

# Using the Dictionary Structure for efficient Convolutional Dictionary Learning

Thomas Moreau – INRIA – Université Paris Saclay



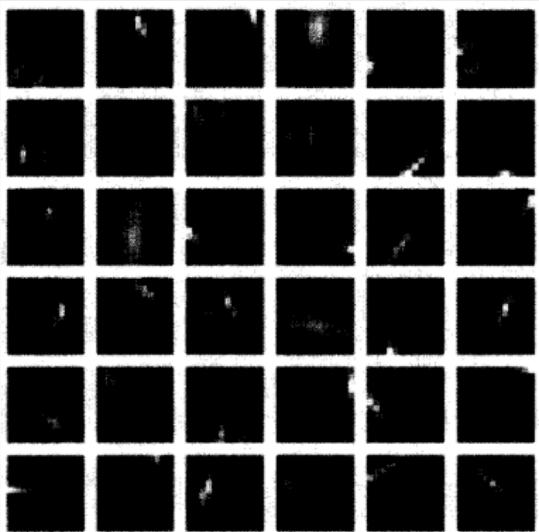
PARIETAL

*informatics mathematics*  
**inria**

Dictionary learning learns a set of atoms (patterns) to sparsely reconstruct a signal,

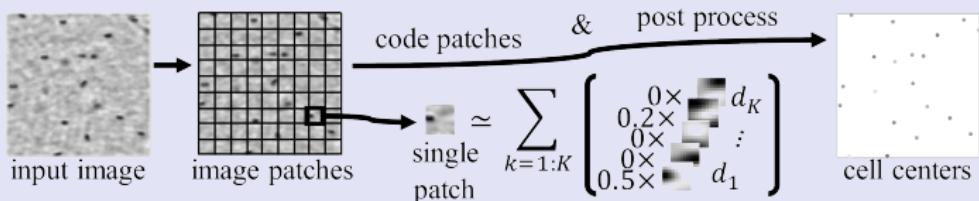
## Goal:

- ▶ Feature extraction,
- ▶ Signal exploration.

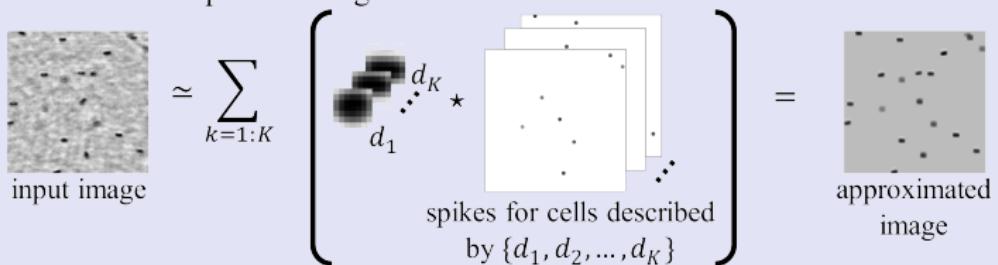


Patches learned with natural images in Olshausen and Field 1997.

Sparse coding



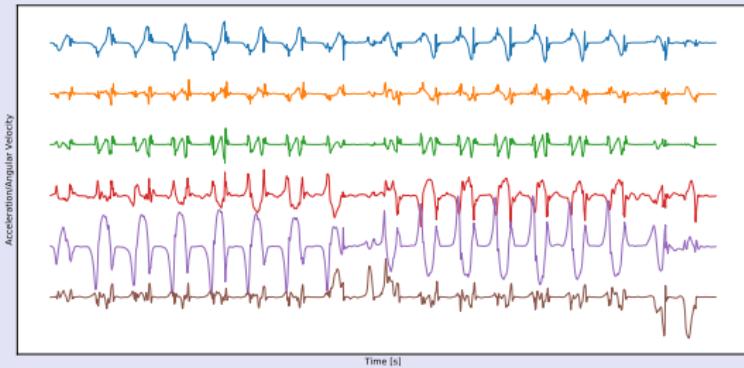
Convolutional sparse coding



**Convolutional** Dictionary Learning learns a set of **shift-invariant** atoms to sparsely reconstruct a signal,

- ▶ Improve sparsity
- ▶ Not all patches are encoded
- ▶ Sharper atoms

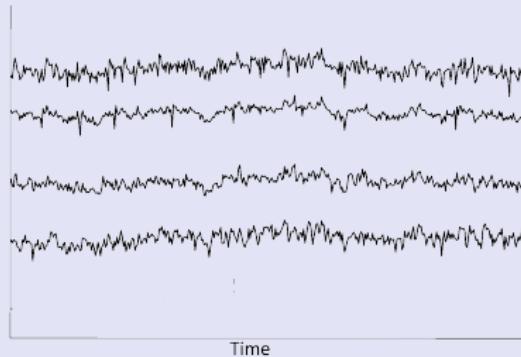
# Application fields



[Oudre et al. 2018, Sensors]

- ▶ Detecting steps in human walk recordings to predict elderly falls.

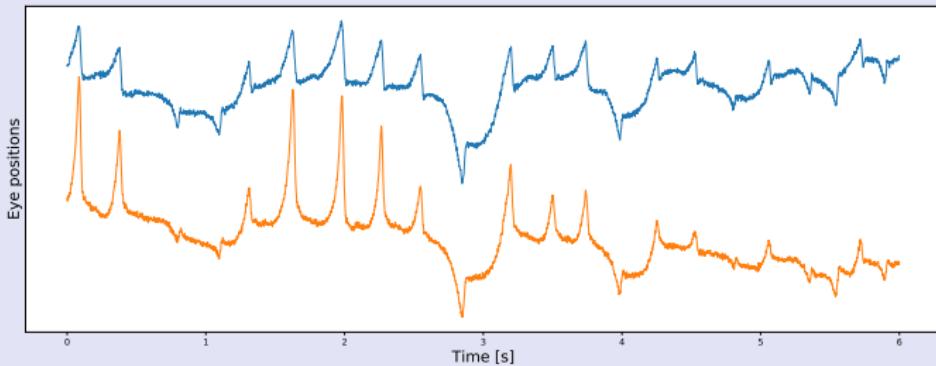
## Application fields



[Dupré la Tour et al. 2018, NeurIPS]

- ▶ Detecting steps in human walk recordings to predict elderly falls.
- ▶ Exploring neurological signals from ECG and MEG,

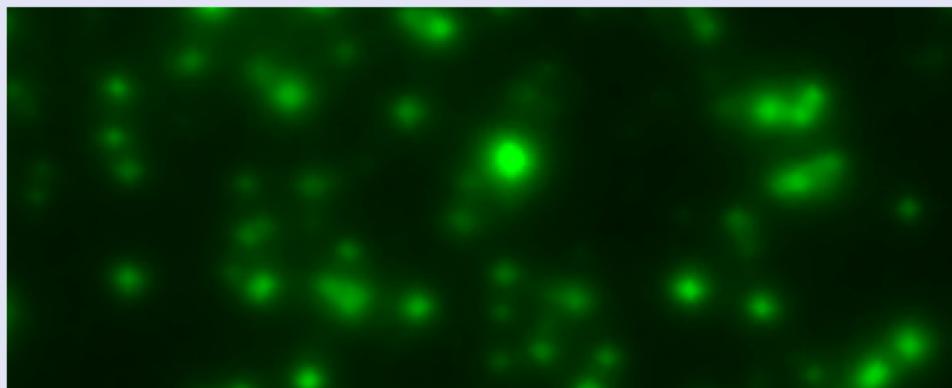
## Application fields



[Robert et al. 2016, preprint]

- ▶ Detecting steps in human walk recordings to predict elderly falls.
- ▶ Exploring neurological signals from ECG and MEG,
- ▶ Classifying pathological eye movements from oculomotor signals.

## Application fields



[del Aguila Pla et al. 2018, IEEE TSP; Yellin et al. 2017, ISBI]

- ▶ Detecting steps in human walk recordings to predict elderly falls.
- ▶ Exploring neurological signals from ECG and MEG,
- ▶ Classifying pathological eye movements from oculomotor signals.
- ▶ Counting cells in biological images.

## Application fields



[del Aguila Pla et al. 2018, ICASSP;  
Beckouche et al. 2013, Astronomy & Astrophysics]

- ▶ Detecting steps in human walk recordings to predict elderly falls.
- ▶ Exploring neurological signals from ECG and MEG,
- ▶ Classifying pathological eye movements from oculomotor signals.
- ▶ Counting cells in biological images.
- ▶ Counting stars and galaxies in telescope images

# Challenges of Convolutional Dictionary Learning

---

- ▶ **Computational:** how to scale with large signals,
  - ▶ by exploiting the structure of the dictionary.
  - ▶ by parallelization.
- ▶ **Modelization:** how to incorporate prior knowledge,
  - ▶ on the activations.
  - ▶ on the patterns.
- ▶ **Evaluation:** how to evaluate the quality of the learned patterns.
- ▶ **Theoretical:** pattern recovery.

# Challenges of Convolutional Dictionary Learning

---

- ▶ **Computational:** how to scale with large signals,
  - ▶ by exploiting the structure of the dictionary.  
[Moreau and Bruna 2017, ICLR]
  - ▶ by parallelization.  
[Moreau et al. 2018, ICML; Moreau and Gramfort 2019, preprint]
- ▶ **Modelization:** how to incorporate prior knowledge,
  - ▶ on the activations.
  - ▶ on the patterns.  
[Dupré la Tour et al. 2018, NeurIPS]
- ▶ **Evaluation:** how to evaluate the quality of the learned patterns.
- ▶ **Theoretical:** pattern recovery.

Convolutional Dictionary Learning

Adaptive Sparse Coding

Scaling up Convolutional Sparse Coding with  
coordinate descent and distributed optimization

Rank-1 Constrained Convolutional Dictionary Learning

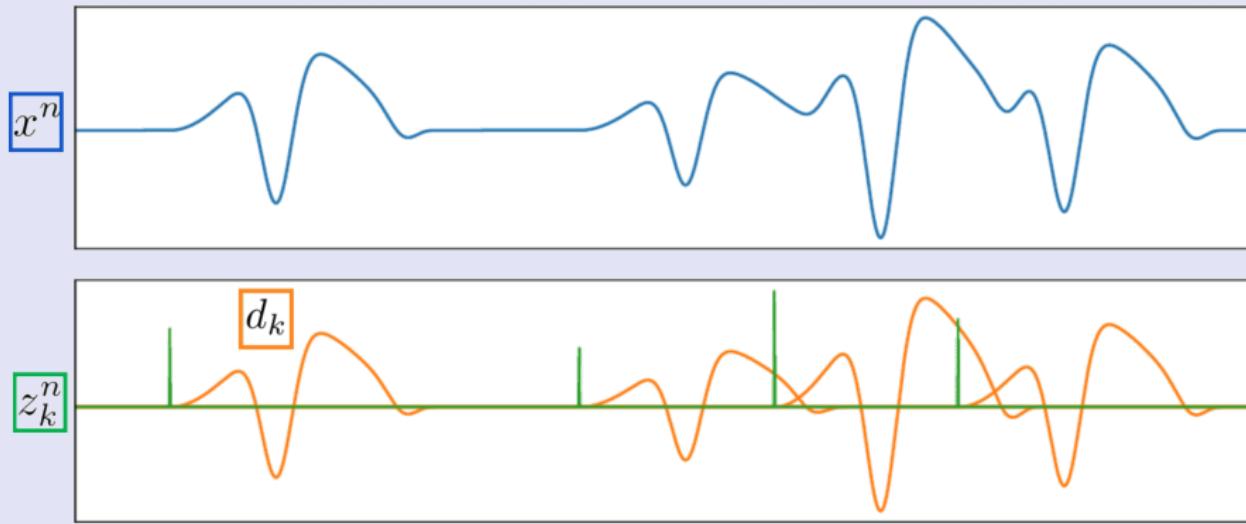
## Convolutional Dictionary Learning

### References

- ▶ Grosse, R., Raina, R., Kwong, H., and Ng, A. Y. (2007). Shift-Invariant Sparse Coding for Audio Classification. *Cortex*, 8:9

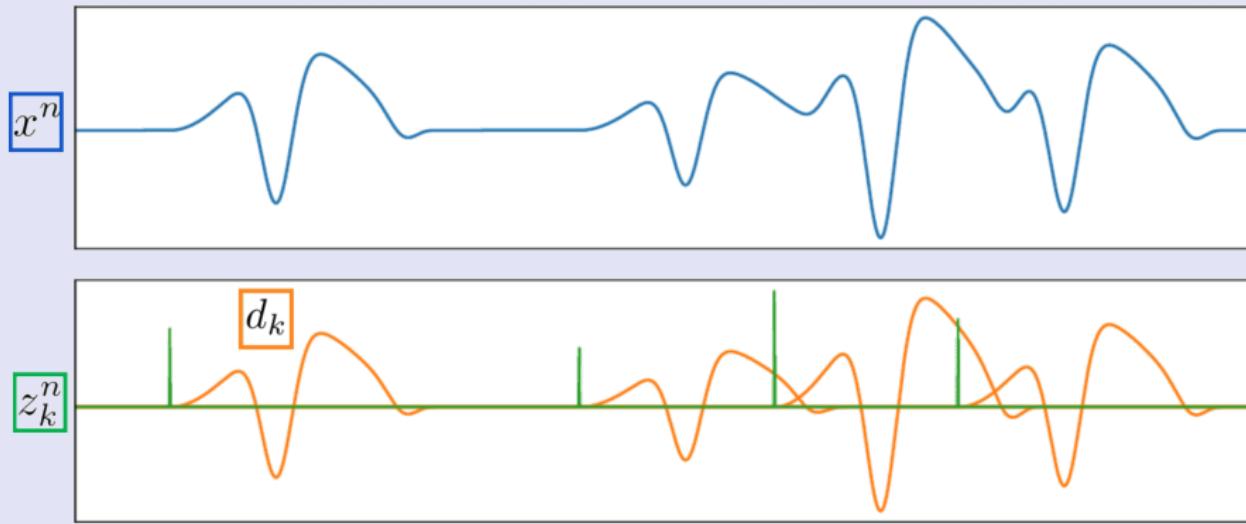
## Extracting shift invariant patterns

**Key idea:** decouple the localization of the patterns and their shape



## Extracting shift invariant patterns

**Key idea:** decouple the localization of the patterns and their shape

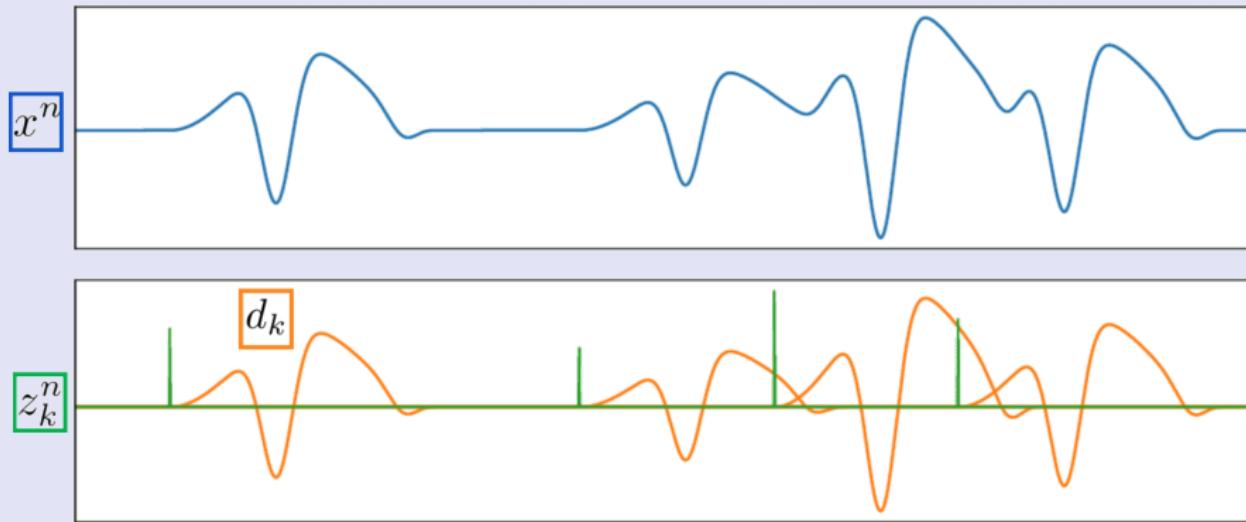


**Convolutional  
Representation:**

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

# Extracting shift invariant patterns

**Key idea:** decouple the localization of the patterns and their shape



**Convolutional  
Dictionary Learning:**

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

## Notation

---

### Sparse Convolutional model:

$$X[t] = \sum_{k=1}^K (\mathcal{D}_k * Z_k)[t] + \mathcal{E}[t]$$

with  $Z$  sparse. Few of its coefficients are non-zero.

- ▶  $X$  is a signal of length  $T$
- ▶  $\mathcal{E}$  is a noise signal of length  $T$
- ▶  $\mathcal{D}$  is a set of  $K$  patterns of length  $L$
- ▶  $Z$  is a signal of length  $\tilde{T} = T - L + 1$  in  $\mathbb{R}^K$

## Convolutional Dictionary Learning

Dictionary learning optimization problem for  $\{X^{[n]}\}_{n=1}^N$

$$\min_{Z, \|\mathcal{D}_k\| \leq 1} \frac{1}{N} \sum_{n=1}^N \underbrace{\left\| X^{[n]} - \sum_{k=1}^K \mathcal{D}_k * Z_k^{[n]} \right\|_2^2}_{E(Z) \text{ data fit}} + \underbrace{\lambda \|Z^{[n]}\|_1}_{\text{penalization}}$$

with a regularization parameter  $\lambda > 0$ .

This problem is bi-convex and an approximate solution is obtained through  
**alternate minimization.** [Engan et al., 1999; Grosse et al., 2007]

## *D*-step: Dictionary updates

→  $Z$  fixed, update  $\mathbf{D}$

$$\mathbf{D}^* = \operatorname{argmin}_{\mathbf{D}} \frac{1}{N} \sum_{n=1}^N \|X^{[n]} - \sum_{k=1}^K \mathbf{D}_k * Z_k^{[n]}\|_2^2 + \mathbf{1}_{\Omega}(D)$$

### Related Algorithms:

- ▶ Proximal Gradient Descent (PDG) [Rockafellar, 1976]
- ▶ Accelerated PGD [Nesterov, 1983]
- ▶ Block Coordinate Descent [Mairal et al., 2010]
- ▶ Alternated Direction Method of Multiplier (ADMM)  
[Gabay and Mercier, 1976]

## $Z$ -step: Sparse coding

$\rightarrow \mathbf{D}$  fixed, update  $Z$

$$Z^{[n],*} = \operatorname*{argmin}_{Z^{[n]}} \|X^{[n]} - \sum_{k=1}^K \mathbf{D}_k * Z_k^{[n]}\|_2^2 + \lambda \|Z^{[n]}\|_1$$

$\Rightarrow$  Independent for each  $n \in [1, N]$

### Related Algorithms:

- ▶ Iterative Soft-Thresholding Algorithm (ISTA)  
[Daubechies et al., 2004; Chalasani et al., 2013]
- ▶ Fast ISTA  
[Beck and Teboulle, 2009; Wohlberg, 2016]
- ▶ Alternated Direction Method of Multiplier (ADMM)  
[Gabay and Mercier, 1976; Bristow et al., 2013]
- ▶ Coordinate Descent (CD)  
[Friedman et al., 2007; Kavukcuoglu et al., 2010]

## Adaptive Sparse Coding

### References

- Moreau, T. and Bruna, J. (2017). Understanding Neural Sparse Coding with Matrix Factorization. In *International Conference on Learning Representation (ICLR)*

## Adaptive Optimization

We have to solve  $N$  problems with a common structure  $D$ .

$$Z^{[n],*} = \operatorname{argmin}_{Z^{[n]}} \|X^{[n]} - \sum_{k=1}^K D_k * Z_k^{[n]}\|_2^2 + \lambda \|Z^{[n]}\|_1$$

**Can we use this structure to accelerate the resolution?**

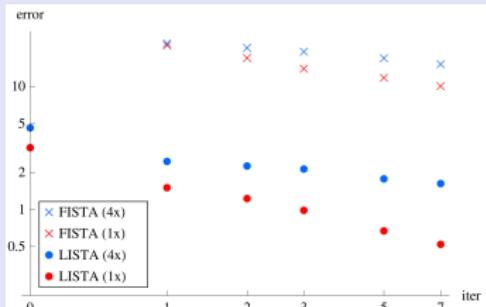
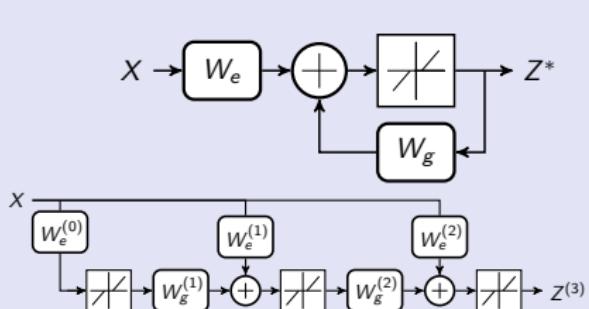
# Adaptive Optimization

We have to solve  $N$  problems with a common structure  $D$ .

$$Z^{[n],*} = \underset{Z^{[n]}}{\operatorname{argmin}} \|X^{[n]} - \sum_{k=1}^K D_k * Z_k^{[n]}\|_2^2 + \lambda \|Z^{[n]}\|_1$$

Can we use this structure to accelerate the resolution?

Yes, with the Learned ISTA [Gregor and Le Cun 2010, NeurIPS]



# Adaptive Optimization

We have to solve  $N$  problems with a common structure  $D$ .

$$Z^{[n],*} = \operatorname{argmin}_{Z^{[n]}} \|X^{[n]} - \sum_{k=1}^K D_k * Z_k^{[n]}\|_2^2 + \lambda \|Z^{[n]}\|_1$$

**Can we use this structure to accelerate the resolution?**

Yes, with the Learned ISTA [Gregor and Le Cun 2010, NeurIPS]

**Why does it work?** (non-convolutional setting).

- ▶ [Xin et al. 2016, NeurIPS]: Improved support recovery.
- ▶ [Giryes et al. 2018, IEEE TSP]: Leverage the sparsity pattern in  $Z$ .
- ▶ [Chen et al. 2018, NeurIPS]: Linear convergence if  $Z$  sparse enough.

# Adaptive Optimization

We have to solve  $N$  problems with a common structure  $\mathcal{D}$ .

$$Z^{[n],*} = \operatorname*{argmin}_{Z^{[n]}} \|X^{[n]} - \sum_{k=1}^K \mathcal{D}_k * Z_k^{[n]}\|_2^2 + \lambda \|Z^{[n]}\|_1$$

**Can we use this structure to accelerate the resolution?**

Yes, with the Learned ISTA [Gregor and Le Cun 2010, NeurIPS]

**Why does it work?** (non-convolutional setting).

- ▶ [Xin et al. 2016, NeurIPS]: Improved support recovery.
- ▶ [Giryes et al. 2018, IEEE TSP]: Leverage the sparsity pattern in  $Z$ .
- ▶ [Chen et al. 2018, NeurIPS]: Linear convergence if  $Z$  sparse enough.
- ▶ [Moreau and Bruna 2017, NeurIPS]: Leverage the structure of  $\mathcal{D}$ .

# Vectorized model

- ▶  $x$  is a vector in  $\mathbb{R}^T$
- ▶  $\epsilon$  is a noise vector in  $\mathbb{R}^T$
- ▶  $D$  is a matrix in  $\mathbb{R}^{T \times LK}$
- ▶  $z$  is a coding vector in  $\mathbb{R}^{LK}$

**Sparse Linear model:**

$$x = Dz + \epsilon$$

with  $z$  sparse.

Few of its coefficients are non-zero.

$$D = \begin{pmatrix} & \begin{matrix} \bar{D} & \bar{0} \\ \bar{0} & \bar{D} \end{matrix} & \begin{matrix} \bar{0} \\ \bar{0} \end{matrix} \\ & \begin{matrix} \bar{0} & \bar{D} & \bar{0} \\ \bar{0} & \bar{D} & \bar{0} \\ \bar{0} & \bar{D} & \bar{0} \end{matrix} & \begin{matrix} \bar{0} \\ \bar{0} \end{matrix} \\ & \vdots & \vdots \\ & \begin{matrix} \bar{0} & \bar{D} & \bar{0} \\ \bar{0} & \bar{D} & \bar{0} \\ \bar{0} & \bar{D} & \bar{0} \end{matrix} & \begin{matrix} \bar{0} \\ \bar{0} \end{matrix} \end{pmatrix}$$

## Notations

Consider the sparse coding problem with a dictionary  $D$ .

$$z^* = \underset{z}{\operatorname{argmin}} F(z) = \underbrace{\frac{1}{2} \|x - Dz\|_2^2}_{E(z)} + \lambda \|z\|_1$$

We denote  $B = D^T D$  is the Gram matrix of  $D$ .

**We introduce a novel class of algorithms – FacNet – based on a sparse factorization of  $B$ .**

Quadratic form:  $Q_S(u, v) = \frac{1}{2}(u - v)^T S(u - v) + \lambda \|u\|_1$  .

Note that  $F(z) = Q_B(z, D^\dagger x)$

For  $S$  is diagonal,  $\operatorname{argmin}_u Q_S(u, v)$  can be efficiently minimized as the problem is separable on each coordinate.

Given an estimate  $z^{(q)}$  of  $z^*$  at iteration  $q$ , we can write:

$$\begin{aligned} F(z) &= E(z) + \lambda \|z\|_1 \\ &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(z, z^{(q)}) , \end{aligned}$$

Given an estimate  $z^{(q)}$  of  $z^*$  at iteration  $q$ , we can write:

$$\begin{aligned} F(z) &= E(z) + \lambda \|z\|_1 \\ &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(z, z^{(q)}) , \end{aligned}$$

ISTA: Replace  $B$  by diagonal matrix  $\textcolor{red}{S} = \|B\|_2 \mathbf{I}_K$

$$F_q(z) = E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_{\textcolor{red}{S}}(z, z^{(q)}) ,$$

$$\min_z F_q(z) \Leftrightarrow \min_z Q_{\textcolor{red}{S}} \left( z, z^{(q)} - \textcolor{red}{S}^{-1} \nabla E(z^{(q)}) \right)$$

Given an estimate  $z^{(q)}$  of  $z^*$  at iteration  $q$ , we can write:

$$\begin{aligned} F(z) &= E(z) + \lambda \|z\|_1 \\ &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(z, z^{(q)}) , \end{aligned}$$

ISTA: Replace  $B$  by diagonal matrix  $\textcolor{red}{S} = \|B\|_2 \mathbf{I}_K$

FacNet: Replace  $B$  by  $A^T S A$  ( $S$  diagonal,  $A$  unitary)

$$\widetilde{F}_q(z) = E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_{\mathbf{S}_q}(\mathbf{A}_q z, \mathbf{A}_q z^{(q)}) ,$$

$$\min_z \widetilde{F}_q(z) \Leftrightarrow \min_z Q_{\mathbf{S}_q} \left( \mathbf{A}_q z, \mathbf{A}_q z^{(q)} - \mathbf{S}_q^{-1} \mathbf{A}_q \nabla E(z^{(q)}) \right)$$

Given an estimate  $z^{(q)}$  of  $z^*$  at iteration  $q$ , we can write:

$$\begin{aligned} F(z) &= E(z) + \lambda \|z\|_1 \\ &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_B(z, z^{(q)}) , \end{aligned}$$

ISTA: Replace  $B$  by diagonal matrix  $\textcolor{red}{S} = \|B\|_2 \mathbf{I}_K$

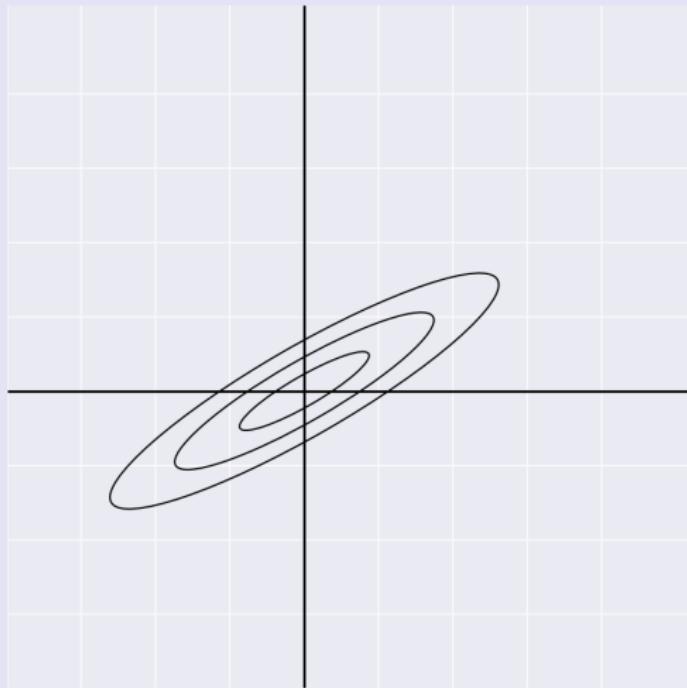
FacNet: Replace  $B$  by  $A^T S A$  ( $S$  diagonal,  $A$  unitary)

$$\widetilde{F}_q(z) = E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_{\textcolor{red}{S}_q}(\textcolor{blue}{A}_q z, \textcolor{blue}{A}_q z^{(q)}) ,$$

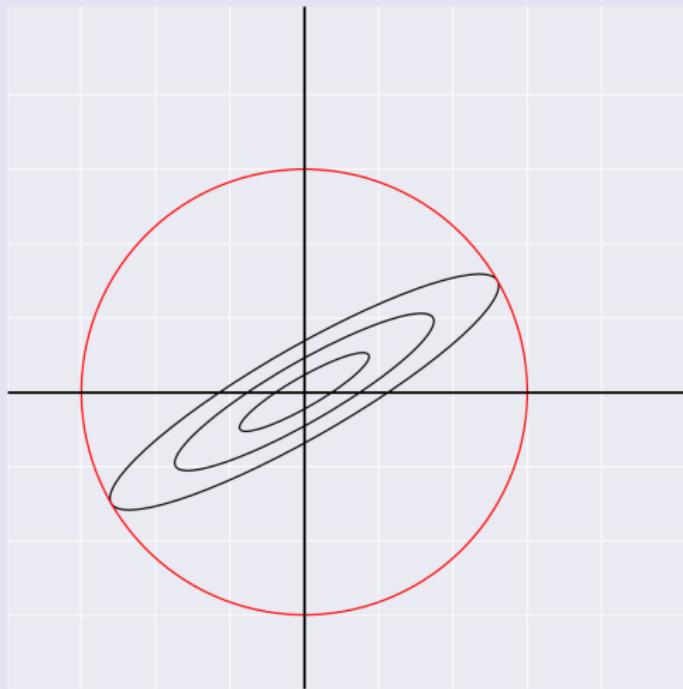
$$\min_z \widetilde{F}_q(z) \Leftrightarrow \min_z Q_{\textcolor{red}{S}_q} \left( \textcolor{blue}{A}_q z, \textcolor{blue}{A}_q z^{(q)} - \textcolor{red}{S}_q^{-1} \textcolor{red}{A}_q \nabla E(z^{(q)}) \right)$$

Can we choose  $A_q, S_q$  to accelerate the optimization compared to ISTA?

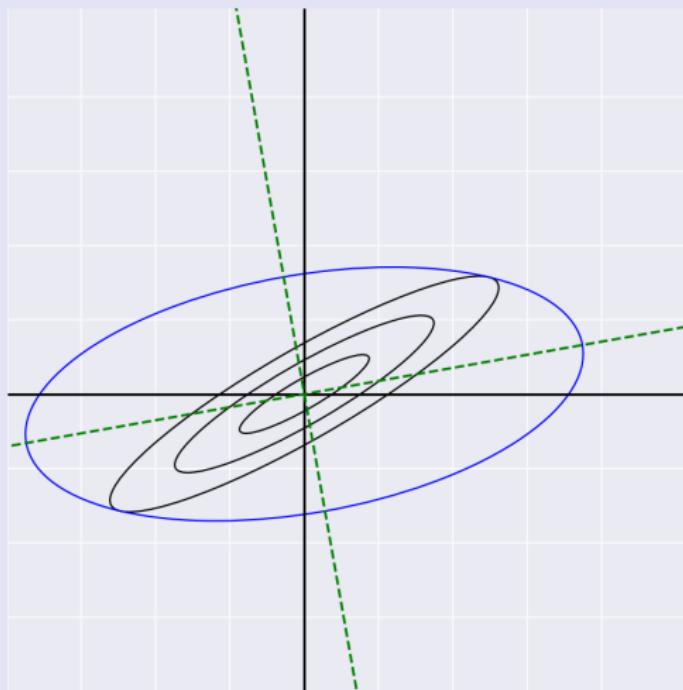
## Toward an adaptive procedure



## Toward and adaptive procedure



## Toward and adaptive procedure



## Toward an adaptive procedure

Similar iterative procedure with steps adapted to the problem topology.

$$\widetilde{F}_q(z) = F(z) + (z - z^{(q)})^T R(z - z^{(q)}) + \delta_A(z)$$

Tradeoff between:

- ▶ Rotation to align the norm  $\|\cdot\|_B$  and the norm  $\|\cdot\|_1$  , Computation

$$R = A^T S A - B$$

- ▶ Deformation of the  $\ell_1$ -norm with the rotation  $A$  . Accuracy

$$\delta_A(z) = \lambda \left( \|Az\|_1 - \|z\|_1 \right)$$

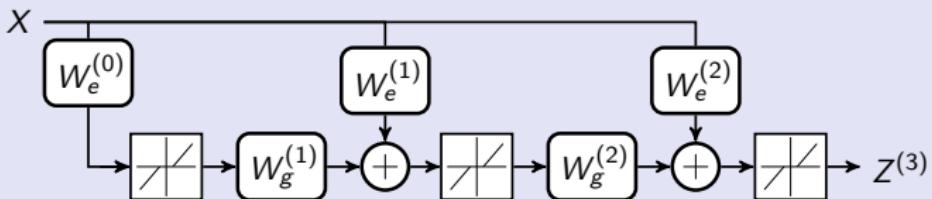
## Theoretical results

- ▶ We showed that FacNet has the same asymptotic convergence rate as ISTA in  $\mathcal{O}(\frac{1}{q})$ .
- ▶ The constant factors are different and can be improved. If the factorization  $(A_q, S_q)$  at iteration  $q$  verifies

$$\|R_q\|_2 + 2 \frac{L_{A_q}(z^{(q+1)})}{\|z^* - z^{(q)}\|_2} \leq \frac{\|B\|_2}{2}$$

and  $A_p = \mathbf{I}_K, S_p = \|B\|_2 \mathbf{I}_K$  for  $p > q$ , then the procedure has improved convergence rate compared to ISTA.

⇒ There is a phase transition when  $\|z^{(q)} - z^*\|_2 \rightarrow 0$

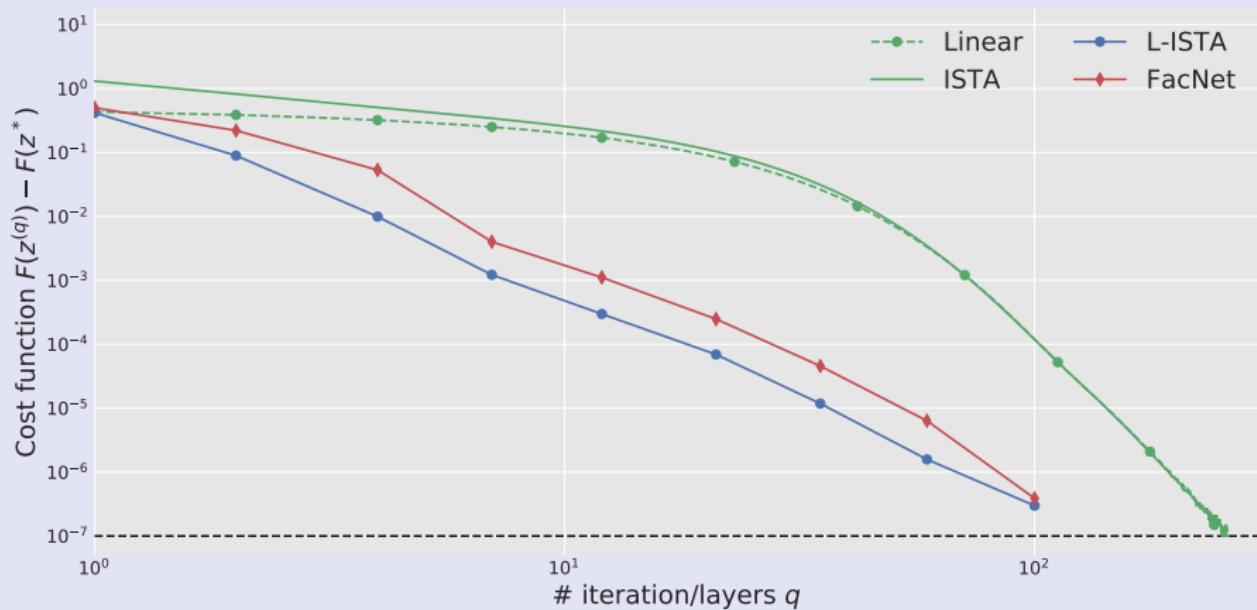


With  $W_e = \frac{D^T}{\|B\|_2}$  and  $W_g = I - \frac{B}{\|B\|_2}$ , this network computes ISTA.

**FacNet:** Specialization of LISTA with  $\begin{cases} W_e &= S^{-1}AD^T \\ W_g &= A^T - S^{-1}ABA^T \end{cases}$

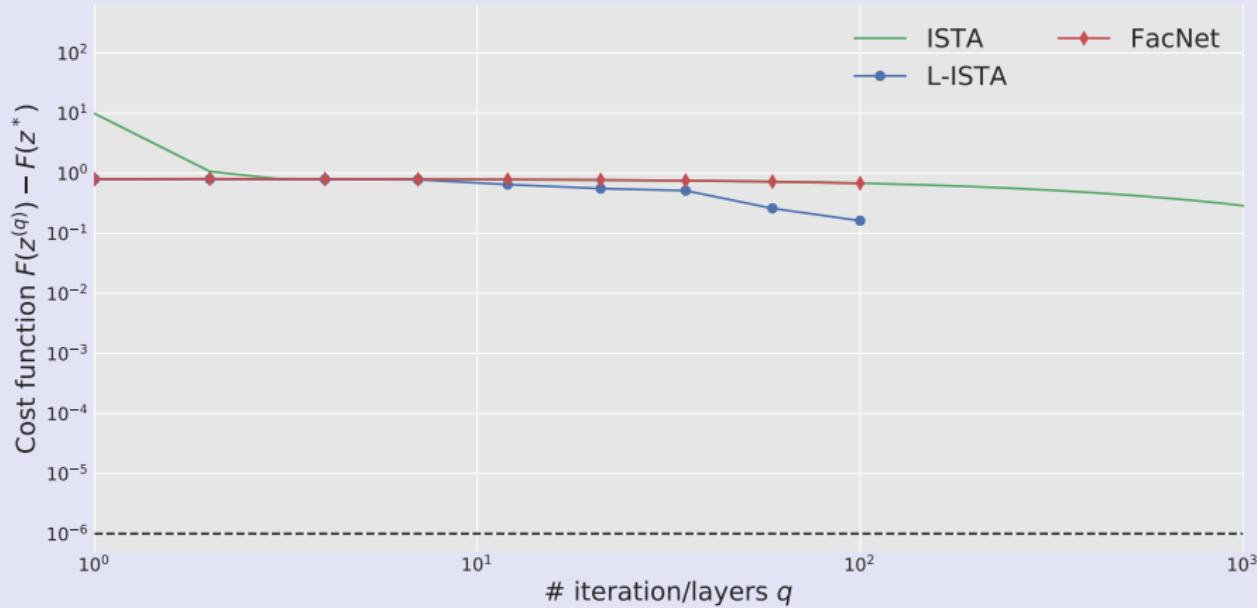
⇒ LISTA can be at least as good as this model.

# Generic Dictionary



$K$  generic atoms (uniform in  $S^{P-1}$ ) with Bernoulli-Gaussian activation.  
Params:  $K = 100$ ,  $P = 64$ ,  $\rho = 1/20$ ,  $\sigma = 10$  and  $\lambda = 0.01$

## Adversarial dictionary



Same parameters with adverse dictionary (dense eigen-spaces).

### Take home message

- ▶ Non asymptotic acceleration of ISTA is possible based on the structure of  $D$ ,
- ▶ Sufficient analysis to explain LISTA acceleration,
- ▶ Empirically showed the structure of  $D$  is necessary for LISTA.

### Ahead of us

- ▶ Improve the factorization formulation for direct optimization,
- ▶ Adaptation of the analysis to convolutional sparse coding,
- ▶ Explore the link with sparse eigenvectors of the gram matrix.

## Scaling up Convolutional Sparse Coding with coordinate descent and distributed optimization

### References

- ▶ Moreau, T., Oudre, L., and Vayatis, N. (2018). [DICOD: Distributed Convolutional Sparse Coding](#). In *International Conference on Machine Learning (ICML)*, pages 3626–3634, Stockholm, Sweden. PMLR (80)
- ▶ Moreau, T. and Gramfort, A. (2019). [Distributed Convolutional Dictionary Learning \(DiCoDiLe\): Pattern Discovery in Large Images and Signals](#). *preprint ArXiv*, 1901.09235

## $Z$ -step: Sparse coding

$\rightarrow \mathbf{D}$  fixed, update  $Z$

$$Z^{[n],*} = \operatorname*{argmin}_{Z^{[n]}} \|X^{[n]} - \sum_{k=1}^K \mathbf{D}_k * Z_k^{[n]}\|_2^2 + \lambda \|Z^{[n]}\|_1$$

$\Rightarrow$  Independent for each  $n \in [1, N]$

### Related Algorithms:

- ▶ Iterative Soft-Thresholding Algorithm (ISTA)  
[Daubechies et al., 2004; Chalasani et al., 2013]
- ▶ Fast ISTA  
[Beck and Teboulle, 2009; Wohlberg, 2016]
- ▶ Alternated Direction Method of Multiplier (ADMM)  
[Gabay and Mercier, 1976; Bristow et al., 2013]
- ▶ Coordinate Descent (CD)  
[Friedman et al., 2007; Kavukcuoglu et al., 2010]

# Coordinate Descent (CD)

Minimize

$$Z^* = \operatorname{argmin}_Z \|X - \sum_{k=1}^K D_k * Z_k\|_2^2 + \lambda \|Z\|_1$$

Update one coordinate at each iteration.

1. Select a coordinate  $(k_0, t_0)$  to update.

Three algorithms for LASSO:

- ▶ Cyclic updates;  $\mathcal{O}(1)$  [Friedman et al., 2007]
- ▶ Random updates;  $\mathcal{O}(1)$  [Nesterov, 2010]
- ▶ Greedy updates;  $\mathcal{O}(KL)$  [Osher and Li, 2009]

## Coordinate Descent (CD)

Minimize

$$Z^* = \operatorname{argmin}_Z \|X - \sum_{k=1}^K D_k * Z_k\|_2^2 + \lambda \|Z\|_1$$

Update one coordinate at each iteration.

1. Select a coordinate  $(k_0, t_0)$  to update.
2. Compute a new value  $Z'_{k_0}[t_0]$  for this coordinate

For convolutional CD, we can use optimal updates:

$$Z'_{k_0}[t_0] = \frac{1}{\|D_{k_0}\|_2^2} \mathbf{ST}(\beta_{k_0}[t_0], \lambda),$$

with  $\mathbf{ST}(y, \lambda) = \operatorname{sign}(y)(|y| - \lambda)_+$ . [Kavukcuoglu et al. \[2010\]](#) showed this can be done efficiently, with  $\mathcal{O}(KW)$  operations.

## Coordinate Descent (CD)

Minimize

$$Z^* = \operatorname{argmin}_Z \|X - \sum_{k=1}^K D_k * Z_k\|_2^2 + \lambda \|Z\|_1$$

Update one coordinate at each iteration.

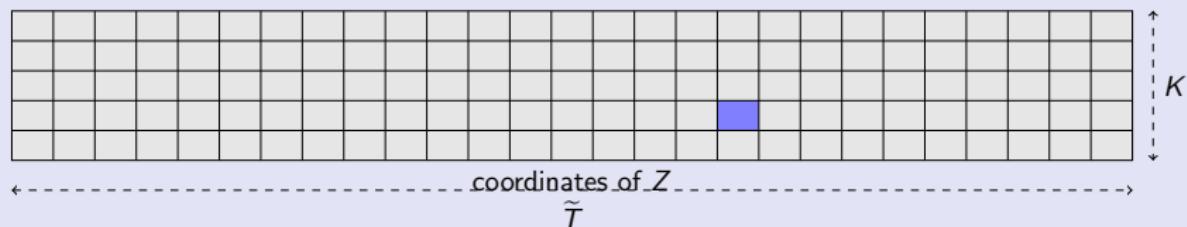
1. Select a coordinate  $(k_0, t_0)$  to update.
2. Compute a new value  $Z'_{k_0}[t_0]$  for this coordinate

$\Rightarrow$  Converges to the optimal point for CSC problem in  $\mathcal{O}\left(\frac{1}{q}\right)$  iterations.

Trade-off between cheap computational complexity (random/cyclic CD) and importance sampling with faster convergence (Greedy CD).

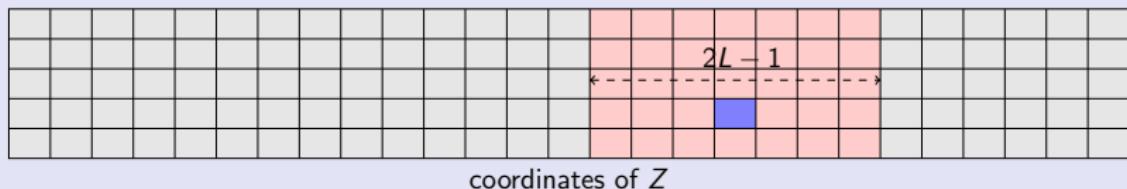
[Nutini et al., 2015; Karimireddy et al., 2019]

We introduced the LGCD method which is an extension of GCD.



GCD has  $\mathcal{O}(K\tilde{T})$  computational complexity.

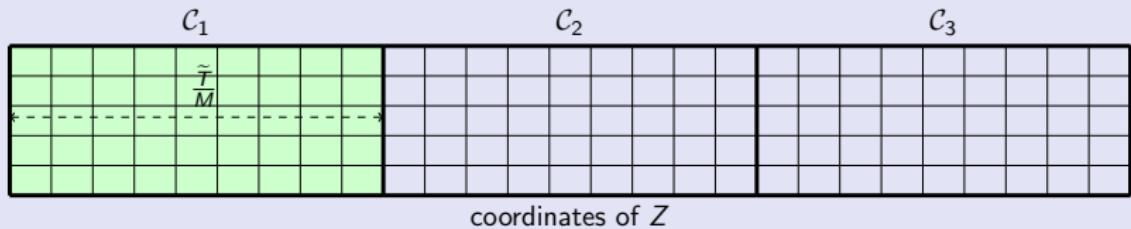
We introduced the LGCD method which is an extension of GCD.



GCD has  $\mathcal{O}(K\tilde{T})$  computational complexity.

But the update itself has complexity  $\mathcal{O}(KL)$

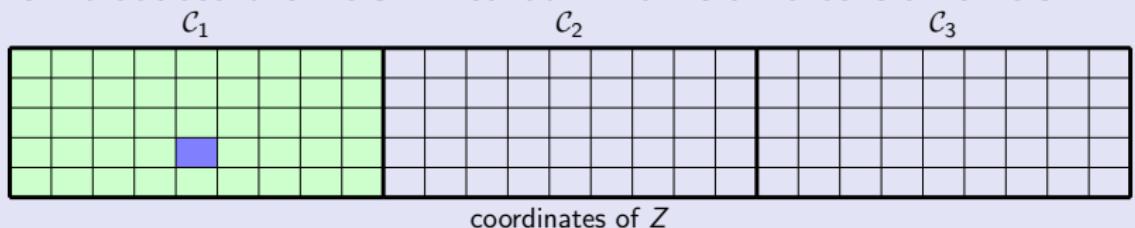
We introduced the LGCD method which is an extension of GCD.



With a partition  $\mathcal{C}_m$  of the signal domain  $[1, K] \times [0, \tilde{T}]$ ,

$$\mathcal{C}_m = [1, K] \times \left[ \frac{(m-1)\tilde{T}}{M}, \frac{m\tilde{T}}{M} \right]$$

We introduced the LGCD method which is an extension of GCD.



With a partition  $\mathcal{C}_m$  of the signal domain  $[1, K] \times [0, \tilde{T}]$ ,

$$\mathcal{C}_m = [1, K] \times \left[ \frac{(m-1)\tilde{T}}{M}, \frac{m\tilde{T}}{M} \right]$$

The coordinate to update is chosen greedily on a sub-domain  $\mathcal{C}_m$

$$\frac{\tilde{T}}{M} = 2L - 1 \Rightarrow \mathcal{O}(\text{Coordinate selection}) = \mathcal{O}(\text{Coordinate Update})$$

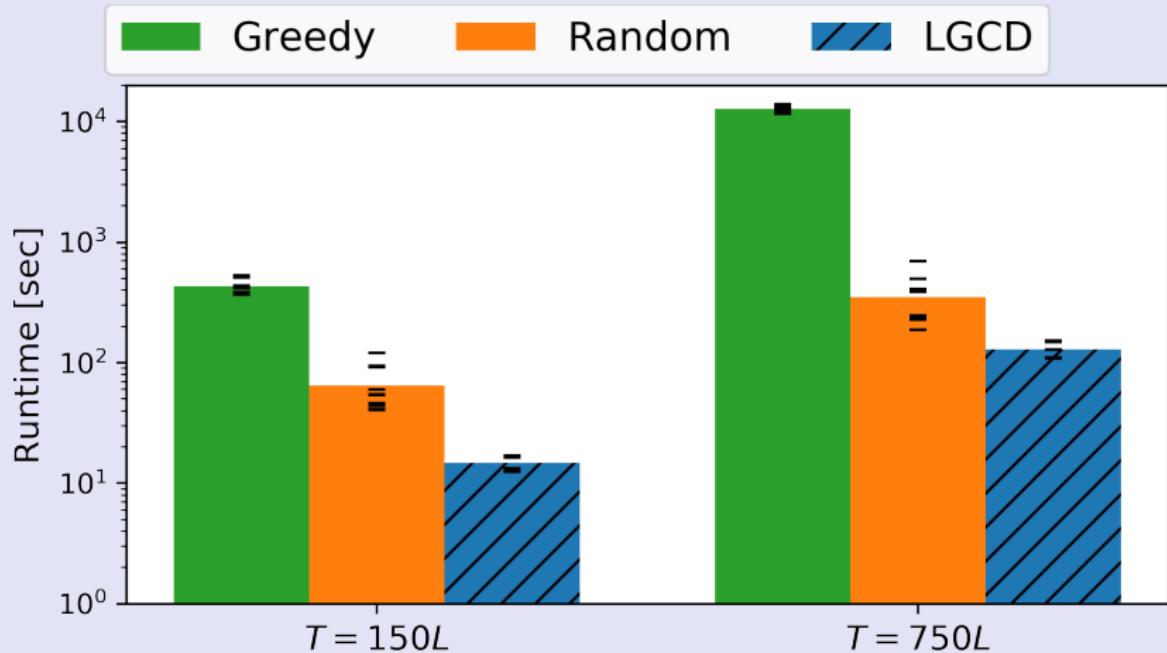
The overall iteration complexity is  $\mathcal{O}(KL)$  instead of  $\mathcal{O}(K\tilde{T})$ .

$\Rightarrow$  Efficient for sparse  $Z$

## Fast optimization

Comparison of the coordinate selection strategy for CD on simulated signals

We set  $K = 10$ ,  $L = 150$ ,  $\lambda = 0.1\lambda_{\max}$



## Weak dependence of the coordinate updates

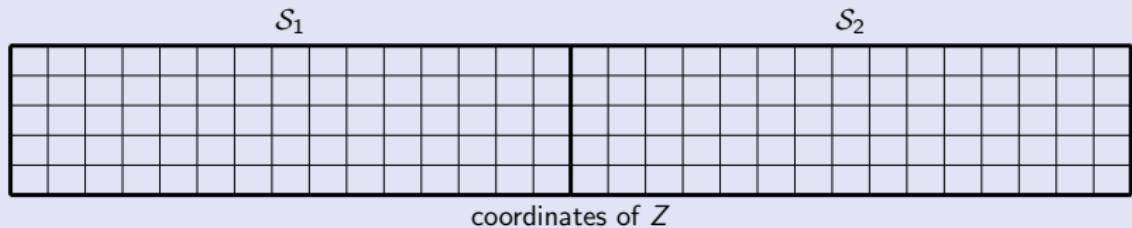
The update of the  $W$  coordinates  $(k_w, \omega_w)_{w=1}^W$  with additive update  $\Delta Z_{k_w}[\omega_w]$  changes the cost by:

$$\Delta E = \underbrace{\sum_{i=1}^W \Delta E_w}_{\text{iterative steps}} - \underbrace{\sum_{w \neq w'} (d_{k_w} * d_{k_{w'}}^\top) [\omega_{w'} - \omega_w] \Delta Z_{k_w}[\omega_w] \Delta Z_{k_{w'}}[\omega_{w'}]}_{\text{interference}},$$

⇒ If the updates are far enough, they can be considered as independent.

# Distributed Convolutional Coordinate Descent

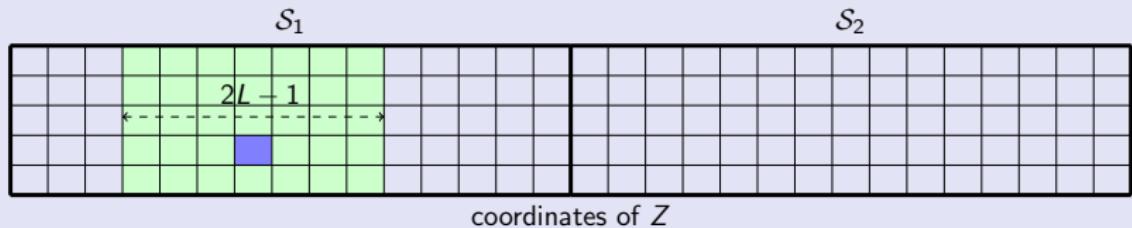
[Moreau et al. 2018, ICML]



- ▶ Split the coordinates in continuous sub-segment  $\mathcal{S}_w = \left[ \frac{(w-1)T}{W}, \frac{wT}{W} \right]$

# Distributed Convolutional Coordinate Descent

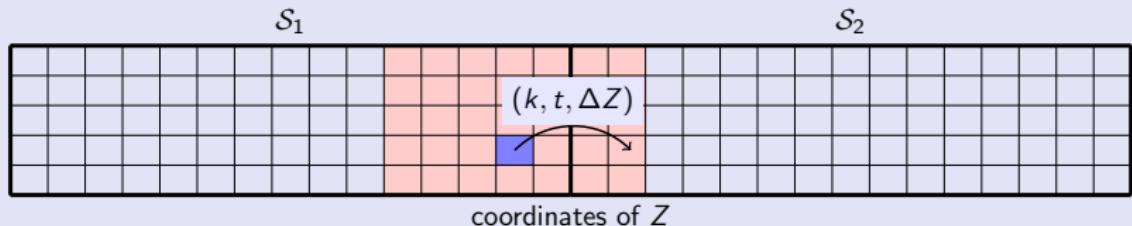
[Moreau et al. 2018, ICML]



- ▶ Split the coordinates in continuous sub-segment  $\mathcal{S}_w = \left[ \frac{(w-1)T}{W}, \frac{wT}{W} \right]$ .
- ▶ Use CD updates in parallel in each sub-segment.

# Distributed Convolutional Coordinate Descent

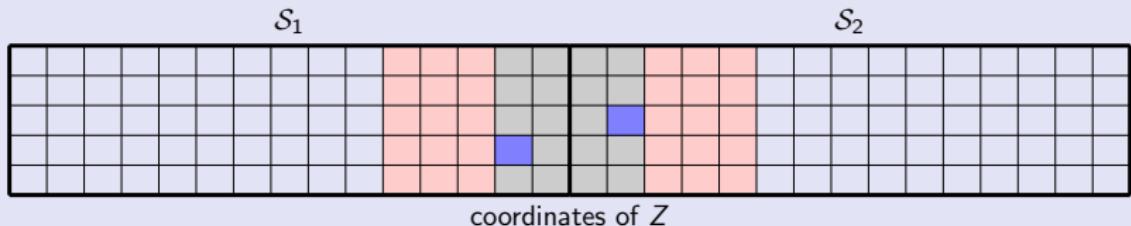
[Moreau et al. 2018, ICML]



- ▶ Split the coordinates in continuous sub-segment  $\mathcal{S}_w = \left[ \frac{(w-1)T}{W}, \frac{wT}{W} \right]$ .
- ▶ Use CD updates in parallel in each sub-segment.
- ▶ Notify neighbor workers when the update is on the border of  $\mathcal{S}_w$ .

# Distributed Convolutional Coordinate Descent

[Moreau et al. 2018, ICML]



- ▶ Split the coordinates in continuous sub-segment  $\mathcal{S}_w = \left[ \frac{(w-1)T}{W}, \frac{wT}{W} \right]$ .
- ▶ Use CD updates in parallel in each sub-segment.
- ▶ Notify neighbor workers when the update is on the border of  $\mathcal{S}_w$ .
- ▶ What do we do when two updates are interfering?

DICOD converges to the solution of the CSC for 1D signals without having a control mechanism on the interference.

### Theorem (Convergence of DICOD)

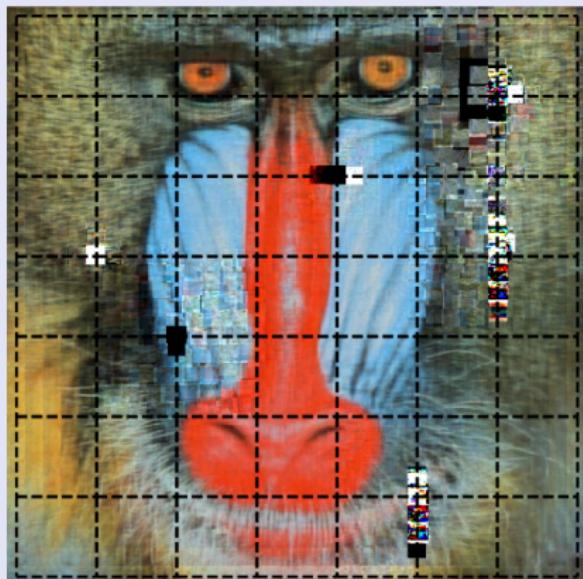
We consider the following assumptions:

- H1:** If the cross correlation between atoms of  $\mathcal{D}$  is strictly smaller than 1.
- H2:** No cores stop before all its coefficients are optimal.
- H3:** If the delay in communication between the processes is inferior to the update time.

Under these assumptions, the DICOD algorithm converges asymptotically to the optimal solution  $Z^*$  of CSC.

# Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)

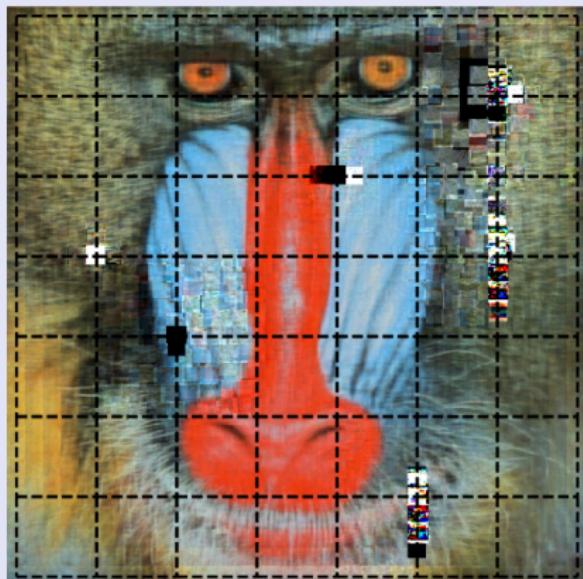
[Moreau and Gramfort 2019, preprint]



- ▶ DICOD does not work for higher dimensional signals.

# Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)

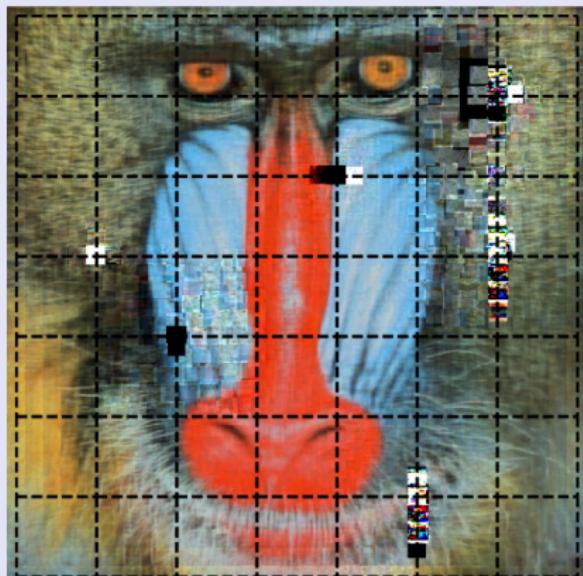
[Moreau and Gramfort 2019, preprint]



- ▶ DICOD does not work for higher dimensional signals.
- ▶ Extension require to control interferences.

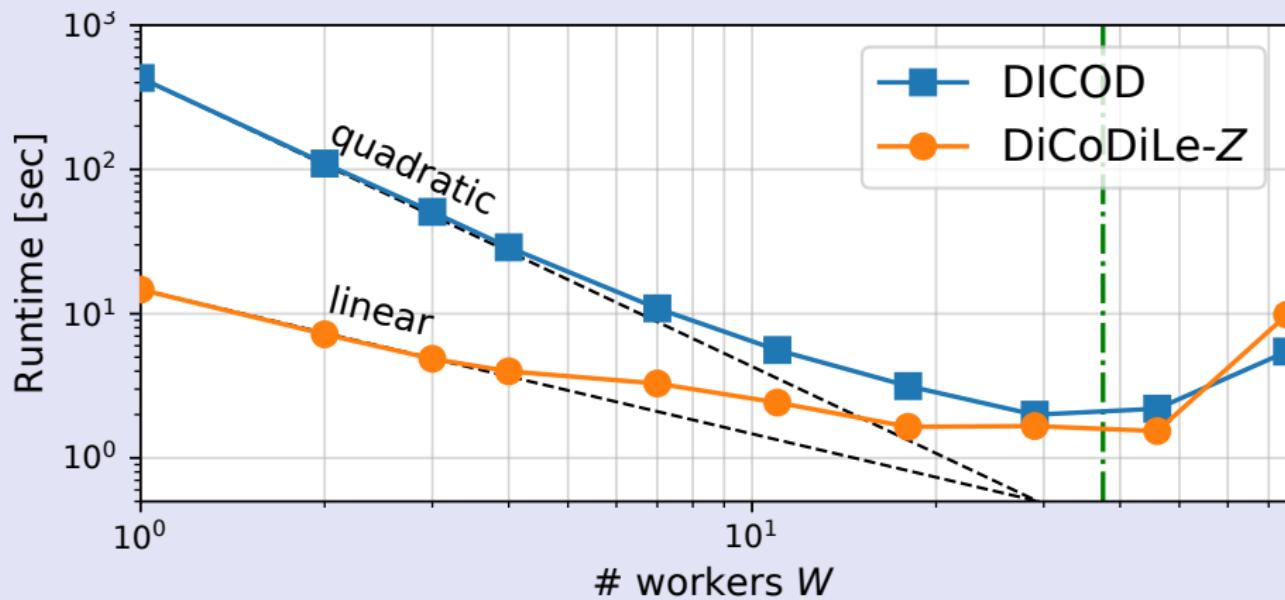
# Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)

[Moreau and Gramfort 2019, preprint]



- ▶ DICOD does not work for higher dimensional signals.
- ▶ Extension require to control interferences.
- ▶ Use asynchronous mechanism:  
Soft-lock.

## Numerical speed-up



Running time as a function fo the number of workers  $W$ .

## Recap Part II

### Take home message

- ▶ LGCD is a very efficient algorithm when working with CSC for long signals.
- ▶ Can be distributed efficiently for multi-dimensional signals,
- ▶ Good scaling properties with the number of workers  $W$  used to distribute the algorithm.

### Ahead of us

- ▶ Extend this algorithm to local penalization such as Group LASSO.
- ▶ This algorithm could be used for algorithm such as MP for  $\ell_0$  or  $\ell_{0,\infty}$  penalties.

## Rank-1 Constrained Convolutional Dictionary Learning

### References

- ▶ Dupré la Tour, T., Moreau, T., Jas, M., and Gramfort, A. (2018).  
[Multivariate Convolutional Sparse Coding for Electromagnetic Brain Signals.](#)  
In *Advances in Neural Information Processing Systems (NeurIPS)*, pages  
3296–3306, Montreal, Canada

## D-step: solving for the atoms

The dictionary update is performed by minimizing

$$\min_{\|\mathbf{D}_k\|_2 \leq 1} E(\{\mathbf{D}_k\}_k) \triangleq \sum_{n=1}^N \frac{1}{2} \|X^n - \sum_{k=1}^K z_k^n * \mathbf{D}_k\|_2^2 . \quad (1)$$

Computing  $\nabla_{\mathbf{D}_k} E(\{\mathbf{D}_k\}_k)$  can be done efficiently

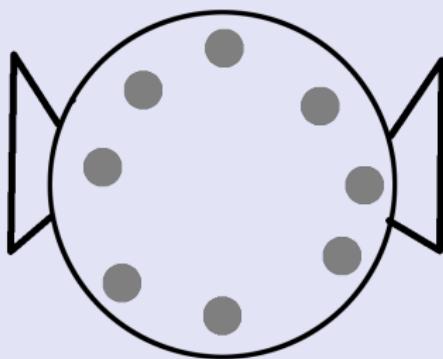
$$\nabla_{d_k} E(\{\mathbf{D}_k\}_k) = \sum_{n=1}^N (z_k^n)^\top * \left( X^n - \sum_{l=1}^K z_l^n * \mathbf{D}_l \right) = \Phi_k - \sum_{l=1}^K \Psi_{k,l} * \mathbf{D}_l ,$$

$\Rightarrow$  Save with Projected Gradient Descent (PGD) with an Armijo backtracking line-search for the D-step [Wright and Nocedal, 1999].

However, this model does not account for the physics of the problem.

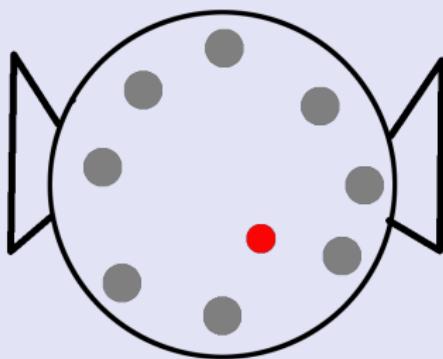
## EM wave diffusion

- ▶ Recording here with 8 sensors



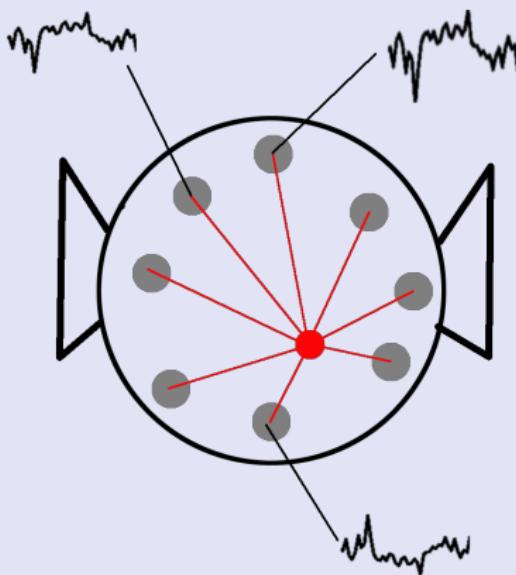
## EM wave diffusion

- ▶ Recording here with 8 sensors
- ▶ EM activity in the brain



## EM wave diffusion

- ▶ Recording here with 8 sensors
- ▶ EM activity in the brain
- ▶ The electric field is spread **linearly** and **instantaneously** over all sensors (Maxwell equations)



## Multivariate CSC with rank-1 constraint

[Dupré la Tour et al. 2018, NeurIPS]

**Idea:** Impose a rank-1 constraint on the dictionary atoms  $D_k$

To make the problem tractable, we decided to use auxiliary variables  $u_k$  and  $v_k$  s.t.  $D_k = u_k v_k^\top$ .

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

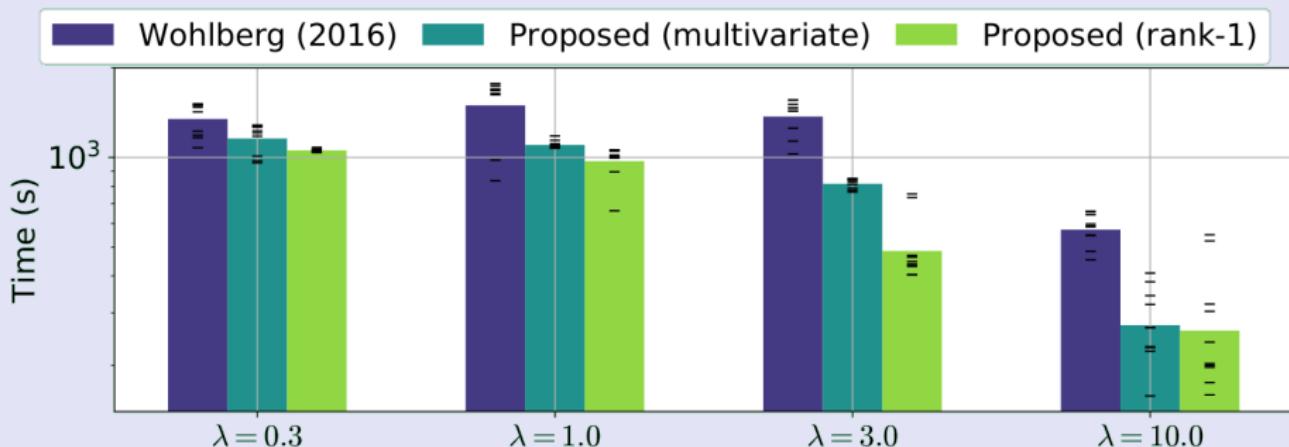
Here,

- ▶  $u_k \in \mathbb{R}^P$  is the spatial pattern of our atom
- ▶  $v_k \in \mathbb{R}^L$  is the temporal pattern of our atom

⇒ Tri-convex optimization problem , solved with alternate minimization.

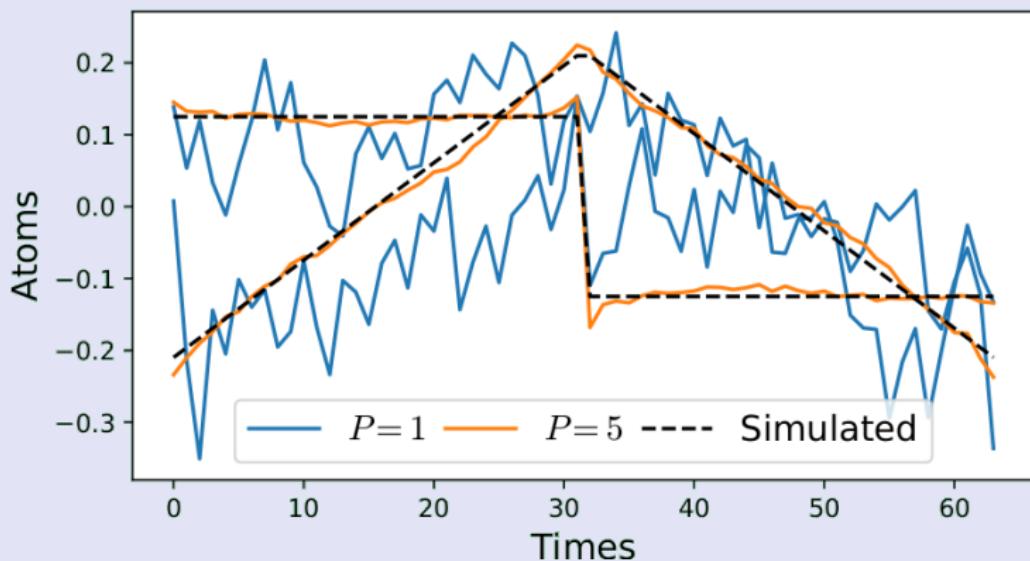
## Fast optimization

Comparison with multivariate methods on somato dataset with  
 $T = 134,700$ ,  $K = 8$ ,  $P = 5$  and  $L = 128$



## Pattern recovery

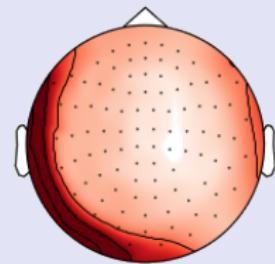
Patterns recovered with  $P = 1$  and  $P = 5$ . The signals were generated with the two simulated temporal patterns and with  $\sigma = 10^{-3}$ .



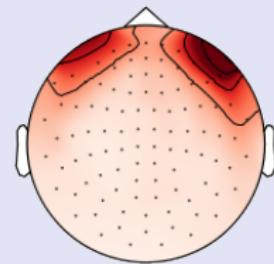
# MNE somatosensory data

A selection of temporal waveforms of the atoms learned on the MNE sample dataset.

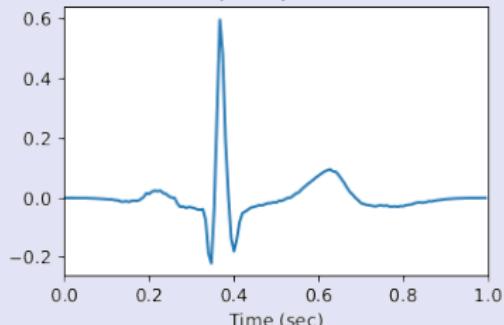
Spatial pattern 0  
Explained variance 5.62 %



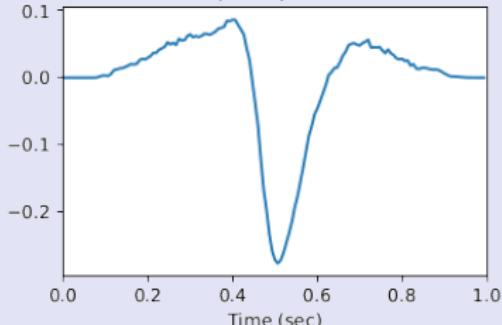
Spatial pattern 1  
Explained variance 2.38 %



Temporal pattern 0

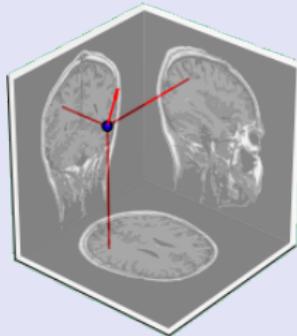
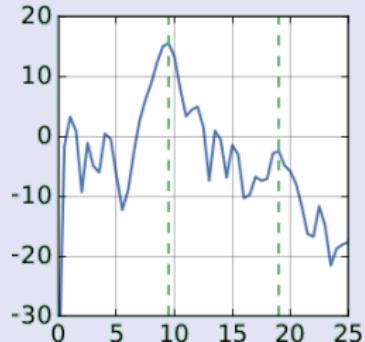
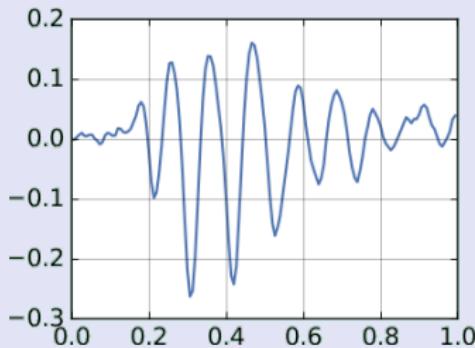


Temporal pattern 1



## MNE somatosensory data

Atoms revealed using the MNE somatosensory data. Note the non-sinusoidal comb shape of the mu rhythm.



## Recap Part III

### Take home message

- ▶ The structure of the learned dictionary can be constrained to improve the interpretability of the recovered patterns.
- ▶ Can lead to more efficient algorithm and better recovery property.
- ▶ Open source package  <https://alphacsc.github.io>

### Ahead of us

- ▶ Analysis of the patterns learned on large MEG database (HCP).
- ▶ Link between the learned waveforms and information propagation properties in the brain.
- ▶ Extension to scale invariant CDL to study frequency coupling in the brain.

## Convolutional Dictionary Learning

- ▶ Flexible pattern extraction technique,
- ▶ Computationally tractable for more and more problems,
- ▶ Some application are already beginning to emerge.

## Challenges

- ▶ Theoretical challenges remains (convergence, recoverability),
- ▶ The evaluation (and thus the parameter choices) is still not clear,
- ▶ Can give some insight for deep learning models?

# Thanks!

Code available online:

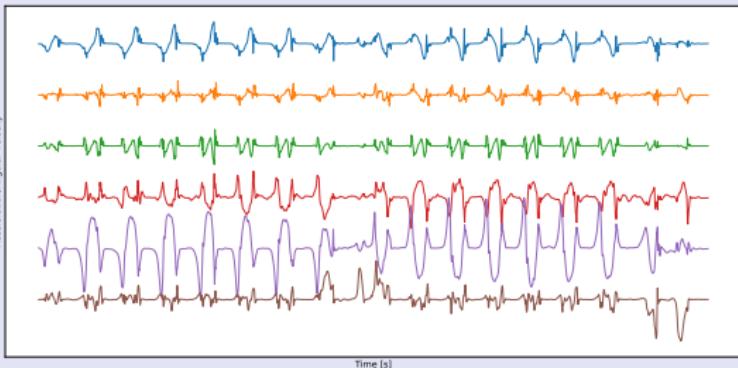
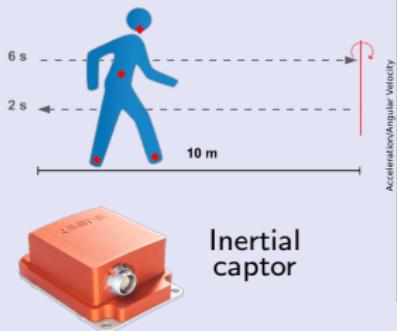
- ⌚ **LISTA** : [github.com/tommoral/AdaptiveOptim](https://github.com/tommoral/AdaptiveOptim)
- ⌚ **DICOD** (& DiCoDiLe soon) : [github.com/tommoral/dicod](https://github.com/tommoral/dicod)
- ⌚ **alphacsc** : [alphacsc.github.io](https://alphacsc.github.io)

Slides are on my web page:

 [tommoral.github.io](https://tommoral.github.io)

 [@tomamoral](https://twitter.com/tomamoral)

# Signals from human walking



- ▶ Shift invariant patterns linked to steps,
- ▶ Manual segmentation of the signal is expensive.

⇒ Can we do better with data-driven approach?

## Experiment

Create a dictionary with 25 Gaussian patterns ( $W = 90$ )

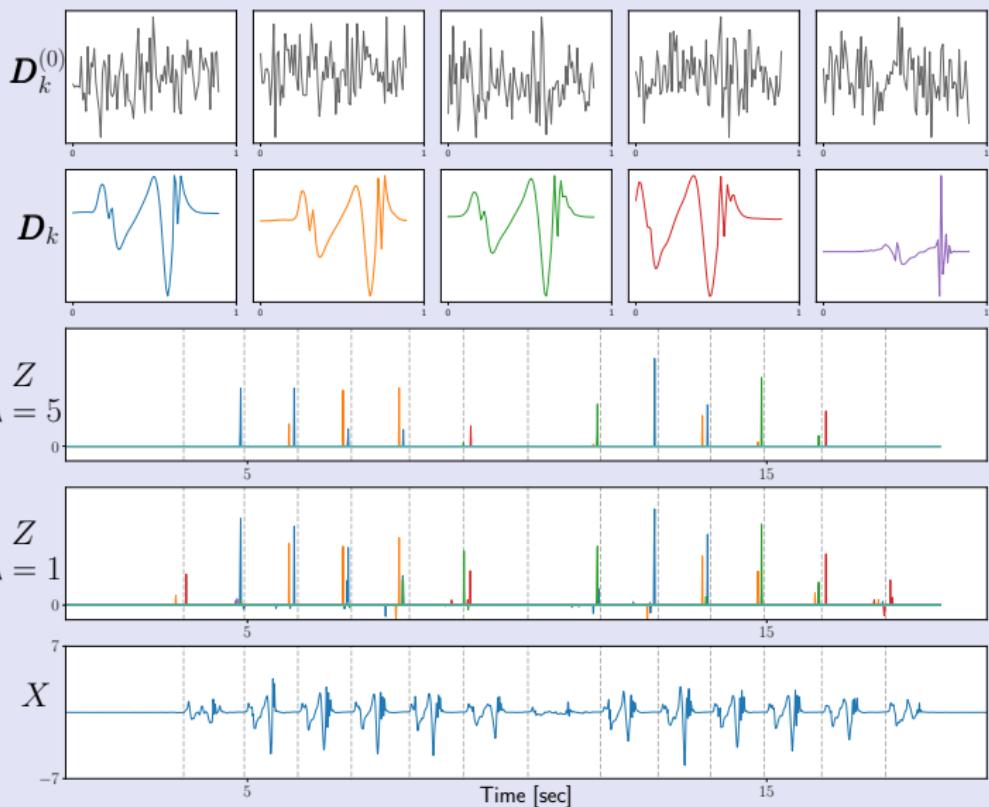
$$\mathcal{D}_k^{(0)} \sim \mathcal{N}(0, I_{90})$$

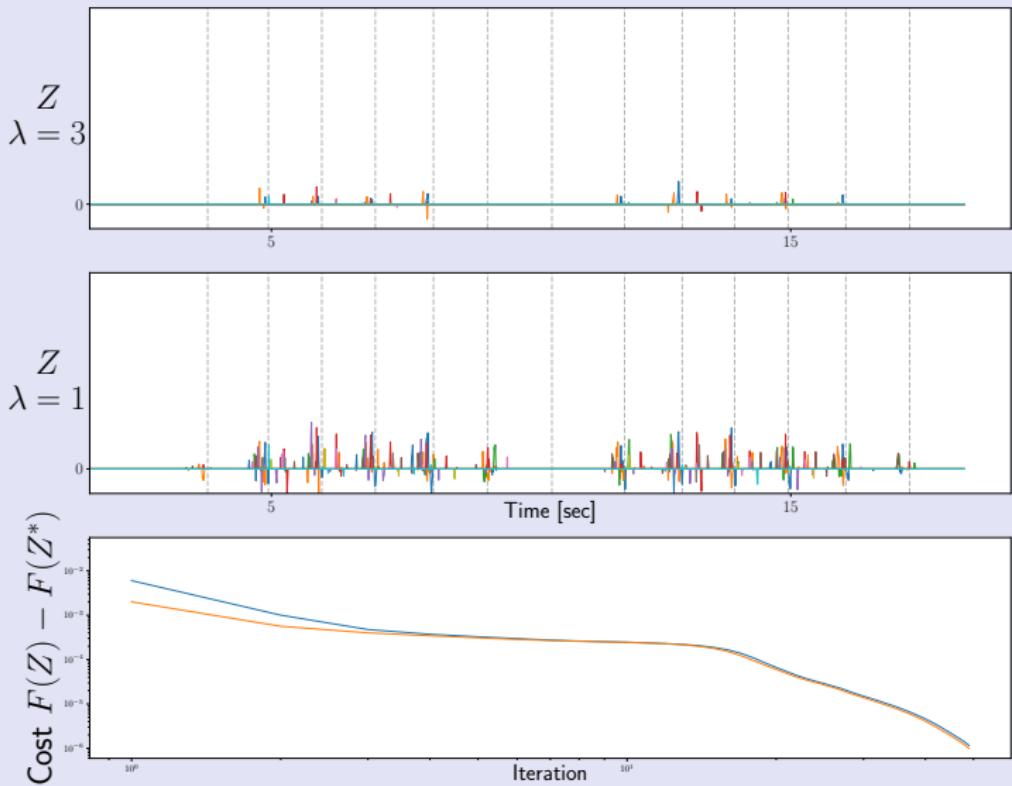
Use the Convolutional Dictionary Learning with DICOD to learn a dictionary  $\mathcal{D}$  on a set of 50 recording of healthy subjects walking.

## Challenges

- ▶ Alignment of the patterns,
- ▶ Detect steps of different amplitude,
- ▶ Handle multivariate signals.

# Experiment





## Related works

---

- ▶ Giryes et al. [2018]: Propose the inexact projected gradient descent and conjecture that LISTA accelerate the LASSO resolution by learning the sparsity pattern of the input distribution.
- ▶ Xin et al. [2016]: Study the Hard-thresholding Algorithm and its capacity to recover the support of a sparse vector.  
The paper relax the RIP conditions for the dictionary.

## Generic Dictionaries

---

A dictionary  $D \in \mathbb{R}^{p \times K}$  is a generic dictionary when its columns  $D_i$  are drawn uniformly over the  $\ell_2$  unit sphere  $\mathcal{S}^{p-1}$ .

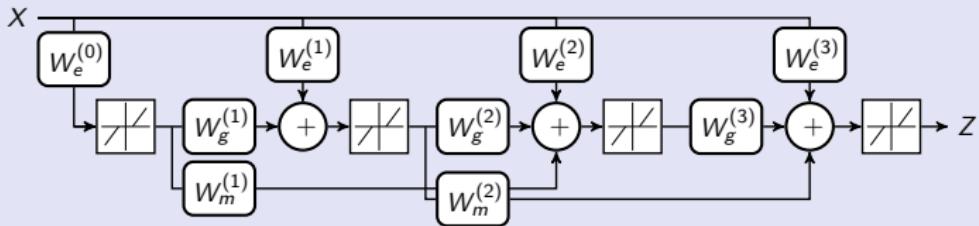
## Theorem (Generic Acceleration)

In expectation over the generic dictionary  $D$ , the factorization algorithm using a diagonally dominant matrix  $A \subset \mathcal{E}_\delta$ , has better performance for iteration  $q + 1$  than the normal ISTA iteration – which uses the identity – when

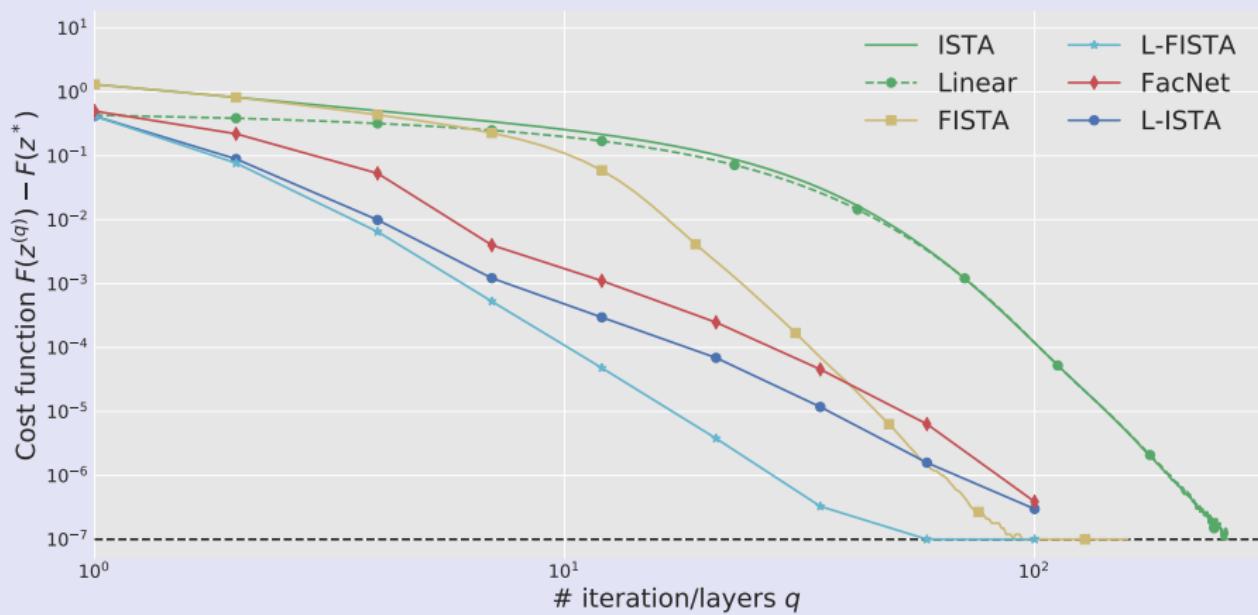
$$\lambda \mathbb{E}_z \left[ \|z^{(q+1)}\|_1 + \|z^*\|_1 \right] \leq \sqrt{\frac{K(K-1)}{p}} \underbrace{\mathbb{E}_z \left[ \|z^{(q)} - z^*\|_2^2 \right]}_{\text{expected resolution at iteration } q}$$

FacNet can improve the performances compared to ISTA when this is verified.

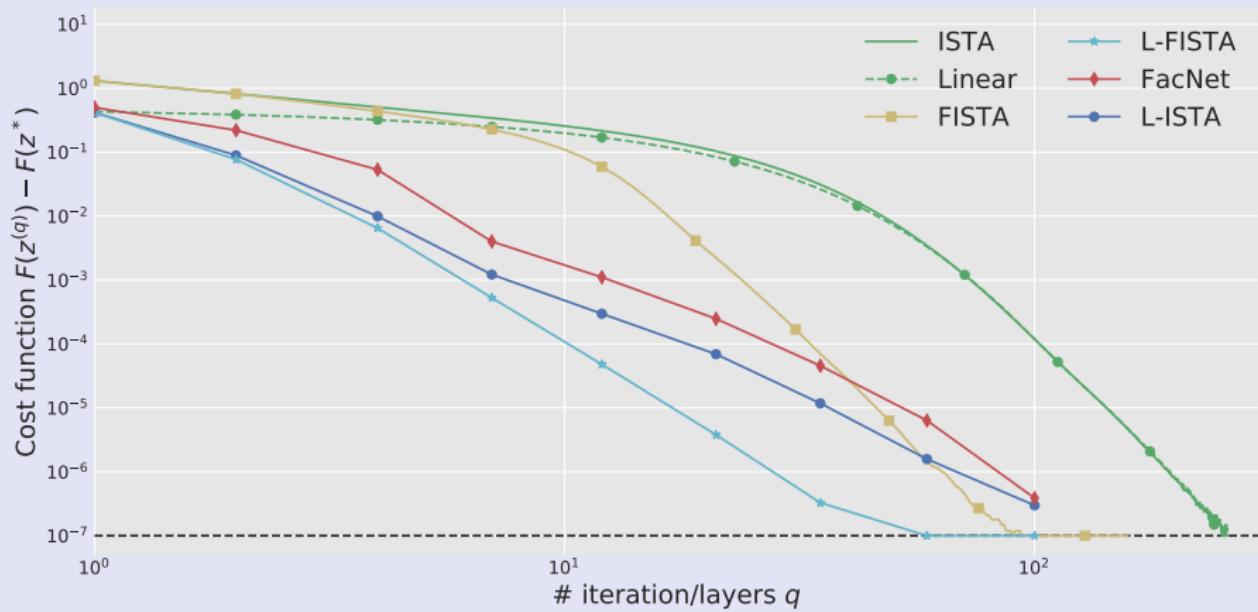
# L-FISTA



Network architecture for L-FISTA.

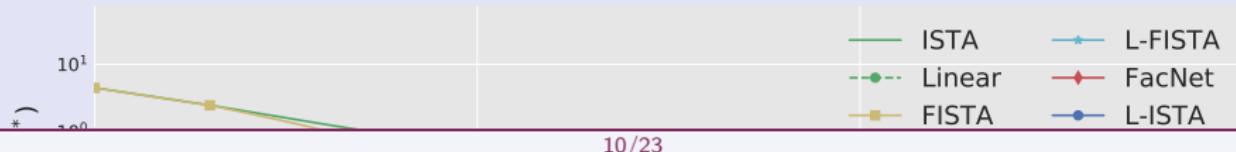


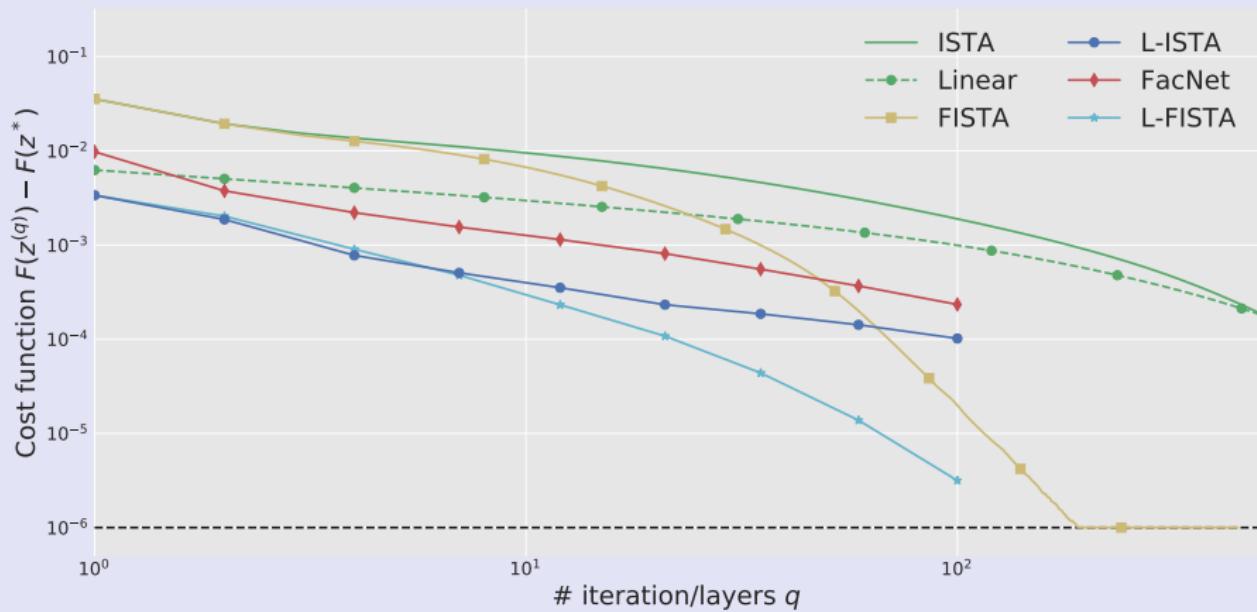
Evolution of the cost function  $F(z^{(q)}) - F(z^*)$  with the number of layers/iterations  $q$  with a denser model



$$\rho = ^1/20.$$

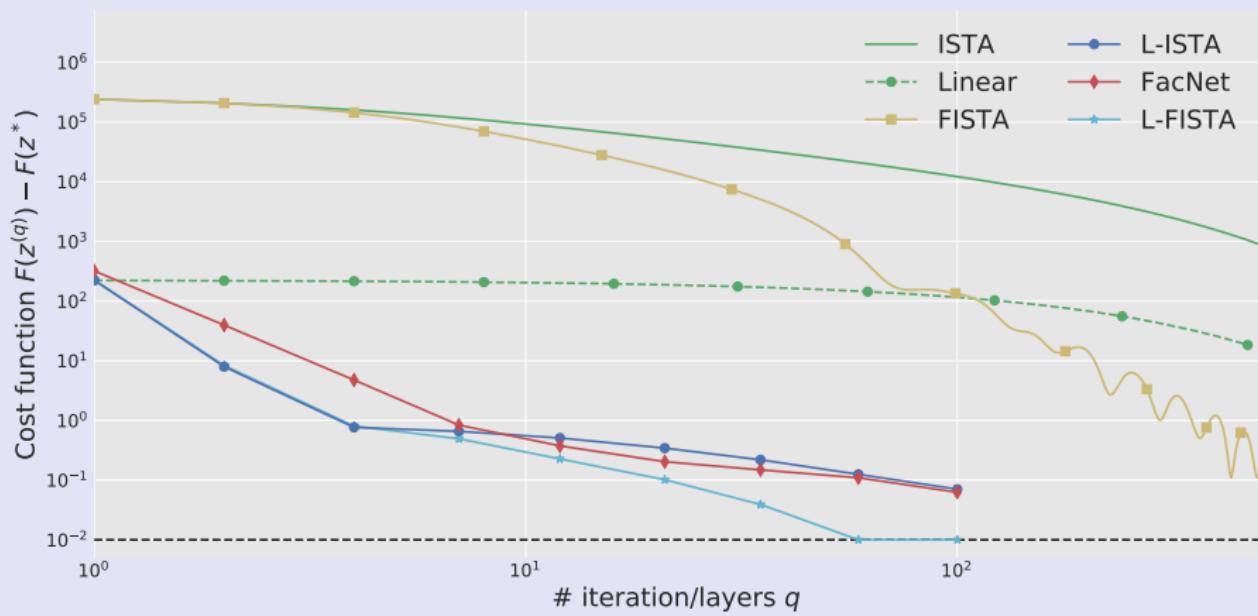
Evolution of the cost function  $F(z^{(q)}) - F(z^*)$  with the number of layers/iterations  $q$  with a denser model





Evolution of the cost function  $F(z^{(q)}) - F(z^*)$  with the number of layers

Dictionary  $D$  with  $K = 100$  atoms learned on 10 000 MNIST samples (17x17) with dictionary learning. LISTA trained with MNIST training set and tested on MNIST test set.



## Adversarial dictionary

---

The dictionary is constructed such that its eigen-vectors are sampled from the Fourier basis, with

$$D_{k,j} = e^{-2i\pi k \zeta_j}$$

for a random subset of frequencies

$$\{\zeta_i\}_{0 \leq i \leq p} \sim \mathcal{U}\left\{\frac{m}{K}; 0 \leq m \leq \frac{K}{2}\right\}$$

Diagonalizing  $B$  implies large deformation of the  $\ell_1$ -norm.

## Finishing the process in a distributed environment

Non trivial point: **How to decide that the algorithm has converged?**

- ▶ Neighbors paused is not enough!
- ▶ Define a master 0 and send probes.  
Wait for  $M$  probes return.
- ▶ Uses the notion of message queue and network flow.  
Maybe we can have better way?

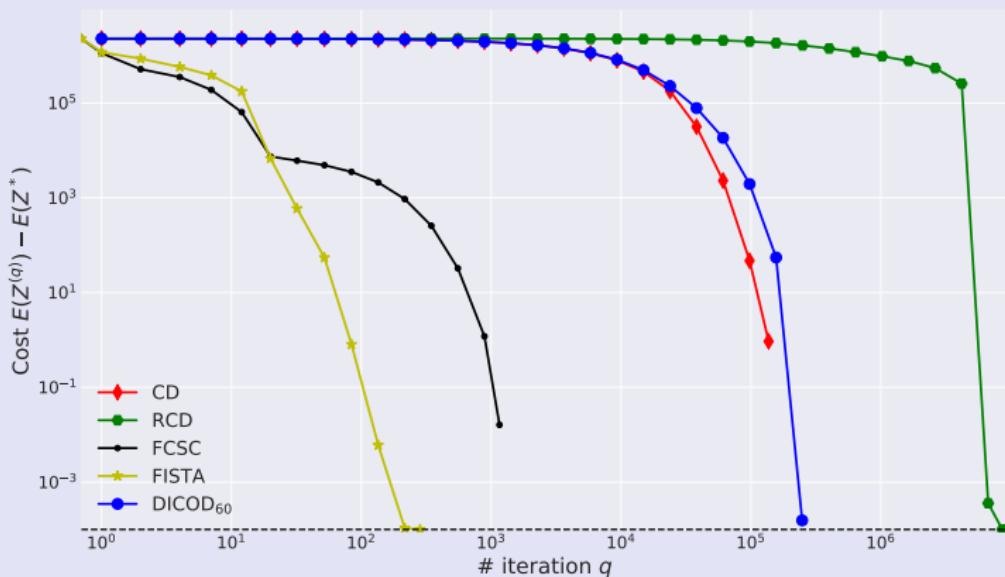
# Numerical Experiments

Test on long signals generated with Bernoulli-Gaussian coding signal  $Z$  and a Gaussian dictionary  $\mathcal{D}$ . Fixed  $K = 25$ ,  $W = 200$  and  $T = 600 * W$ ,

## Algorithms implemented for benchmark

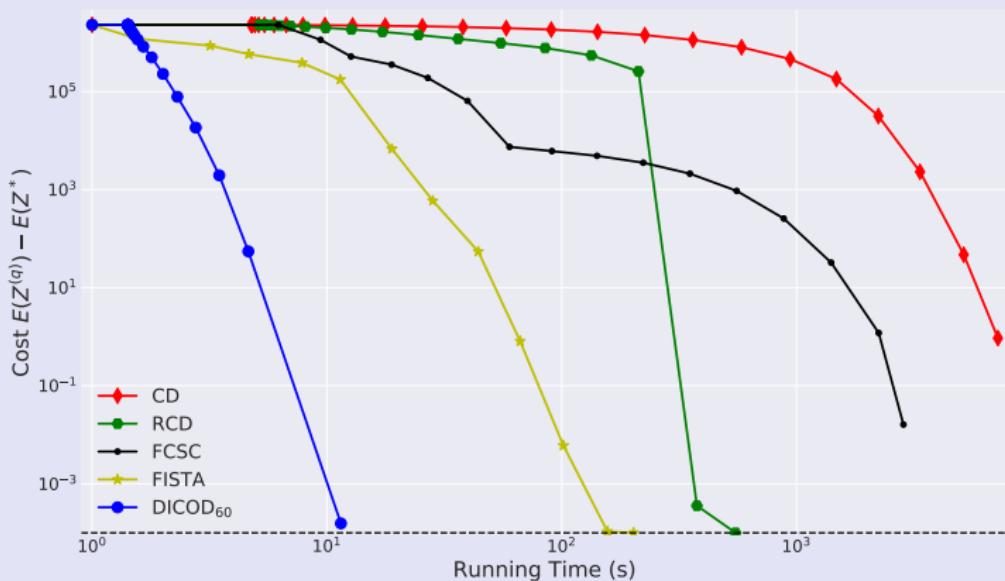
- ▶ Coordinate Descent (CD) [Kavukcuoglu et al., 2010]
- ▶ Randomized Coordinate Descent (RCD) [Nesterov, 2010]
- ▶ Fast Convolutional Sparse Coding (FCSC) [Bristow et al., 2013]
- ▶ Fast Iterative Soft-Thresholding Algorithm (FISTA) [Chalasani et al., 2013; Wohlberg, 2016]
- ▶ DICOD with 60 cores

# Numerical convergence



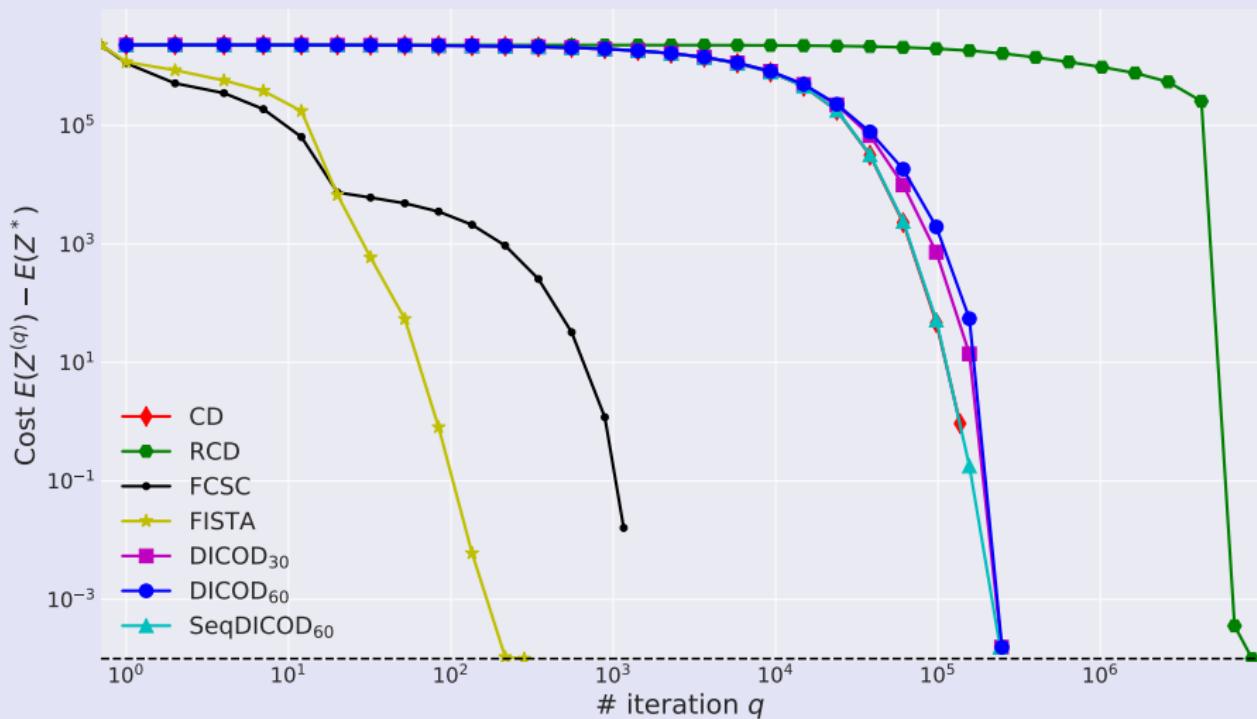
Cost as a function of the iterations

# Numerical convergence



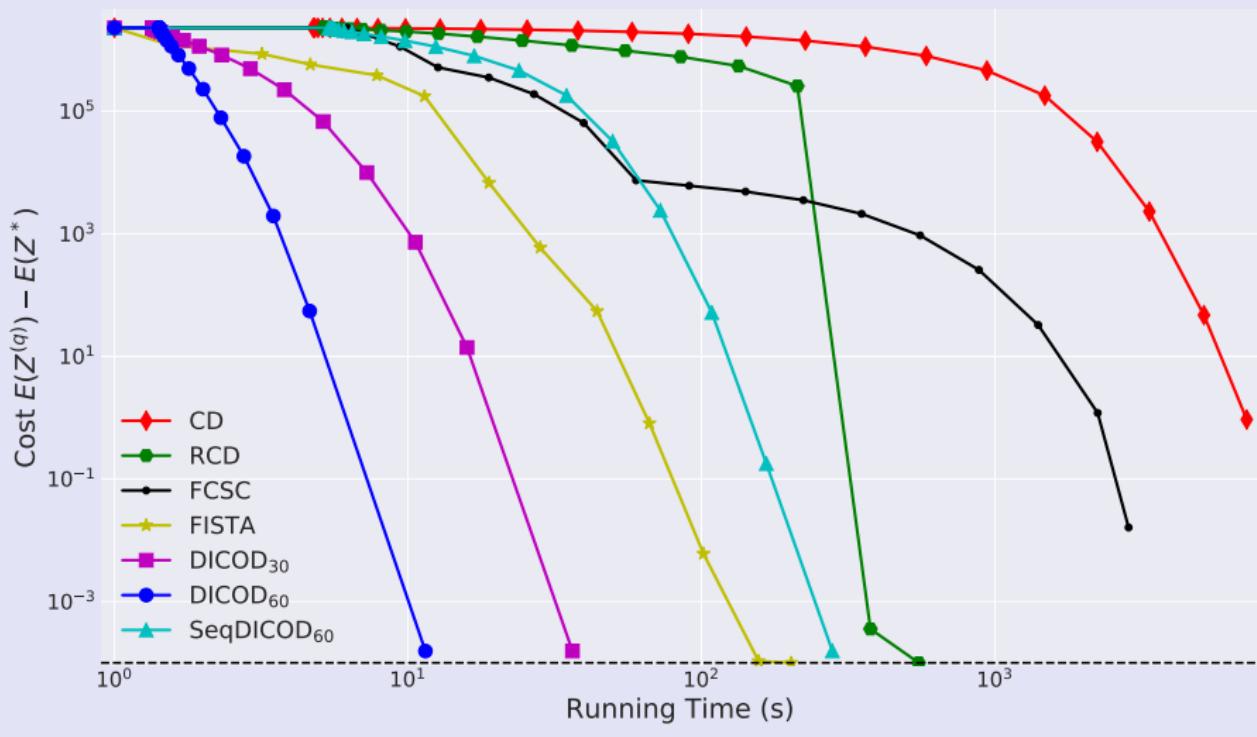
Cost as a function of the runtime

## DICOD: numerical convergence



Cost as a function of the iterations

## DICOD: numerical convergence



Cost as a function of the time

# Complexity Analysis

Two sources of acceleration:

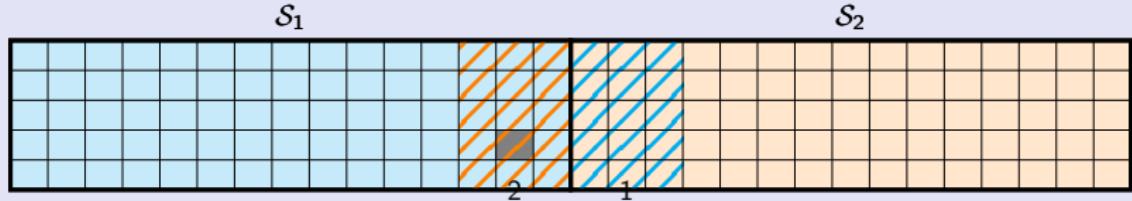
- ▶ Perform  $M$  updates in parallel,
- ▶ Each update is computed on a segment of size  $\frac{L}{M}$   
Iteration complexity of  $\mathcal{O}\left(K \frac{L}{M}\right)$  instead of  $\mathcal{O}(KL)$

Limitations:

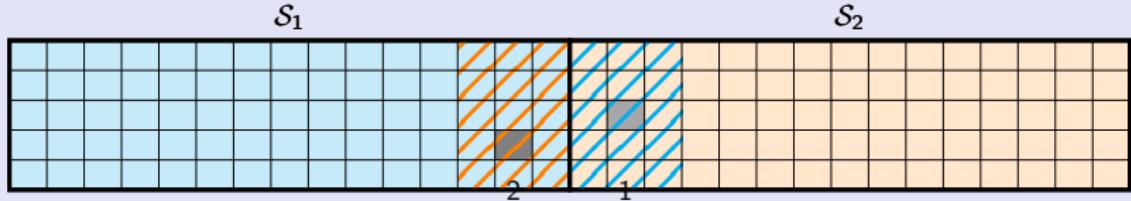
- ▶ Interfering updates, with probability  $\alpha^2 = \left(\frac{WM}{T}\right)^2$ 
$$\mathbb{E}[Q_{dicod}] \underset{\alpha \rightarrow 0}{\gtrsim} M(1 - 2\alpha^2 M^2 + \mathcal{O}(\alpha^4 M^4)) .$$
- ▶ Cost of the update of  $\beta$  in  $\mathcal{O}(KW)$



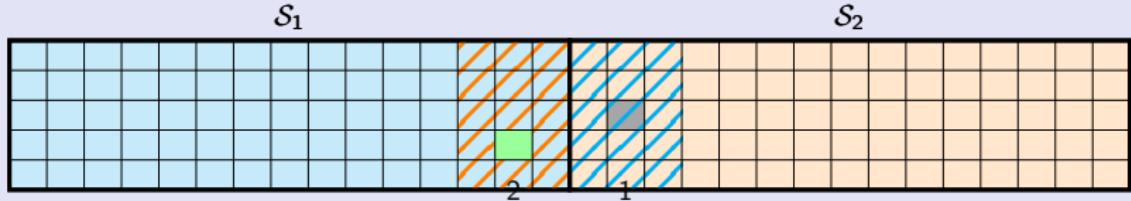
- ▶ Keep track of the value of the optimal update in an extended zone of size  $L - 1$ .



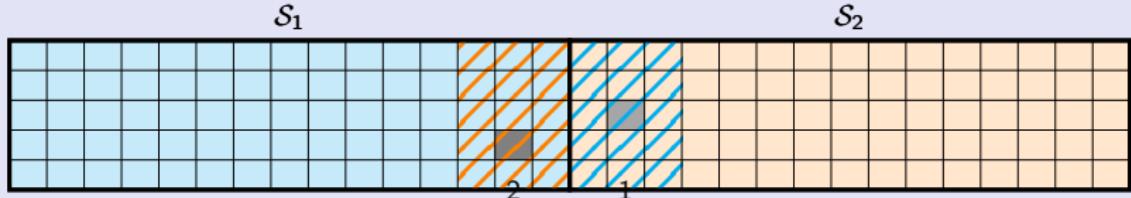
- ▶ Keep track of the value of the optimal update in an extended zone of size  $L - 1$ .
- ▶ Select an update candidate with LGCD.



- ▶ Keep track of the value of the optimal update in an extended zone of size  $L - 1$ .
- ▶ Select an update candidate with LGCD.
- ▶ If it is in the interfering zone, compare the value of the update with the value potential updates in the other worker.



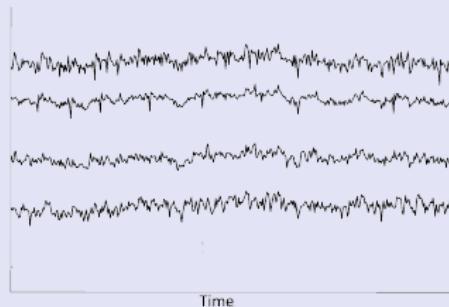
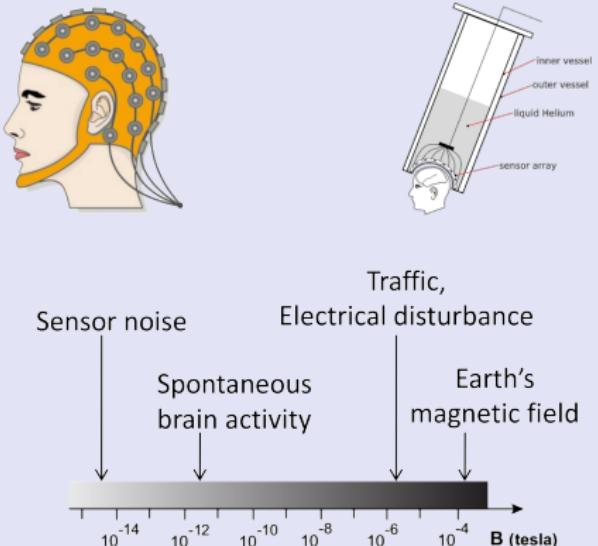
- ▶ Keep track of the value of the optimal update in an extended zone of size  $L - 1$ .
- ▶ Select an update candidate with LGCD.
- ▶ If it is in the interfering zone, compare the value of the update with the value potential updates in the other worker.
- ▶ Only perform the update if it is larger than the other update.



- ▶ Keep track of the value of the optimal update in an extended zone of size  $L - 1$ .
- ▶ Select an update candidate with LGCD.
- ▶ If it is in the interfering zone, compare the value of the update with the value potential updates in the other worker.
- ▶ Only perform the update if it is larger than the other update.
  - ⇒ Give an update order asynchronously.

# Studying brain activity through electromagnetic signals

- ▶ Brain (electrical) activity produces an electromagnetic field.
- ▶ This can be measured with EEG or MEG.



## Goal: Study Oscillation in Neural Data

---

Oscillations are believed to play an important role in cognitive functions.

Many studies rely on Fourier or wavelet analyses:

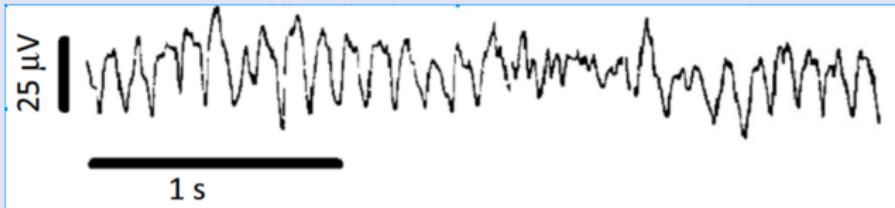
- ▶ Easy interpretation,
- ▶ Standard analysis e.g. canonical bands alpha, beta or theta.

[Buzsaki, 2006]

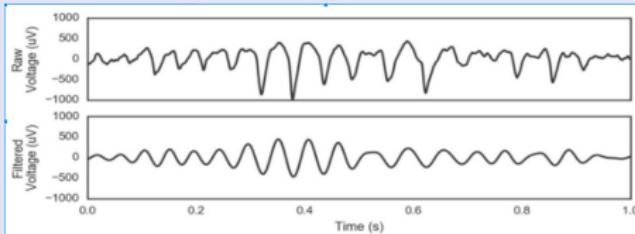
# Goal: Study Oscillation in Neural Data

However, some brain rhythms are not sinusoidal, e.g. mu-waves.

[Hari, 2006]



and filtering degrades waveforms



⇒ Can we do better with data-driven approach?

## Pattern recovery

Evolution of the recovery loss with  $\sigma$  for different values of  $P$ . Using more channels improves the recovery of the original patterns.

