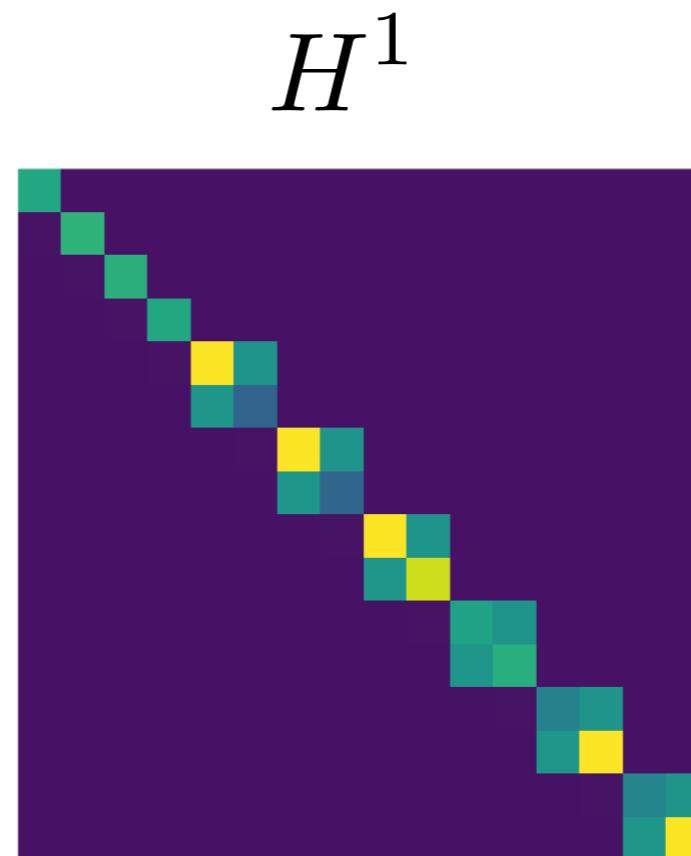
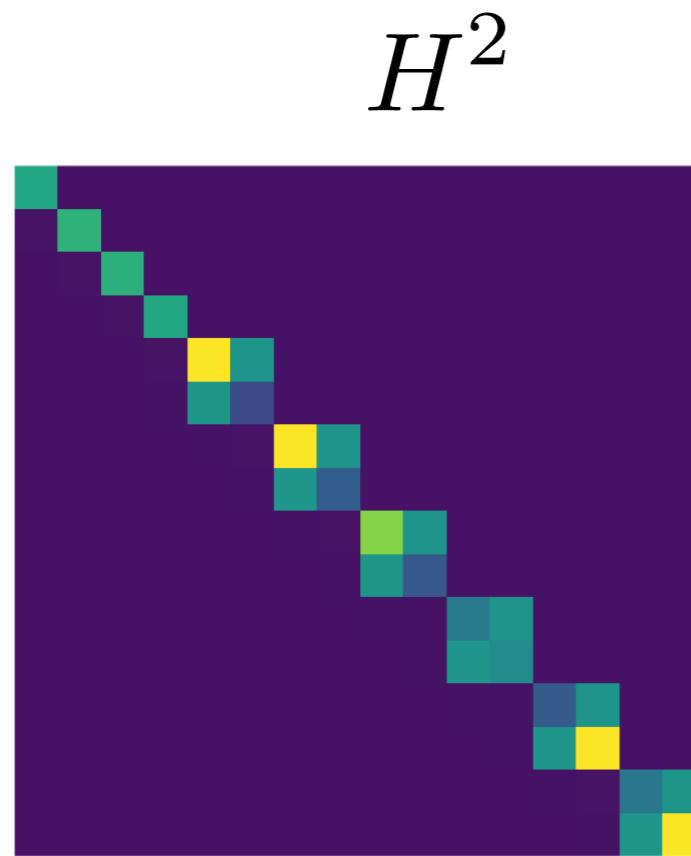
- BFGS builds an approximation of the Hessian using only the past function and gradient evaluations
- Does low-rank corrections of an initial guess $H^{(0)}$ of the Hessian

$$H^{(q)} = \boxed{\text{I}} \boxed{-} + \boxed{\text{I}} \boxed{-} + H^{(q-1)} \text{ (rank-2 updates)}$$

- Initial guess $H^{(0)}$ should be easy to invert and is commonly a multiple of identity

[Liu, D. C., & Nocedal, J. « On the limited memory BFGS method for large scale optimization. » *Mathematical programming*, 1989]

L-BFGS algorithm with Hessian approx.



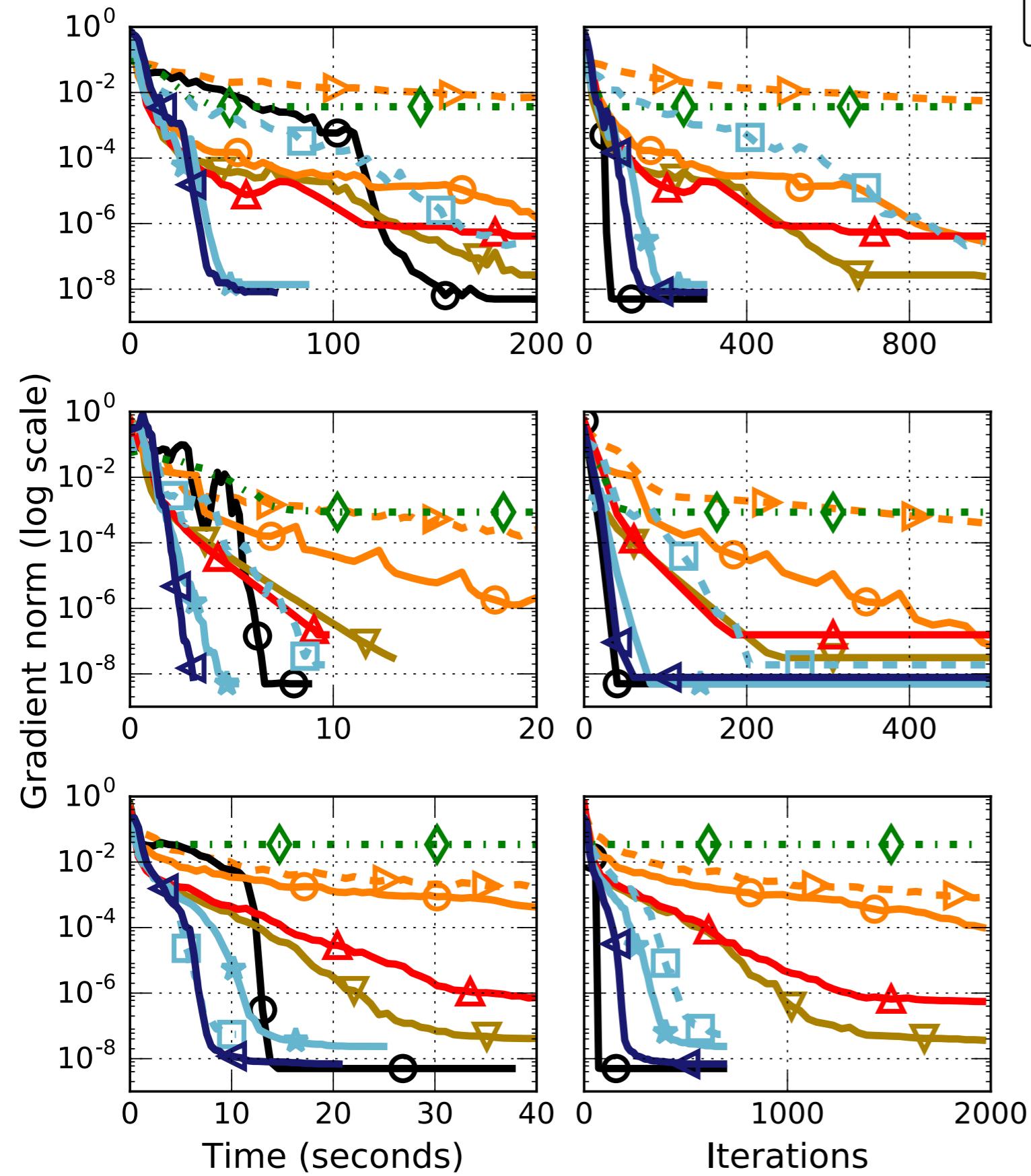
Use these matrices as $H^{(0)}$ in L-BFGS

[P.Ablin, J.-F.Cardoso, A. Gramfort,
Faster ICA by preconditioning with Hessian approximations, IEEE Trans. Signal Processing]

What are other state-of-the art solvers?

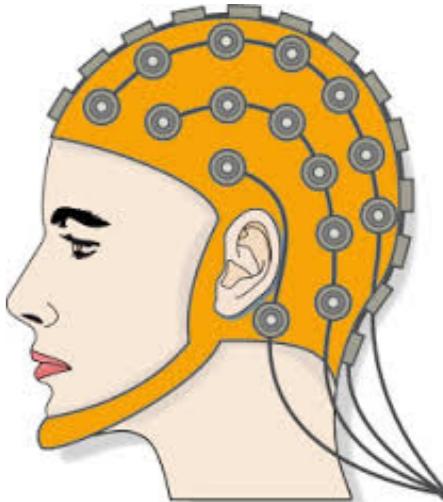
- Stochastic (relative) gradient a.k.a. Infomax
[Bell and Sejnowski 1995]
- Batch (relative) gradient descent
- Truncated Newton with full Hessian (using conjugate gradient solver and “Hessian free” products)
[Tillet et al. 2017]
- Quasi-Newton with H1 or H2 Hessians
close to [Palmer et al. 2012]
- Trust region ICA with H2 approximation
[Choi et al. 2007]

Real data



►► Oracle gradient descent	◆◆ Infomax
○○ Truncated Newton method	□□ L-BFGS
▼▼ Simple quasi-Newton H2	★★ Picard H1
○○ Simple quasi-Newton H1	◀◀ Picard H2
▲▲ Trust region ICA	

EEG



Functional MRI

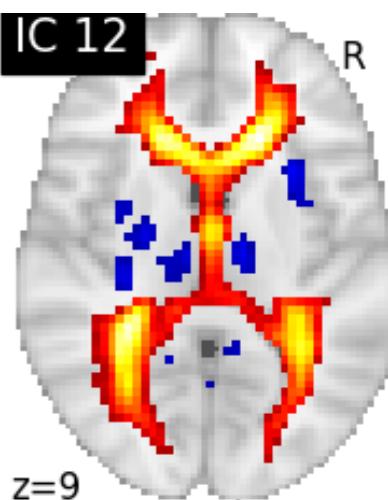
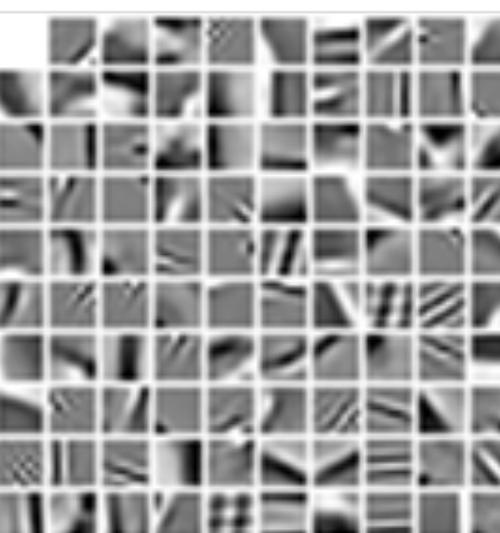


Image patches

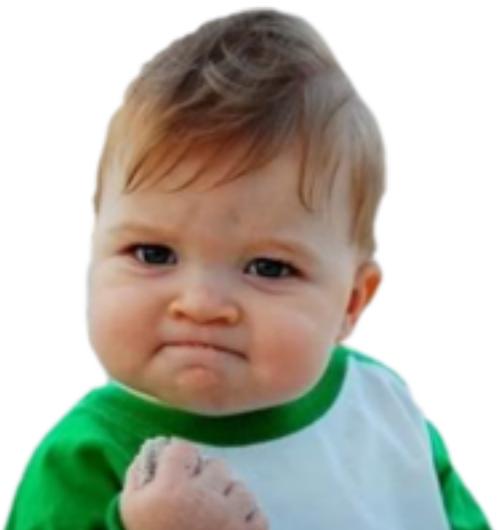


Can we reuse the Hessian approx. idea?

[P.Ablin, J.F. Cardoso and A. Gramfort.
Beyond Pham's algorithm for joint diagonalization.
Proc. ESANN 2019, <https://hal.archives-ouvertes.fr/hal-01936887v1>]

[P.Ablin, D. Fagot, H. Wendt, A. Gramfort and C. Févotte.
A Quasi-Newton algorithm on the orthogonal manifold for NMF with transform learning.
Proc. ICASSP 2019, <https://hal.archives-ouvertes.fr/hal-01912918/document>]

[P.Ablin, J.F. Cardoso and A. Gramfort.
Faster ICA under orthogonal constraint.
Proc. ICASSP'18, <https://hal.inria.fr/hal-01651842/document>]



Problem 1: Picard is a batch method

Problem 2: SGD (here Infomax)

- has no descent guarantee
- requires careful tuning of learning rate

Problem 1: Picard is a batch method

Problem 2: SGD (here Infomax)

- has no descent guarantee
- requires careful tuning of learning rate

Question: Can we find an online algorithm not more costly than SGD with descent guarantee and no learning rate?

ICA, EM and quadratic majorant

Assumption: $p_i \stackrel{\text{def}}{=} d$ Same density per source

We denote: $G(y) = -\log(d(y))$

Objective function:

$$\mathcal{L}(W) = -\log|\det(W)| + \frac{1}{T} \sum_{i=1}^N \sum_{j=1}^T G([WX]_{ij})$$

ICA, EM and quadratic majorant

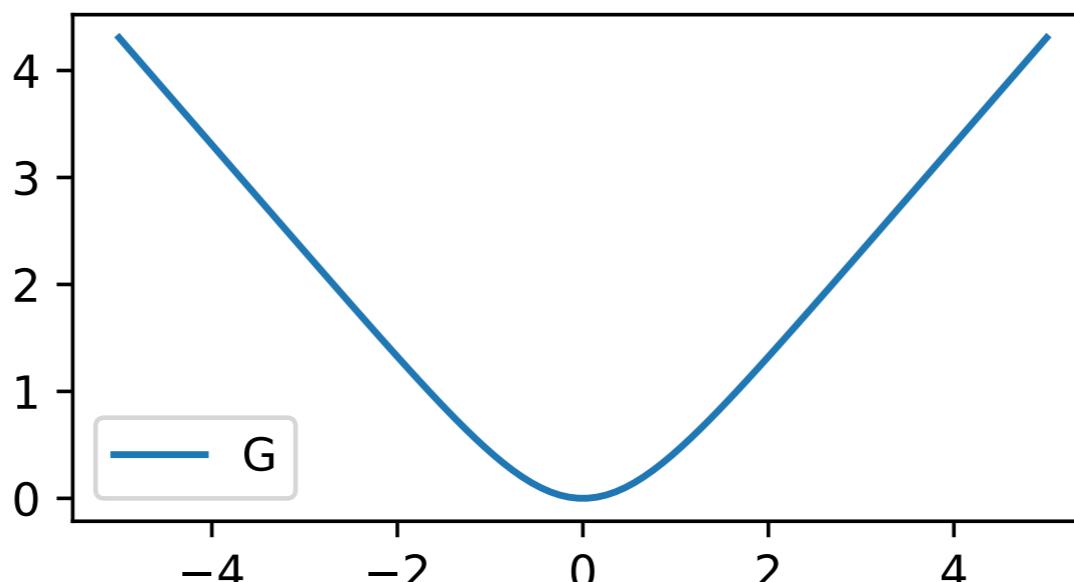
Assumption: $p_i \stackrel{\text{def}}{=} d$ Same density per source

We denote: $G(y) = -\log(d(y))$

Objective function:

$$\mathcal{L}(W) = -\log|\det(W)| + \frac{1}{T} \sum_{i=1}^N \sum_{j=1}^T G([WX]_{ij})$$

Key assumption: $G(\sqrt{\cdot})$ is concave (super-Gaussian source)

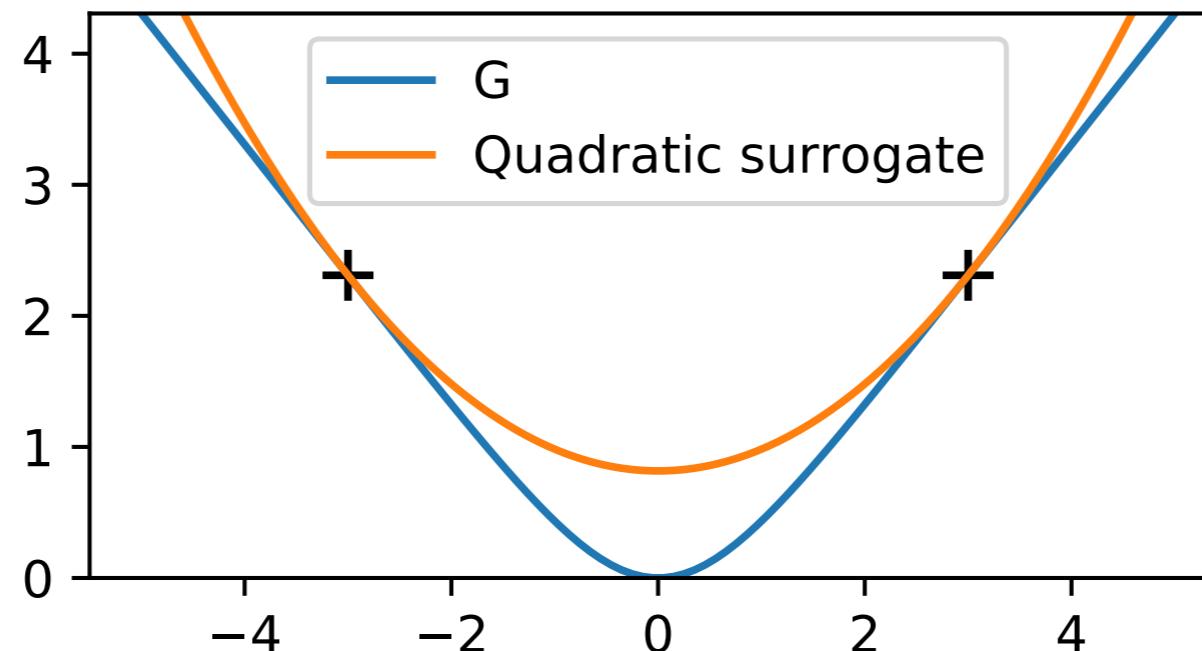


ICA, EM and quadratic majorant

If $G(\sqrt{\cdot})$ is concave (super-Gaussian source)

There exists a function f such that:

$$G(y) = \min_{u \geq 0} \frac{uy^2}{2} + f(u) \quad (\text{cf. Fenchel duality})$$



Minimum is reached at $u^*(y) = \frac{G'(y)}{y}$

ICA, EM and quadratic majorant

The new function reads:

$$\mathcal{L}(W, \textcolor{blue}{U}) =$$

$$-\log|\det(W)| + \frac{1}{2T} \sum_{i=1}^N \sum_{j=1}^T \textcolor{blue}{U}_{ij} [WX]_{ij}^2 + \frac{1}{T} \sum_{i=1}^N \sum_{j=1}^T f(\textcolor{blue}{U}_{ij})$$

ICA, EM and quadratic majorant

The new function reads:

$$\begin{aligned}\mathcal{L}(W, \textcolor{blue}{U}) = \\ -\log|\det(W)| + \frac{1}{2T} \sum_{i=1}^N \sum_{j=1}^T \textcolor{blue}{U}_{ij} [WX]_{ij}^2 + \frac{1}{T} \sum_{i=1}^N \sum_{j=1}^T f(\textcolor{blue}{U}_{ij})\end{aligned}$$

One can minimize by alternate minimization

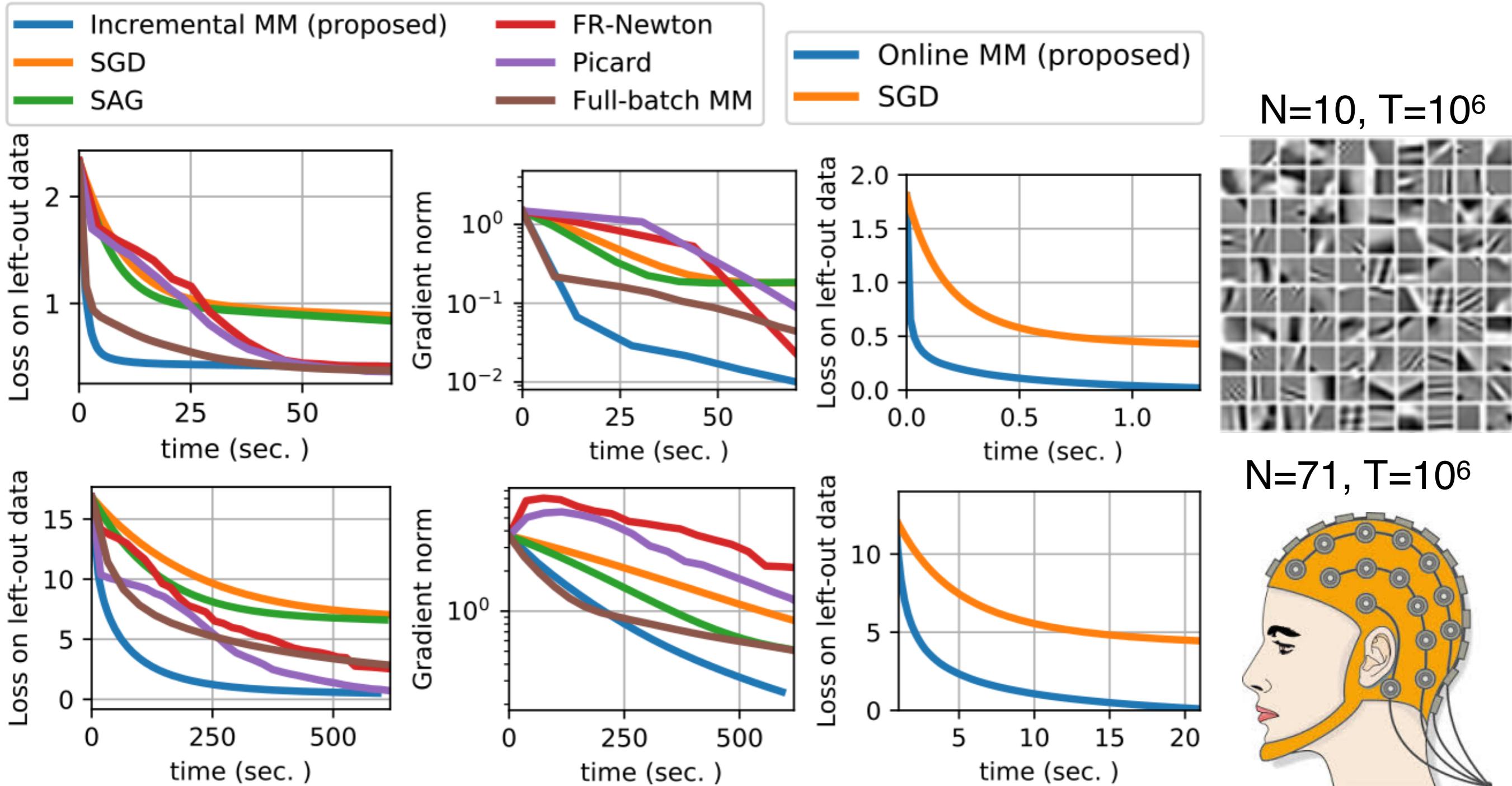
... using ideas from variance reduced stochastic methods (eg. SAG)

... with memory and some accumulation of the right quantities

... and carefully coordinate descent steps

[EM algorithms for ICA, P.Ablin, A. Gramfort, J.F. Cardoso and F.Bach, AISTATS 2019]

Experiments on real data



Learning patterns / waveforms from (multivariate) signals

K-complex

Spindle

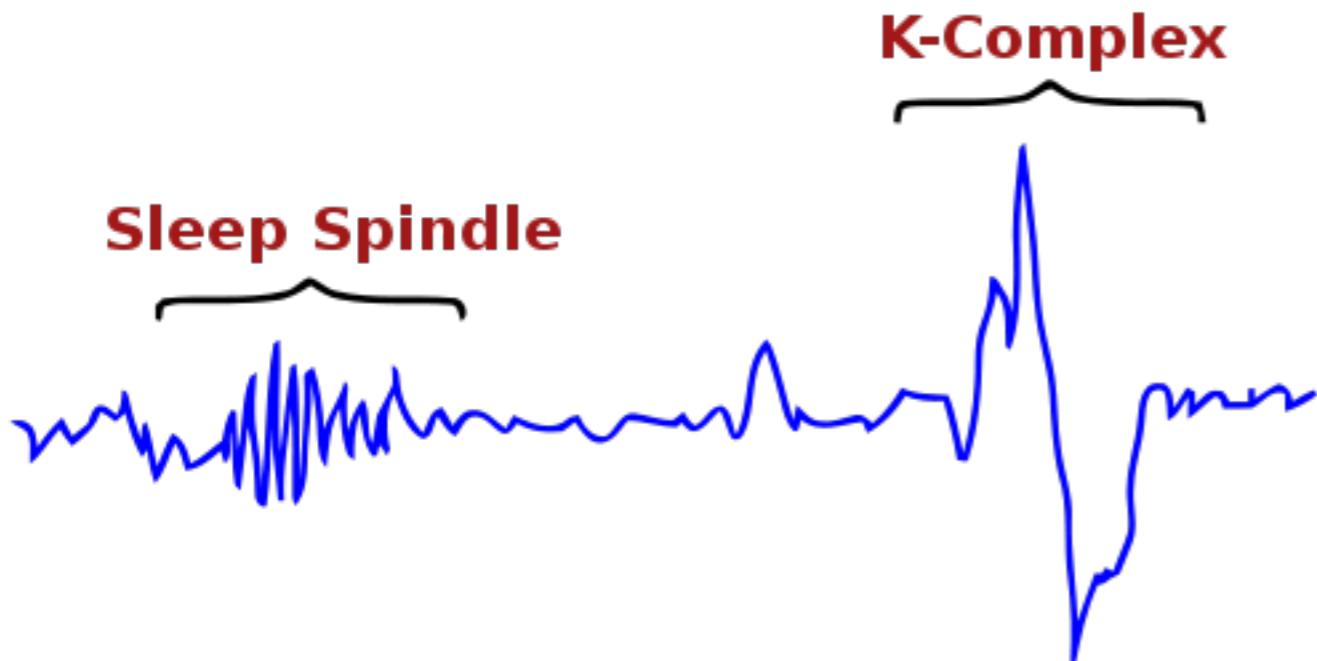
Objective:
Physics informed model
Fast solvers

[*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.*]

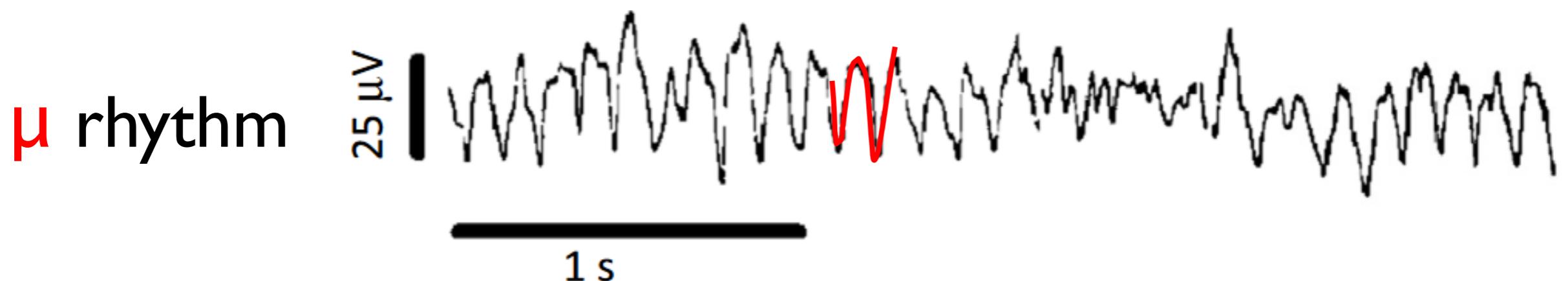
[*Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.*]

[*Distributed Convolutional Dictionary Learning (DiCoDiLe): Pattern Discovery in Large Images and Signals (2019), T. Moreau and A. Gramfort, ArXiv.*]

Code: <https://alphacsc.github.io>

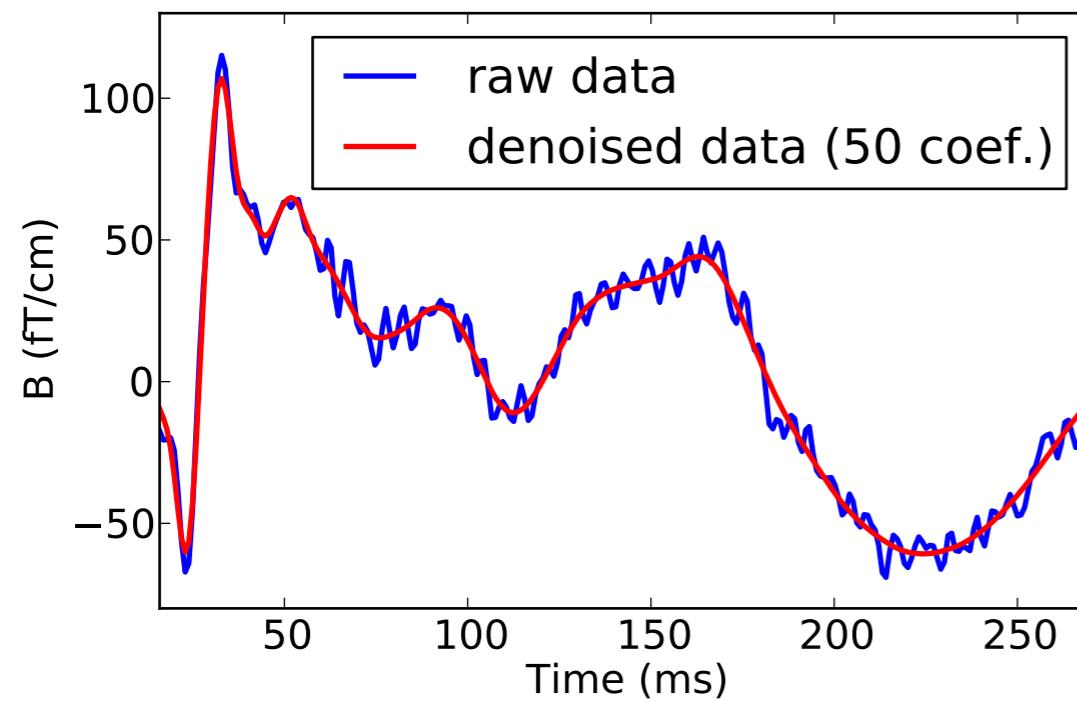
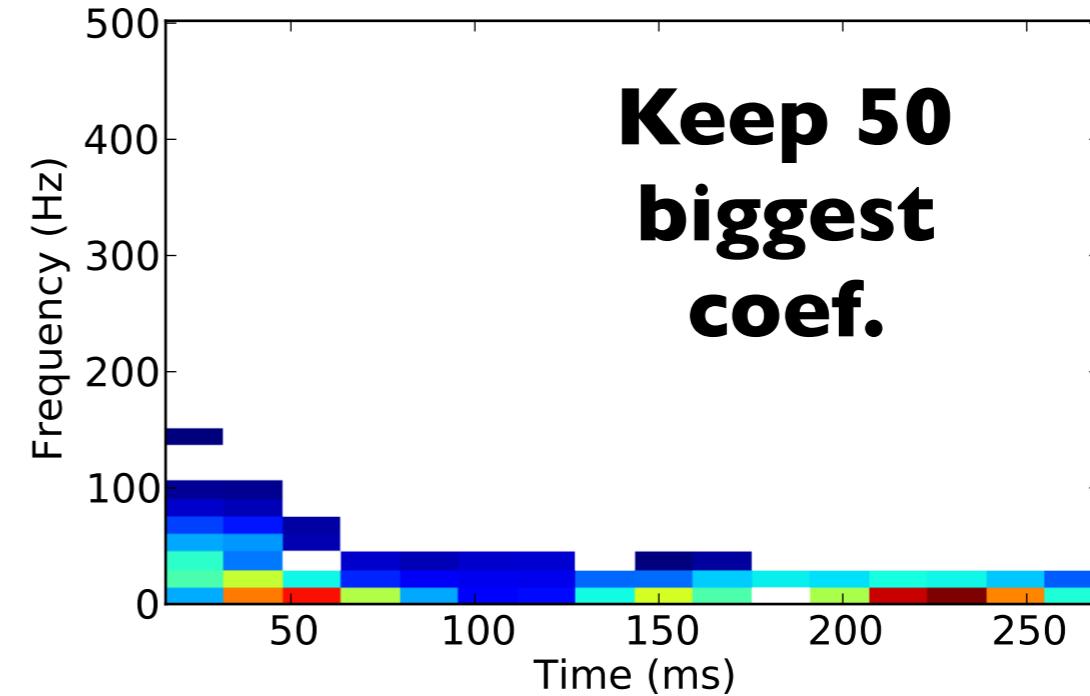
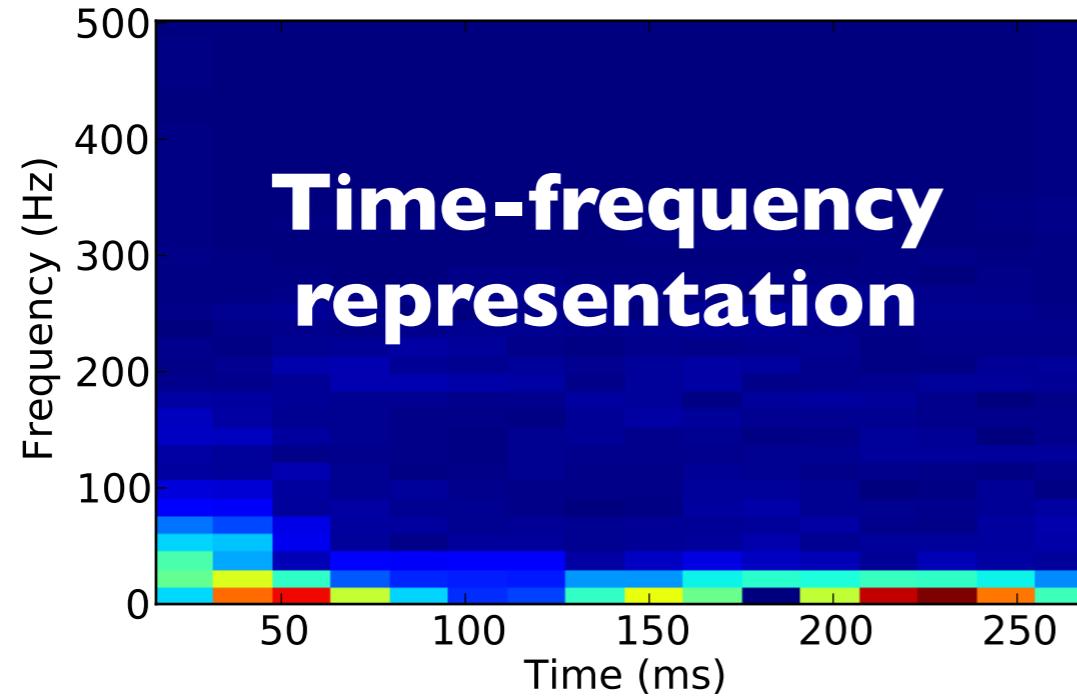


Neural signals exhibit diverse and complex morphologies



CFC: High frequency bursts coupled with slow waves

Why signal representations?



Having adequate signal representation help:
denoising, prediction,
inverse problems etc.

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]

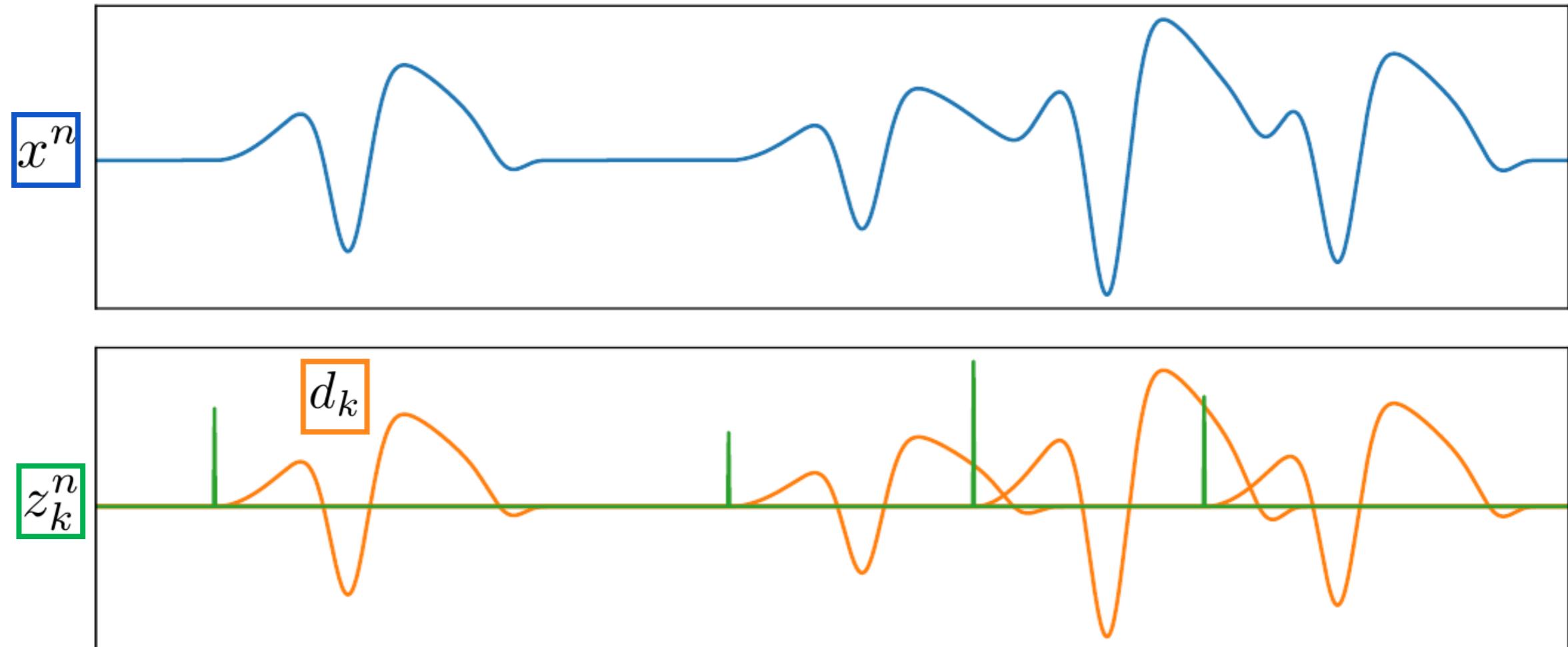
■ Shift-invariant representations

[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]

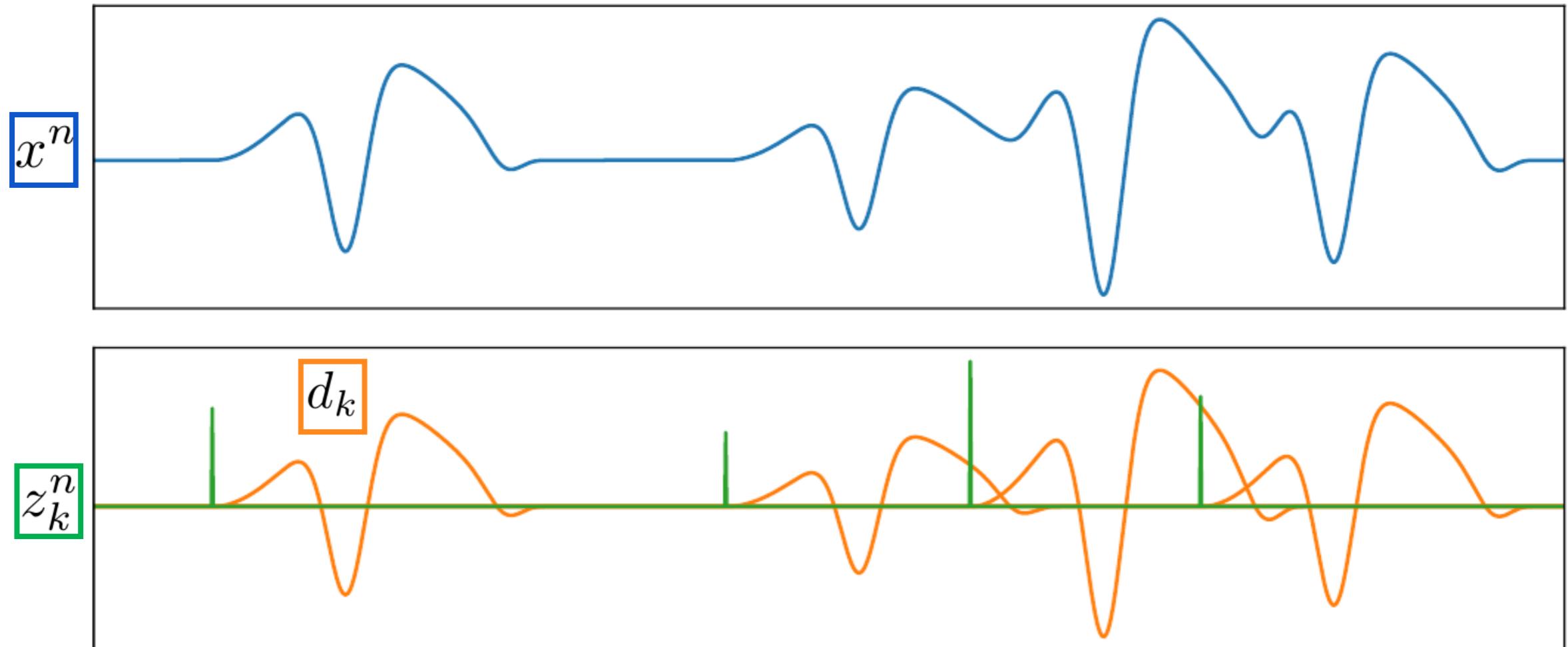
Convolutional sparse coding



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

[Grosse et al, 2007]

Convolutional sparse coding



$$\begin{aligned} \min_{d,z} \sum_{n=1}^N \frac{1}{2} \left\| \boxed{x^n} - \sum_{k=1}^K \boxed{z_k^n} * \boxed{d_k} \right\|_2^2 + \lambda \sum_{k=1}^K \|\boxed{z_k^n}\|_1, \\ \text{s.t. } \|\boxed{d_k}\|_2^2 \leq 1 \text{ and } \boxed{z_k^n} \geq 0. \end{aligned}$$

[Grosse et al, 2007]

Optimization strategy

$$\begin{aligned} \min_{d,z} \sum_{n=1}^N \frac{1}{2} \left\| x^n - \sum_{k=1}^K z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^K \|z_k^n\|_1, \\ \text{s.t. } \|d_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0. \end{aligned}$$

Block-coordinate descent (update Z, D, Z, D, Z...):

Optimization strategy

$$\begin{aligned} \min_{d,z} \sum_{n=1}^N \frac{1}{2} \left\| x^n - \sum_{k=1}^K z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^K \|z_k^n\|_1, \\ \text{s.t. } \|d_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0. \end{aligned}$$

Block-coordinate descent (update Z, D, Z, D, Z...):

- Z-step

- GCD [Kavukcuoglu et al, 2010]
- FISTA [Chalasani et al, 2013]
- ADMM [Bristow et al, 2013]
- ADMM + FFT [Wohlberg, 2016]
- L-BFGS [Jas et al, 2017]
- LGCD [Dupré la Tour et al, 2018]

Optimization strategy

$$\begin{aligned} \min_{d,z} \sum_{n=1}^N \frac{1}{2} \left\| x^n - \sum_{k=1}^K z_k^n * d_k \right\|_2^2 + \lambda \sum_{k=1}^K \|z_k^n\|_1, \\ \text{s.t. } \|d_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0. \end{aligned}$$

Block-coordinate descent (update Z, D, Z, D, Z...):

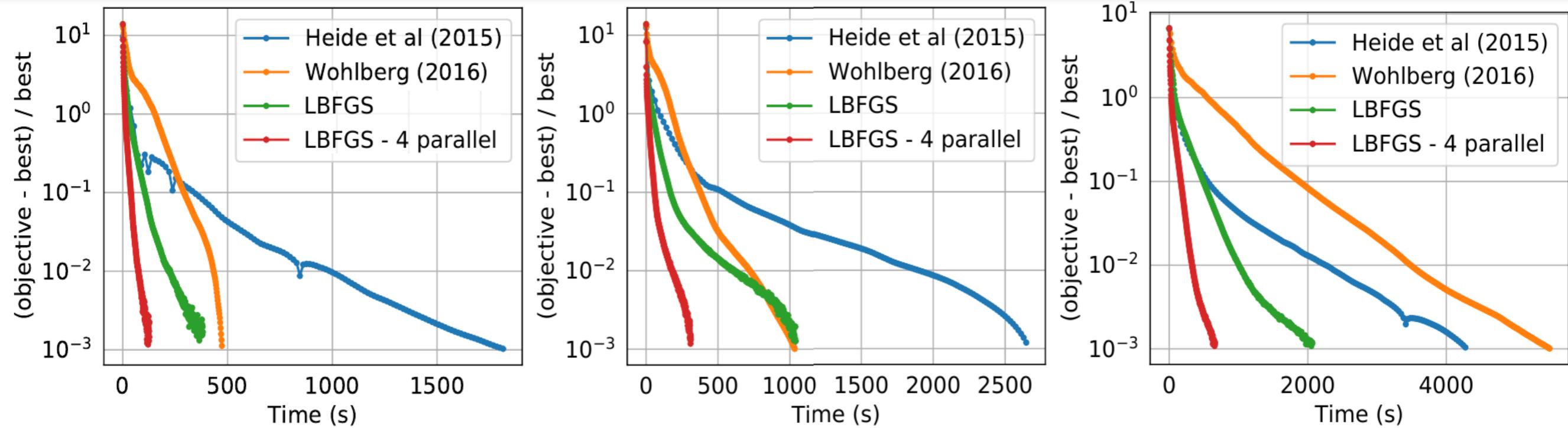
■ Z-step

- GCD [Kavukcuoglu et al, 2010]
- FISTA [Chalasani et al, 2013]
- ADMM [Bristow et al, 2013]
- ADMM + FFT [Wohlberg, 2016]
- L-BFGS [Jas et al, 2017]
- LGCD [Dupré la Tour et al, 2018]

■ D-step

- FFT [Grosse et al, 2007]
- ADMM + FFT [Heide et al, 2015]
- ADMM + FFT [Wohlberg, 2016]
- L-BFGS (dual) [Jas et al, 2017]
- PGD [Dupré la Tour et al, 2018]

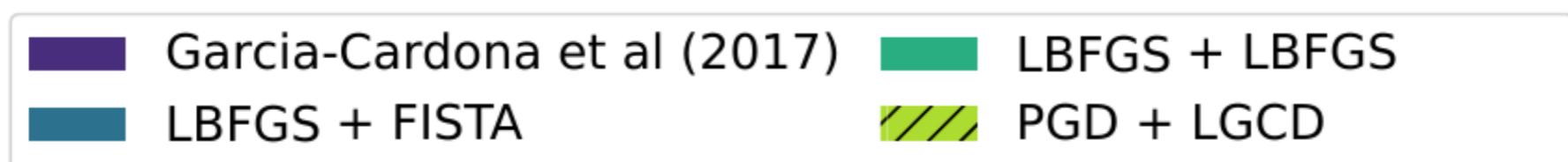
Speed benchmarks



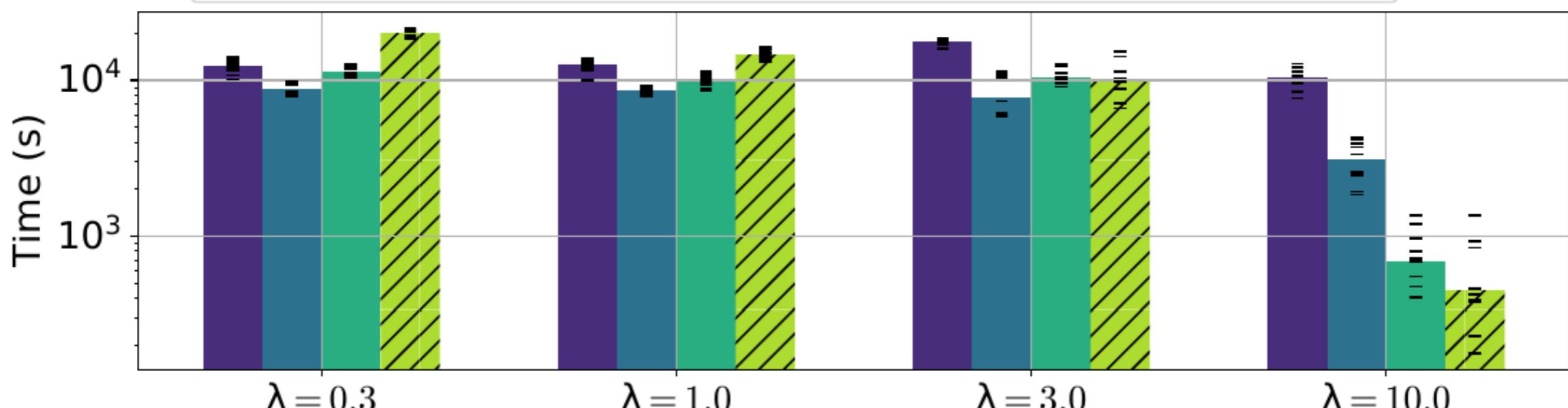
(a) $K = 2, L = 32$.

(b) $K = 2, L = 128$.

(c) $K = 10, L = 32$.



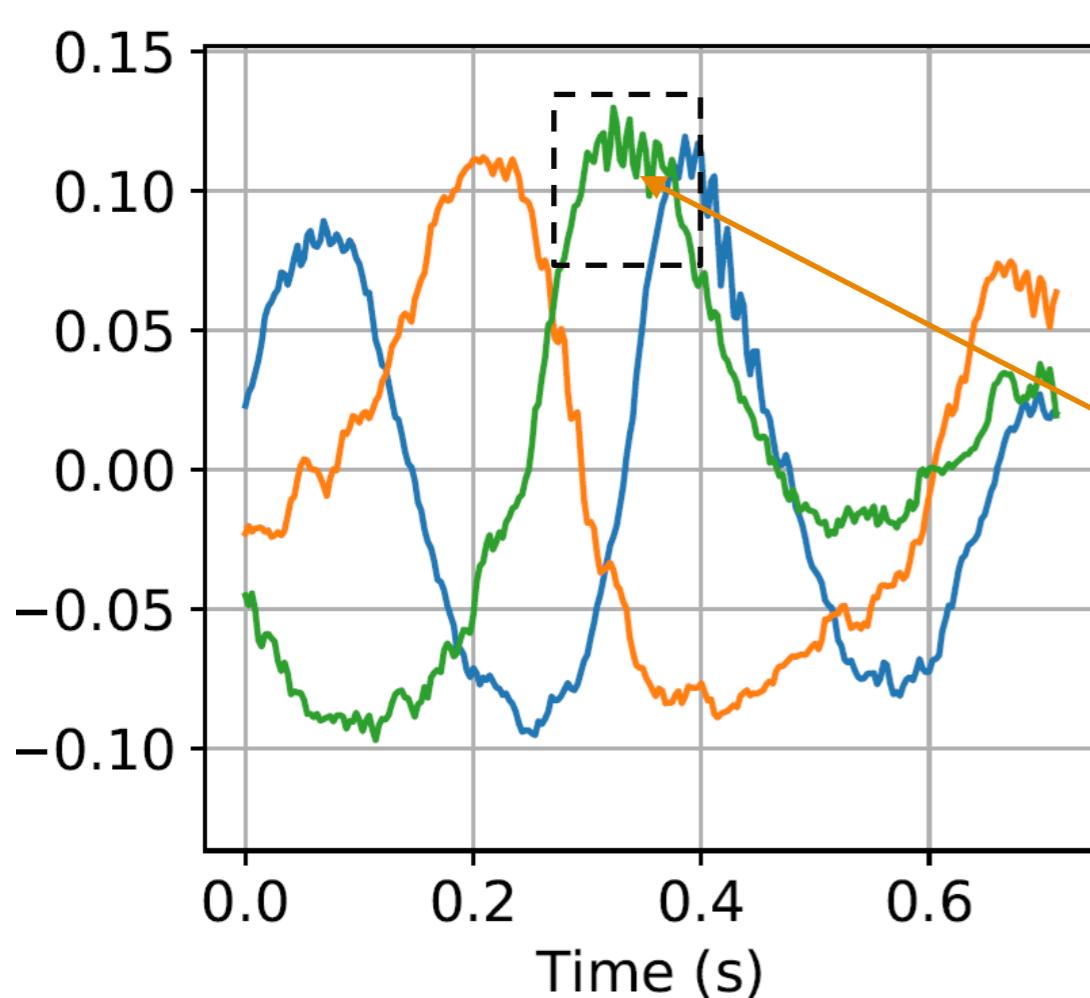
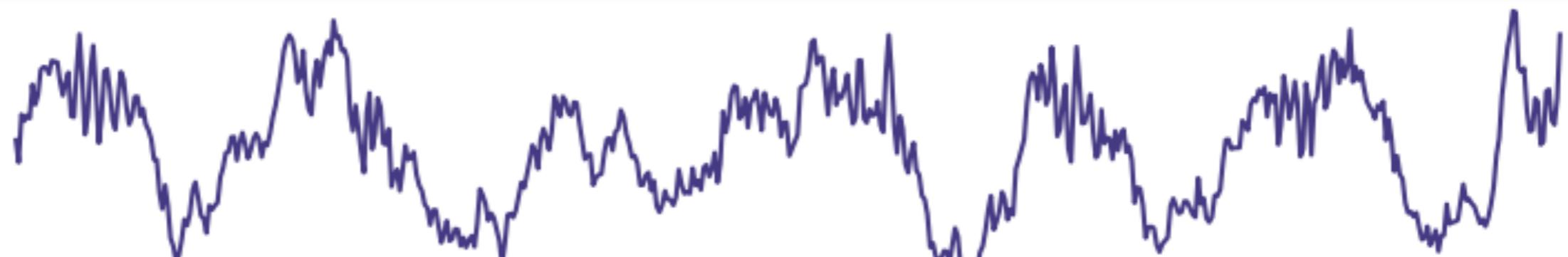
Jas et al. 2017



[Dupré la Tour et al. 2018]

Learned atoms

Data:



**How about if I
have many
channels?**

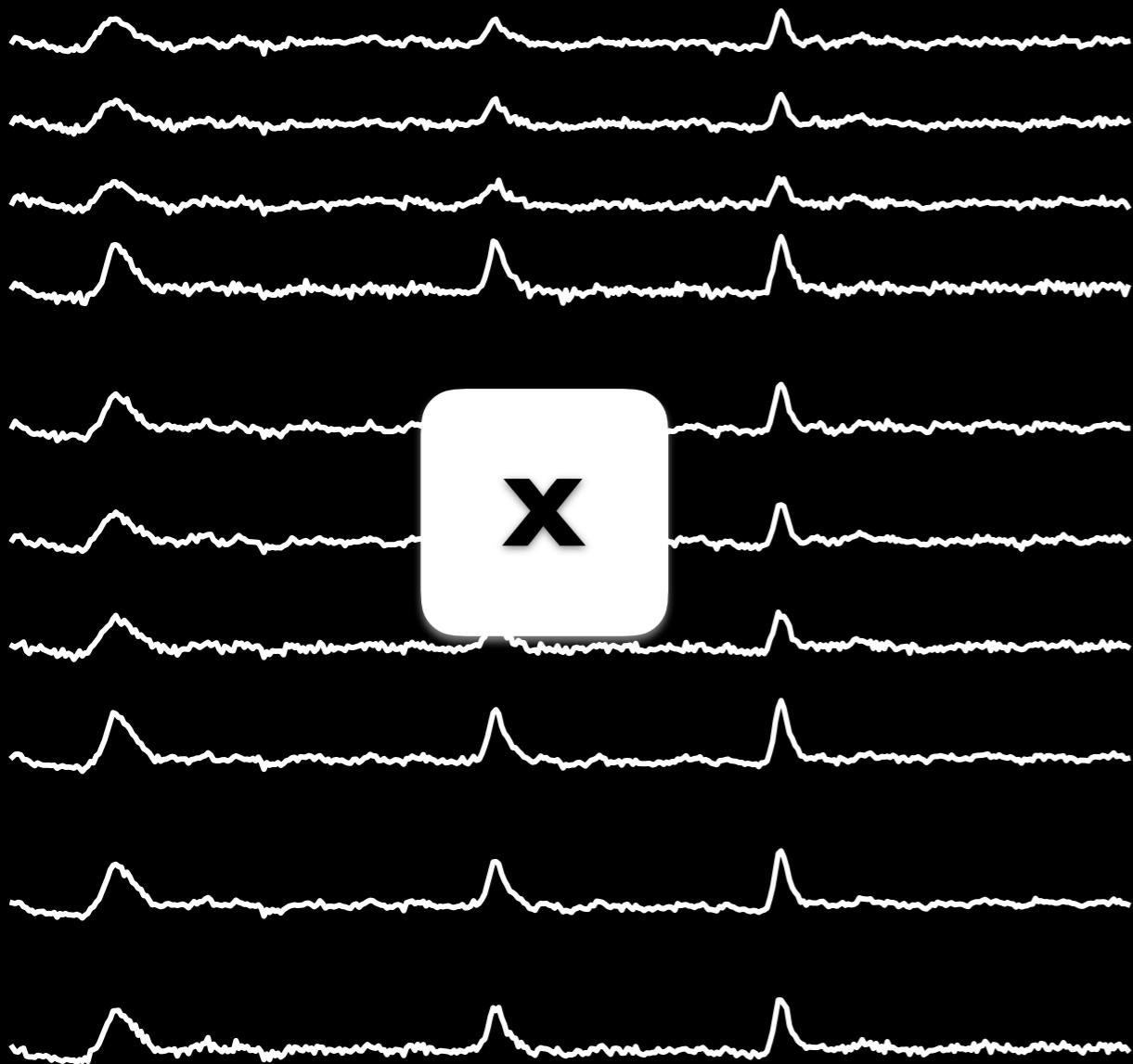
CSC reveals “nested oscillations”

[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.]

From ICA to CSC

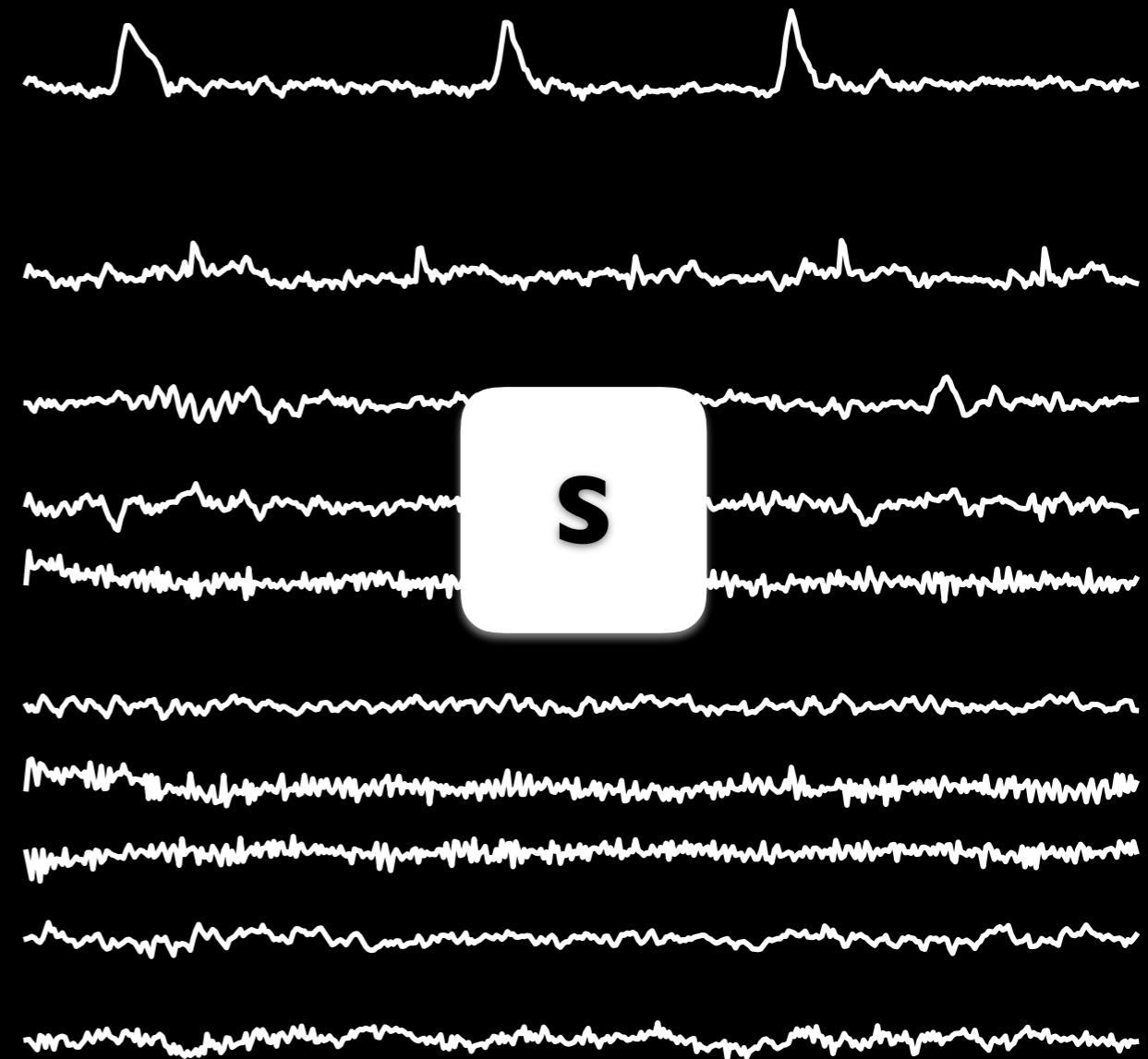
Independent Component Analysis (ICA)

Observations (raw EEG)



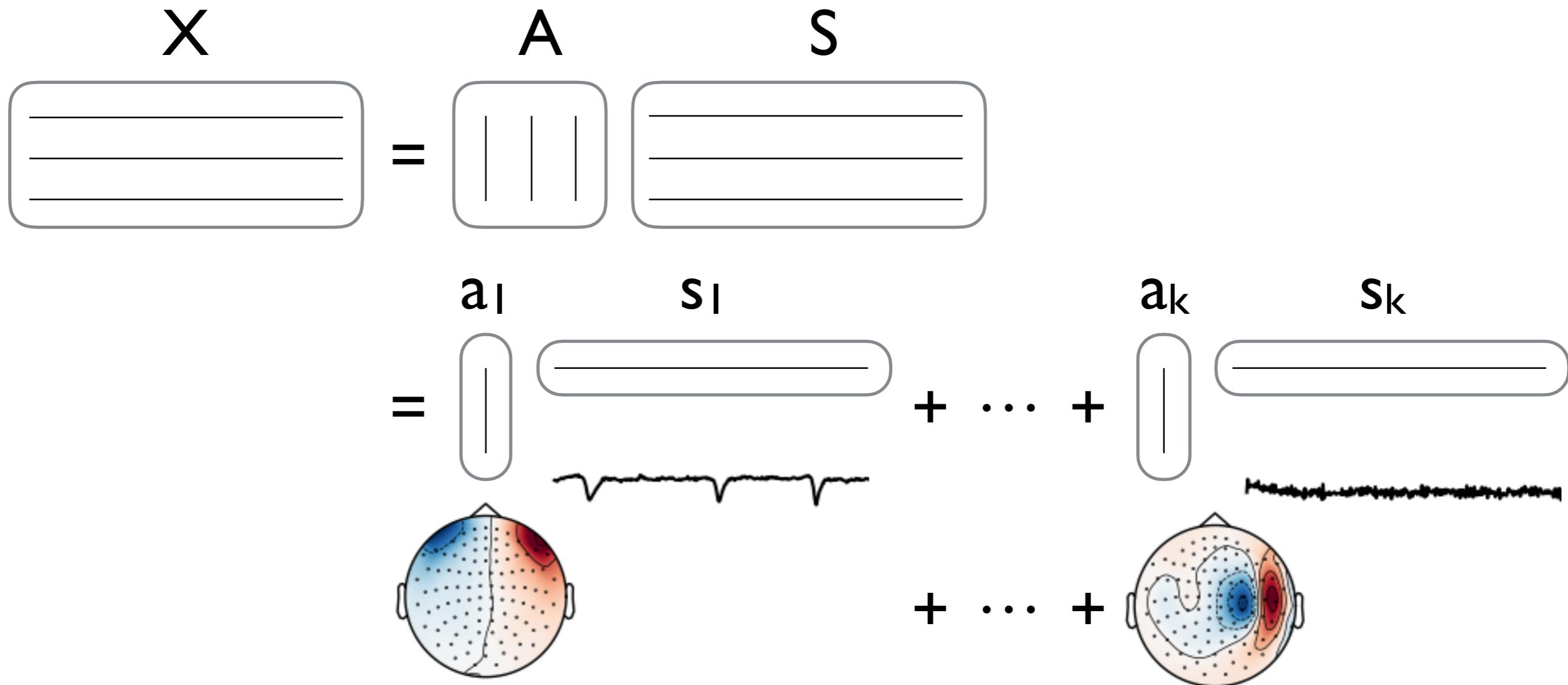
X

ICA recovered sources



S

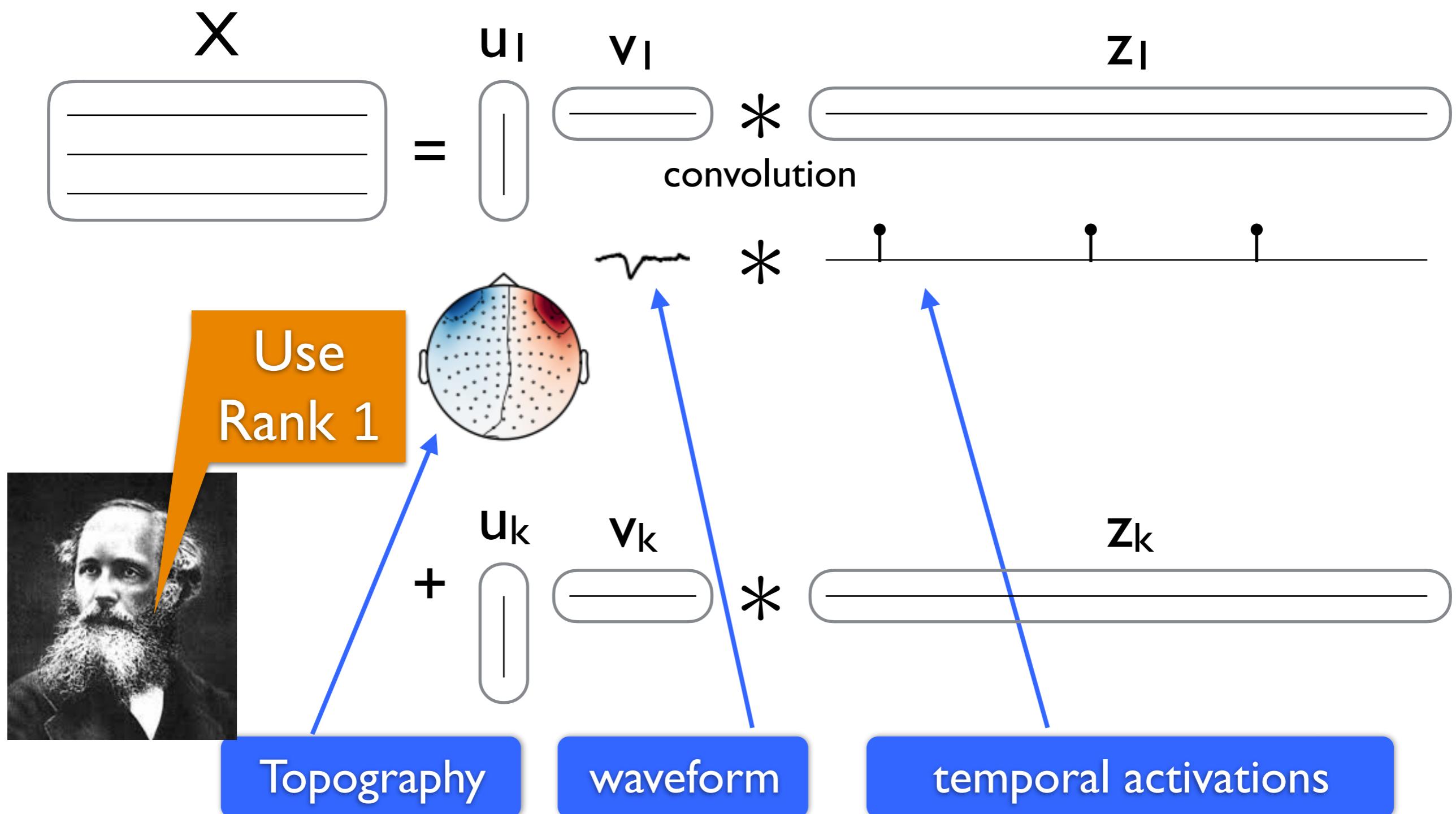
From ICA...



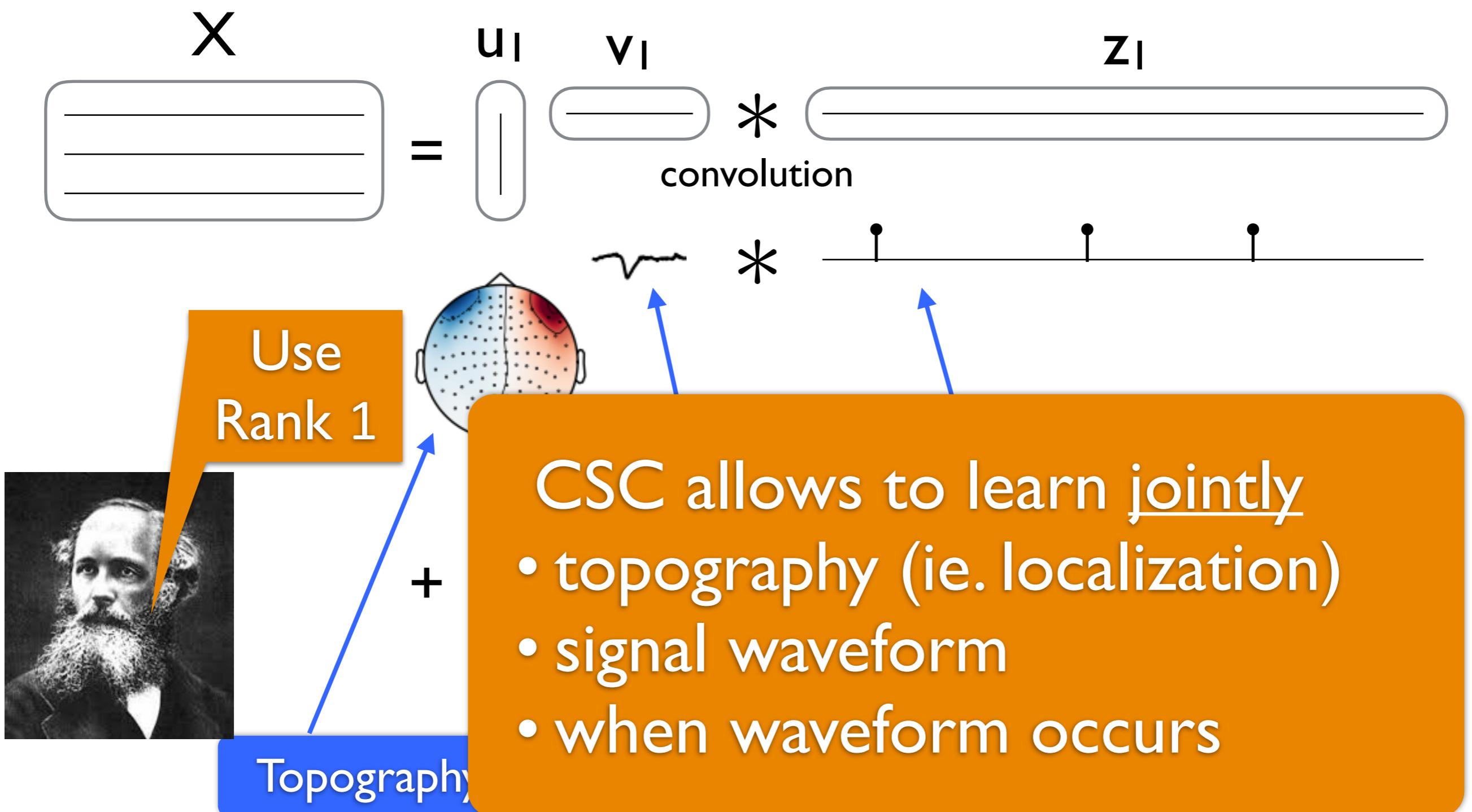
https://www.martinos.org/mne/stable/auto_tutorials/plot_artifacts_correction_ica.html

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

... to CSC



... to CSC



Multivariate CSC

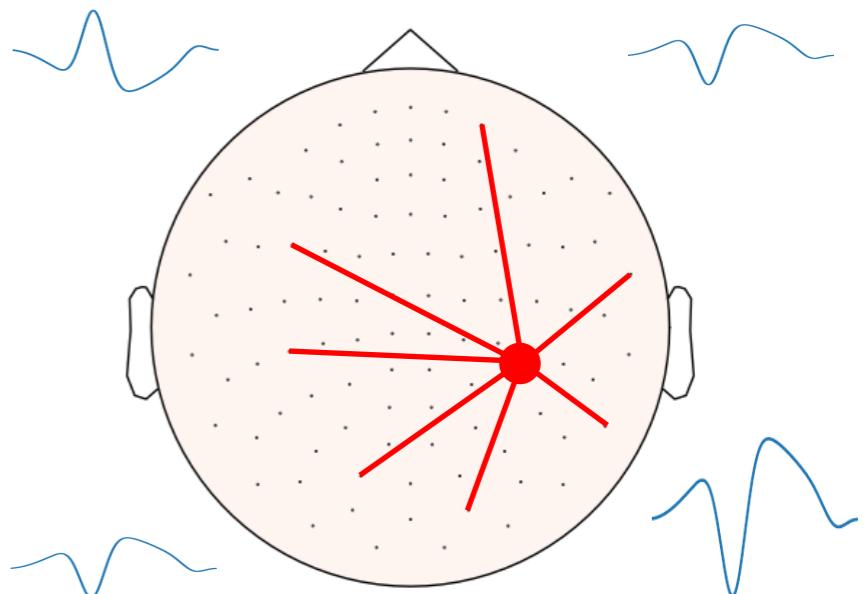
$$\begin{aligned} \min_{D, z} \sum_{n=1}^N \frac{1}{2} \left\| X^n - \sum_{k=1}^K z_k^n * D_k \right\|_2^2 + \lambda \sum_{k=1}^K \|z_k^n\|_1, \\ \text{s.t. } \|D_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0. \end{aligned}$$

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

Multivariate CSC

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

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



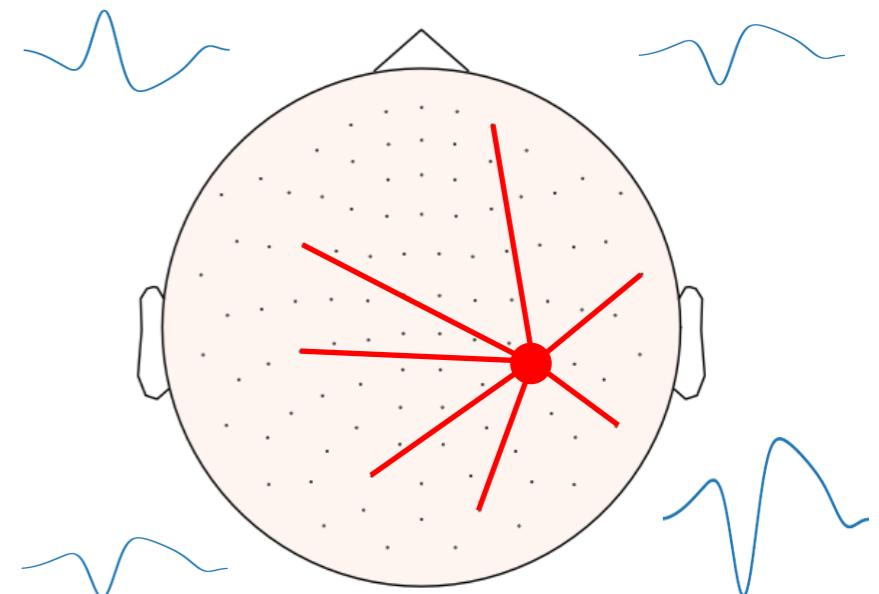
[*Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018),
T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.*]

Multivariate CSC

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

Rank 1 constraint:

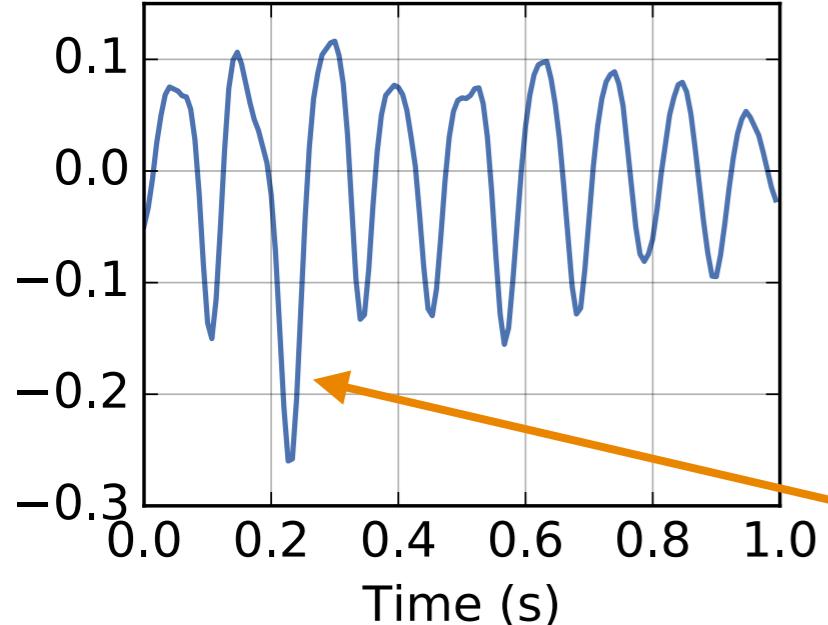
$$\begin{aligned} \min_{u, v, z} \sum_{n=1}^N \frac{1}{2} \left\| X^n - \sum_{k=1}^K z_k^n * (u_k v_k^\top) \right\|_2^2 + \lambda \sum_{k=1}^K \|z_k^n\|_1, \\ \text{s.t. } \|u_k\|_2^2 \leq 1, \|v_k\|_2^2 \leq 1 \text{ and } z_k^n \geq 0. \end{aligned}$$



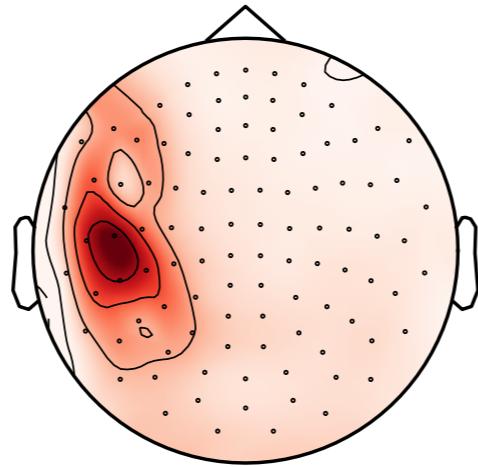
[*Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals, (2018), T. Dupré la Tour, T. Moreau, M. Jas, A. Gramfort, Proc. NeurIPS Conf.*]

CSC on MEG

A. Temporal waveform

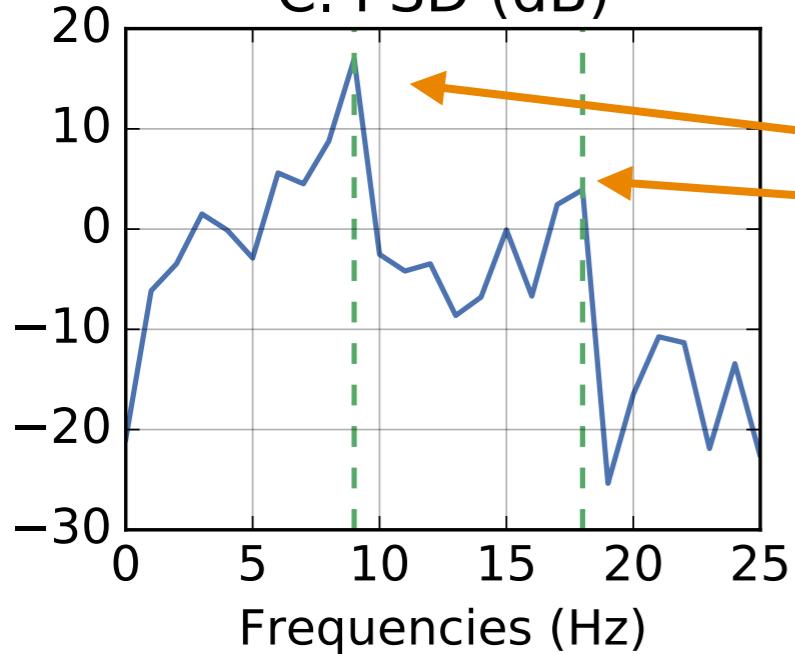


B. Spatial pattern

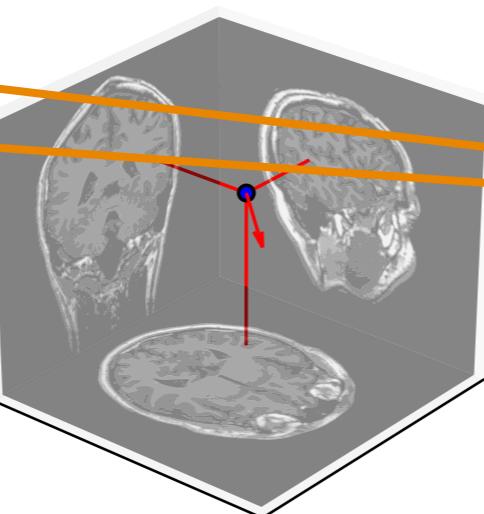


- MEG vectorview
- Median nerve stim.

C. PSD (dB)



D. Dipole fit

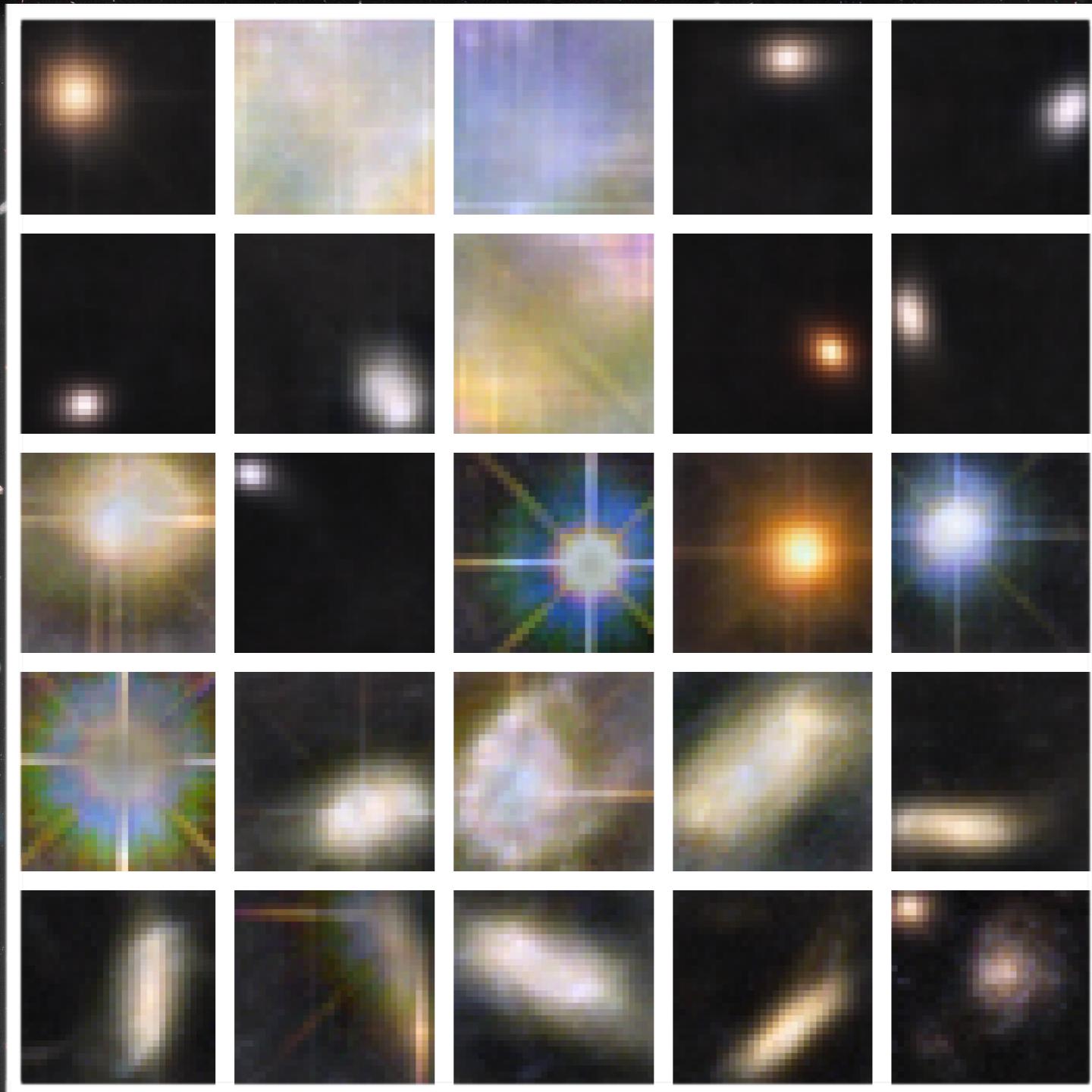


CSC reveals mu-shaped waveforms

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.*]

Example using astrophysics

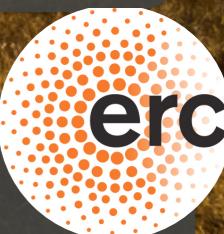


[*Distributed Convolutional Dictionary Learning (DiCoDiLe): Pattern Discovery in Large Images and Signals (2019), T. Moreau and A. Gramfort, ArXiv <https://arxiv.org/abs/1901.09235>*]

Data: <http://hubblesite.org/image/3920/news/58-hubble-ultra-deep-field> (6000x3664 pixels)

Conclusion

- Contrib 1: Use Hessian approximations to inform a quasi-Newton solver (L-BFGS)
- Contrib 2: Convolutional sparse coding model for multivariate series with fast solvers
- Statistical machine learning for neuroscience data
- High dimensional inference with complex noise and non-stationarities
- Fast algorithms for large scale non-smooth problems
- Software tools



Thanks !

Especially to:
Mainak Jas (PhD 2014-2017)
Tom Dupré La Tour (PhD 2015-2018)
Pierre Ablin (PhD 2016-...)
Thomas Moreau (Post-doc)
Francis Bach (DR, Inria)
Umut Şimşekli (MdC, Télécom ParisTech)
Jean-François Cardoso (DR, CNRS)

Contact

<http://alexandre.gramfort.net>

GitHub : @agramfort



Twitter : @agramfort



Support

ANR-NSF Grant Thalameeg
European Research Council (ERC SLAB-YStG-676943)

