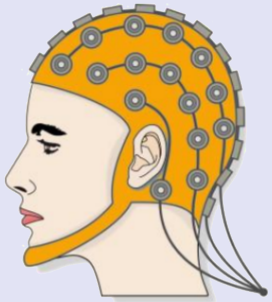
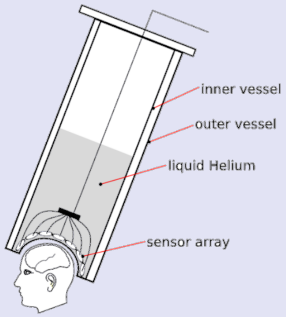
Using the Dictionary Structure for efficient Convolutional Dictionary Learning

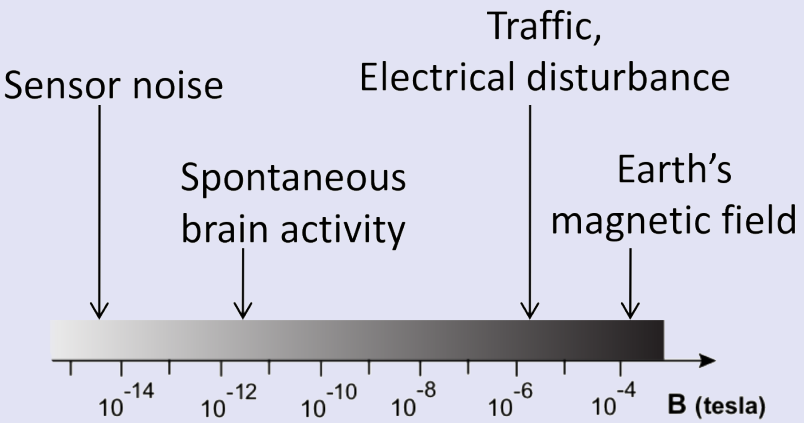
Thomas Moreau INRIA Saclay – parietal

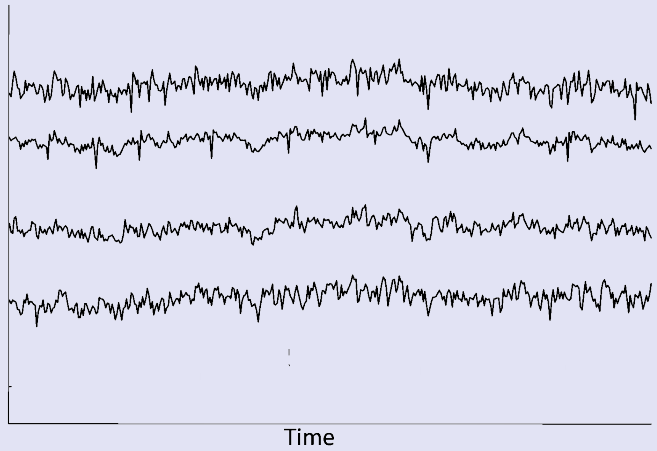
Studying brain activity through electromagnetic signals

- 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:

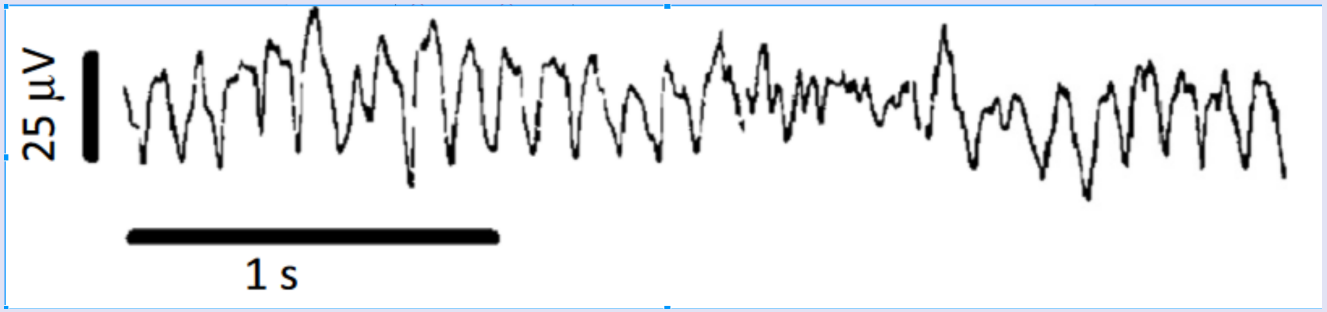
- Easy interpretation,
- Standard analysis *e.g.* canonical bands alpha, beta or theta.

[Buzsaki, 2006]

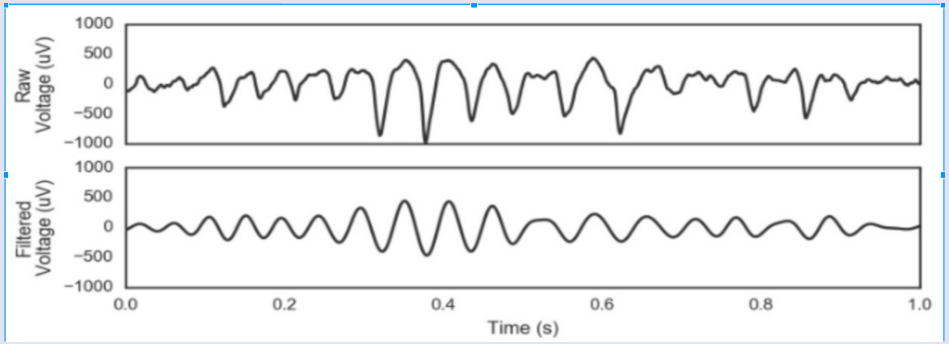
Goal: Study Oscillation in Neural Data

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

[Hari, 2006]



and filtering degrades waveforms

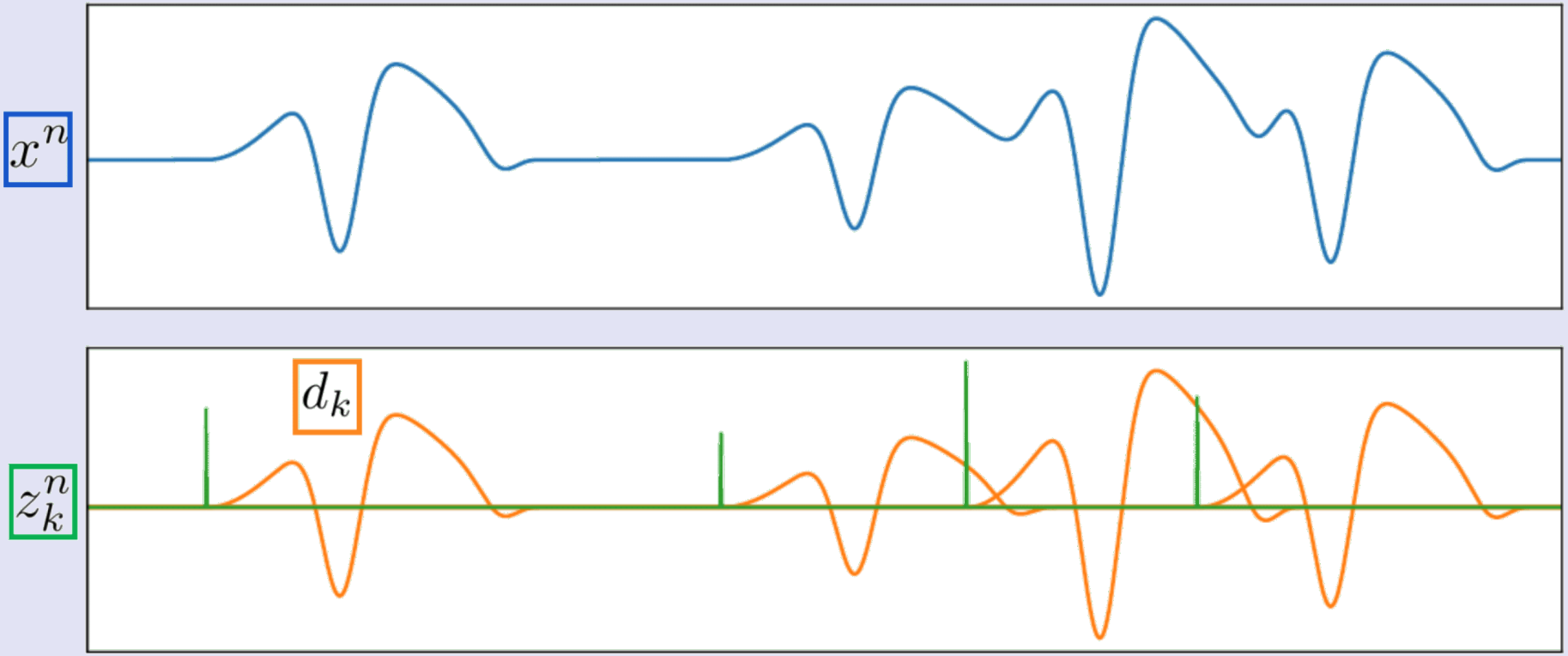


$\Rightarrow$ Can we do better with data-driven approach?

Convolutional representations

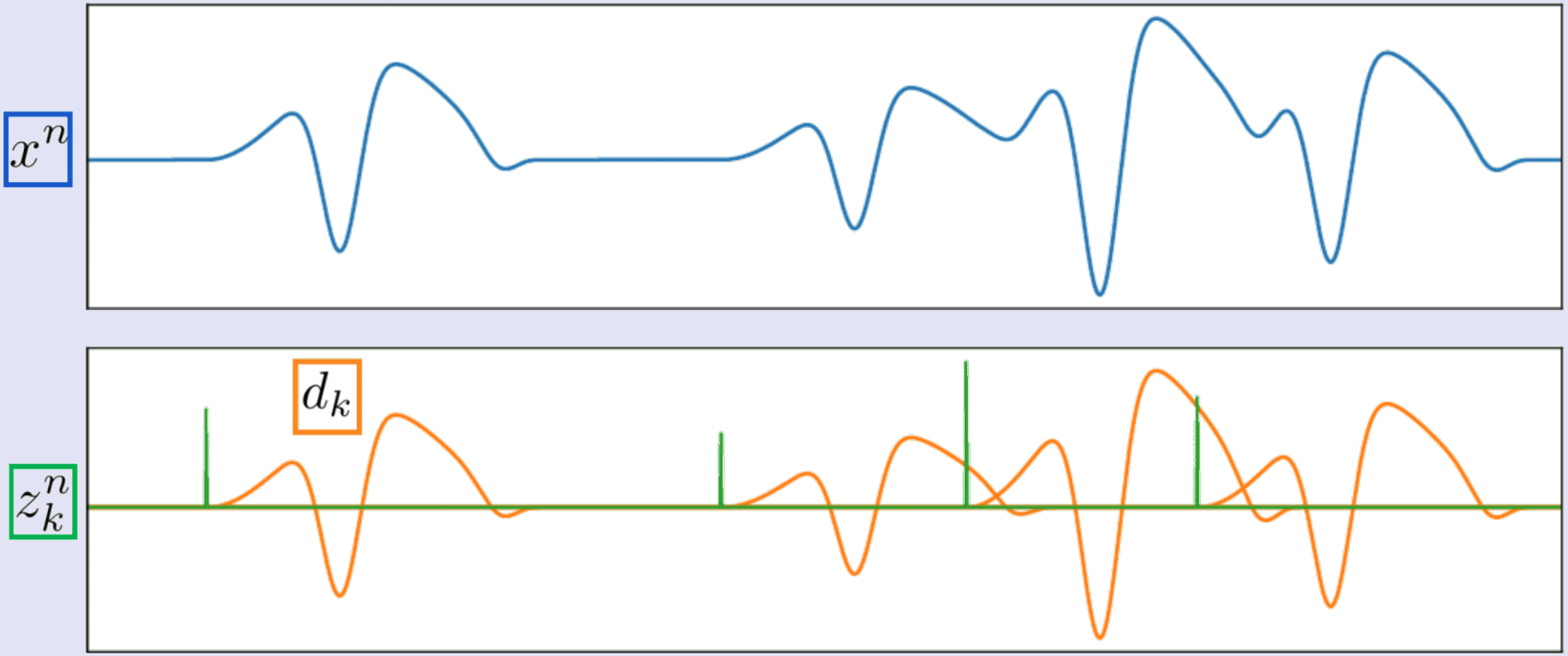
Extracting shift invariant patterns

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

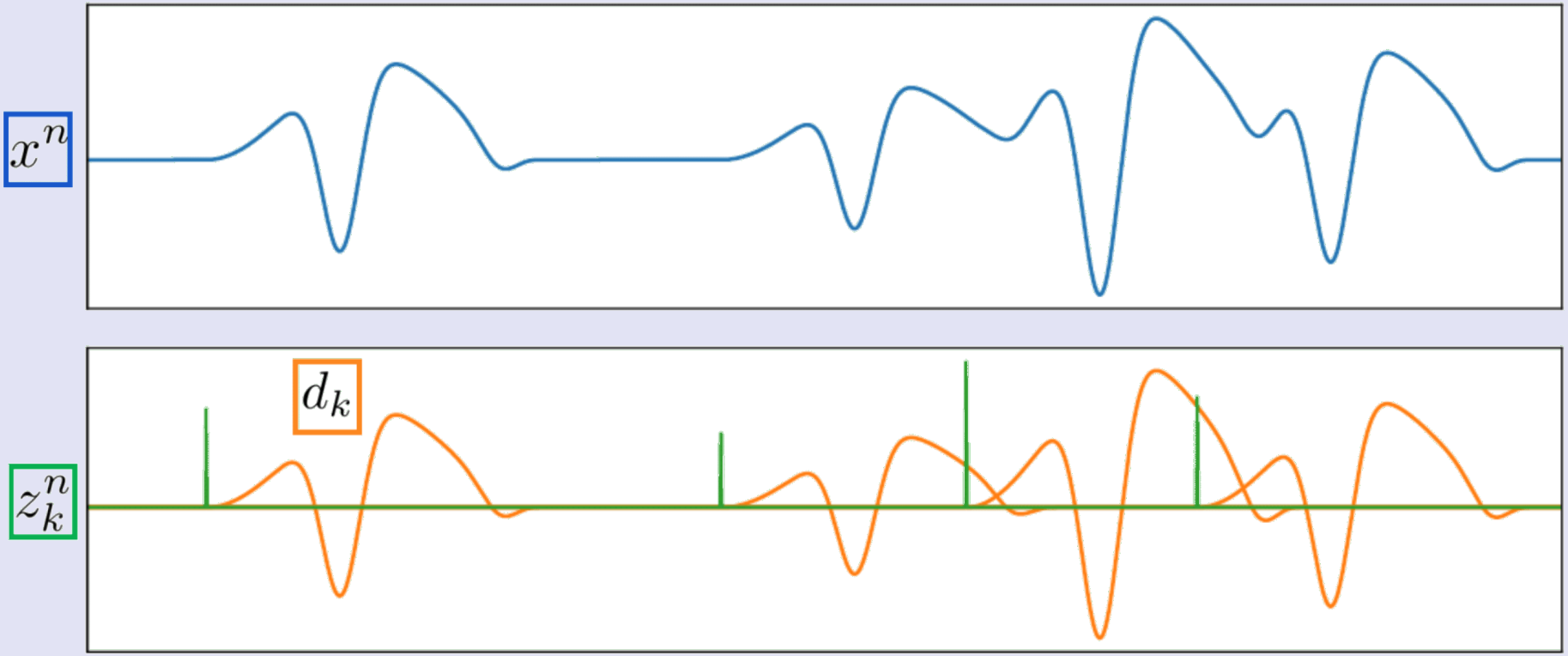


Convolutional Representation:

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

Extracting shift invariant patterns

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



Convolutional Dictionary Learning:

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

$$\text{s.t.} \quad \|d_k\|_2^2 \leq 1$$

Notation

Sparse Convolutional model:

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

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
- $\boldsymbol{D}$ is a set of K patterns of length W
- Z is a signal of length $L = T - W + 1$ in $\mathbb{R}^K$

Convolutional Dictionary Learning

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

$$(Z^*, \boldsymbol{D}^*) = \underset{Z, \boldsymbol{D}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} \underbrace{\|X^{[n]} - \sum_{k=1}^{K} \boldsymbol{D}_k * Z_k^{[n]}\|_2^2}_{E(Z) \text{ data fit}} + \underbrace{\lambda \|Z^{[n]}\|_1 + \mathbf{1}_\Omega(D)}_{\text{penalizations}}$$

with a convex constraint set Ω and a regularization parameter $\lambda > 0$.
Classic constraint sets Ω are the ℓ_2-ball or sphere.

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

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

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

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

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

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

$$\Rightarrow \text{Independent for each } n \in [\![1, N]\!]$$

Related Algorithms:

- Iterative Soft-Thresholding Algorithm (ISTA)
  [Daubechies et al., 2004]
- Fast ISTA
  [Beck and Teboulle, 2009]
- Alternated Direction Method of Multiplier (ADMM)
  [Gabay and Mercier, 1976]
- Coordinate Descent (CD)
  [Friedman et al., 2007]

We have to solve N independent problems with a common structure $\boldsymbol{D}$,

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

Can we use this structure to accelerate the resolution?

We have to solve N independent problems with a common structure $\boldsymbol{D}$,

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

Can we use this structure to accelerate the resolution?

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

Why does it work?

We have to solve N independent problems with a common structure $\boldsymbol{D}$,

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

Can we use this structure to accelerate the resolution?

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

Why does it work? Analysis in the context of sparse coding.

(no convolution)

Part II: Coordinate Descent for CSC

Coordinate descent only performs local updates at each iteration.

$\Rightarrow$ More efficient for convolutional model with long signals.

Can we improve it with the structure of our problem?

Improving CD efficiency for the convolutional structure:

- Locally Greedy Coordinate Descent.
- Asynchronous and distributed algorithms: DICOD and DiCoDiLe.

For electrophisiological signals, the CDL reads,

$$\min_{Z,\boldsymbol{D}} \frac{1}{N} \sum_{n=1}^{N} \|X^{[n]} - \sum_{k=1}^{K} \boldsymbol{D}_k * Z_k^{[n]}\|_2^2 + \lambda \|Z^{[n]}\|_1 + \mathbf{1}_\Omega(D)$$

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

Can we constrain the structure of the dictionary to learn more interpretable atoms?

$\Rightarrow$ Use rank-1 constraints to accomodate Maxwell's equations.

Adaptive Sparse Coding

References

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

Vectorized model

- x is a vector in $\mathbb{R}^T$
- ϵ 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.

$$D = \begin{pmatrix} \bar{D} & \bar{0} & & 0 & \\ \bar{0} & \bar{D} & \bar{0} & & \\ & \bar{0} & \bar{D} & \bar{0} & \\ & & & \ddots & \\ & 0 & & \bar{0} & \bar{D} & \bar{0} \\ & & & & \bar{0} & \bar{D} & \bar{0} \\ & & & & & \bar{0} & \bar{D} \end{pmatrix}.$$

Adaptive Optimization

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

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

Can we use this structure to accelerate the resolution?

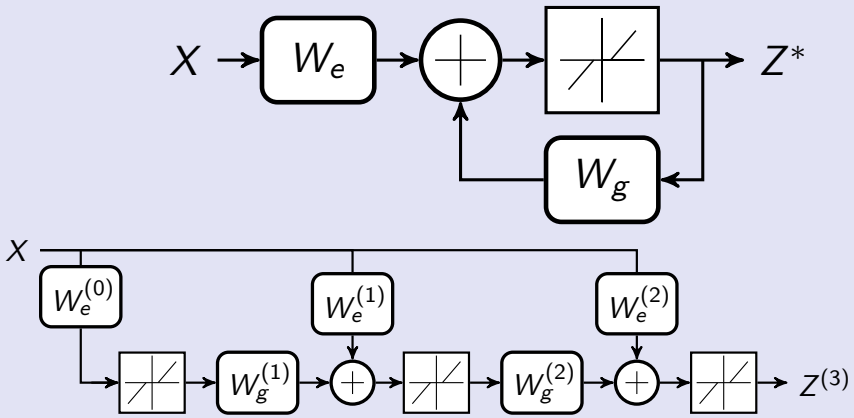
Adaptive Optimization

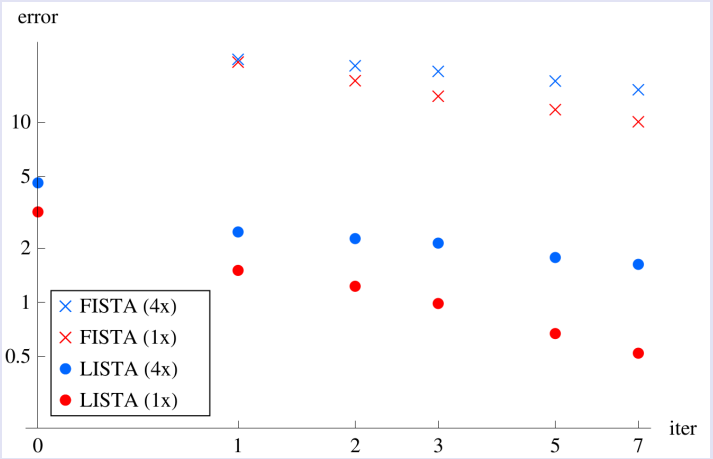
We have to solve N problems with a common structure $\boldsymbol{D}$.

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

Can we use this structure to accelerate the resolution?

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





Adaptive Optimization

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

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

Can we use this structure to accelerate the resolution?

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

Open problem: Why does it work?

- Can we leverage the structure of $\boldsymbol{D}$?
- Can we get a non-asymptotic acceleration of ISTA?
- How to explain LISTA performance?

[Giryes et al., 2018; Xin et al., 2016]

Notations

Consider the sparse coding problem with a dictionary D.

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

We denote $B = D^\mathsf{T} D$ is the Gram matrix of D.

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

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

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:

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

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 $S = \|B\|_2 \boldsymbol{I}_K$

$$\begin{aligned} F_q(z) &= E(z^{(q)}) + \left\langle \nabla E(z^{(q)}), z - z^{(q)} \right\rangle + Q_S\ (\ z,\ z^{(q)}) \ , \\ \min_z F_q(z) &\Leftrightarrow \min_z Q_S \left(\ z,\ z^{(q)} - S^{-1} \nabla E(z^{(q)}) \right) \end{aligned}$$

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 $S = \|B\|_2 \boldsymbol{I}_K$

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

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

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

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 $S = \|B\|_2 \boldsymbol{I}_K$

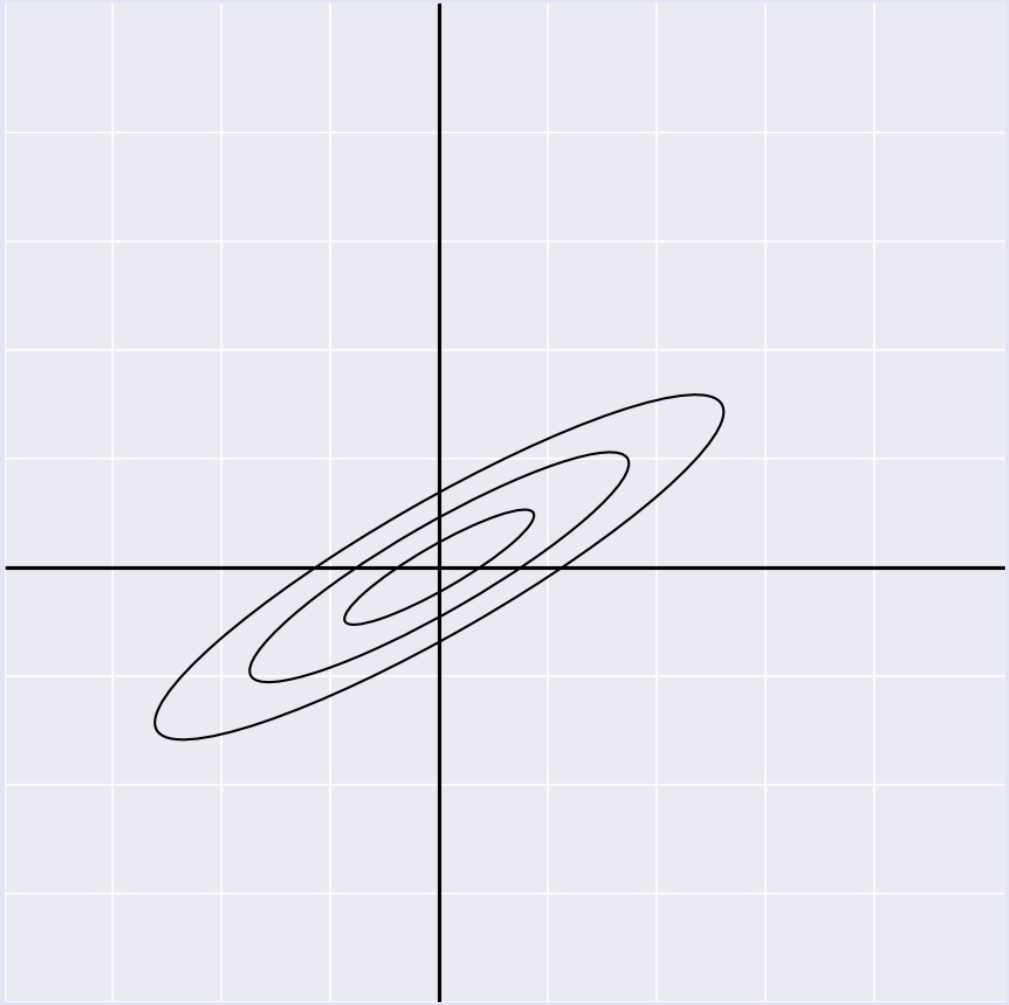
FacNet: Replace B by $A^T S A$ (S diagonal, A unitary)

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

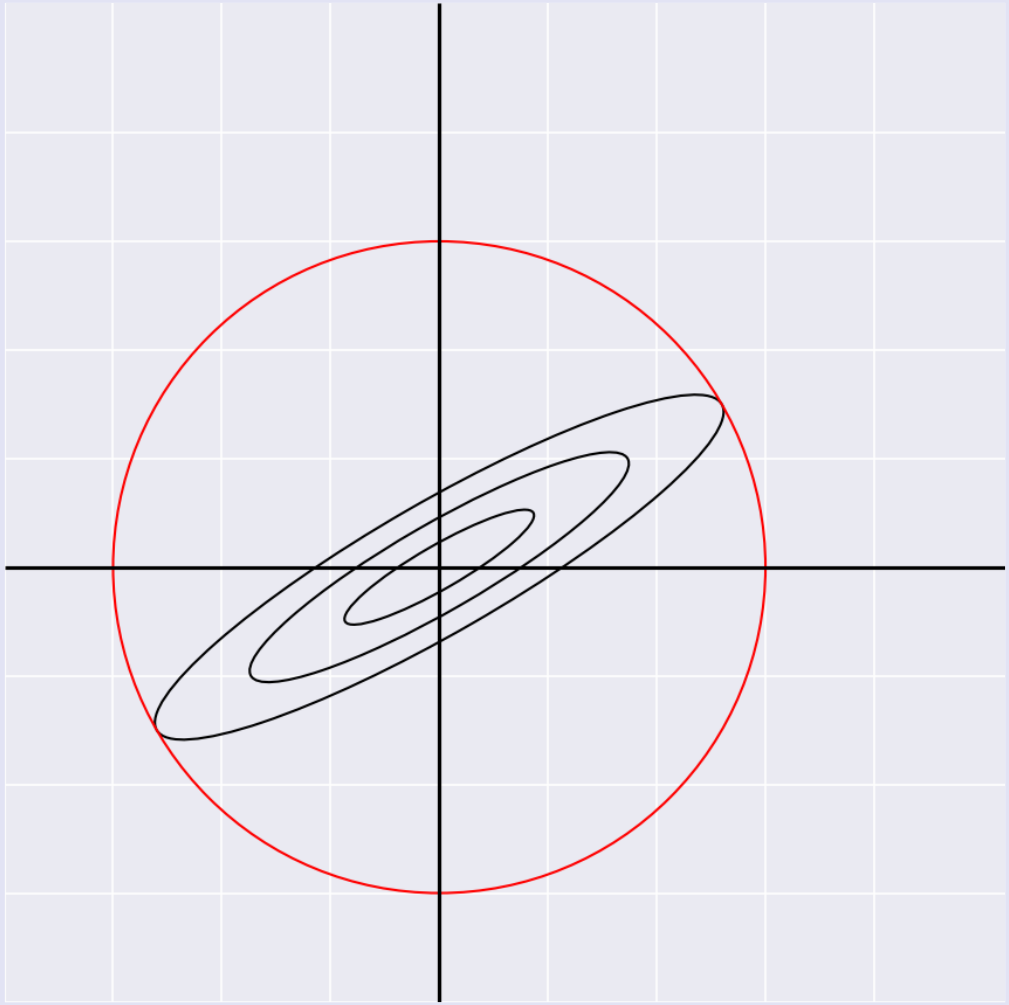
$$\min_z \widetilde{F}_q(z) \Leftrightarrow \min_z Q_{S_q}\left(\boldsymbol{A}_q z, \boldsymbol{A}_q z^{(q)} - S_q^{-1}\boldsymbol{A}_q \nabla E(z^{(q)})\right)$$

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

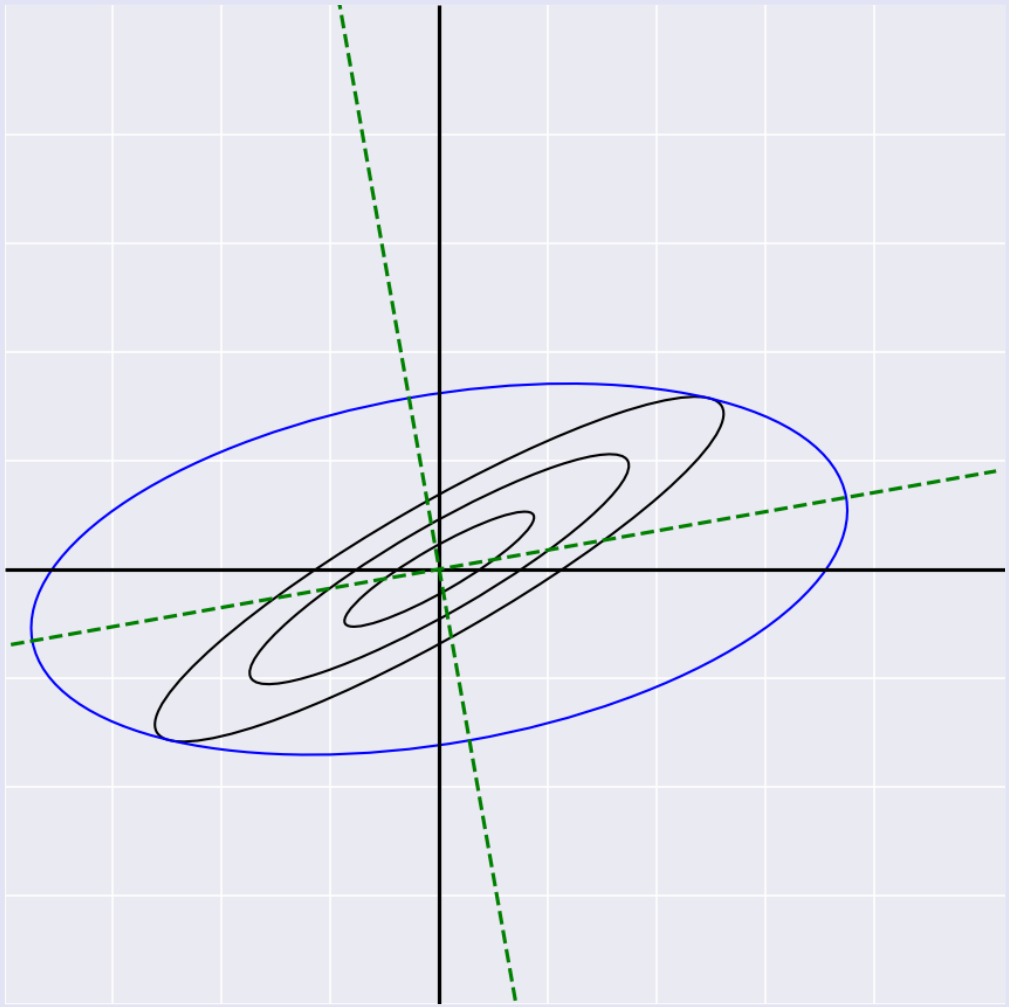
Toward and adaptive procedure



Toward and adaptive procedure



Toward and adaptive procedure



Toward an adaptive procedure

Similar iterative procedure with steps adapted to the problem topology.

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

Tradeoff between:

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

$$R = A^{\mathsf{T}} S A - B$$

- Deformation of the ℓ_1-norm with the rotation A . Accuracy

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

One step improvement

Suppose that $R = A^{\mathsf{T}} S A - B \succ 0$ is positive definite, and define

$$z^{(q+1)} = \arg\min_{z} \widetilde{F_q}(z) \ ,$$

Then

$$F(z^{(q+1)}) - F(z^*) \leq \frac{1}{2}(z^{(q)} - z^*)^{\mathsf{T}} R (z^{(q)} - z^*) + \delta_A(z^*) - \delta_A(z^{(q+1)}) \ .$$

We are interested in factorization (A, S) for which $\|R\|_2$ and δ_A are small.

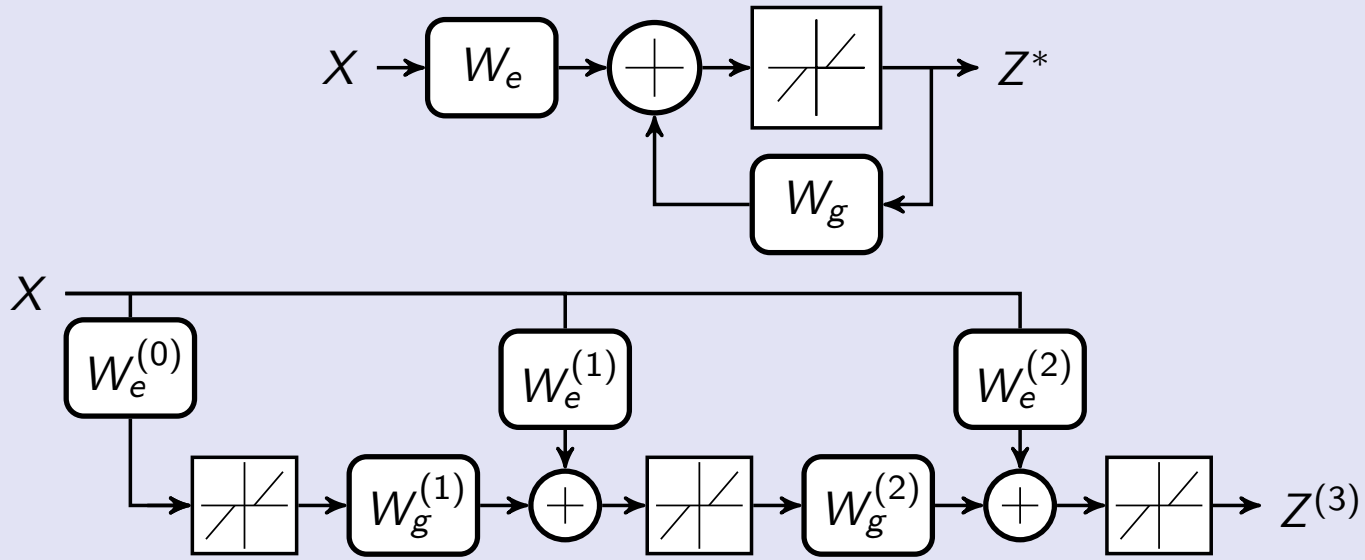
Theoretical results

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

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



Network architecture for ISTA/LISTA. LISTA is the unfolded version of the RNN of ISTA, trainable with back-propagation.

With $W_e = \frac{D^\mathsf{T}}{\|B\|_2}$ and $W_g = I - \frac{B}{\|B\|_2}$, this network computes exactly 2 iterations of ISTA.

Specialization of LISTA

$$z^{(q+1)} = A^{\mathsf{T}} \underset{S}{\text{prox}}(Az^{(q)} - S^{-1}AB(z^{(q)} - y)) \ ,$$

with A unitary and S diagonal.
Same architecture with more constraints on the parameter space:

$$\begin{cases} W_e &= S^{-1}AD^{\mathsf{T}} \\ W_g &= A^{\mathsf{T}} - S^{-1}ABA^{\mathsf{T}} \end{cases}$$

$\Rightarrow$ LISTA can be at least as good as this model.

Generic Dictionary

- Generic dictionary uniformly sample in unit ball,

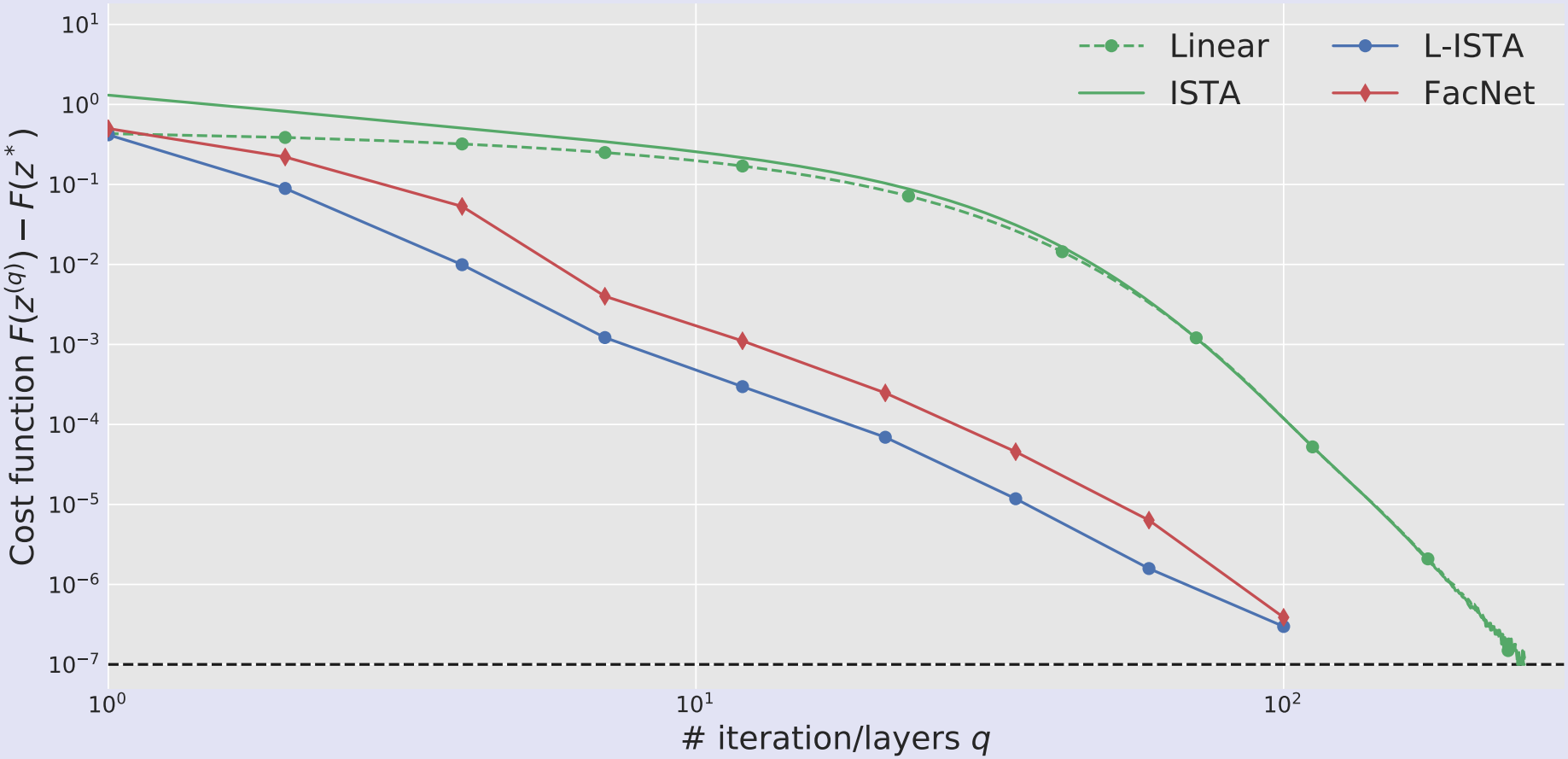
$$D \sim \mathcal{S}^{p-1} \ ,$$

- Sparse code generated with Bernouilli-Gaussian model, *s.t.*

$$z_k = b_k a_k, \qquad b_k \sim \mathcal{B}(\rho) \text{ and } a \sim \mathcal{N}(0, \sigma \boldsymbol{I}_K)$$

Fixed: $K = 100$, $P = 64$, $\sigma = 10$ and $\lambda = 0.01$

Generic Dictionary



$$\rho = {}^{1}/_{20}.$$

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

Adversarial dictionary

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

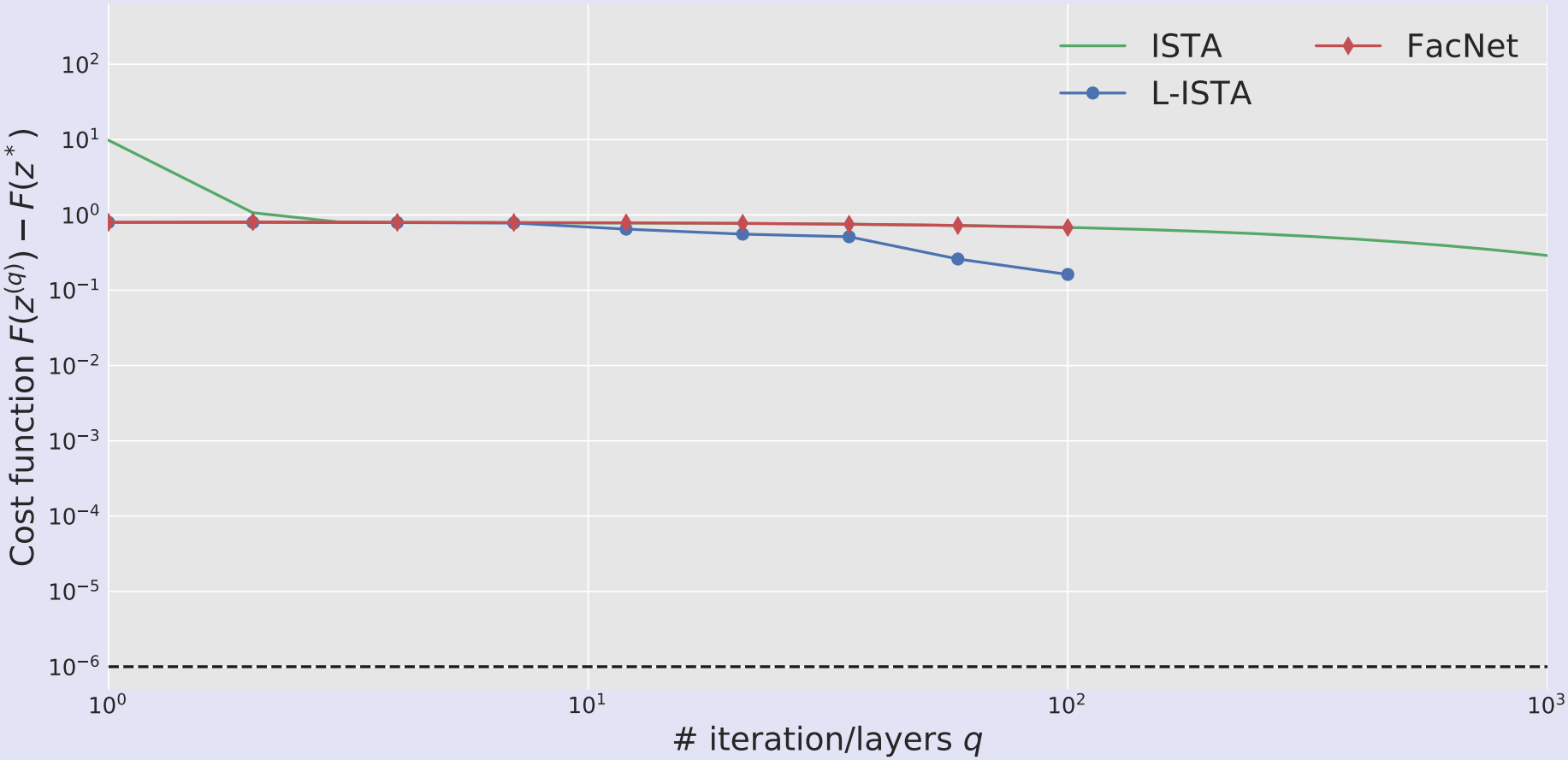
$$D_{k,j} = e^{-2i\pi k\zeta_j}$$

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 ℓ_1-norm.

Adversarial dictionary



Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers/iterations q with an adversarial dictionary.

Recap Part I

Take home message

- Non asymptotic acceleration of ISTA is possible based on the structure of $\boldsymbol{D}$,
- 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.

Adapting CD to Convolutional Sparse Coding

References

- Moreau, T., Oudre, L., and Vayatis, N. (2018). DICOD: Distributed Convolutional Sparse Coding. In *International Conference on Machine Learning (ICML)*, pages 3626–3634, Stockohlm, Sweden. PMLR (80)

Coordinate Descent (CD)

Minimize

$$Z^* = \underset{Z}{\operatorname{argmin}} \|X - \sum_{k=1}^{K} \boldsymbol{D}_k * Z_k\|_2^2 + \lambda \|Z\|_1$$

Update one coordinate at each iteration.

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

Three algorithms:

- 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

$$Z^* = \underset{Z}{\operatorname{argmin}} \|X - \sum_{k=1}^{K} \boldsymbol{D}_k * Z_k\|_2^2 + \lambda \|Z\|_1$$

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:

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

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

$$Z^* = \underset{Z}{\operatorname{argmin}} \|X - \sum_{k=1}^{K} \boldsymbol{D}_k * Z_k\|_2^2 + \lambda \|Z\|_1$$

Update one coordinate at each iteration.

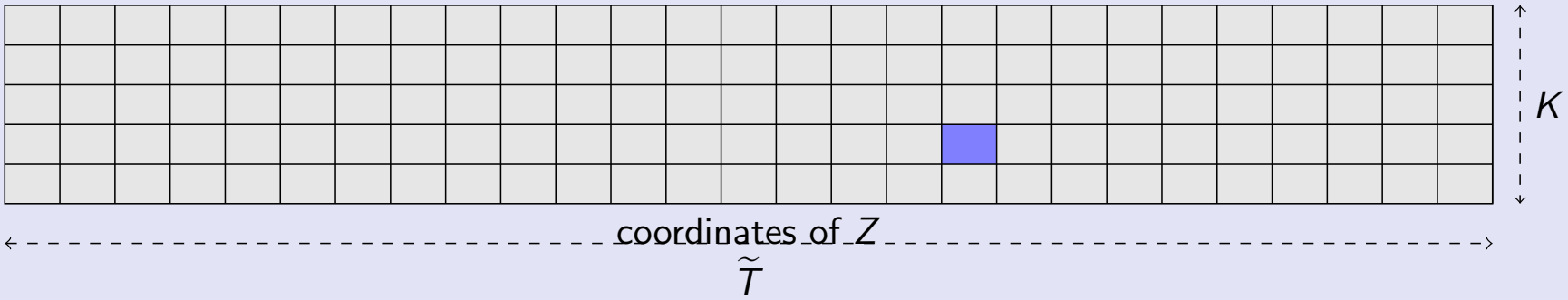
1. Select a coordinate (k_0, t_0) to update.
2. Compute a new value $Z'_{k_0}[t_0]$ for this coordinate

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

Coordinate selection is a trade-off between cheap computational complexity (random/cyclic CD) and importance sampling with faster convergence (Greedy CD). [Nutini et al., 2015]

Locally greedy coordinate descent (LGCD) [Moreau et al., 2018]

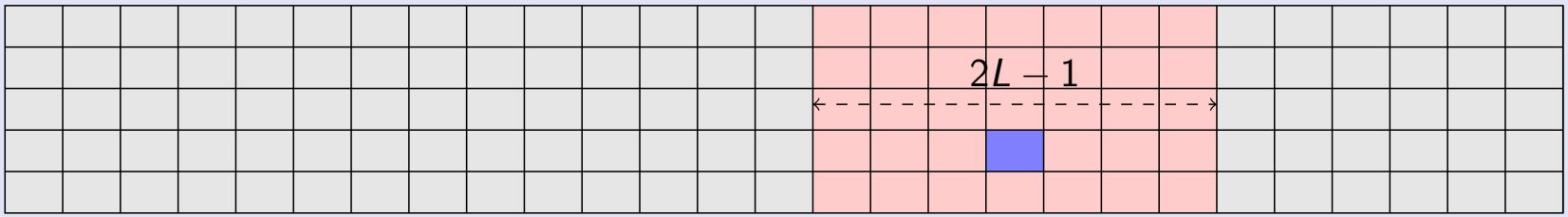
We introduced the LGCD method which is an extension of GCD.



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

Locally greedy coordinate descent (LGCD) [Moreau et al., 2018]

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

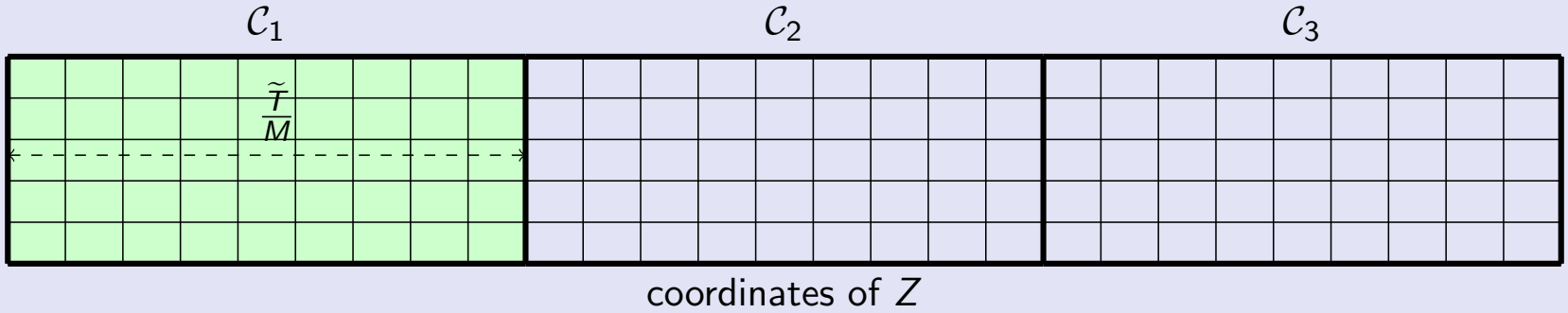


GCD has $\mathcal{O}(K\widetilde{T})$ computational complexity.

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

Locally greedy coordinate descent (LGCD) [Moreau et al., 2018]

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

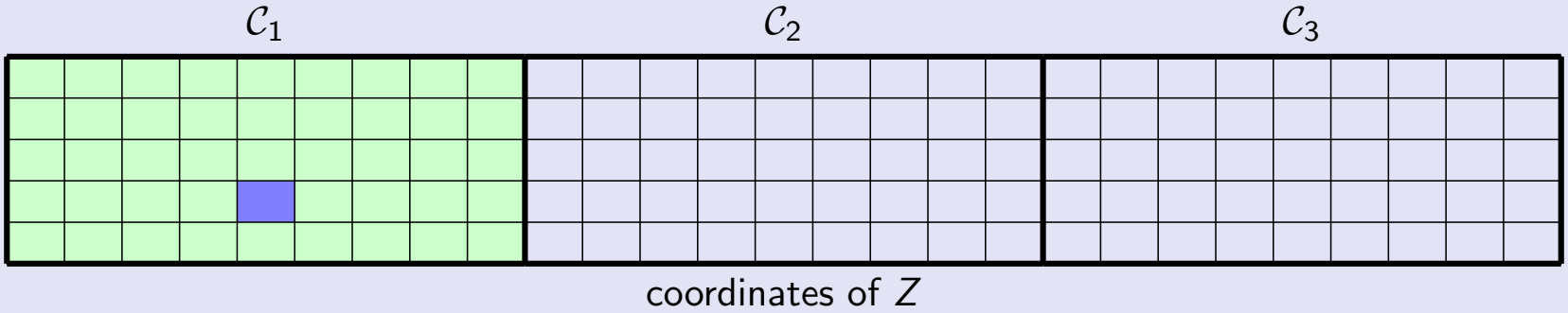


With a partition $\mathcal{C}_m$ of the signal domain $[1, K] \times [0, \widetilde{T}[$,

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

Locally greedy coordinate descent (LGCD) [Moreau et al., 2018]

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



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

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

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

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

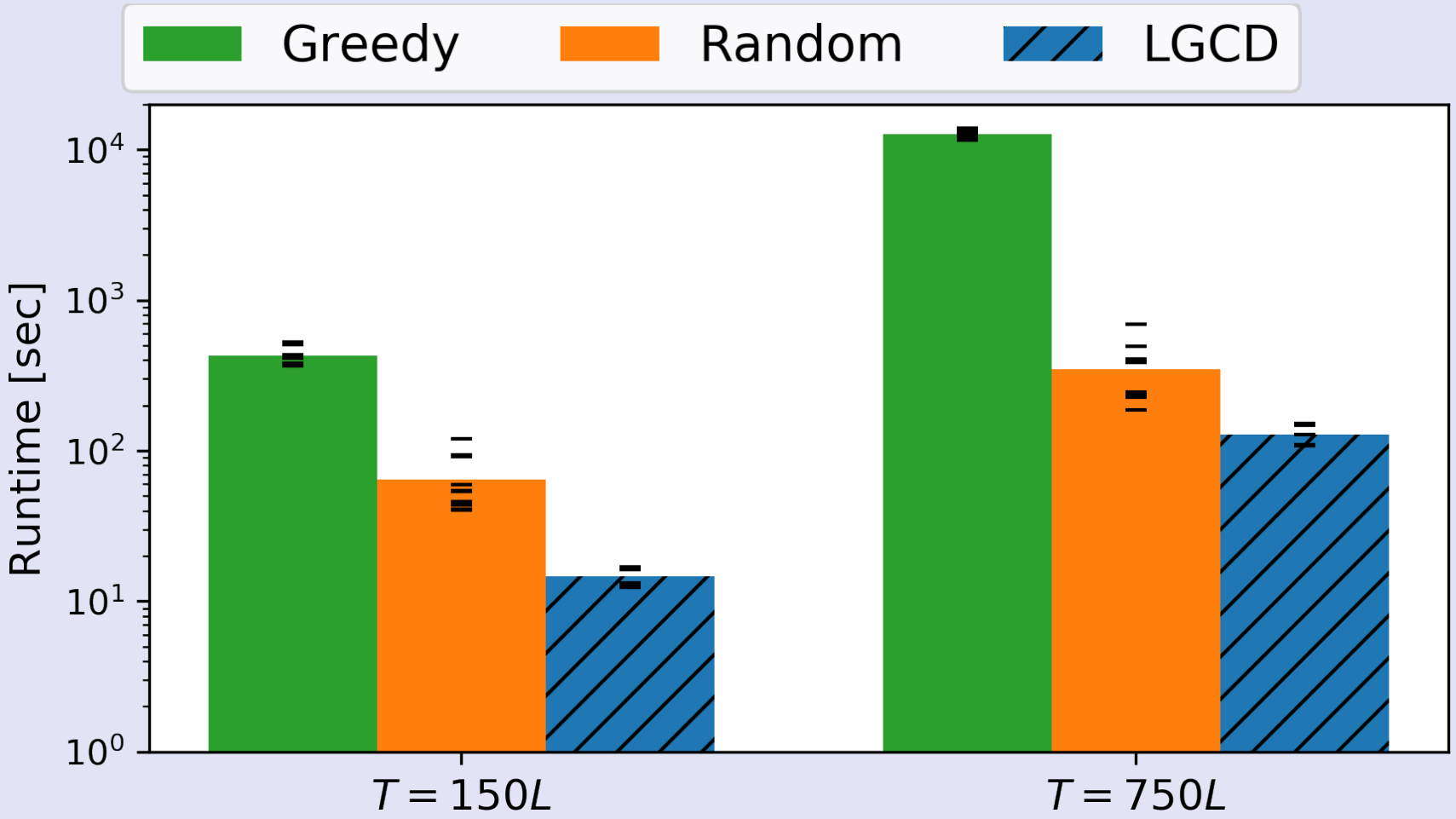
The overall iteration complexity is $\mathcal{O}(KL)$ instead of $\mathcal{O}(K\widetilde{T})$.

$$\Rightarrow \text{Efficient for sparse } Z$$

Fast optimization

Comparison of the coordinate selection strategy for CD on simulated signals

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



Distributed optimization for CSC

References

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

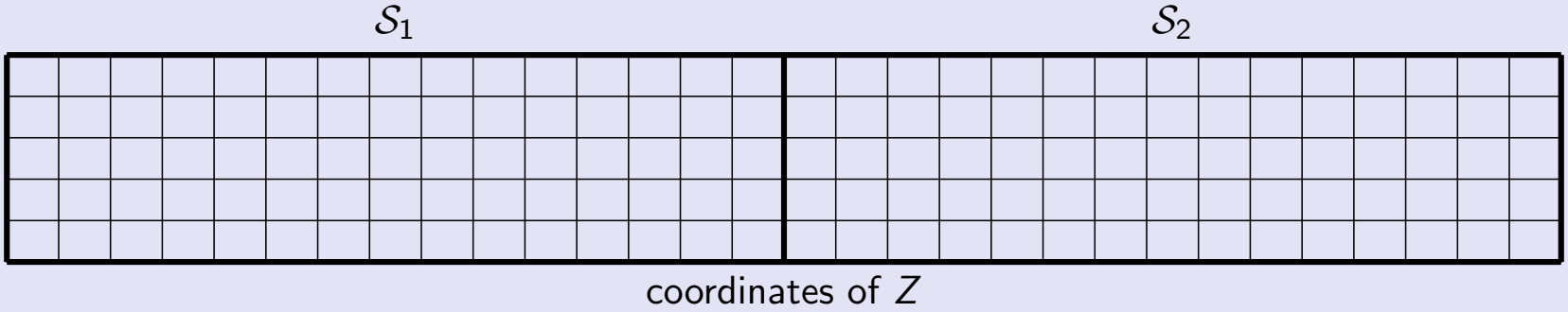
Weak dependence of the coordinate updates

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:

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

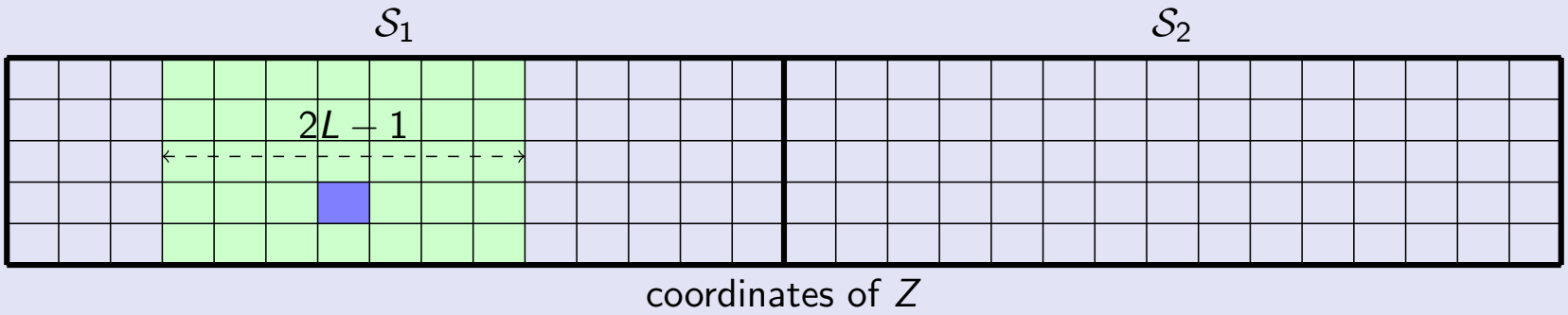
$\Rightarrow$ If the updates are far enough, they can be considered as independent.

Distributed Convolutional Coordinate Descent (DICOD)



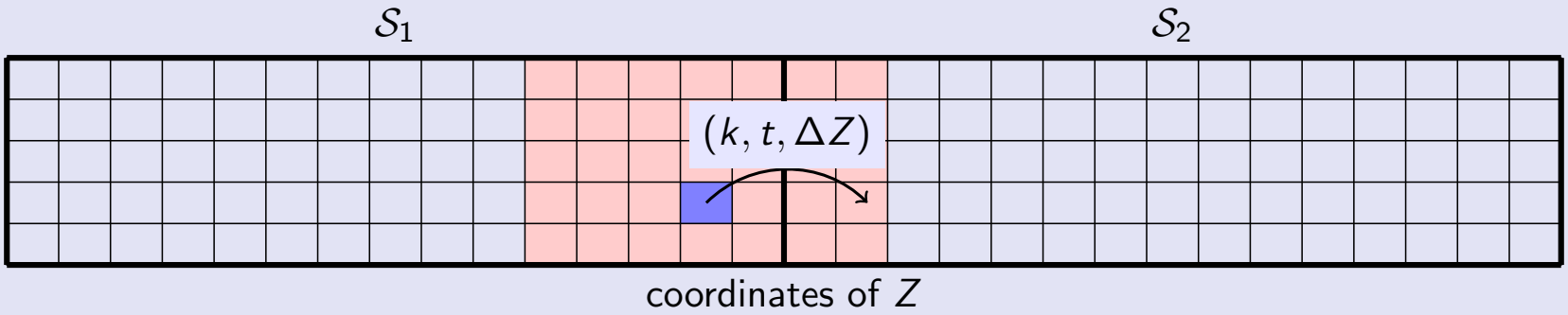
- Split the coordinates in continuous sub-segment $\mathcal{S}_w = \left[\frac{(w-1)T}{W}, \frac{wT}{W}\right[$.

Distributed Convolutional Coordinate Descent (DICOD)



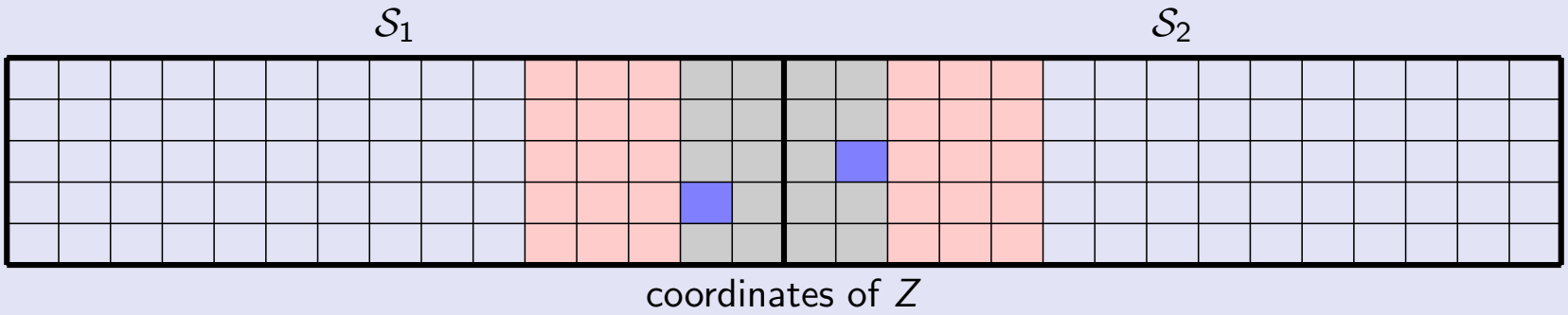
- Split the coordinates in continuous sub-segment $\mathcal{S}_w = \left[\frac{(w-1)T}{W}, \frac{wT}{W}\right[$.
- Use CD updates in parallel in each sub-segment.

Distributed Convolutional Coordinate Descent (DICOD)



- 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 (DICOD)



- 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:

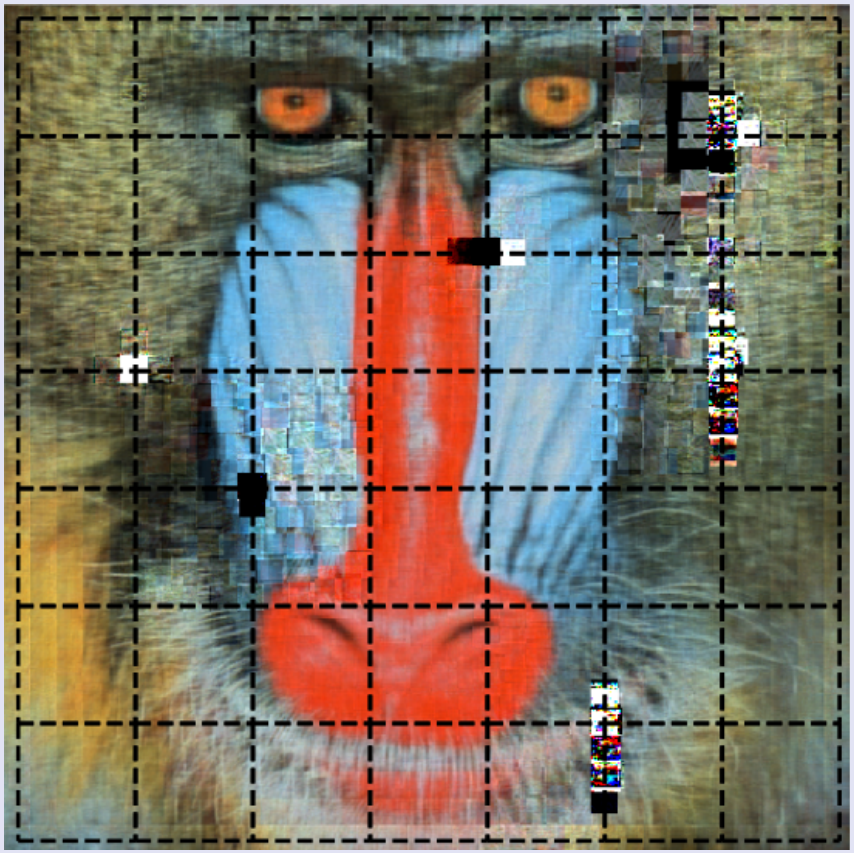
H1: If the cross correlation between atoms of $\boldsymbol{D}$ is strictly smaller than 1.

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.

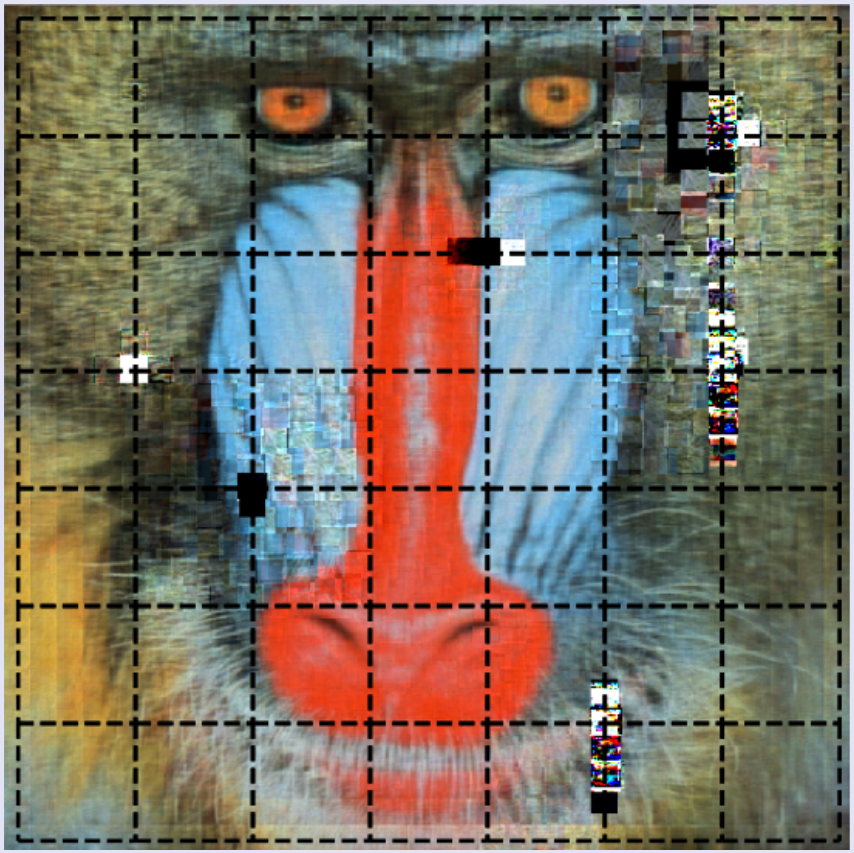
Under these assumptions, the DICOD algorithm converges asymptotically to the optimal solution Z^* of CSC.

Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)



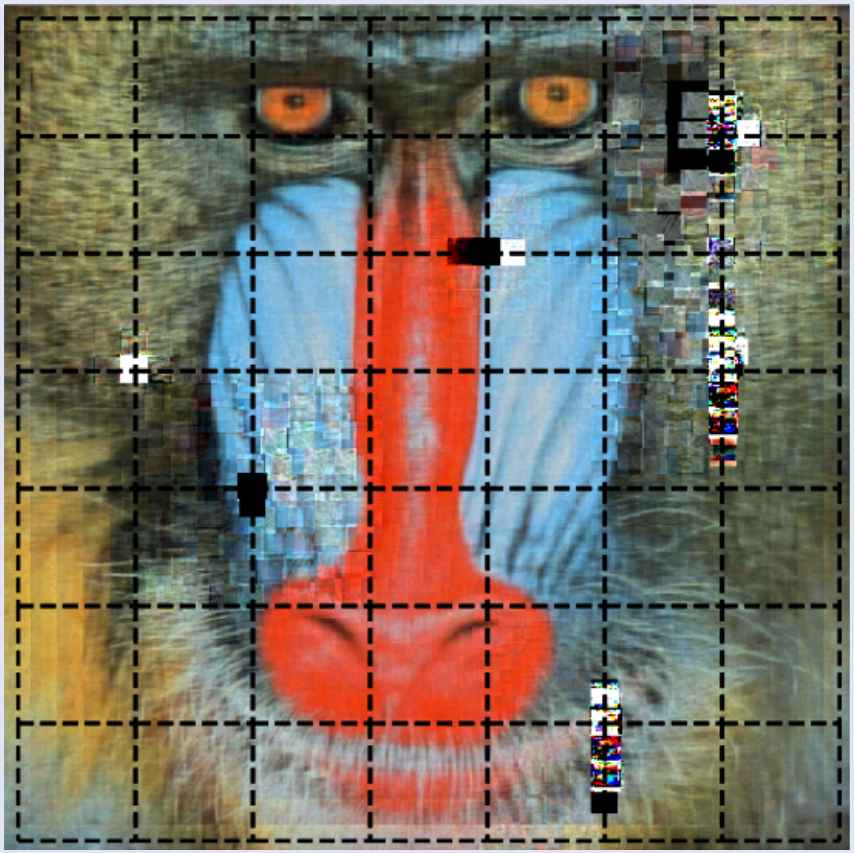
- DICOD does not work for higher dimensional signals.

Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)



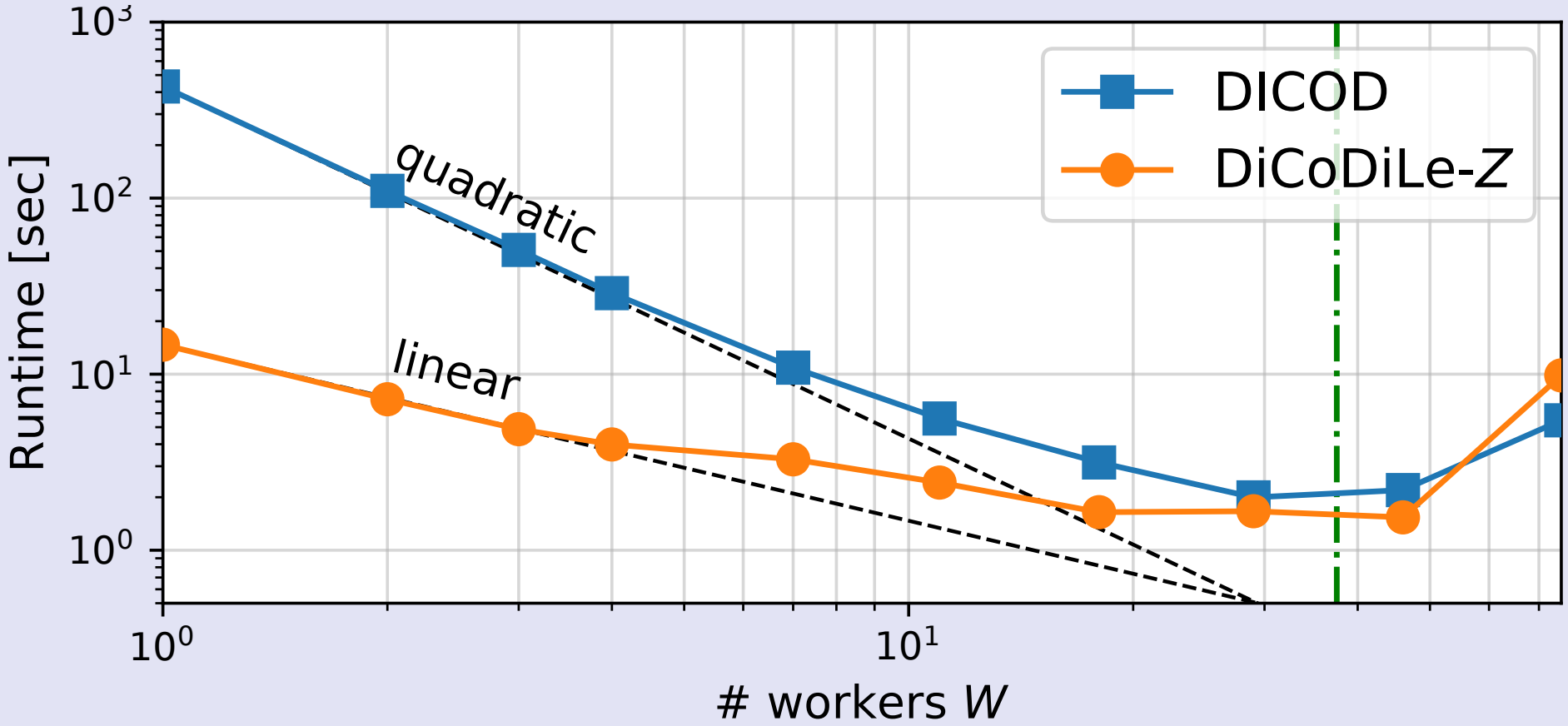
- DICOD does not work for higher dimensional signals.
- Extension require to control interferences.

Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)



- 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 ℓ_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

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

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

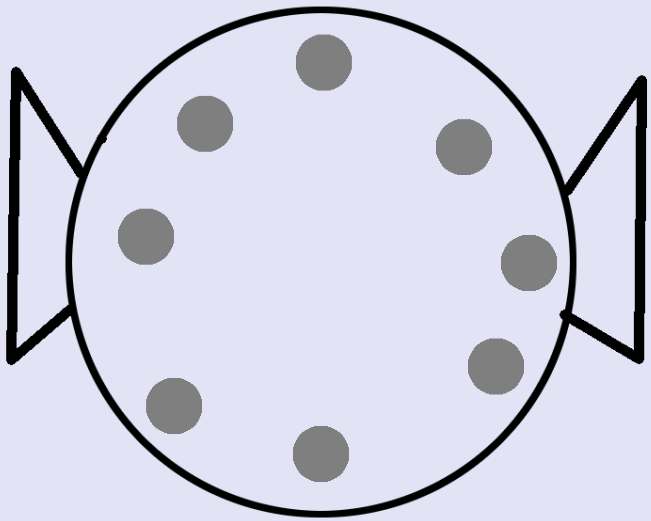
$$\nabla_{d_k} E(\{\boldsymbol{D}_k\}_k) = \sum_{n=1}^{N} (z_k^n)^{\leftharpoonup} * \left(x^n - \sum_{l=1}^{K} z_l^n * \boldsymbol{D}_l \right) = \Phi_k - \sum_{l=1}^{K} \Psi_{k,l} * \boldsymbol{D}_l \ ,$$

$\Rightarrow$ 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.

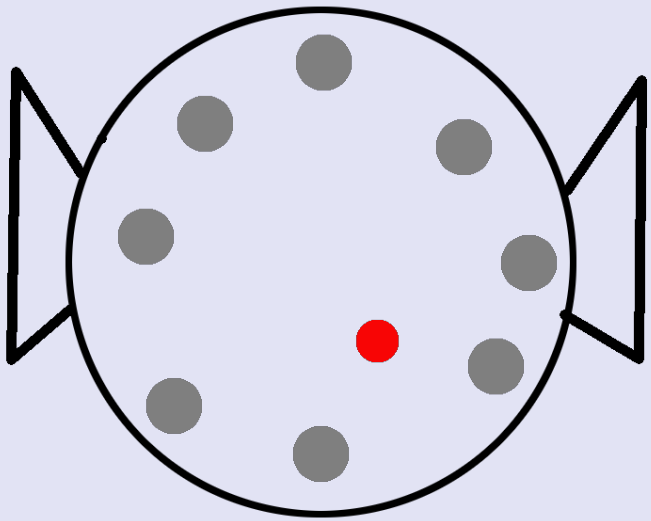
EM wave diffusion

- Recording here with 8 sensors



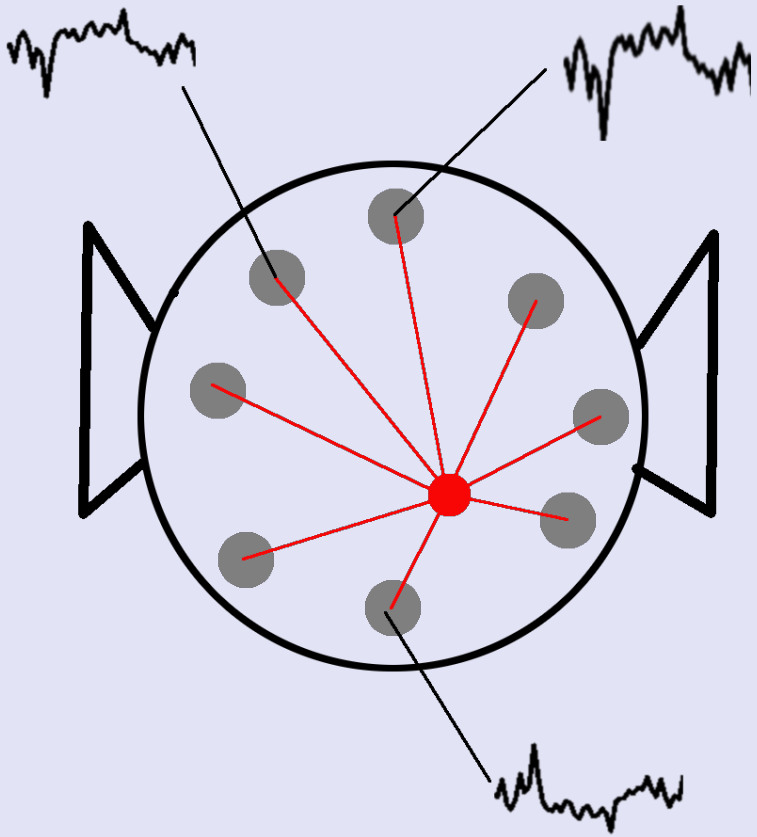
EM wave diffusion

- 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

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.* $D_k = u_k v_k^\top$.

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

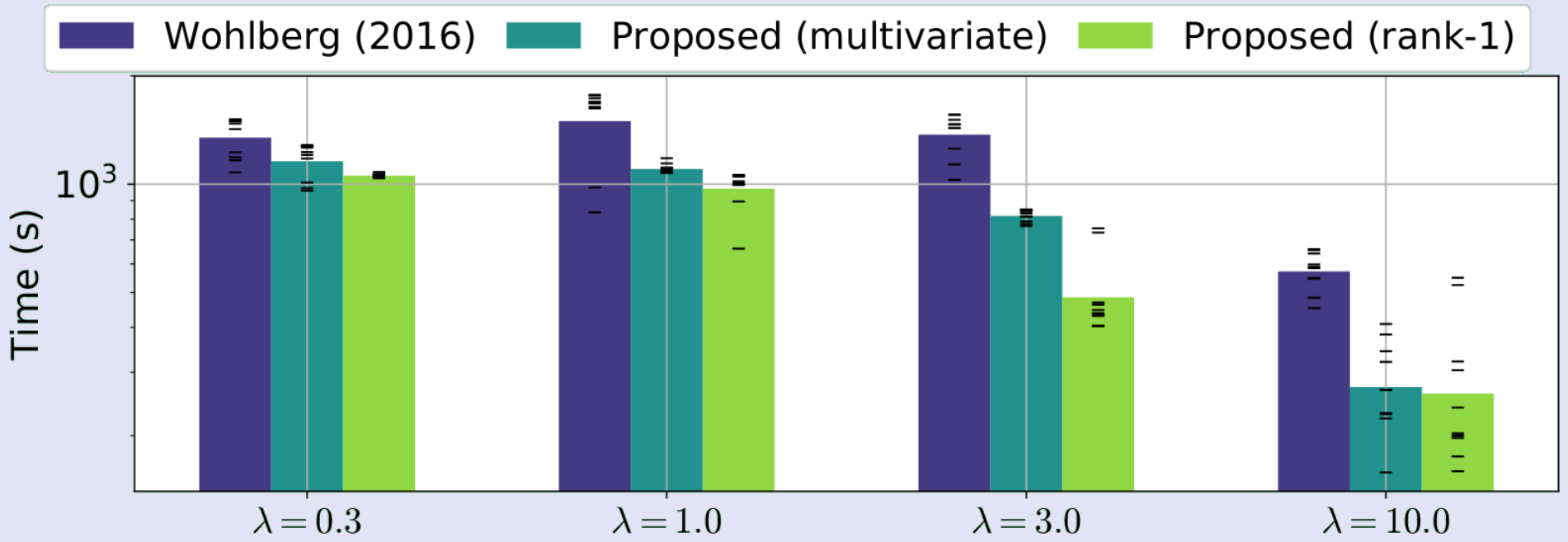
Here,

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

$\Rightarrow$ Tri-convex optimization problem , solved with alternate minimization.

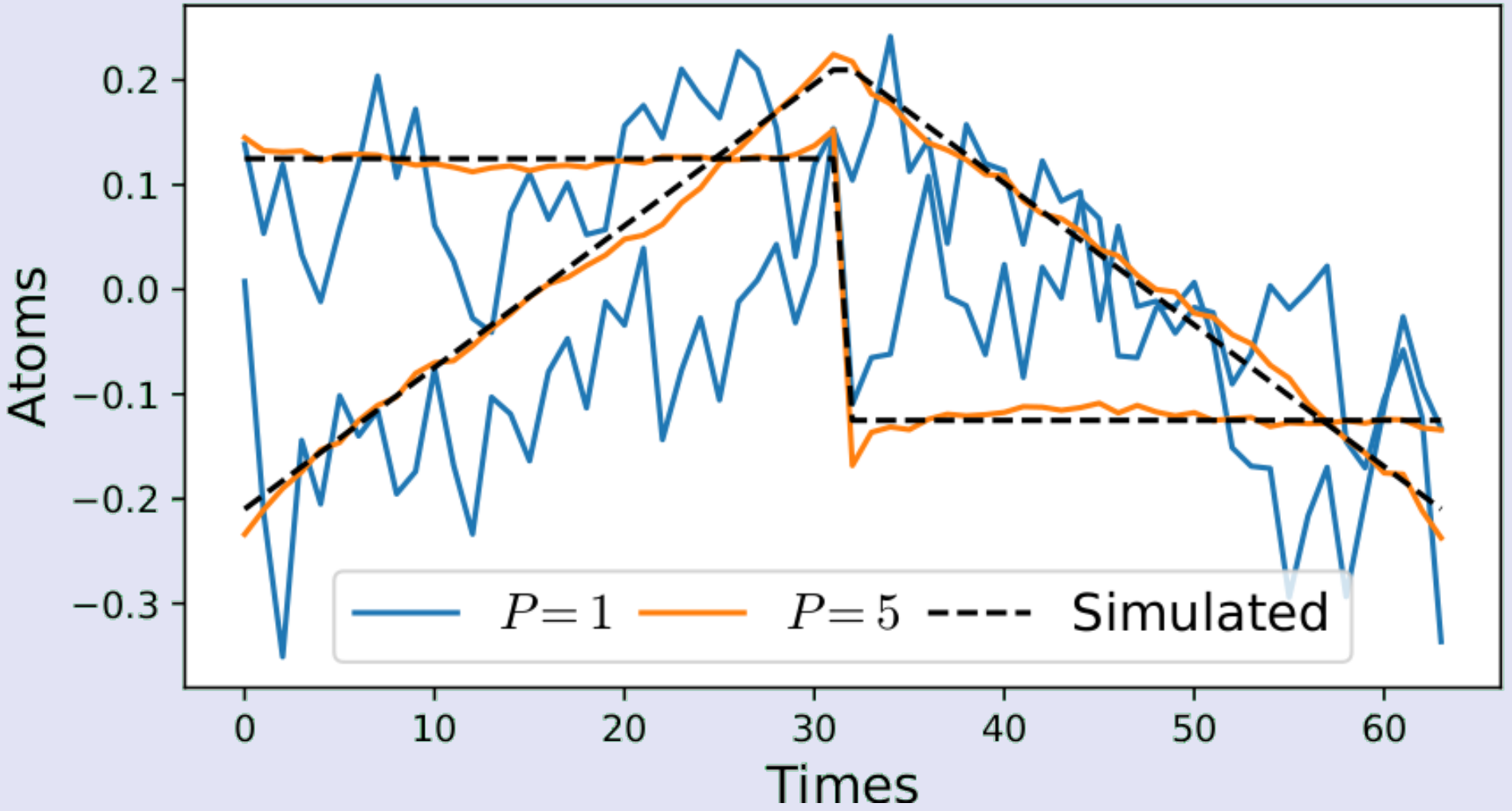
Fast optimization

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



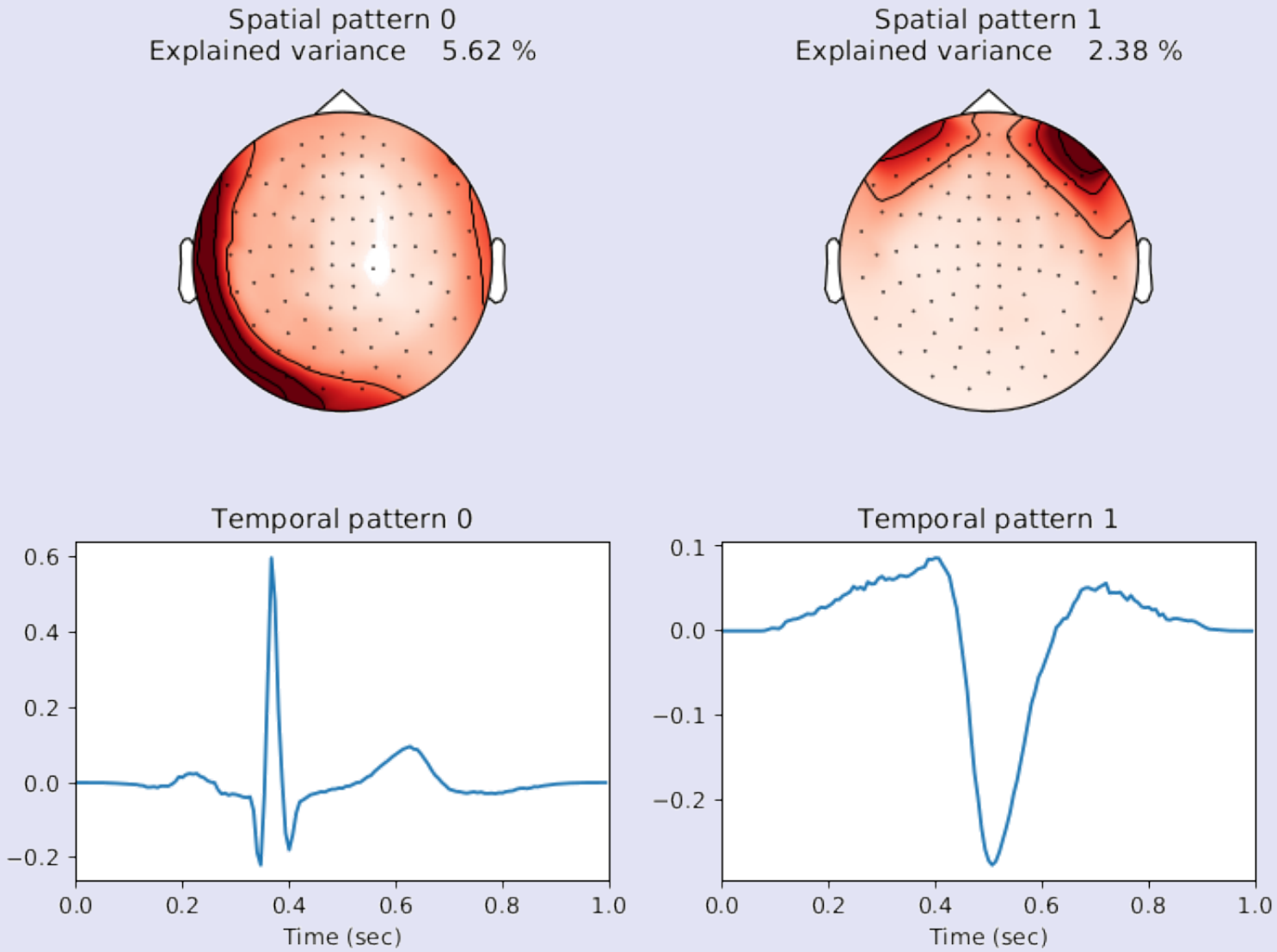
Pattern recovery

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



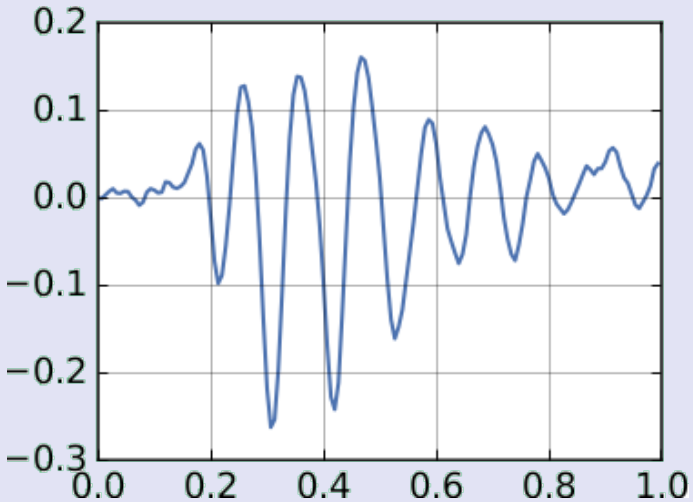
MNE somatosensory data

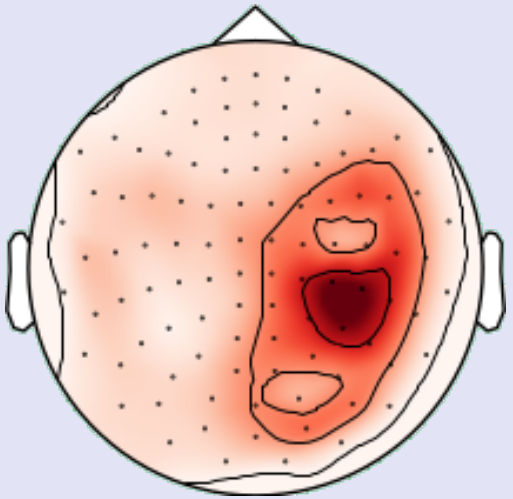
A selection of temporal waveforms of the atoms learned on the MNE sample dataset.

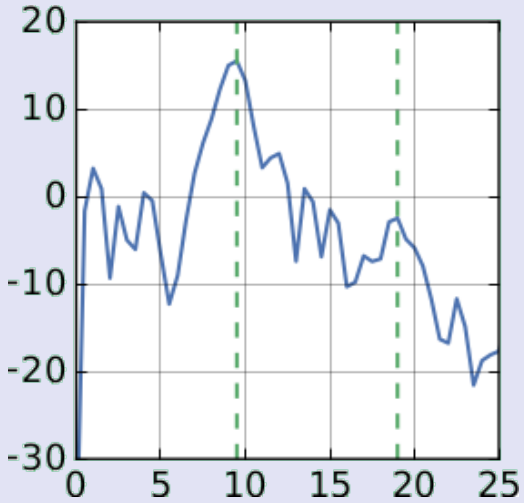


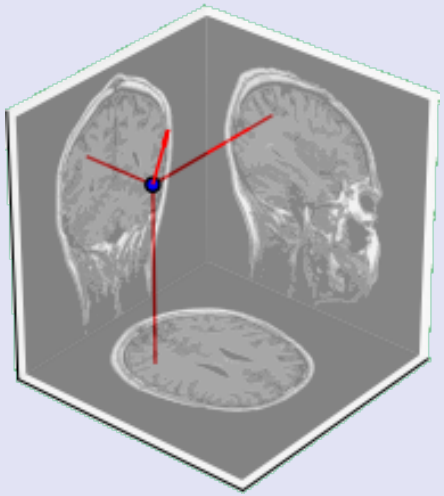
MNE somatosensory data

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

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.

Thanks!

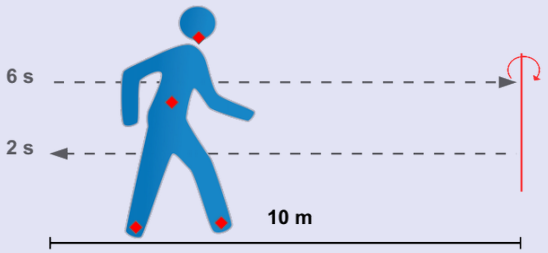
Code available online:

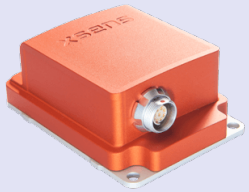
- **LISTA** : github.com/tommoral/AdaptiveOptim
- **DICOD** (& DiCoDiLe soon) : github.com/tommoral/dicod
- **alphacsc** : alphacsc.github.io

Slides are on my web page:

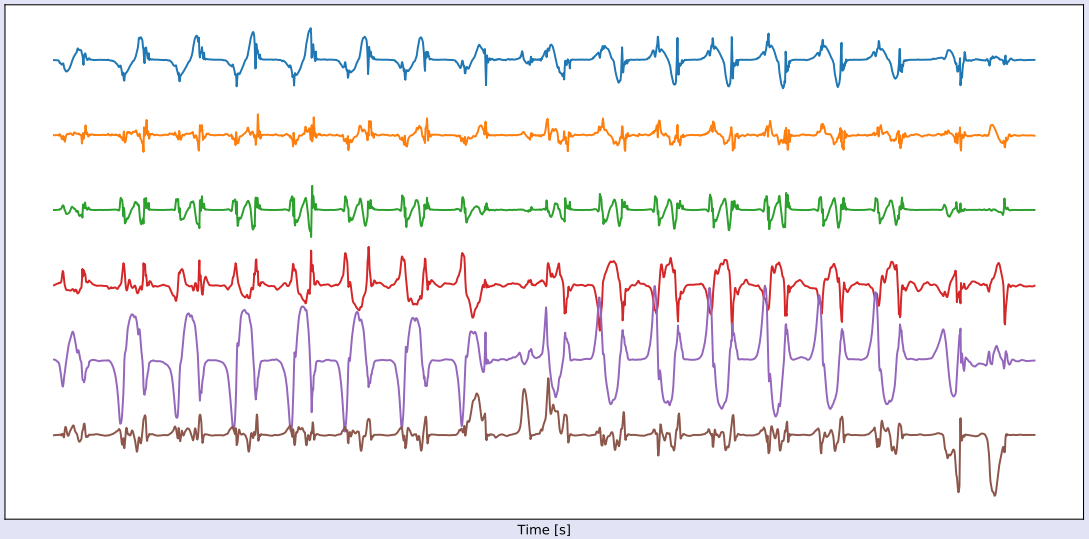
tommoral.github.io @tomamoral

Signals from human walking





Inertial captor



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

$\Rightarrow$ Can we do better with data-driven approach?

Experiment

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

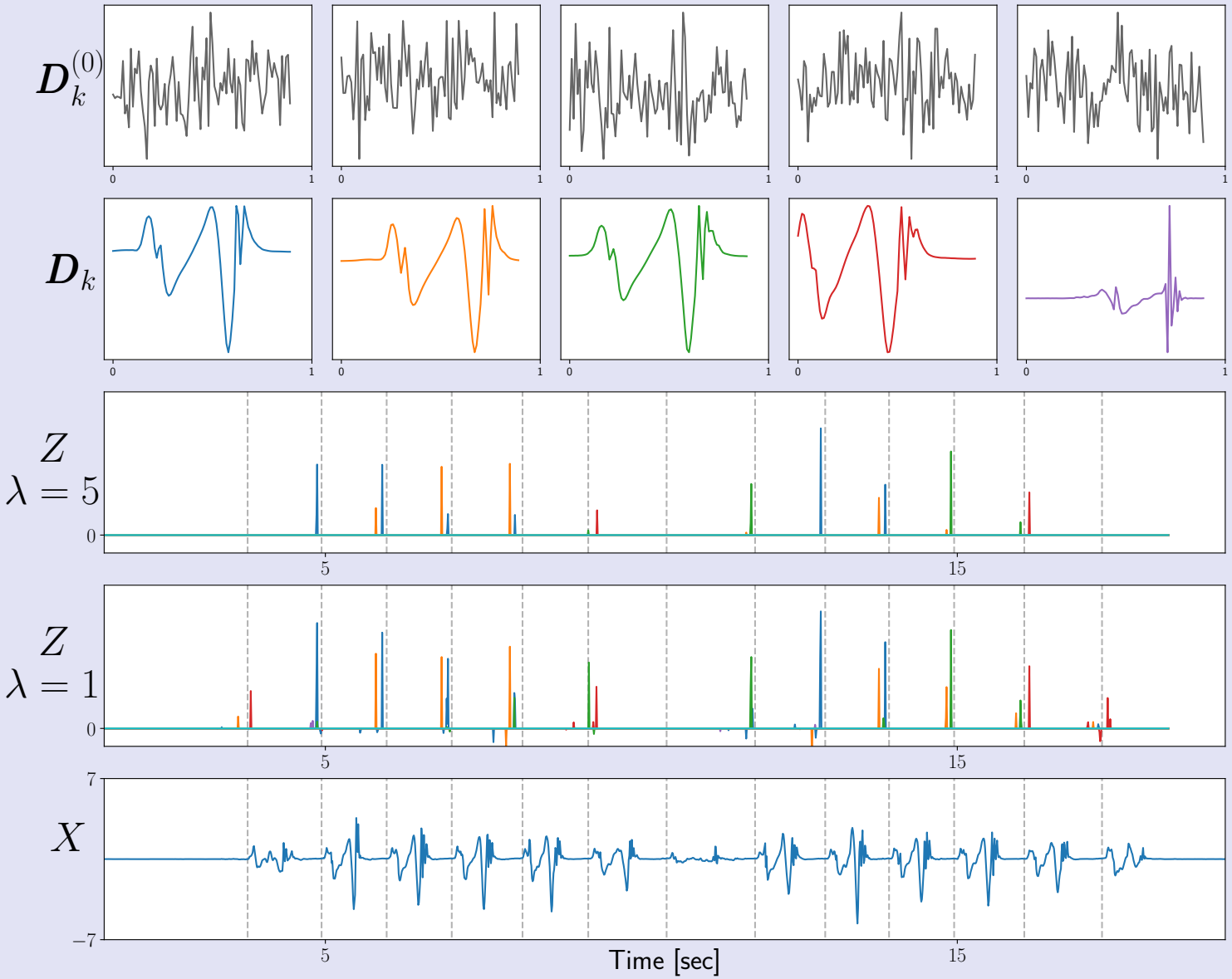
$$\boldsymbol{D}_k^{(0)} \sim \mathcal{N}(0, I_{90})$$

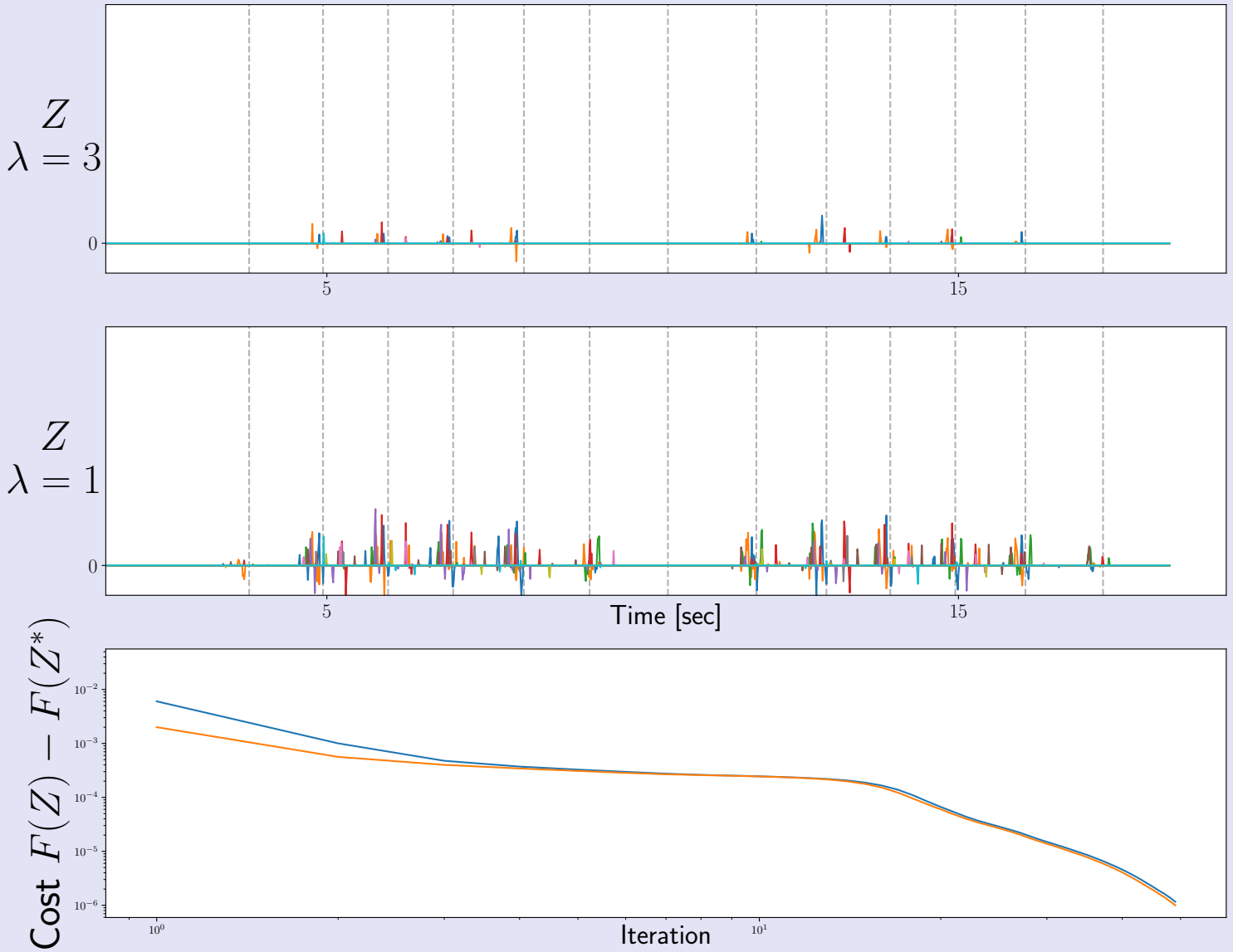
Use the Convolutional Dictionary Learning with DICOD to learn a dictionary $\boldsymbol{D}$ on a set of 50 recording of healthy subjects walking.

Challenges

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

Experiment




Z
$\lambda = 3$
0
5
15
Z
$\lambda = 1$
0
5
Time [sec]
15
Cost $F(Z) - F(Z^*)$
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{0}
10^{1}
Iteration

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 ℓ_2 unit sphere $\mathcal{S}^{p-1}$.

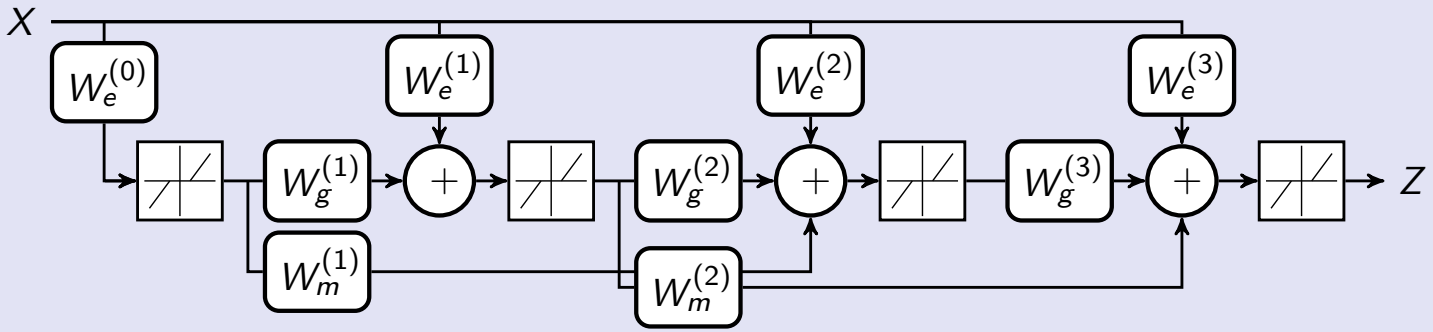
Theorem (Generic Acceleration)

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

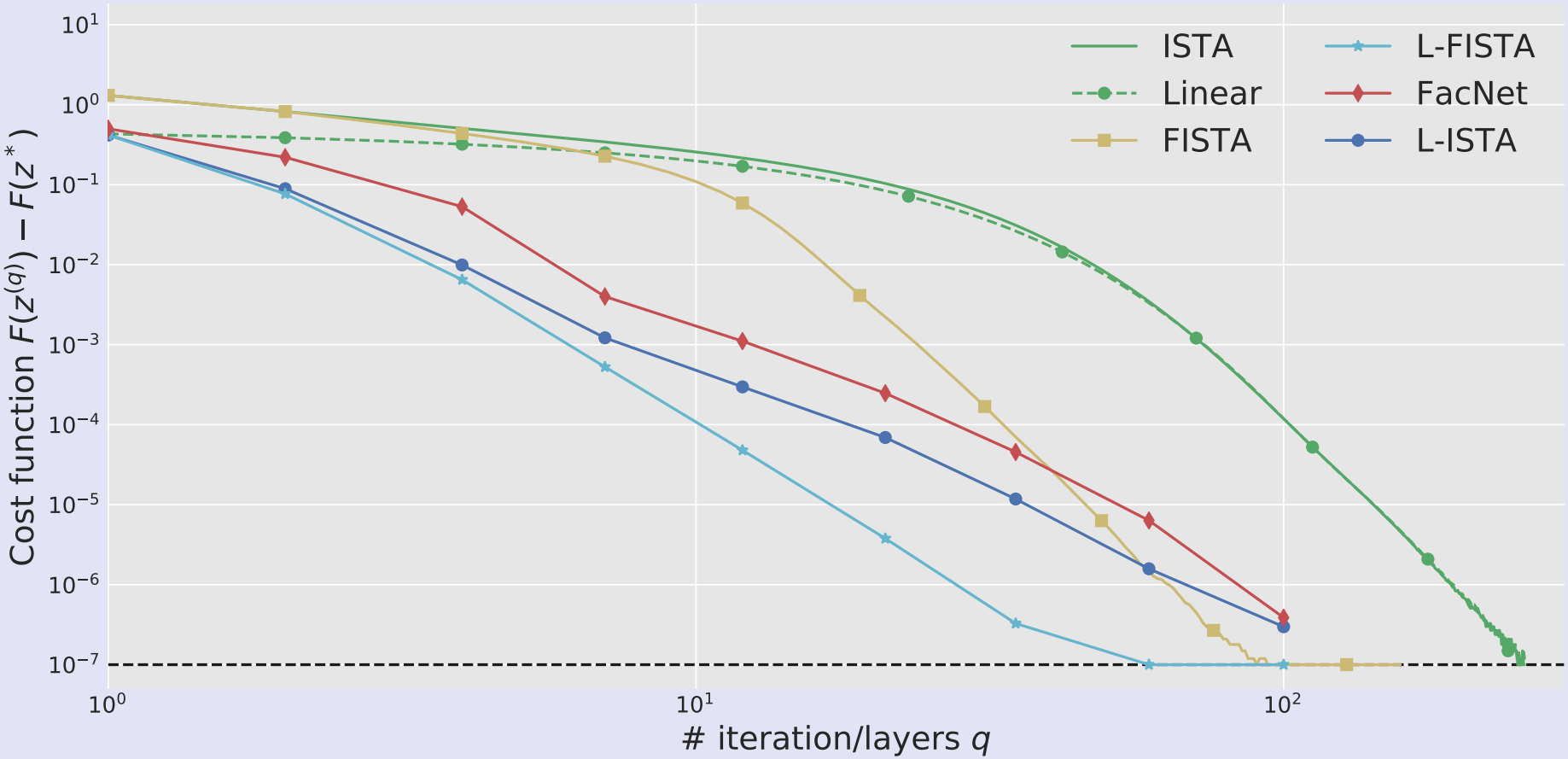
$$\lambda \mathbb{E}_z \left[\|z^{(q+1)}\|_1 + \|z^*\|_1 \right] \leq \sqrt{\frac{K(K-1)}{p}} \underbrace{\mathbb{E}_z \left[\|z^{(q)} - z^*\|_2^2 \right]}_{\substack{\text{expected resolution} \\ \text{at iteration } q}}$$

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

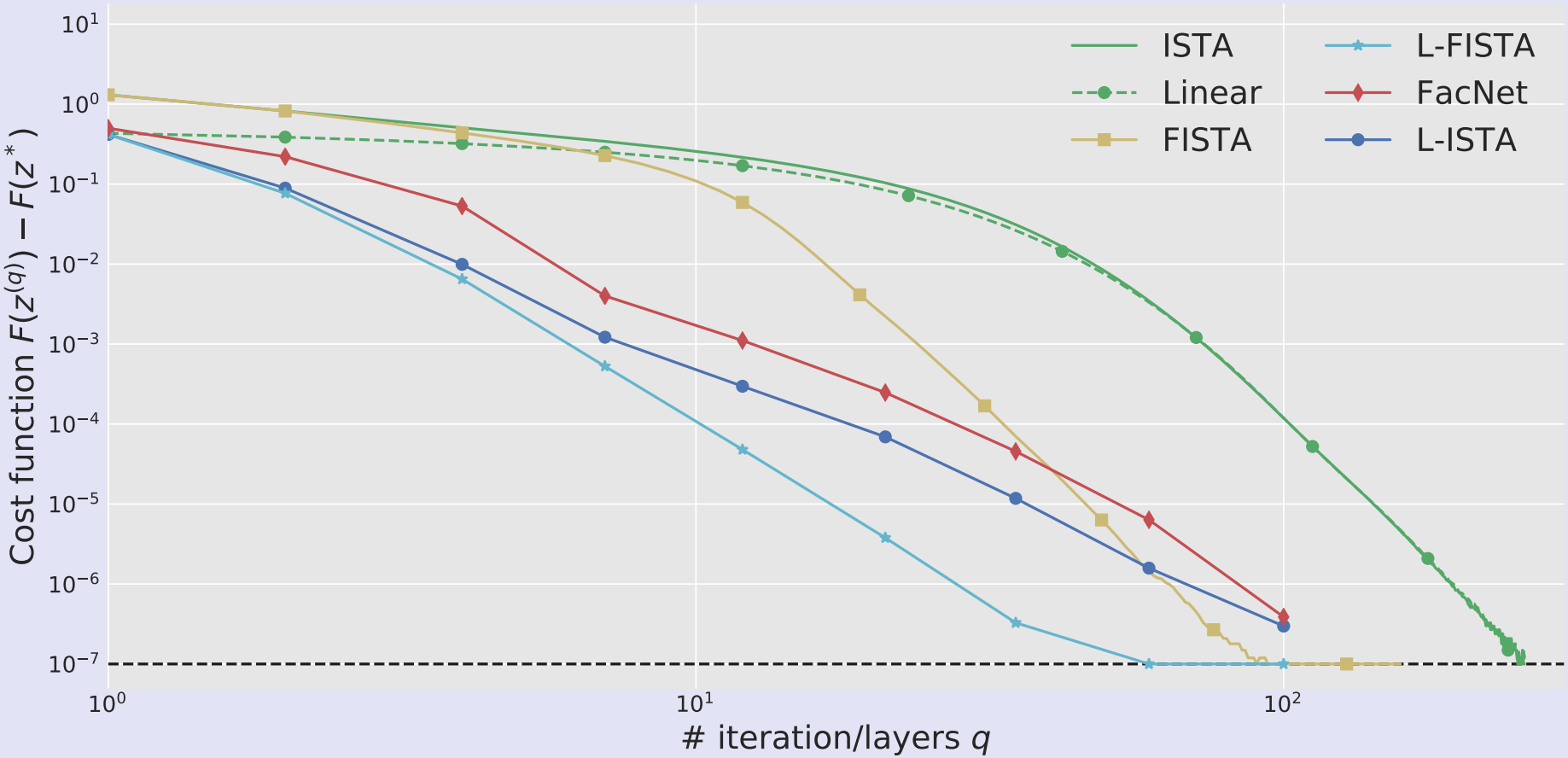
L-FISTA



Network architecture for L-FISTA.

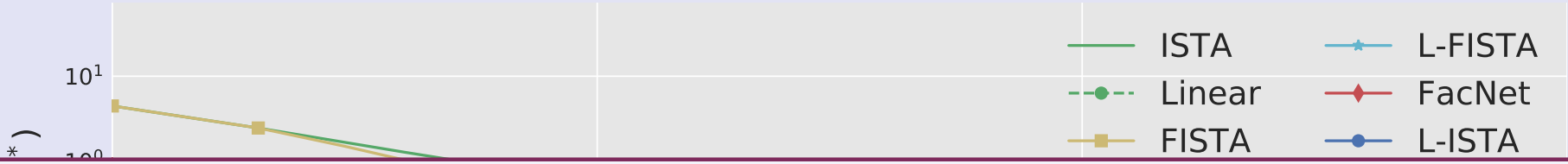


Evolution of the cost function $F(z^{(q)}) - F(z^*)$ with the number of layers/iterations q with a denser model

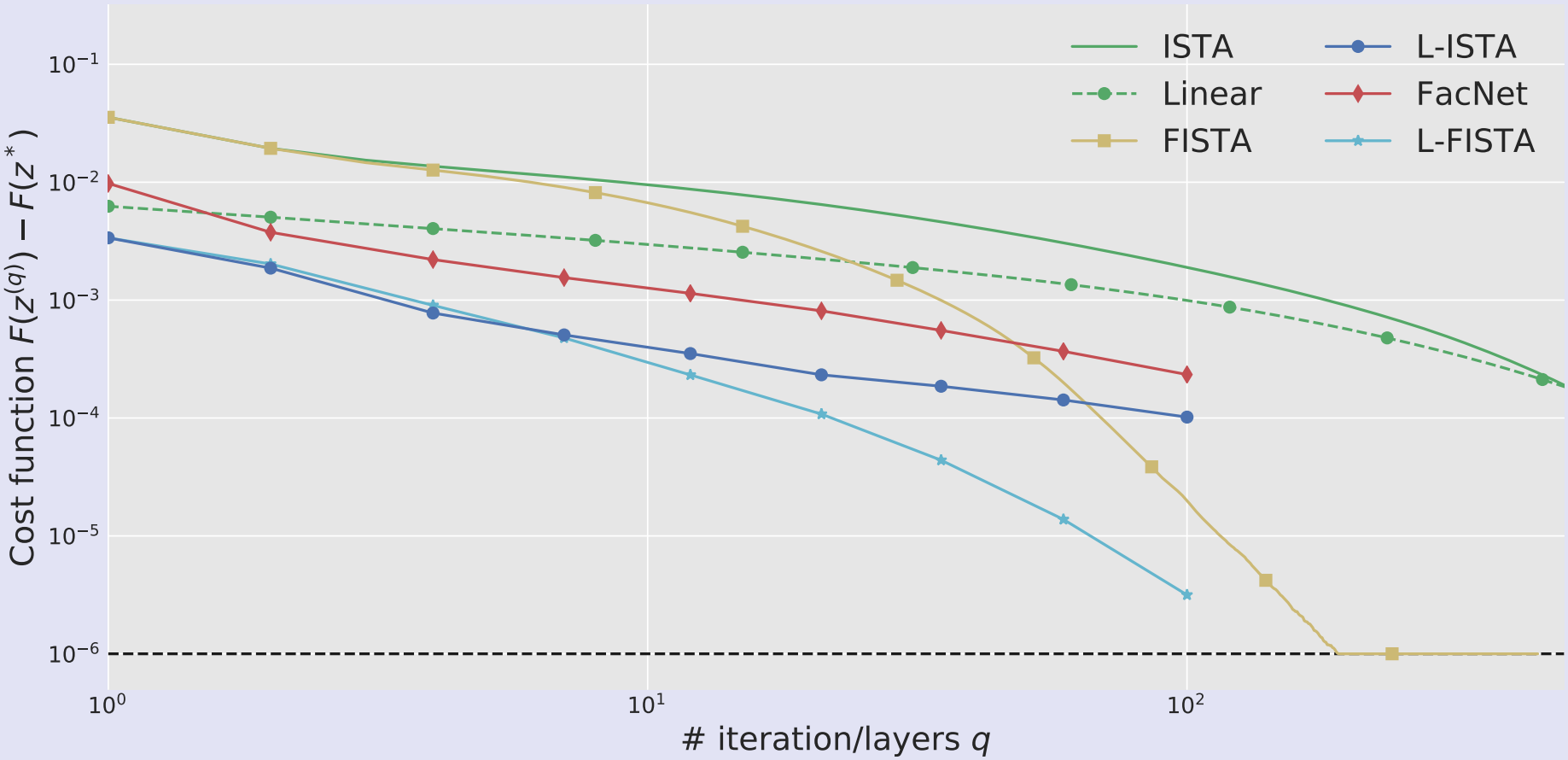


$$\rho = {}^{1}/_{20}.$$

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



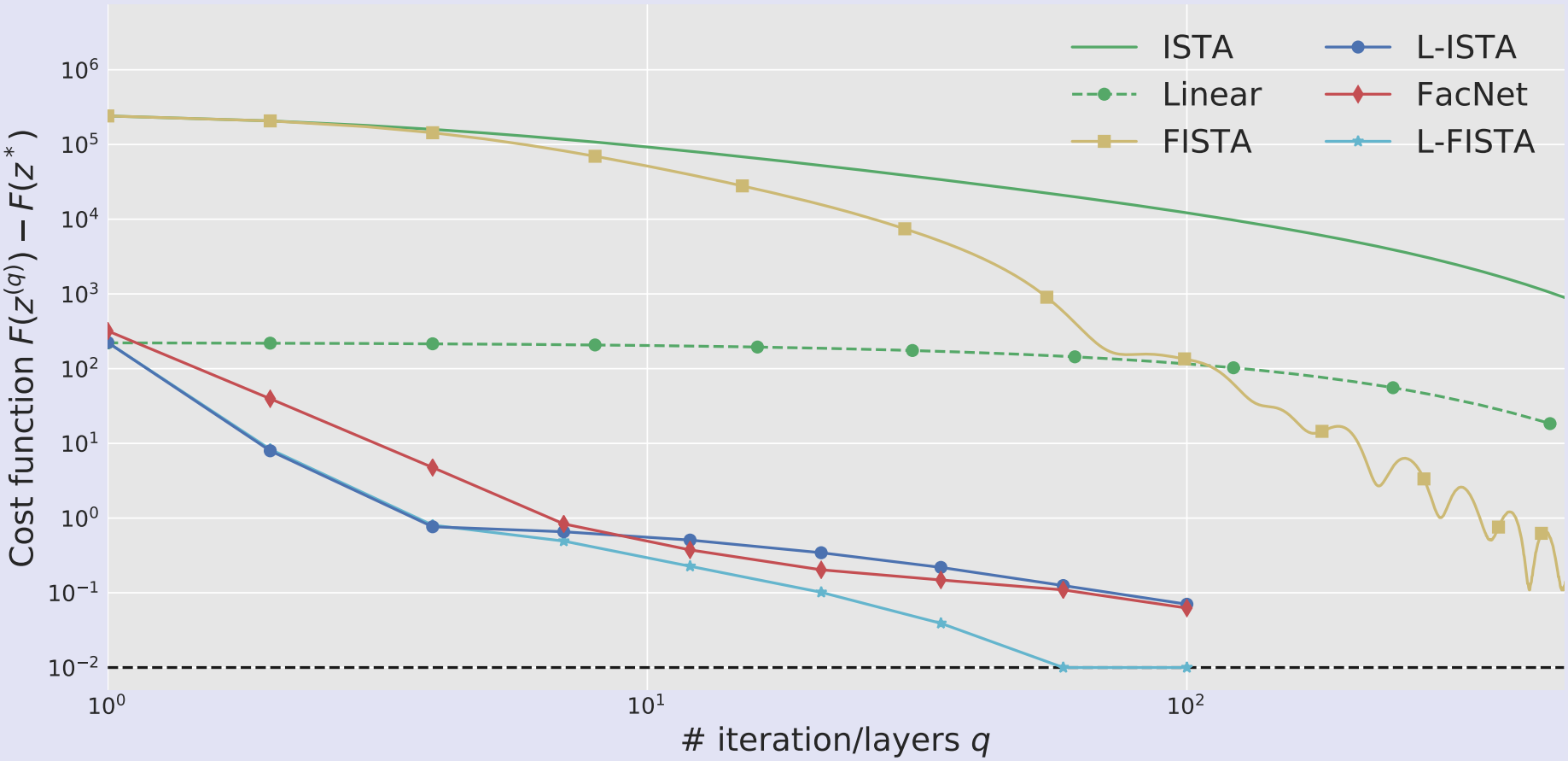
PASCAL 08



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

MNIST

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.



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?

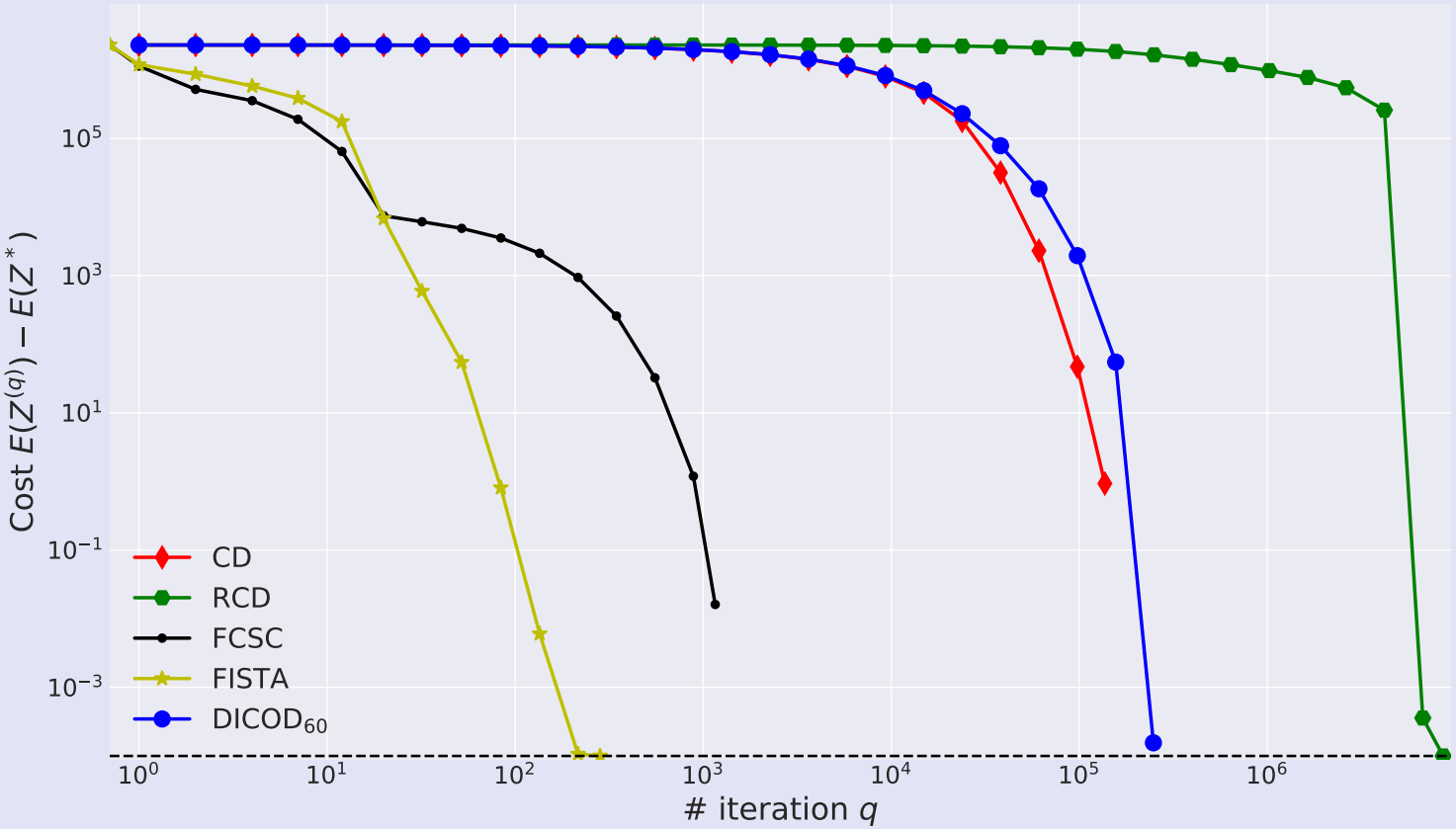
Numerical Experiments

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

Algorithms implemented for benchmark

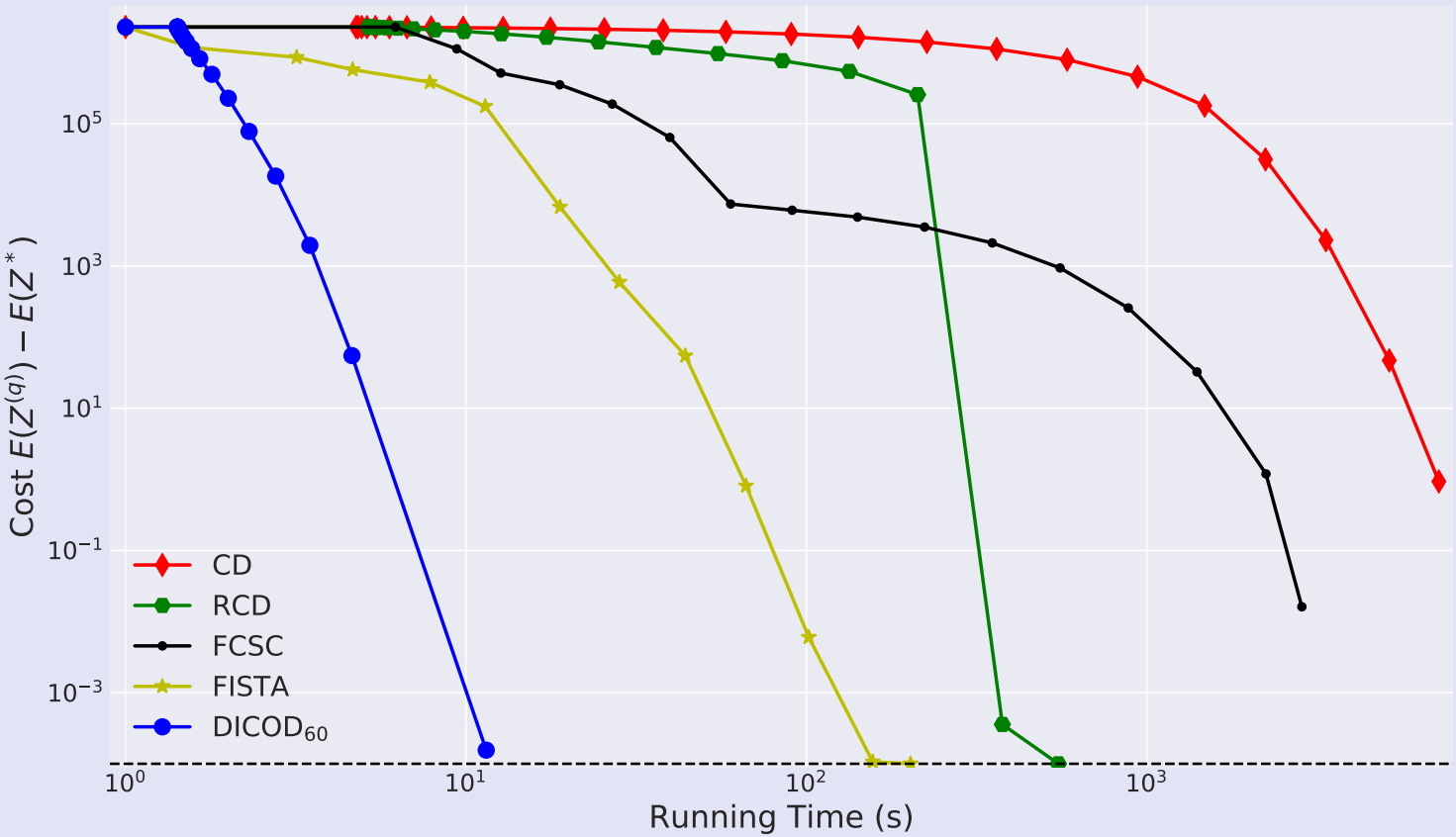
- Coordinate Descent (CD)
  [Kavukcuoglu et al., 2010]
- Randomized Coordinate Descent (RCD)
  [Nesterov, 2010]
- Fast Convolutional Sparse Coding (FCSC)
  [Bristow et al., 2013]
- Fast Ierative Soft-Thresholding Algorithm (FISTA)
  [Chalasani et al., 2013; Wohlberg, 2016]
- DICOD with 60 cores

Numerical convergence



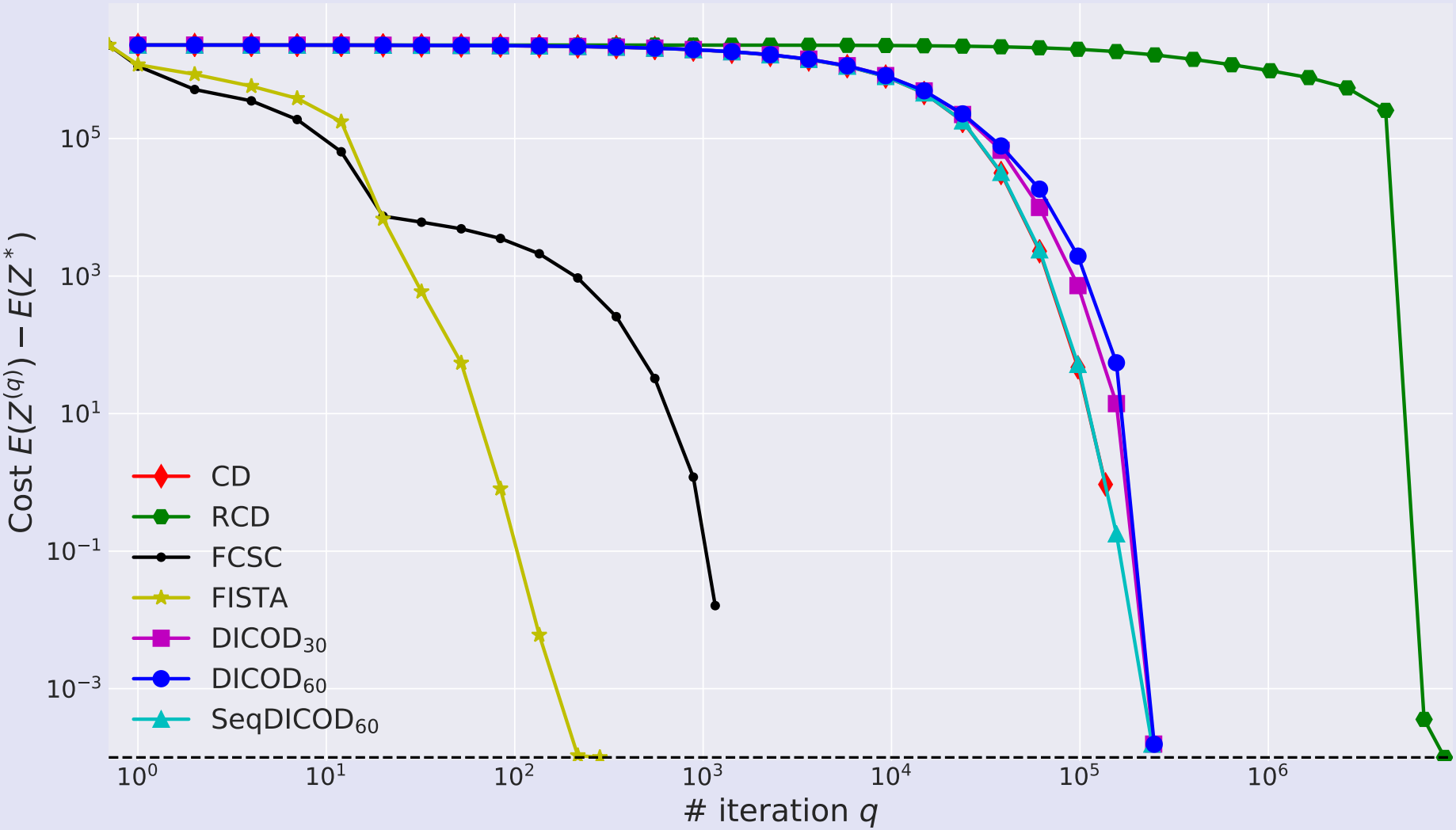
Cost as a function of the iterations

Numerical convergence



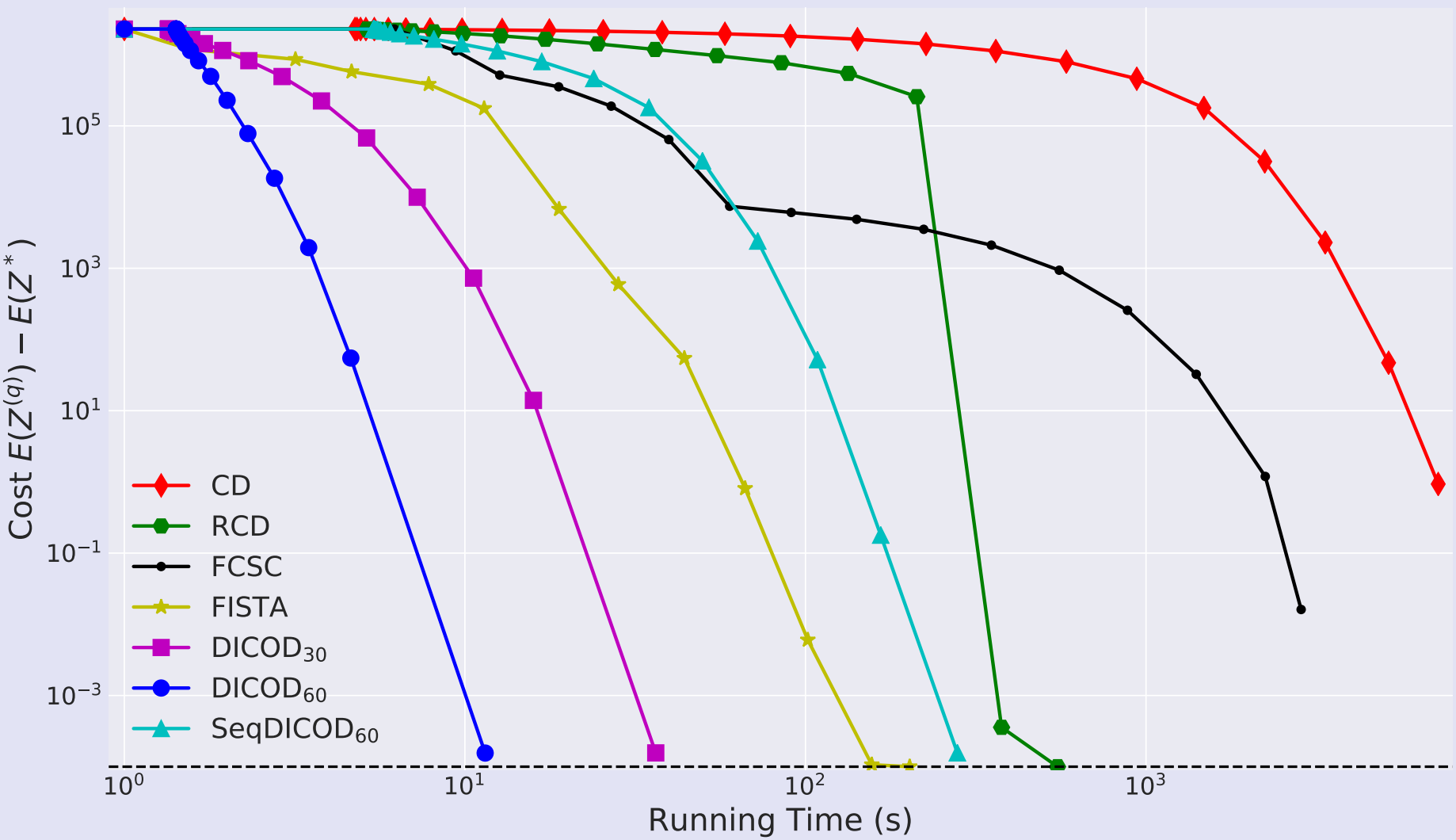
Cost as a function of the runtime

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:

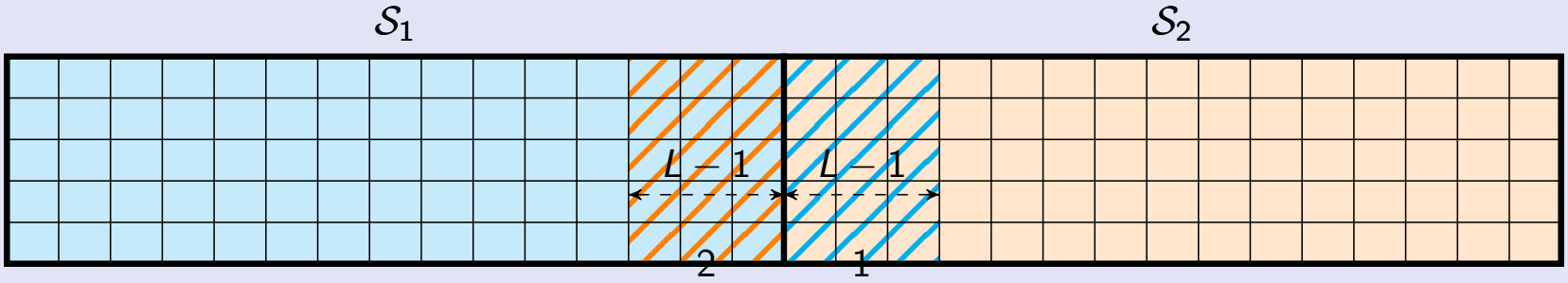
- Perform M updates in parallel,
- Each update is computed on a segment of size $\frac{L}{M}$
  Iteration complexity of $\mathcal{O}\left(K\frac{L}{M}\right)$ instead of $\mathcal{O}\left(KL\right)$

Limitations:

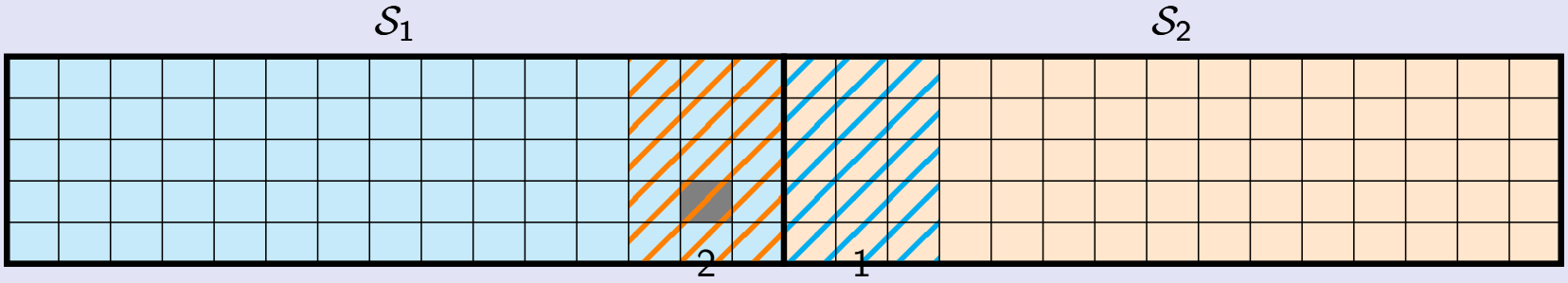
- Interfering updates, with probability $\alpha^2 = \left(\frac{WM}{T}\right)^2$

$$\mathbb{E}[Q_{dicod}] \underset{\alpha \to 0}{\gtrsim} M(1 - 2\alpha^2 M^2 + \mathcal{O}(\alpha^4 M^4)) \ .$$

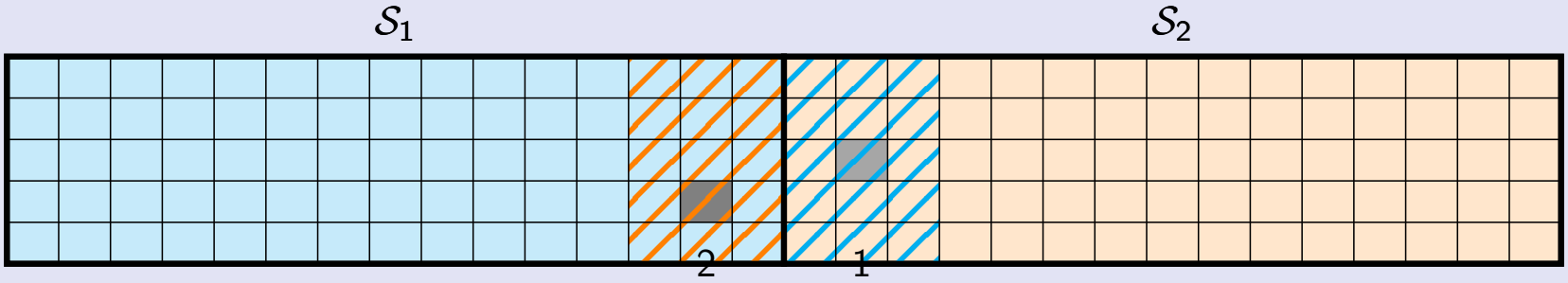
- Cost of the update of β in $\mathcal{O}\left(KW\right)$



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

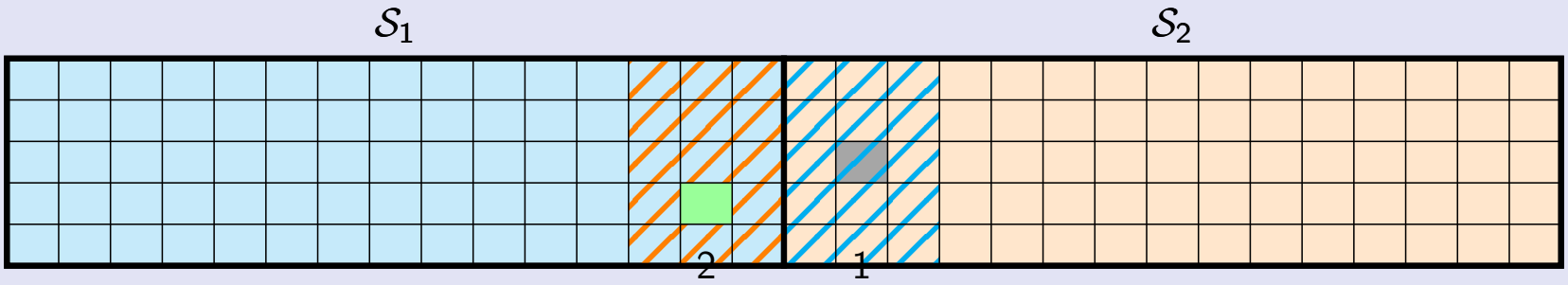


- Keep track of the value of the optimal update in an extended zone of size $L - 1$.
- Select an update candidate with LGCD.

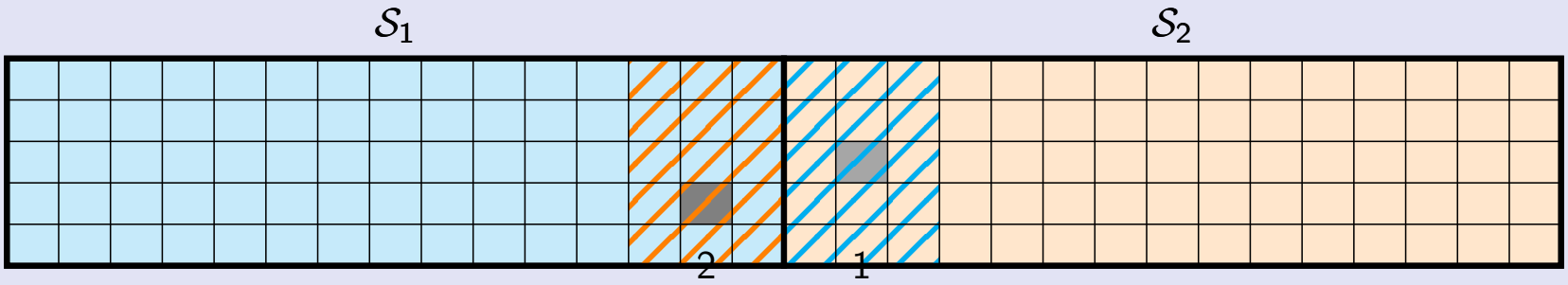


- Keep track of the value of the optimal update in an extended zone of size $L - 1$.
- Select an update candidate with LGCD.
- If it is in the interfering zone, compare the value of the update with the value potential updates in the other worker.

Soft-Locks [Moreau and Gramfort, 2019]



- 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 udpate 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 udpate if it is larger than the other update.

$\Rightarrow$ Give an update order asynchronously.

Pattern recovery

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