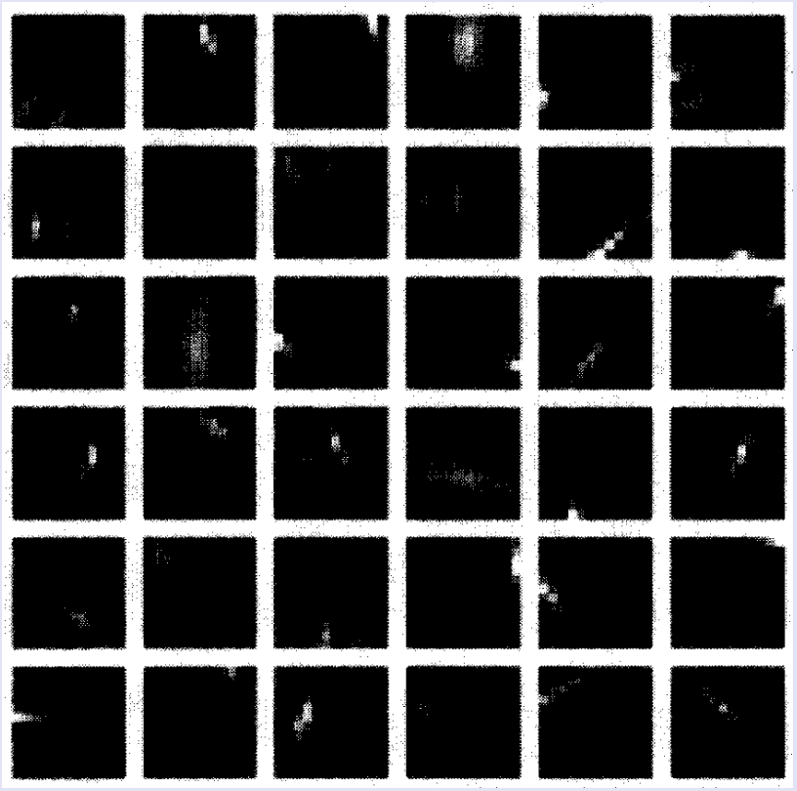
Distributed Convolutional Dictionary Learning (DiCoDiLe): Pattern Recognition in Large Signals

Thomas Moreau – INRIA – Université Paris Saclay

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

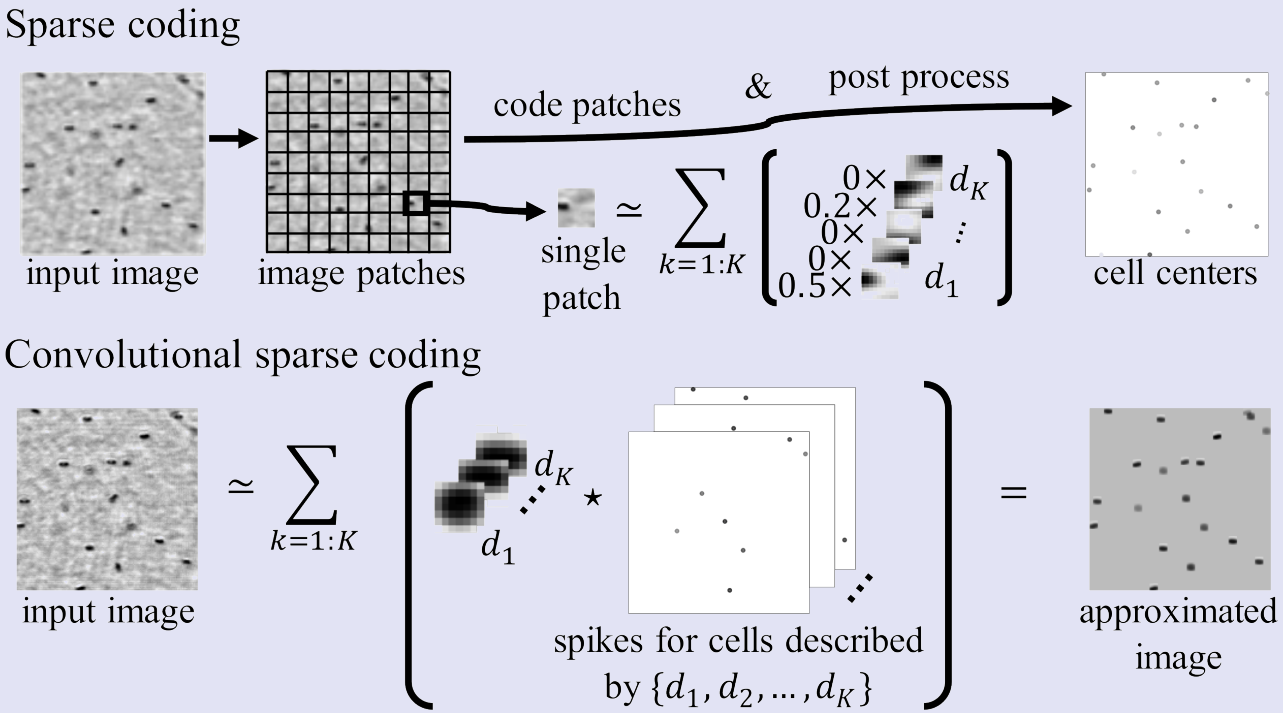
Goal:

- Feature extraction,
- Signal exploration.



Patches learned with natural images in Olshausen and Field 1997.

Convolutional Dictionary Learning [Grosse et al. 2007, UAI]

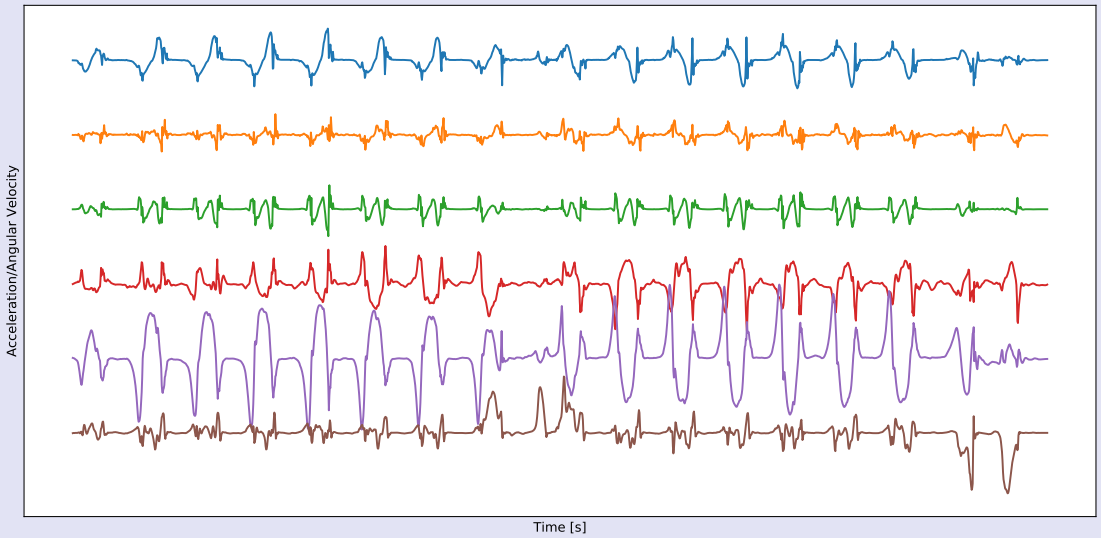


(Credit to Yellin et al. 2017)

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

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

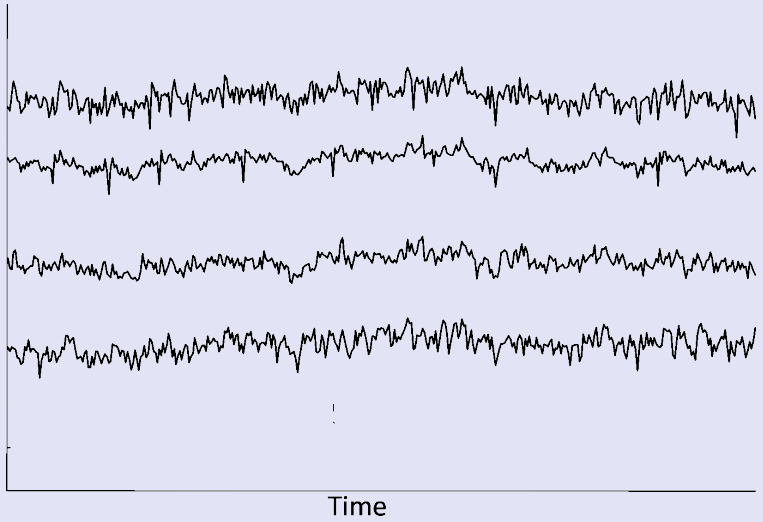
Application fields



[Oudre et al. 2018, Sensors]

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

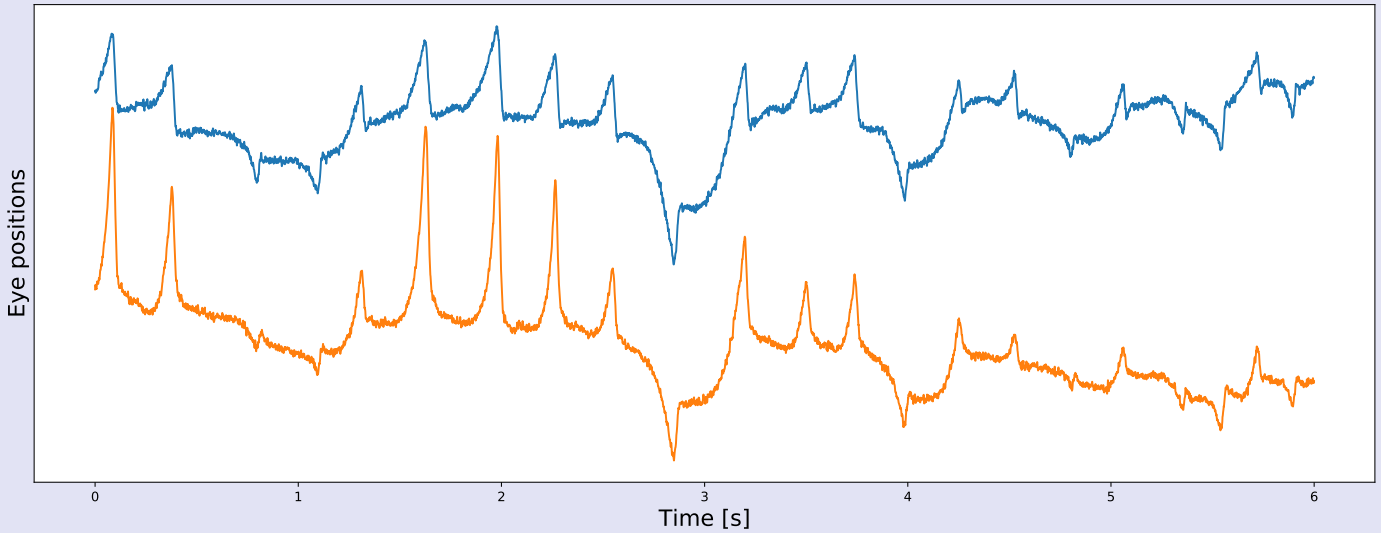
Application fields



[Dupré la Tour et al. 2018, NeurIPS; Cherkaoui et al. 2019, ICASSP]

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

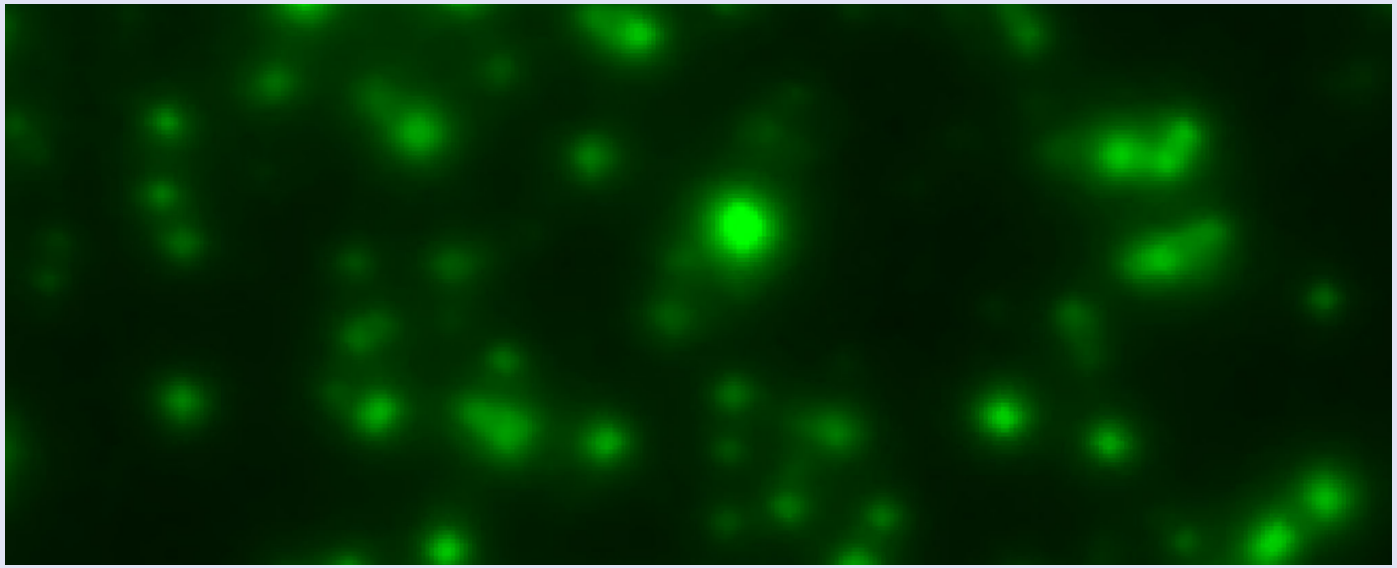
Application fields



[Robert et al. 2016, preprint]

- Detecting steps in human walk recordings to predict elderly falls.
- Exploring neurological signals from ECG, MEG or fMRI,
- Classifying pathological eye movements form 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, MEG or fMRI,
- Classifying pathological eye movements form 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, MEG or fMRI,
- Classifying pathological eye movements form 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.
    [Moreau and Bruna 2017, ICLR]
  - by parallellization.
    [Moreau et al. 2018, ICML]; [Moreau and Gramfort 2019, preprint]
- **Modelization:** how to incorporate prior knowledge,
  - on the activations.
    [Cherkaoui et al. 2019, ICASSP]
  - on the patterns.
    [Dupré la Tour et al. 2018, NeurIPS]
- **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 parallellization.**
    [Moreau et al. 2018, ICML]; [Moreau and Gramfort 2019, preprint]
- **Modelization:** how to incorporate prior knowledge,
  - on the activations.
    [Cherkaoui et al. 2019, ICASSP]
  - on the patterns.
    [Dupré la Tour et al. 2018, NeurIPS]
- **Evaluation:** how to evaluate the quality of the learned patterns.
- **Theoretical:** pattern recovery.

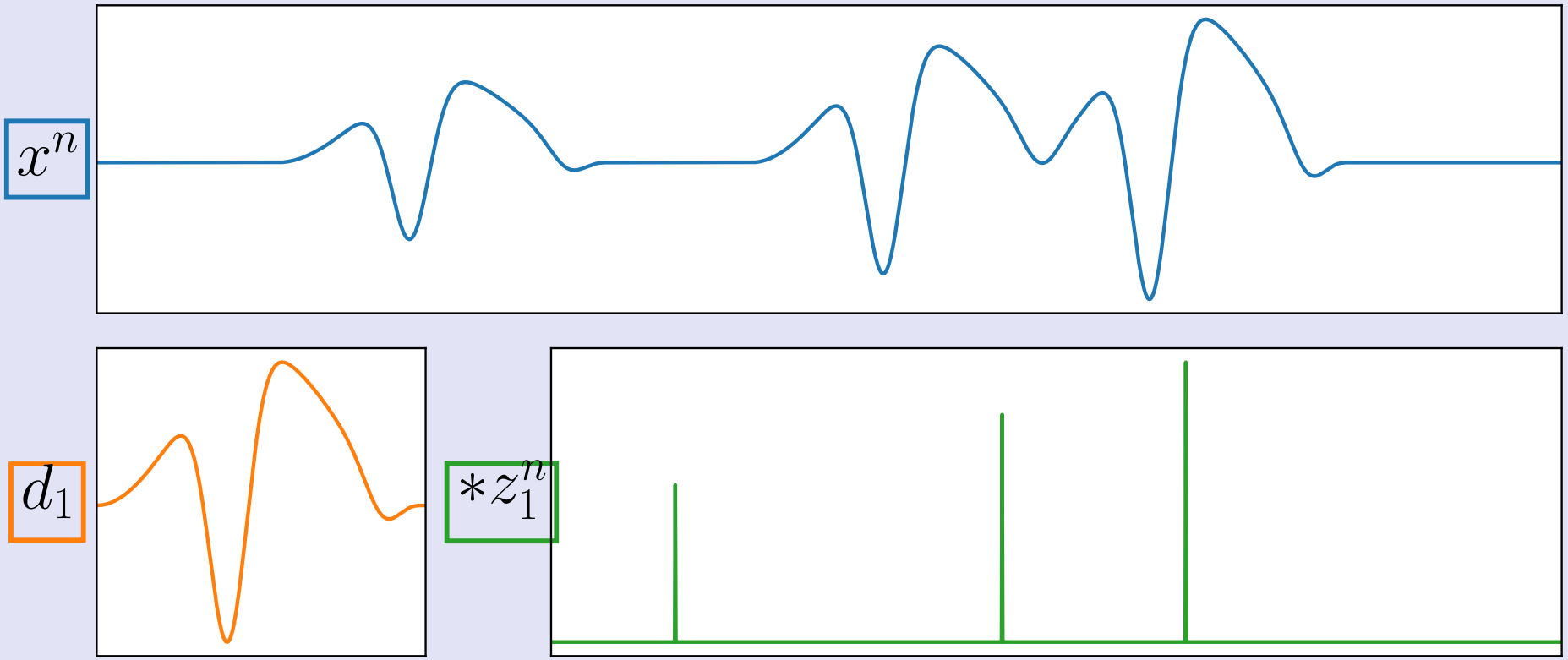
Convolutional Dictionary Learning

References

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

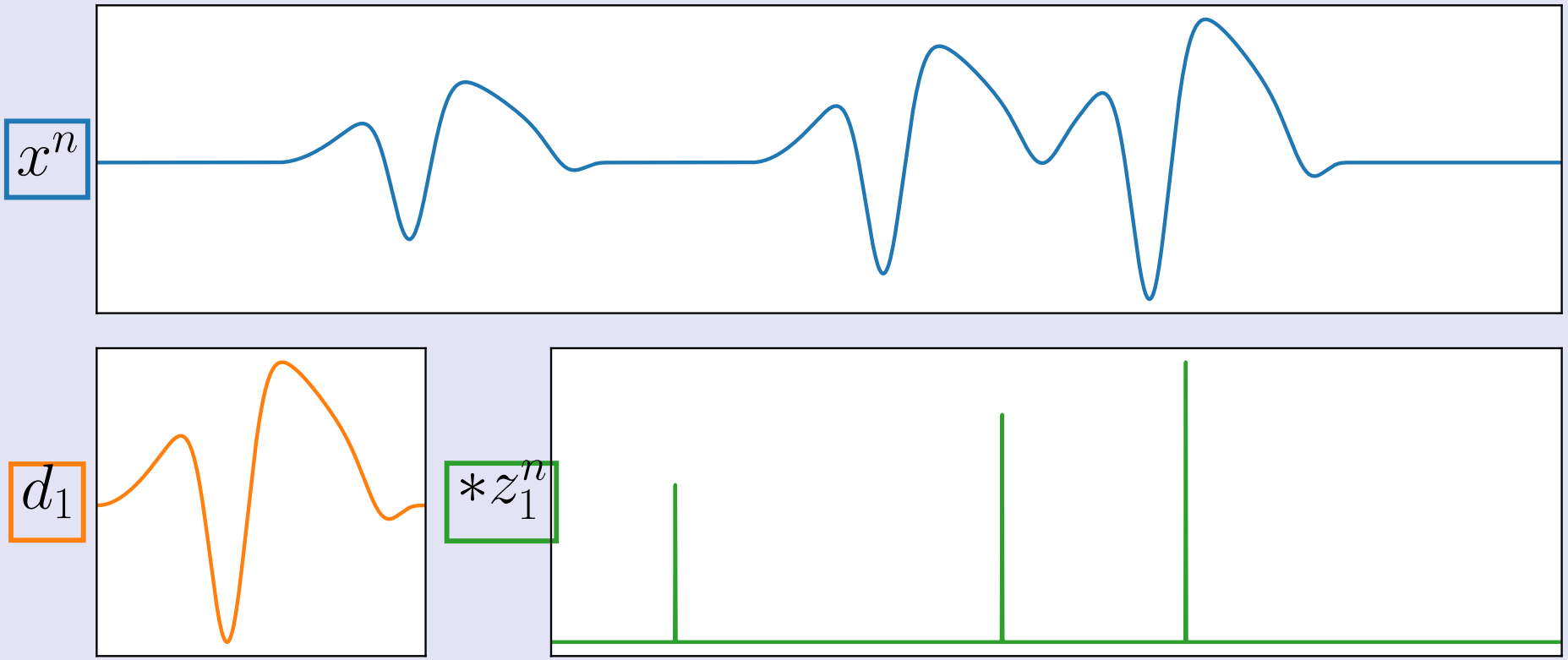
Extracting shift invariant patterns

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

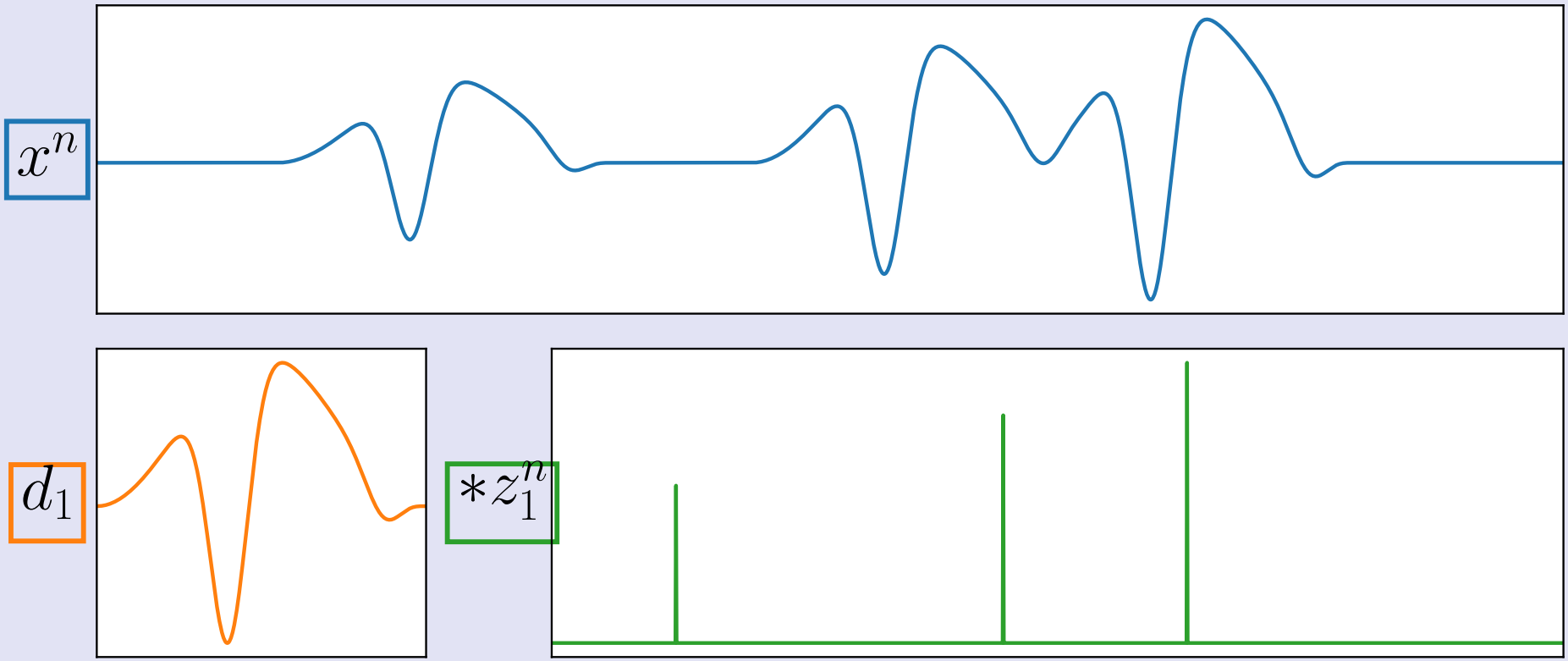


Convolutional Representation:

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

Extracting shift invariant patterns

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



Convolutional Dictionary Learning:

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

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

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 with P channels on a domain Ω of size T
- $\mathcal{E}$ is a noise signal on the same domain Ω
- $\boldsymbol{D}$ is a set of K patterns with small support $\Theta \subset \Omega$ of size L
- Z is a signal with K channels on the domain Ω.

Convolutional Dictionary Learning

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

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

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

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

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

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: Convolutional Sparse Coding (CSC)

$\rightarrow \boldsymbol{D}$ 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; 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]

Locally greedy Coordinate Descent

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)

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; 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]

Complexity Summary

- Full gradient methods: $\mathcal{O}\left(\frac{K^2 TL}{\epsilon}\right)$
  - Iteration complexity with pre-computation : $\mathcal{O}\left(K^2 TL\right)$
  - Convergence in $\mathcal{O}\left(\frac{1}{q}\right)$
- Coordinate Descent: $\mathcal{O}\left(\frac{K^2 TL}{\epsilon}\right)$
  - Iteration complexity with pre-computation : $\mathcal{O}\left(KL\right)$
  - Convergence in $\mathcal{O}\left(\frac{1}{qTK}\right)$

$\Rightarrow$ Does not seem to be much to gain using CD.

But the convergence rate could be pessimistic for CD...
If the updates are performed on the "right" coefficients, the cvg rate becomes $\mathcal{O}\left(\frac{1}{qS}\right)$ where S is the number of non-zero in Z^*.

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, ω_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; $\mathcal{O}(KT)$ [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, ω_0) to update.
2. Compute a new value $Z'_{k_0}[\omega_0]$ for this coordinate

For convolutional CD, we can use optimal updates:

$$Z'_{k_0}[\omega_0] = \frac{1}{\|\boldsymbol{D}_{k_0}\|_2^2} \mathbf{ST}(\beta_{k_0}[\omega_0], \lambda),$$

with $\mathbf{ST}(y, \lambda) = \operatorname{sign}(y)(|y| - \lambda)_+$. Kavukcuoglu et al. [2010] showed this can be done efficiently, with $\mathcal{O}(KL)$ 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, ω_0) to update.
2. Compute a new value $Z'_{k_0}[\omega_0]$ for this coordinate

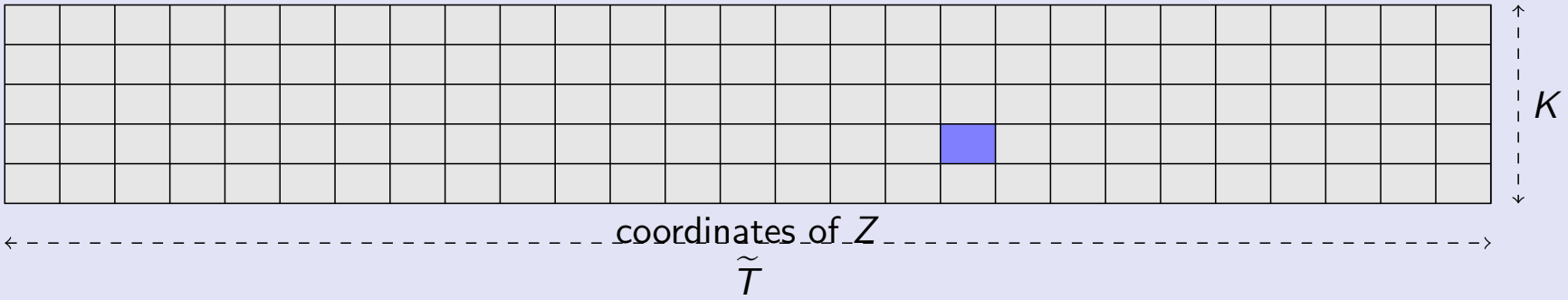
$\Rightarrow$ Converges to an optimal point for CSC problem with rate $\mathcal{O}\left(\frac{1}{q}\right)$.

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]

Locally Greedy Coordinate Descent [Moreau et al. 2018, ICML]

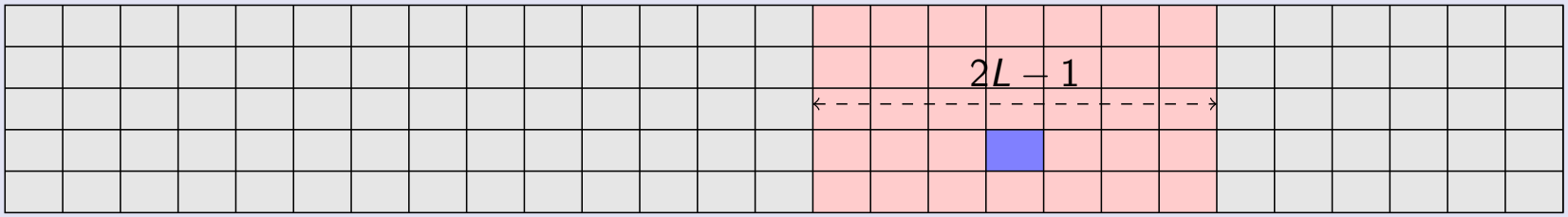
We introduced the LGCD method which is an extension of GCD.



GCD has $\mathcal{O}(KT)$ computational complexity.

Locally Greedy Coordinate Descent [Moreau et al. 2018, ICML]

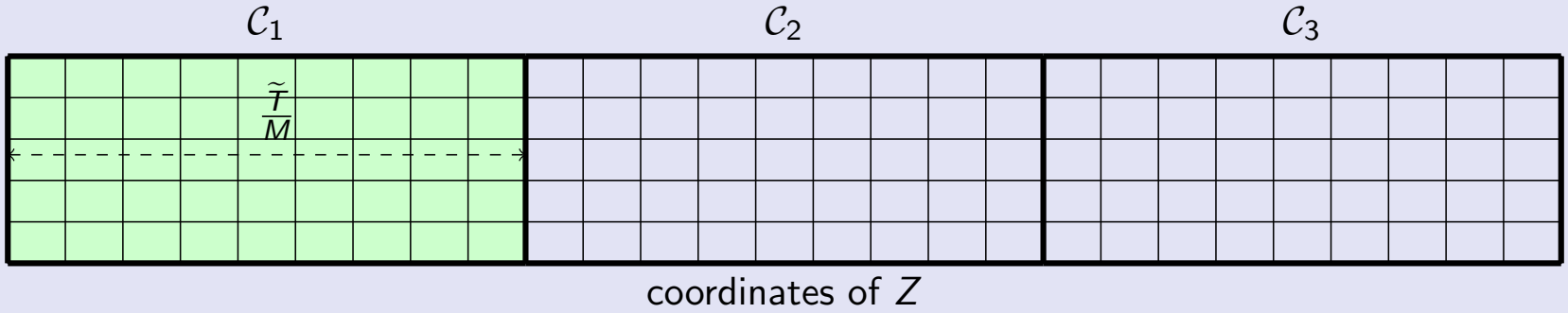
We introduced the LGCD method which is an extension of GCD.



GCD has $\mathcal{O}(KT)$ computational complexity.

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

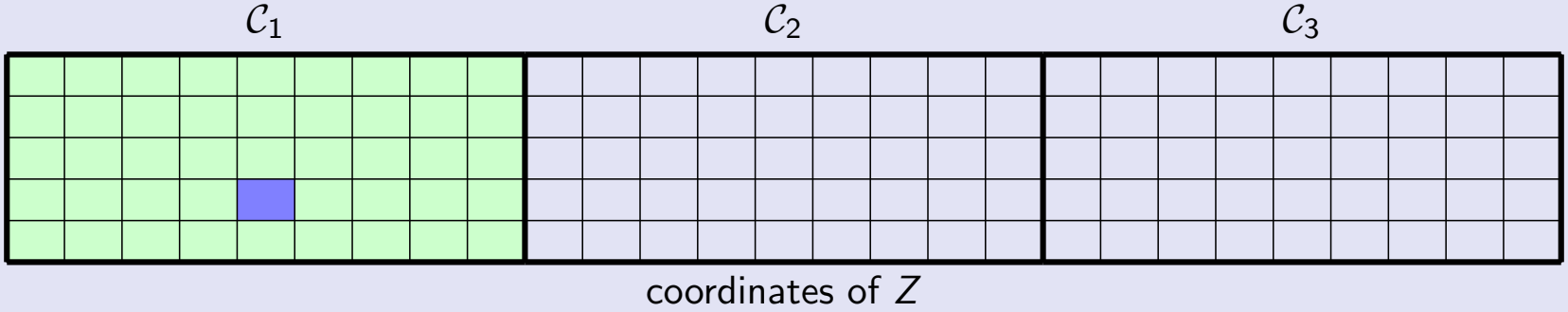
We introduced the LGCD method which is an extension of GCD.



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

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

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



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

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

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

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

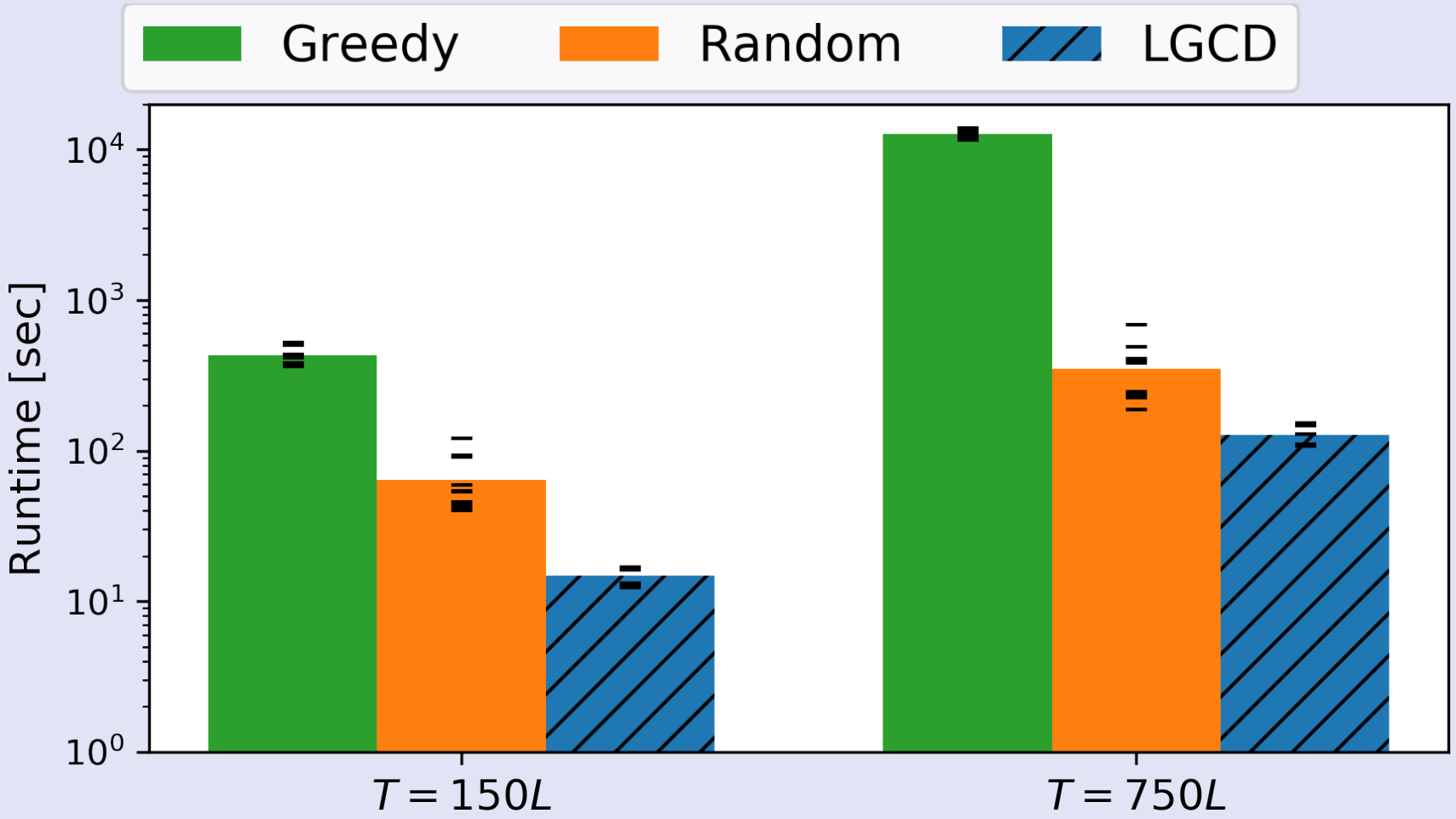
The overall iteration complexity is $\mathcal{O}(KL)$ instead of $\mathcal{O}(KT)$.

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

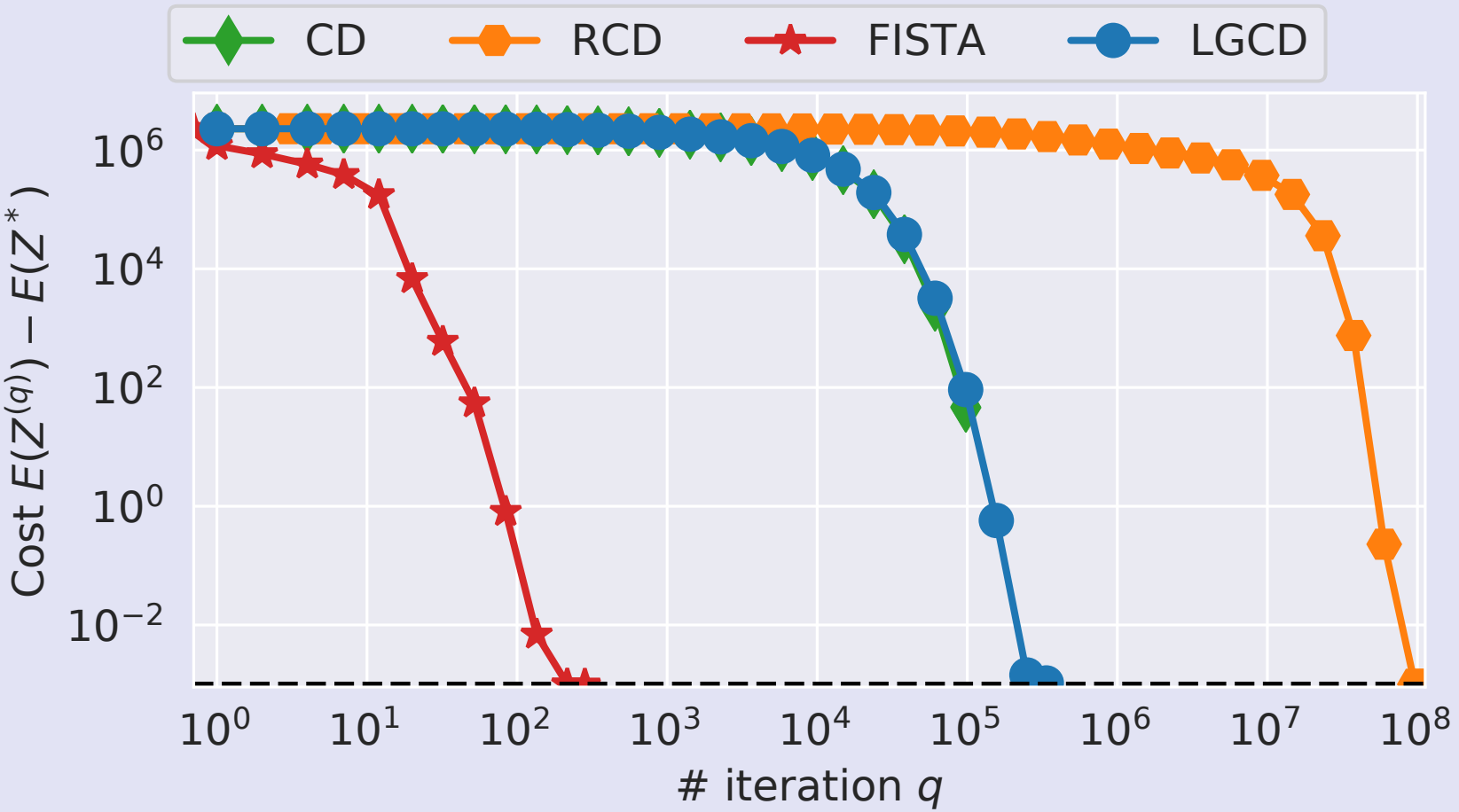
Fast optimization

Comparison of the coordinate selection strategy for CD on simulated signals

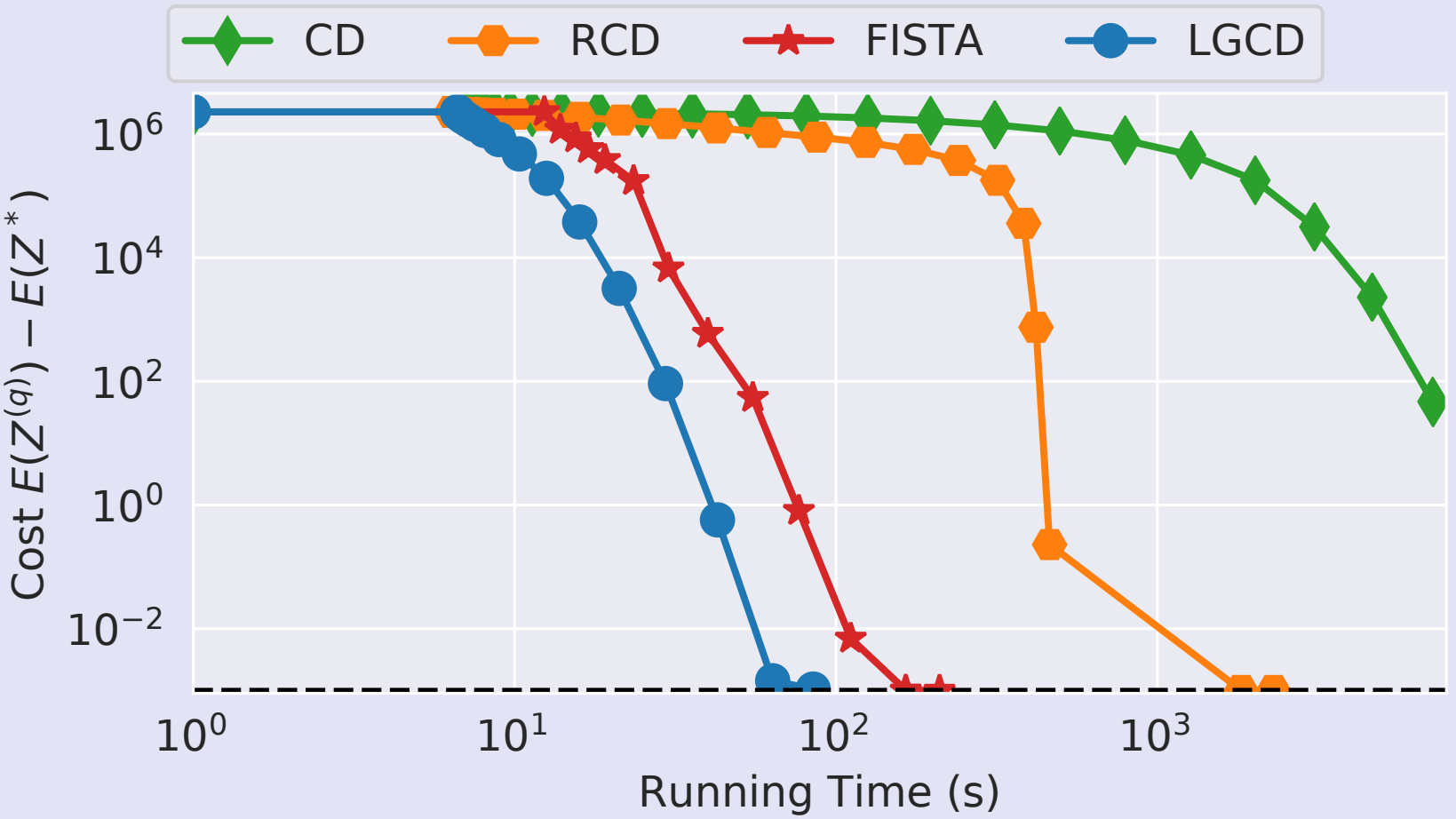
We set $K = 10$, $L = 150$, $\lambda = 0.1\lambda_{\max}$



Comparison with full batch method



Comparison with full batch method



Distributed algorithm for sparse coding: DICOD and DiCodile-Z

Weak dependence of the coordinate updates

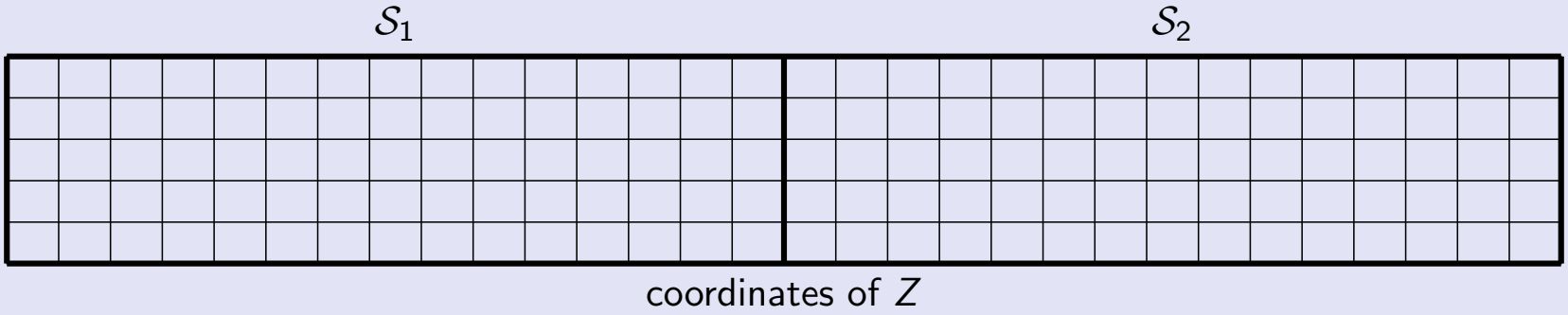
The update of the W coordinates $(k_w, \omega_w)_{w=1}^{W}$ with additive update $\Delta Z_{k_w}[\omega_w]$ changes the cost by:

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

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

Distributed Convolutional Coordinate Descent

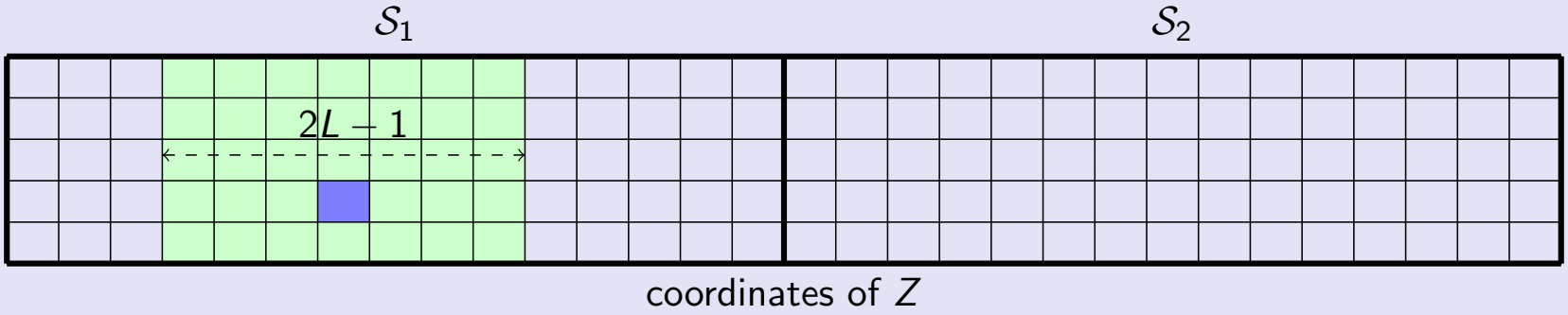
[Moreau et al. 2018, ICML]



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

Distributed Convolutional Coordinate Descent

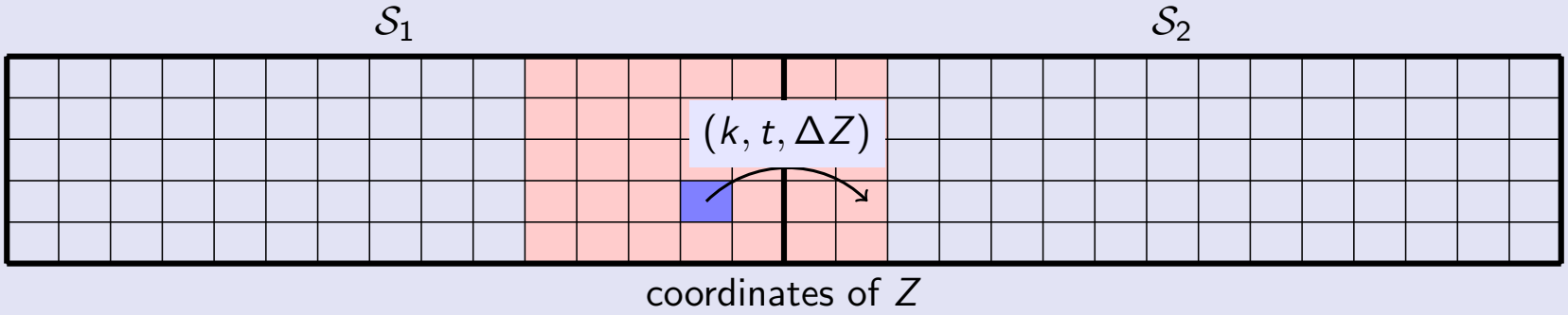
[Moreau et al. 2018, ICML]



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

Distributed Convolutional Coordinate Descent

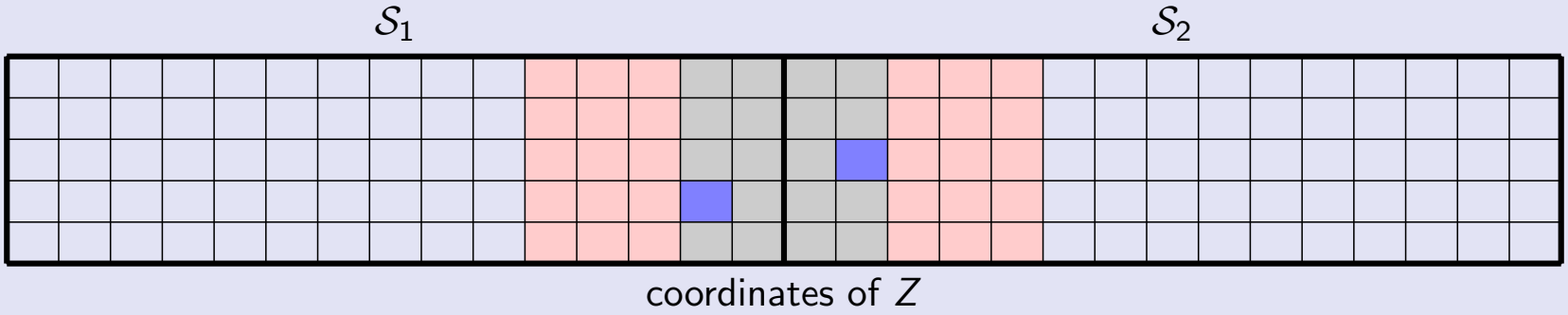
[Moreau et al. 2018, ICML]



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

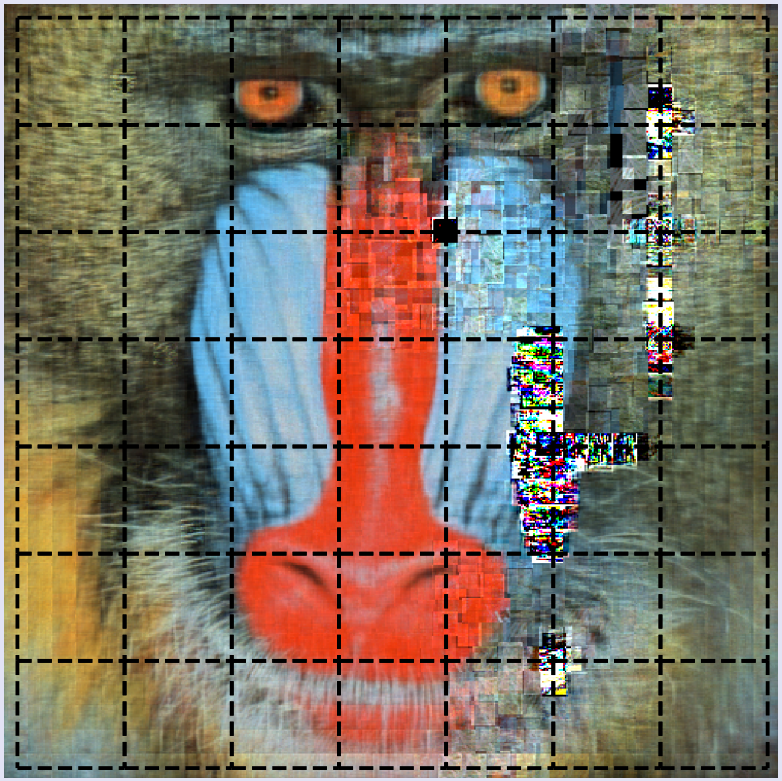
H2: No cores stop before all its coefficients are optimal.

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)

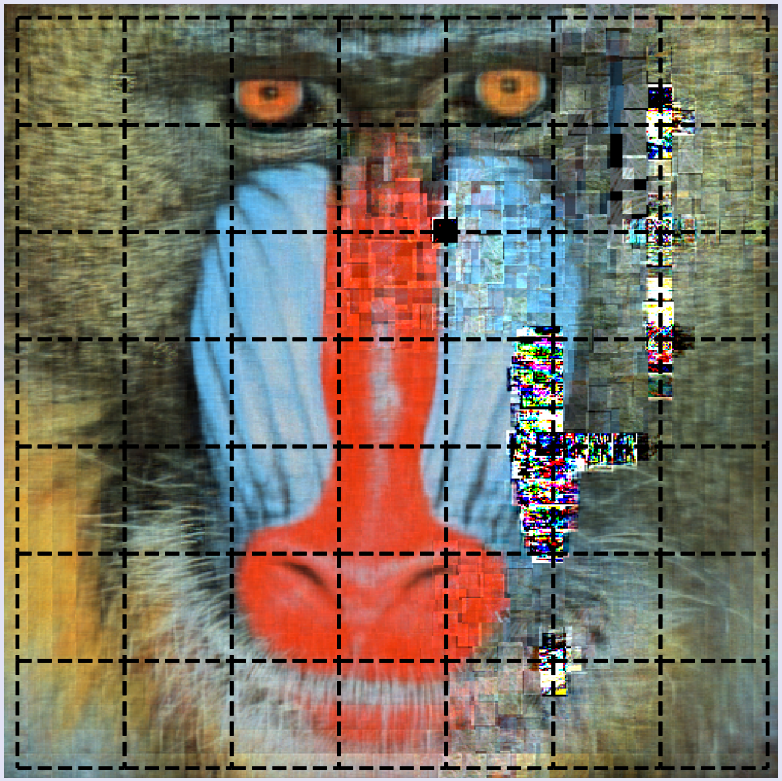
[Moreau and Gramfort 2019, preprint]



- DICOD does not work for higher dimensional signals.

Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)

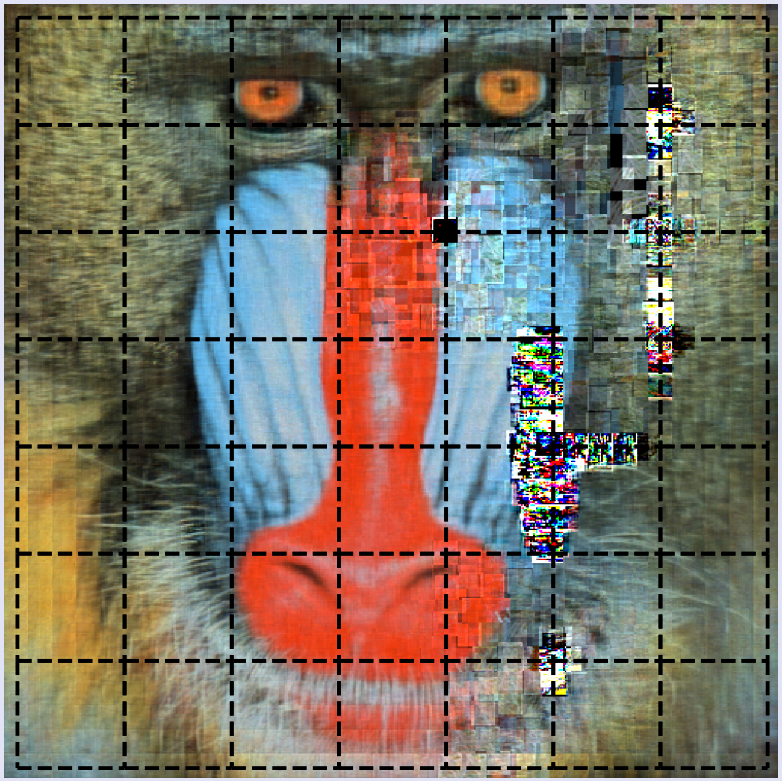
[Moreau and Gramfort 2019, preprint]



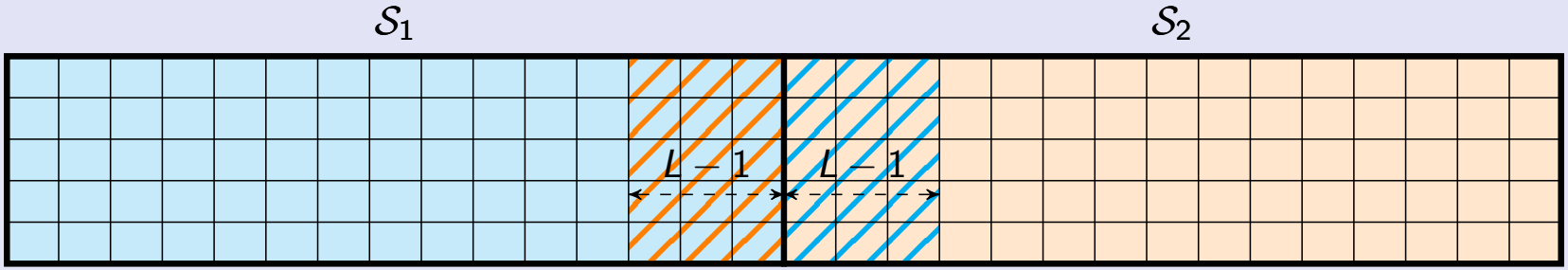
- DICOD does not work for higher dimensional signals.
- Extension require to control interference with more than 2 workers.

Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)

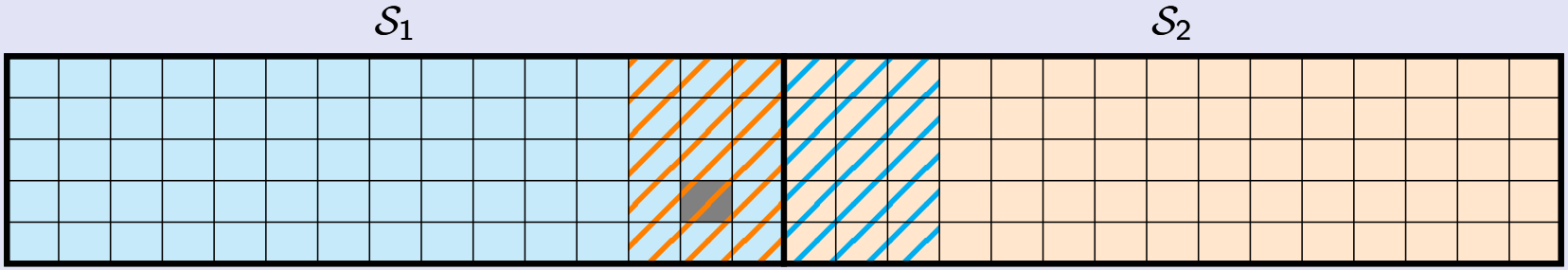
[Moreau and Gramfort 2019, preprint]



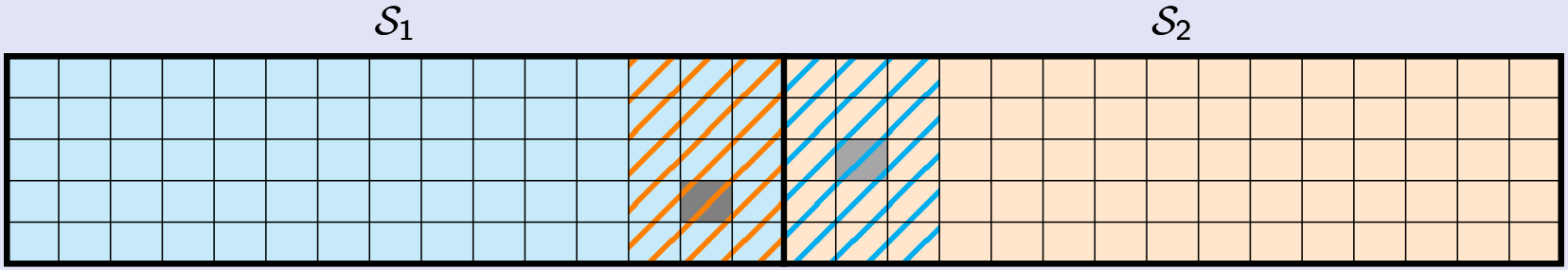
- DICOD does not work for higher dimensional signals.
- Extension require to control interference with more than 2 workers.
- Use asynchronous mechanism: Soft-lock.



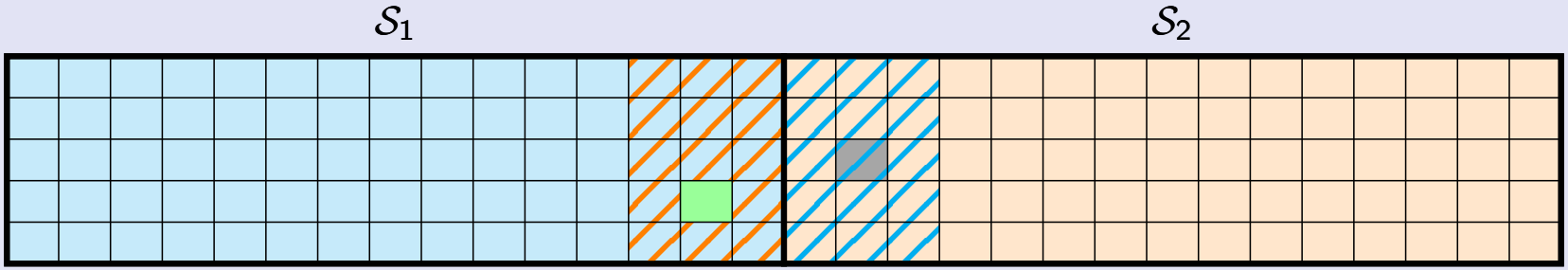
- Keep track of the value of the optimal update in an extended zone of size $L-1$.



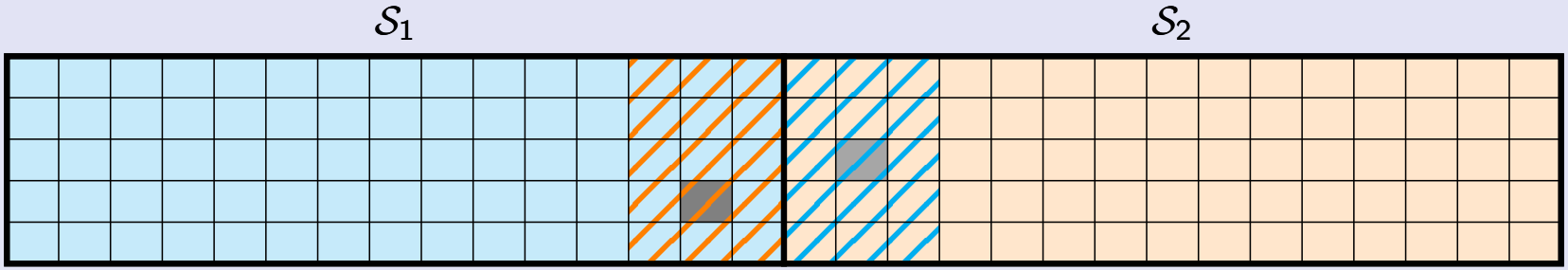
- Keep track of the value of the optimal update in an extended zone of size $L - 1$.
- Select an update candidate with LGCD.



- 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 udpate if it is larger than the other update.

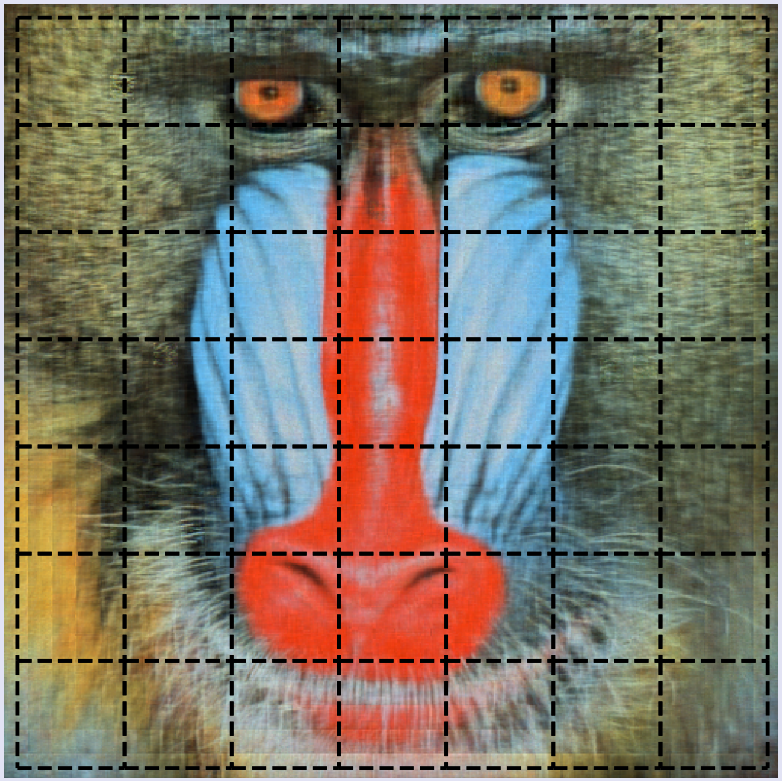


- Keep track of the value of the optimal update in an extended zone of size $L - 1$.
- Select an update candidate with LGCD.
- If it is in the interfering zone, compare the value of the update with the value potential updates in the other worker.
- Only perform the udpate if it is larger than the other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

$\Rightarrow$ Give an update order asynchronously.

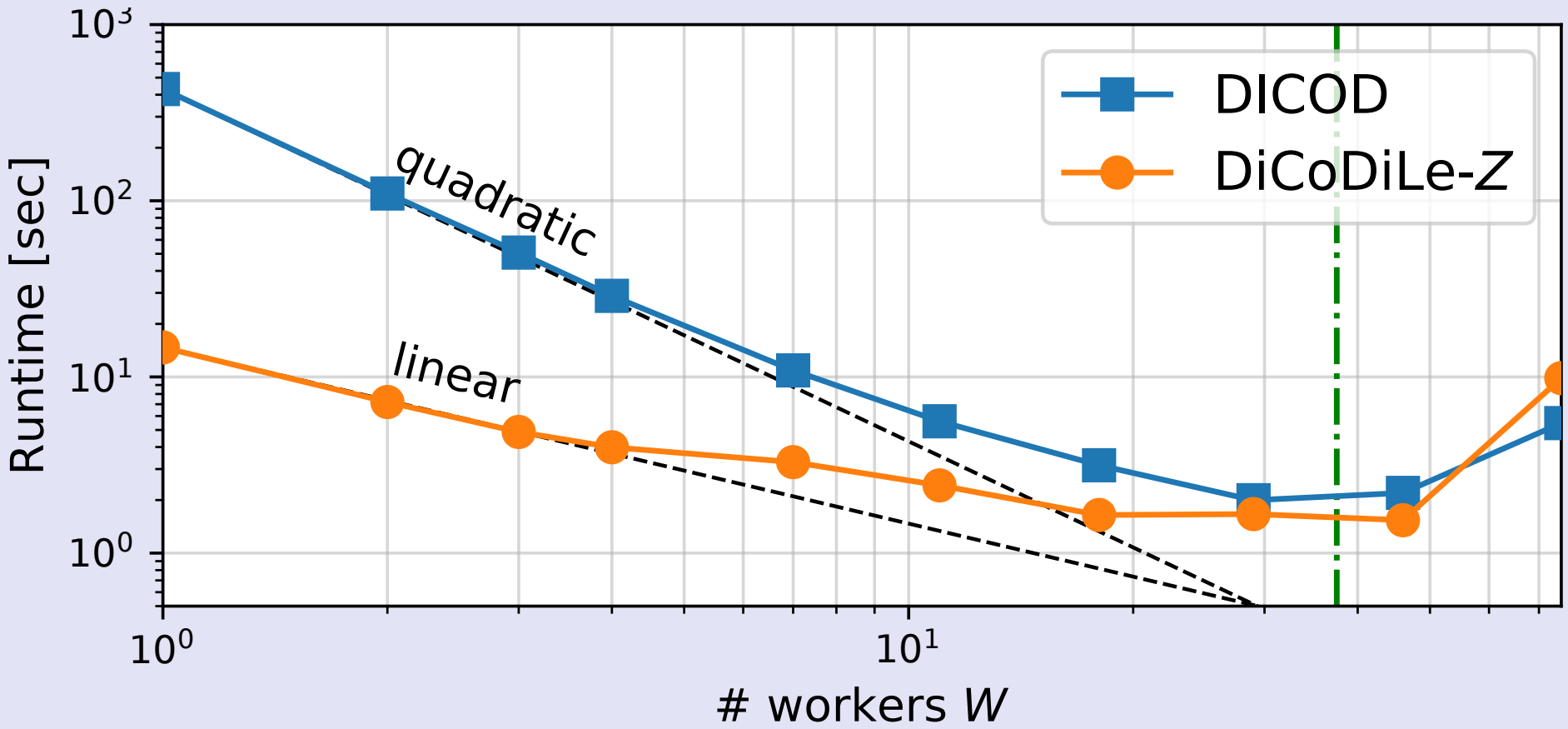
Distributed Convolutional Dictionary Learning (DiCoDiLe-Z)

[Moreau and Gramfort 2019, preprint]



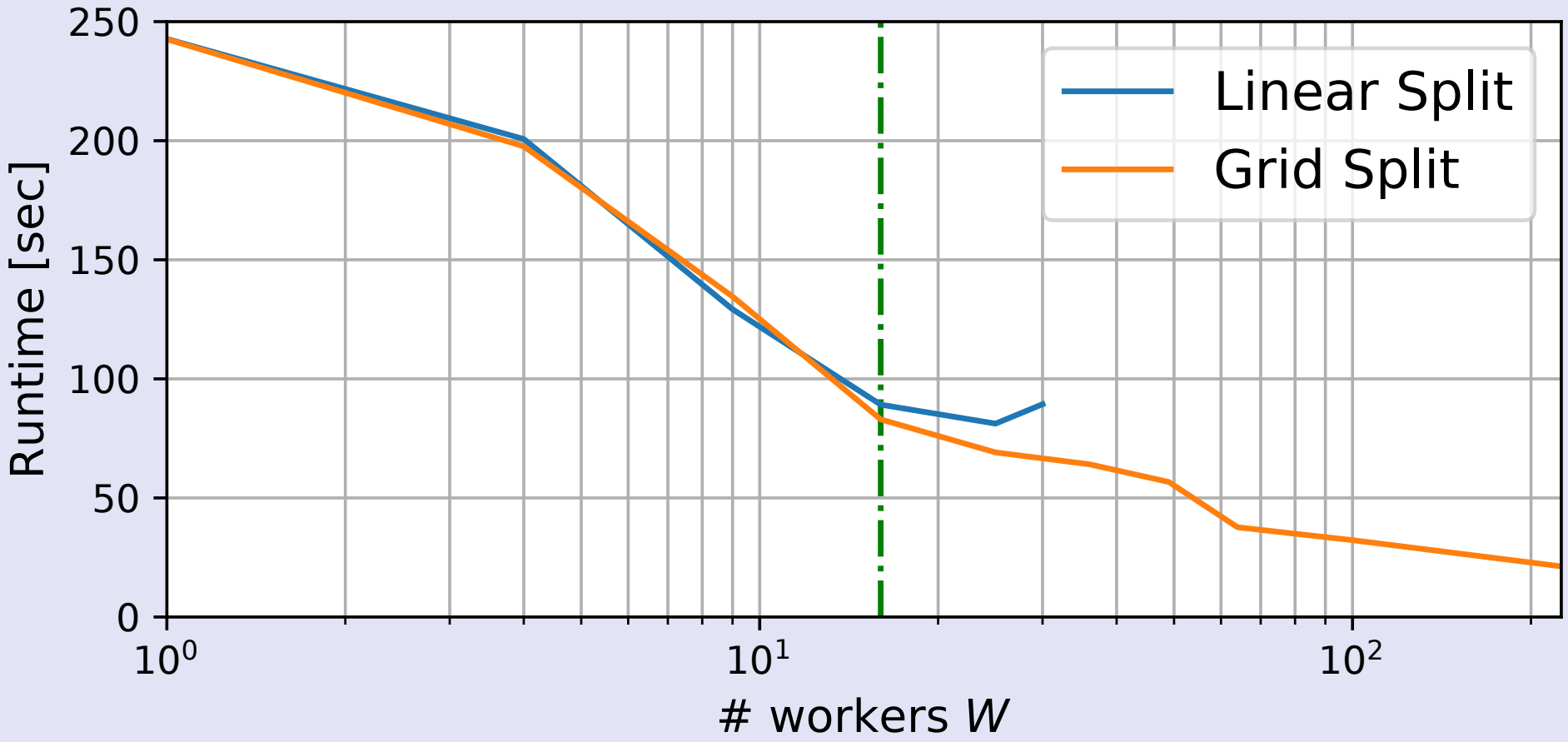
- DICOD does not work for higher dimensional signals.
- Extension require to control interference with more than 2 workers.
- Use asynchronous mechanism: Soft-lock.

Numerical speed-up



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

Scaling on a grid



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

Leveraging distributed computation for dictionary updates

D-step: solving for the atoms

The dictionary update is performed by minimizing

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

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

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

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

However, the computation of Ψ and Φ can be costly.

As each worker is handeling a disjoint sub-domain $\mathcal{S}_w$ of the signal domain Ω such that $\cup_{w=1}^{W} \mathcal{S}_w = \Omega$,

$$\Psi_{k,l}[\tau] = Z_k^{\curvearrowleft} * Z_l[\tau] = \sum_{\omega \in \Omega} Z_k[\omega] Z_l[\tau + \omega]$$

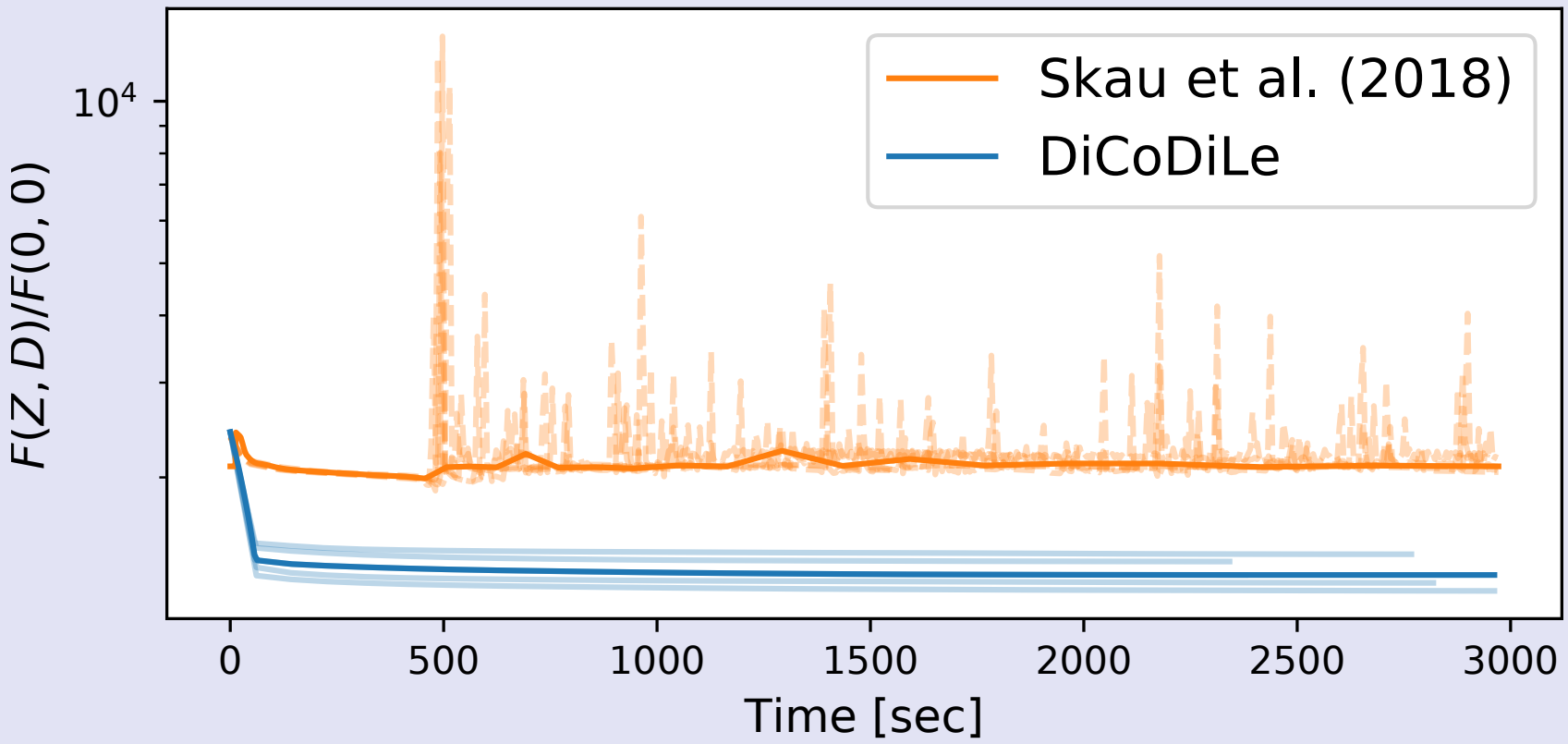
$$= \sum_{w=1}^{W} \sum_{\omega \in \mathcal{S}_w} Z_k[\omega] Z_l[\tau + \omega]$$

$$\Phi_k[\tau] = Z_k^{\curvearrowleft} * X[\tau] = \sum_{w=1}^{W} \sum_{\omega \in \mathcal{S}_w} Z_k[\omega] X[\tau + \omega]$$

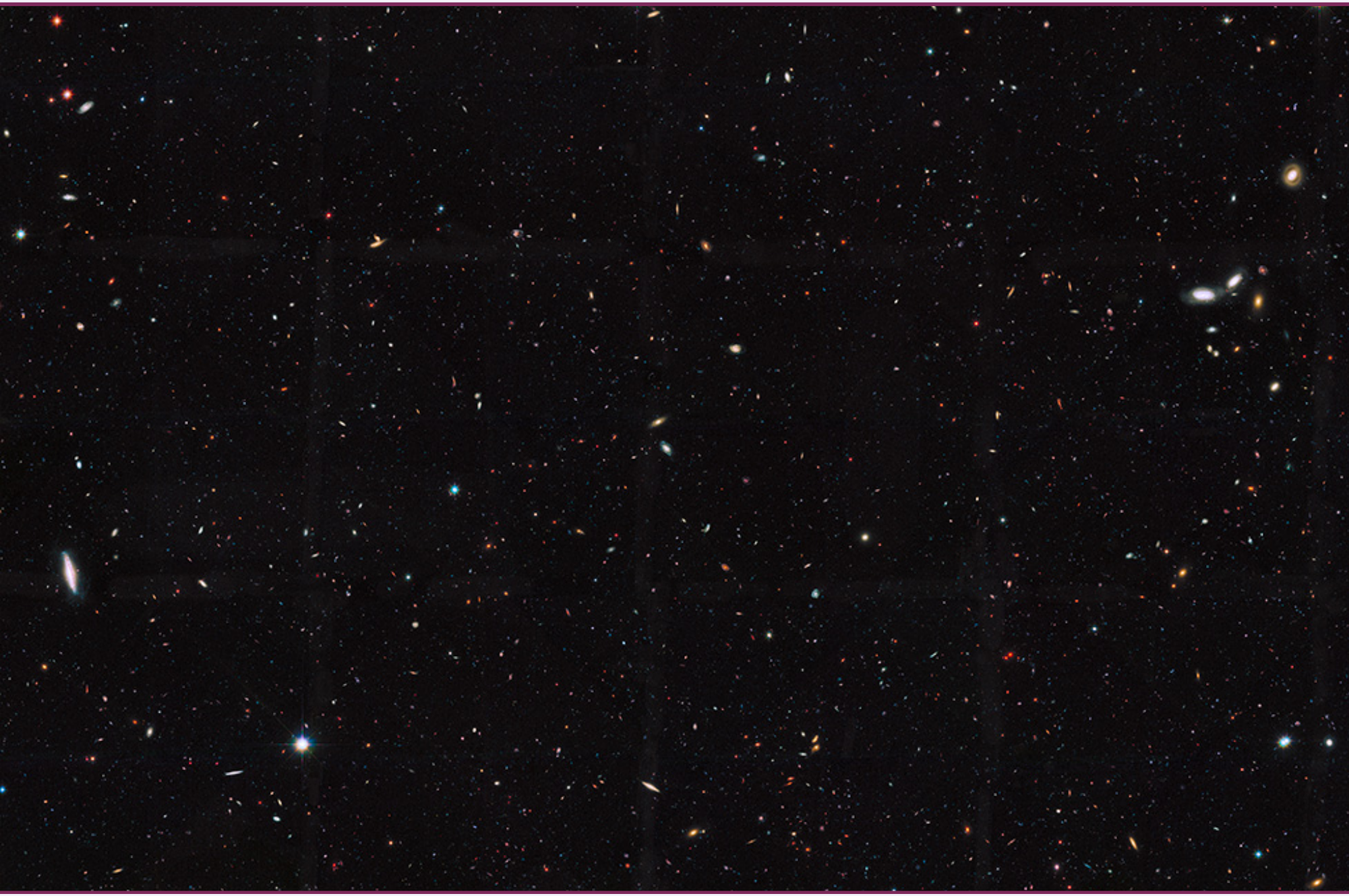
$\Rightarrow$ Computational cost is $\mathcal{O}\left(K(K+P)L(\frac{T}{W} + WL)\right)$

Numerical Experiments

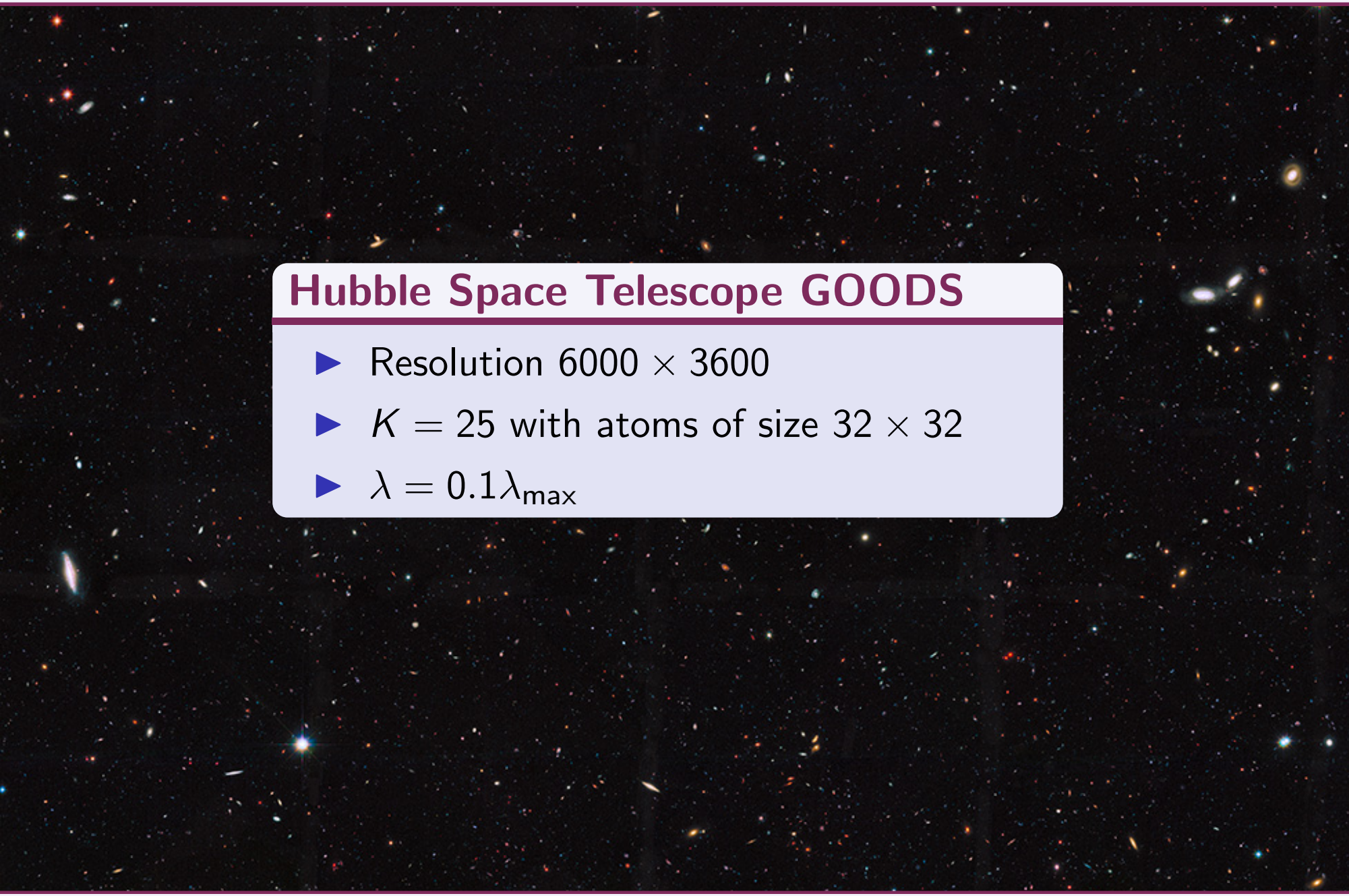
Comparition with distributed CDL



Images from Hubble Space Telescope



Images from Hubble Space Telescope



Hubble Space Telescope GOODS

- Resolution 6000×3600
- $K = 25$ with atoms of size 32×32
- $\lambda = 0.1\lambda_{\max}$

Images from Hubble Space Telescope

Conclusion

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.

Conclusion

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:

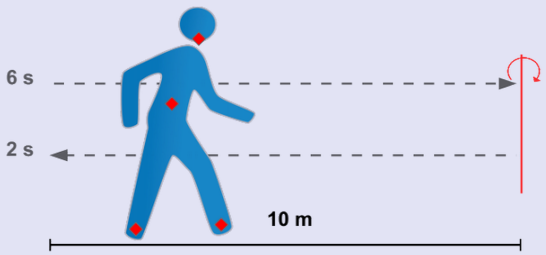
DiCoDiLe : `github.com/tommoral/dicodile`

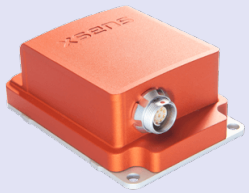
alphacsc : `alphacsc.github.io`

Slides are on my web page:

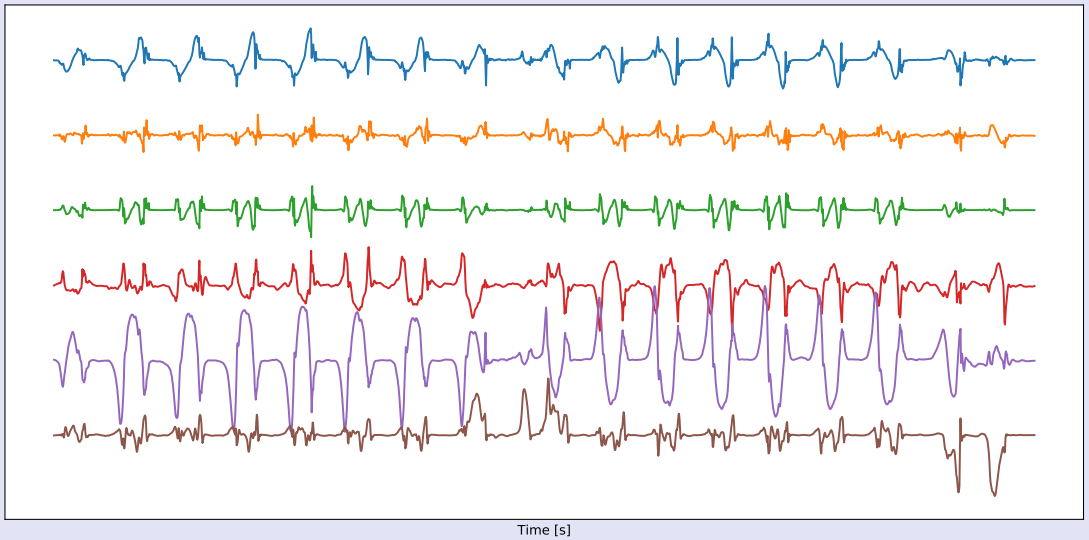
`tommoral.github.io` `@tomamoral`

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 ($W = 90$)

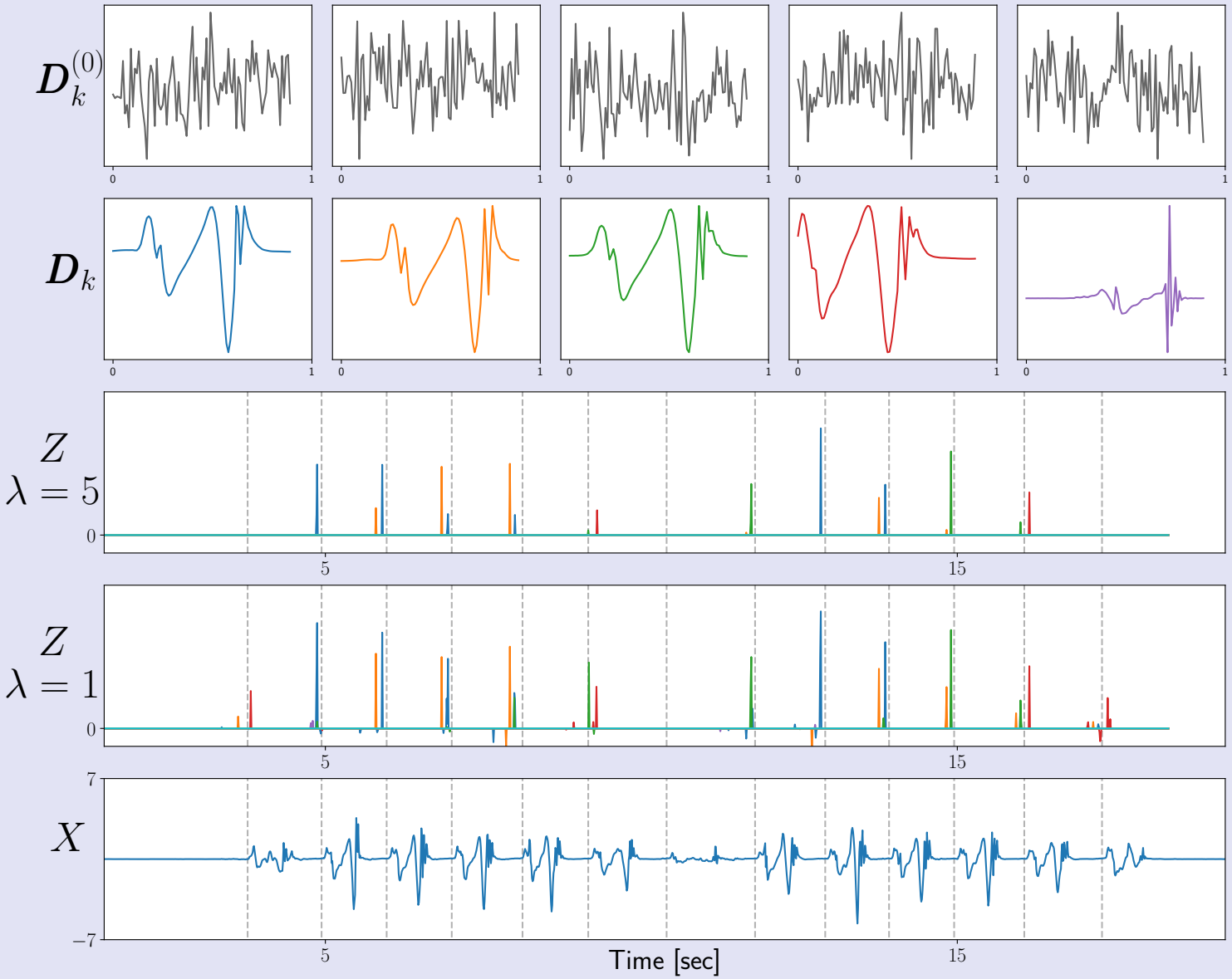
$$\boldsymbol{D}_k^{(0)} \sim \mathcal{N}(0, I_{90})$$

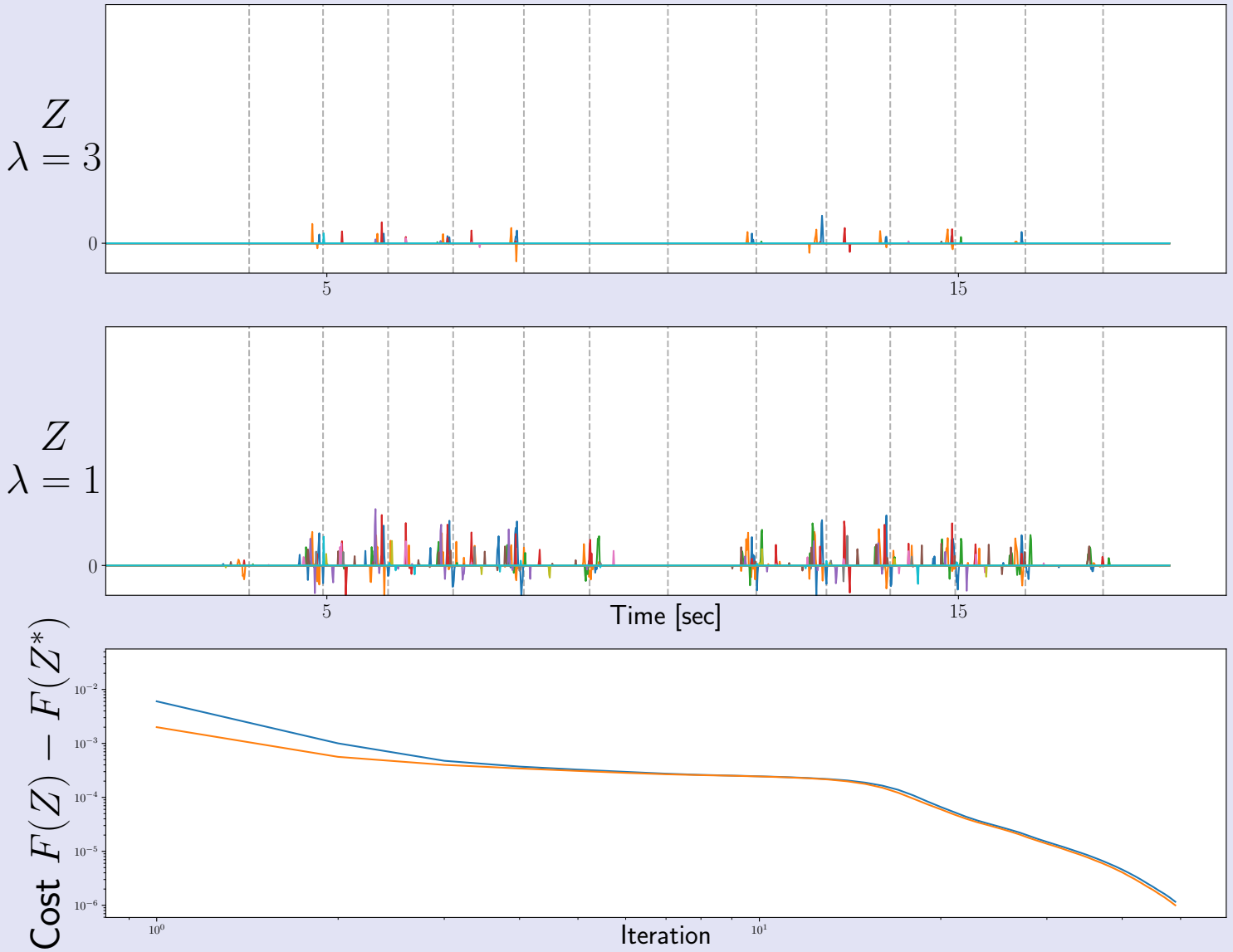
Use the Convolutional Dictionary Learning with DICOD to learn a dictionary $\boldsymbol{D}$ on a set of 50 recording of healthy subjects walking.

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

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)

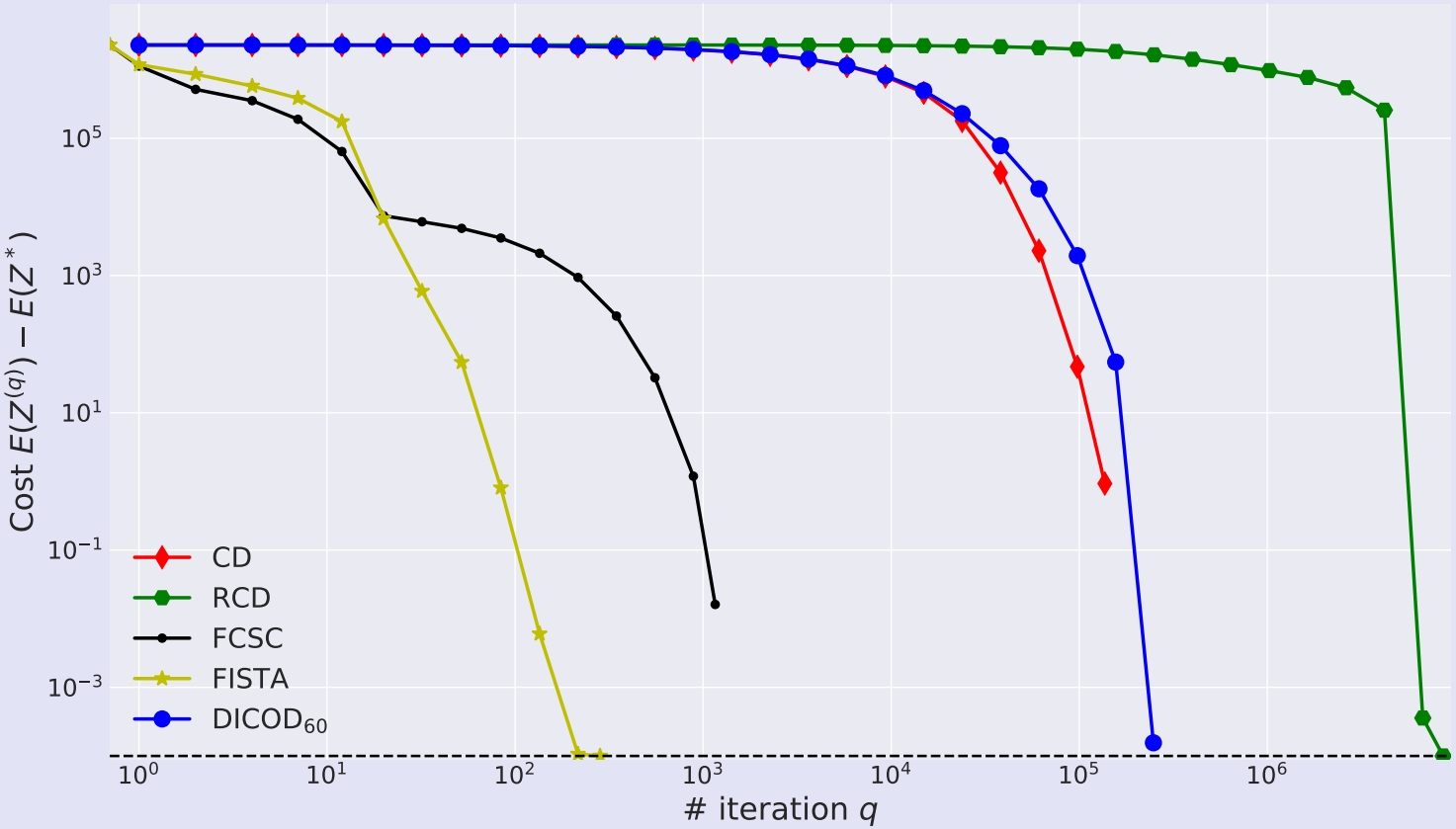
  [Bristow et al., 2013]

- Fast Ierative Soft-Thresholding Algorithm (FISTA)

  [Chalasani et al., 2013; Wohlberg, 2016]

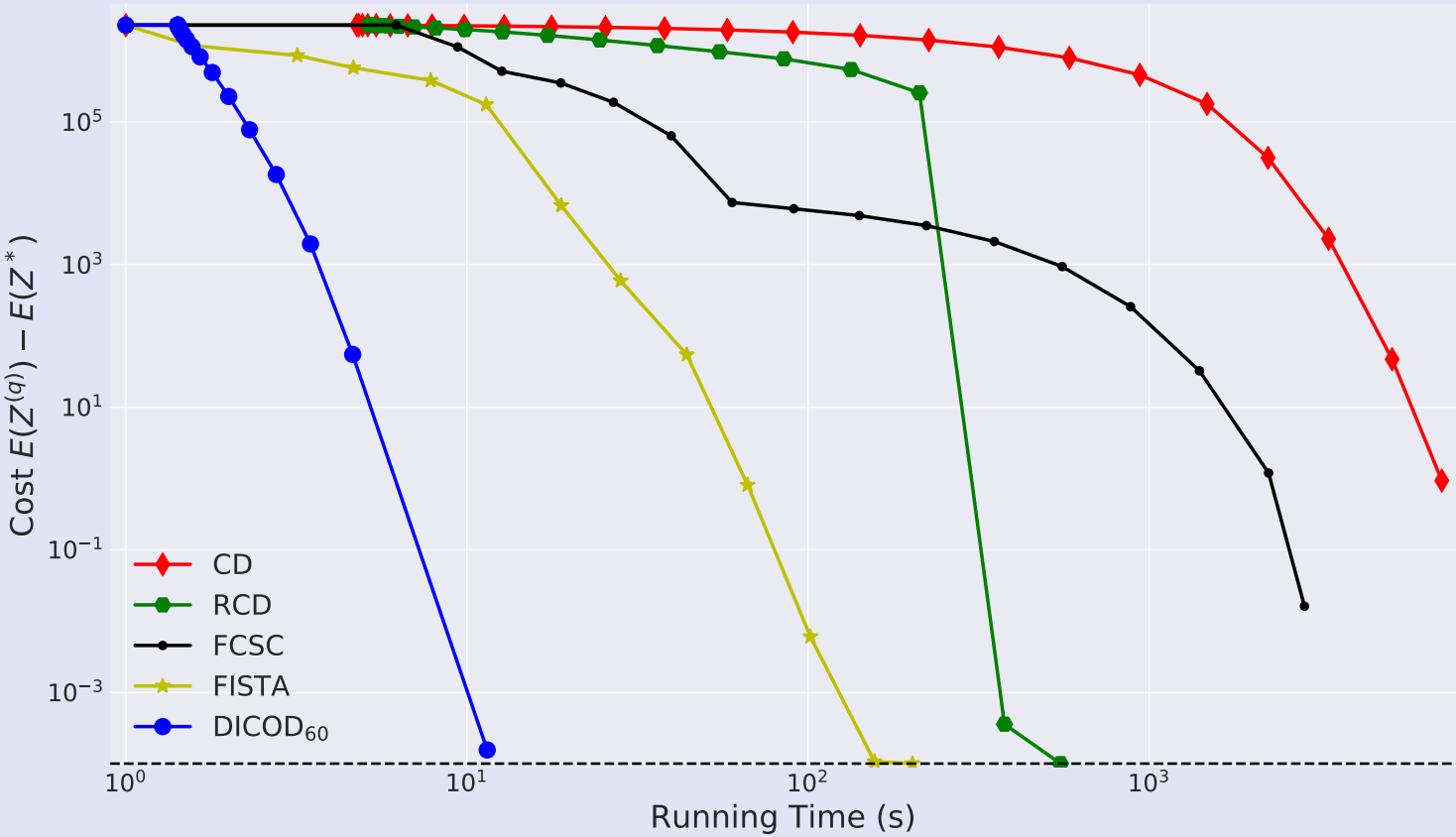
- DICOD with 60 cores

Numerical convergence



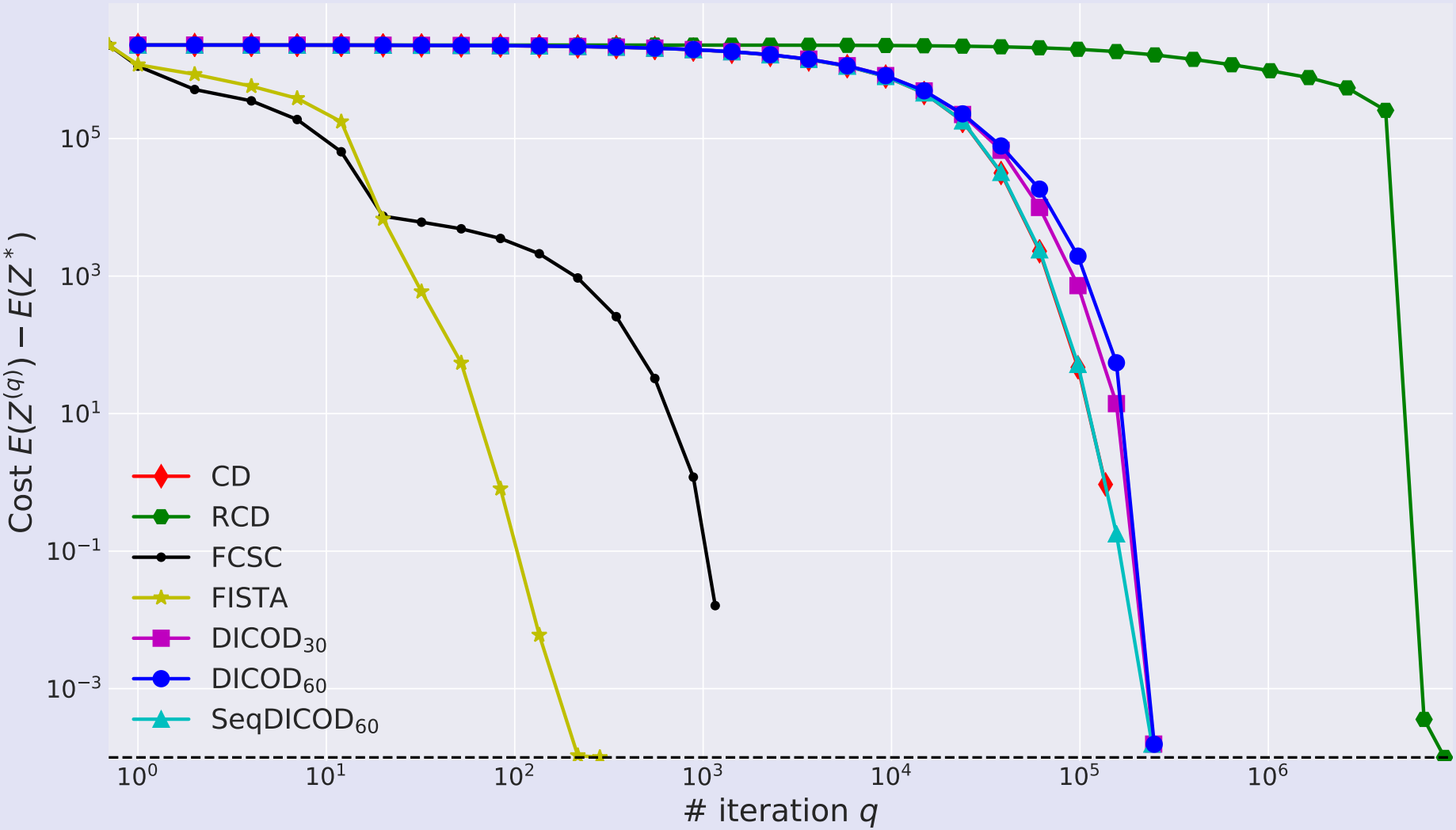
Cost as a function of the iterations

Numerical convergence



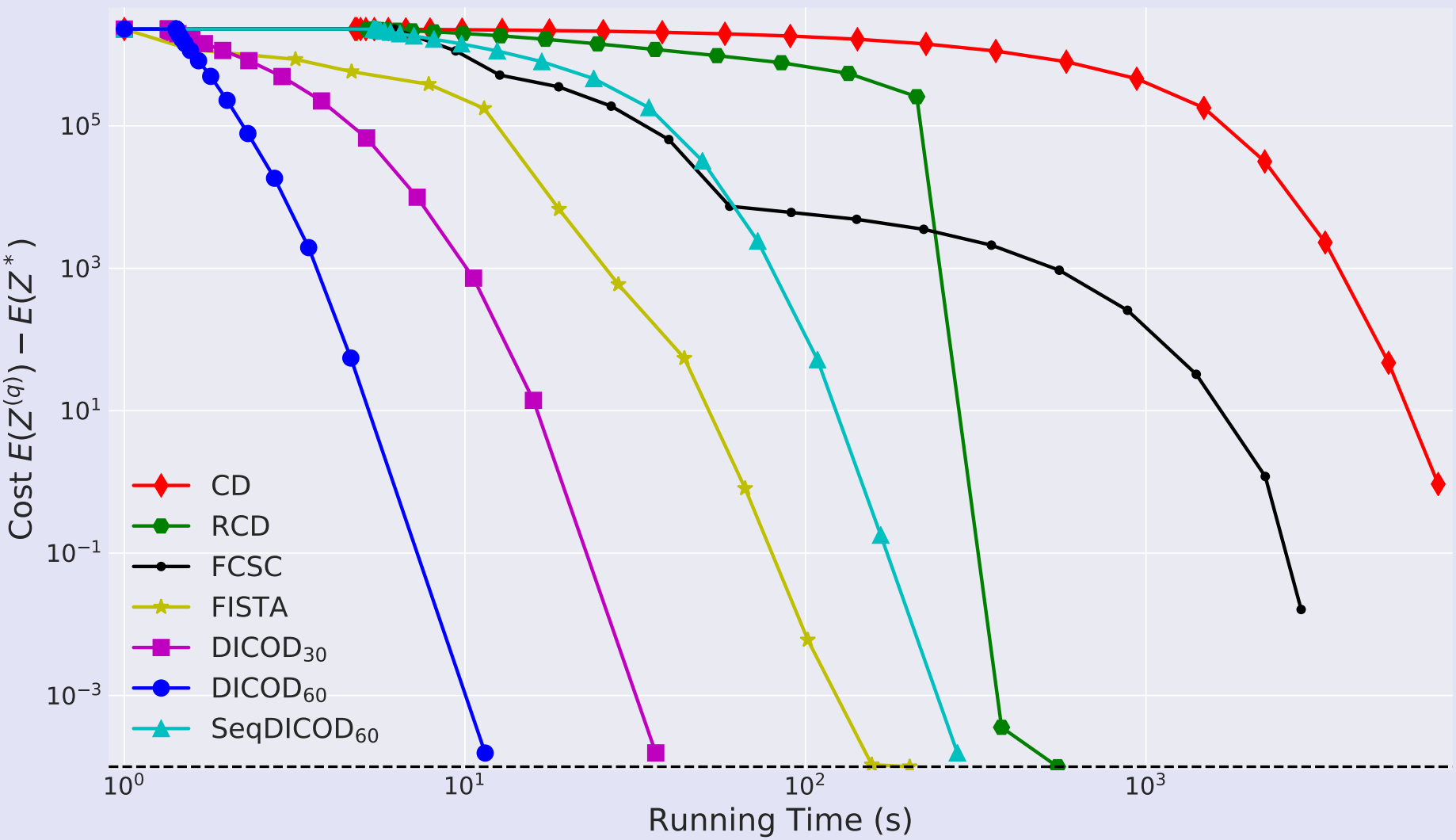
Cost as a function of the runtime

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 $\mathcal{O}\left(KW\right)$