Interpretable and scalable deep learning methods for imaging inverse problems

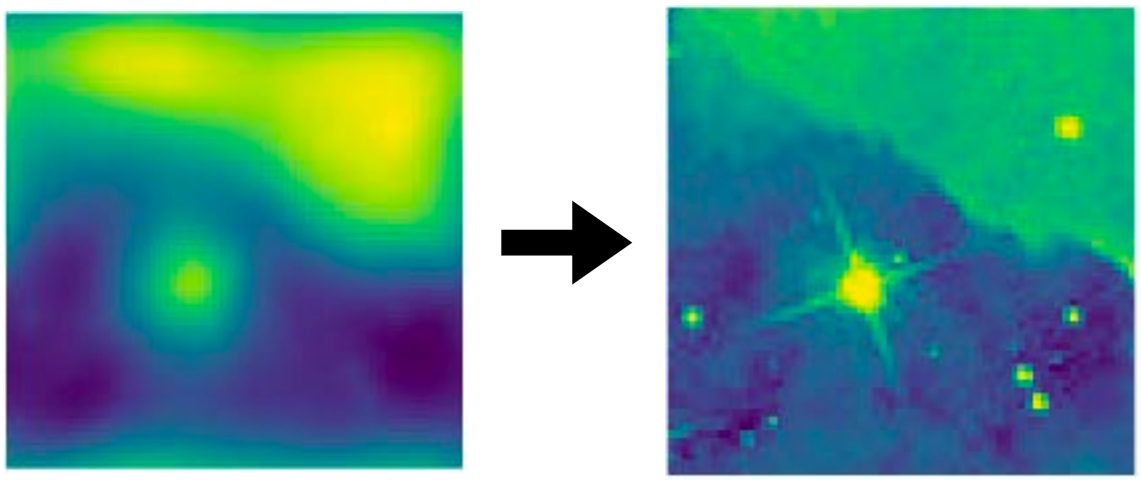
Christophe Kervazo

christophe.kervazo@telecom-paris.fr

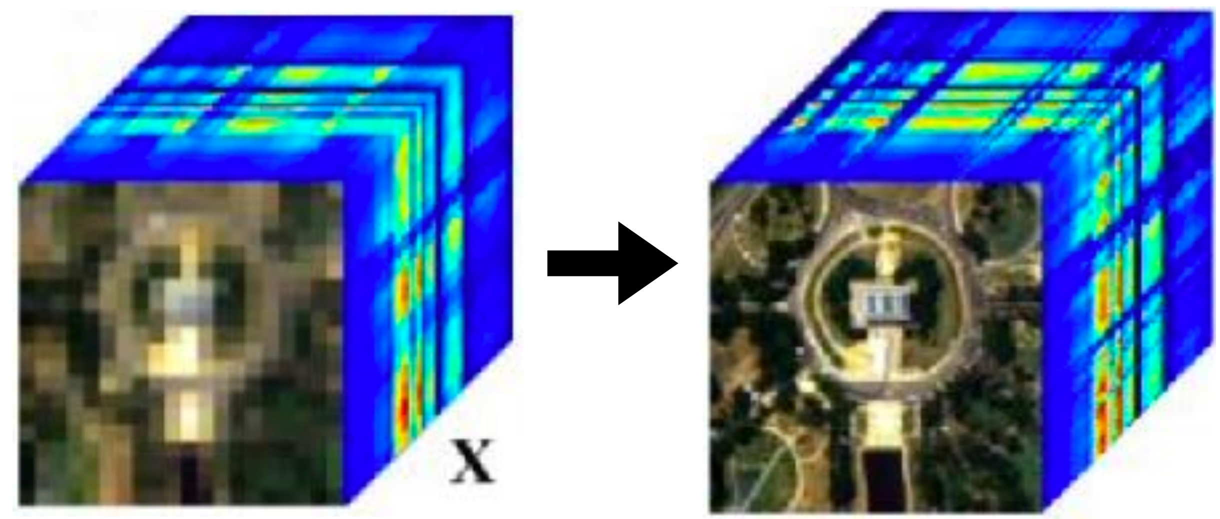
Joint works with J. Bobin, A. Chetoui, J. Cohen, M. Fahes, R. Hadjeres, F. Tupin, and others…

Inverse problems (1/2)

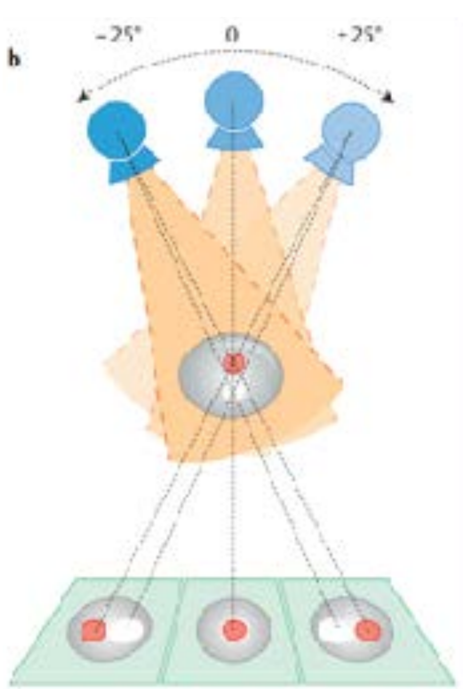
Deconvolution

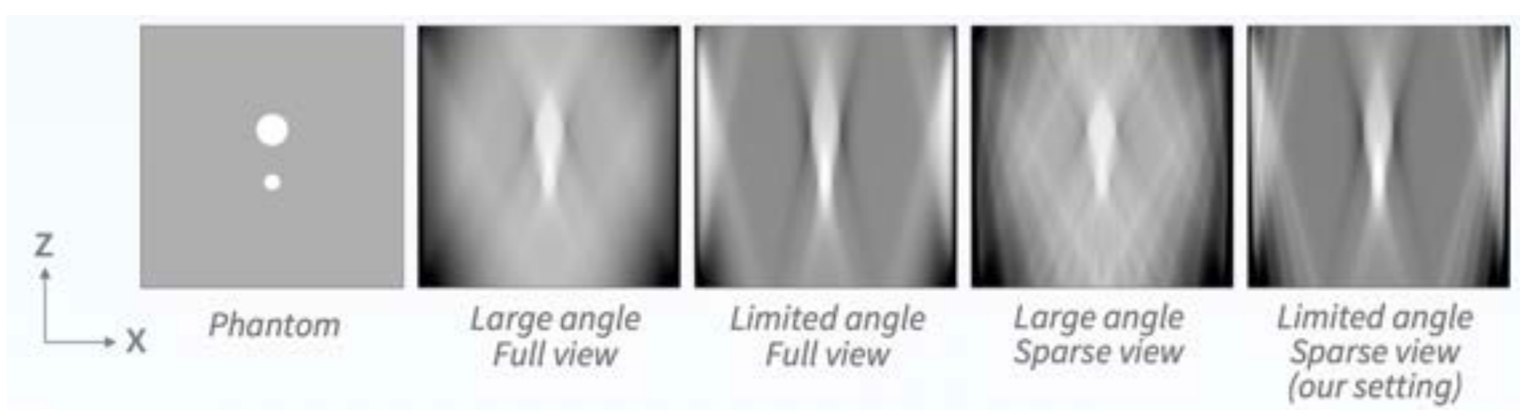


Super-resolution



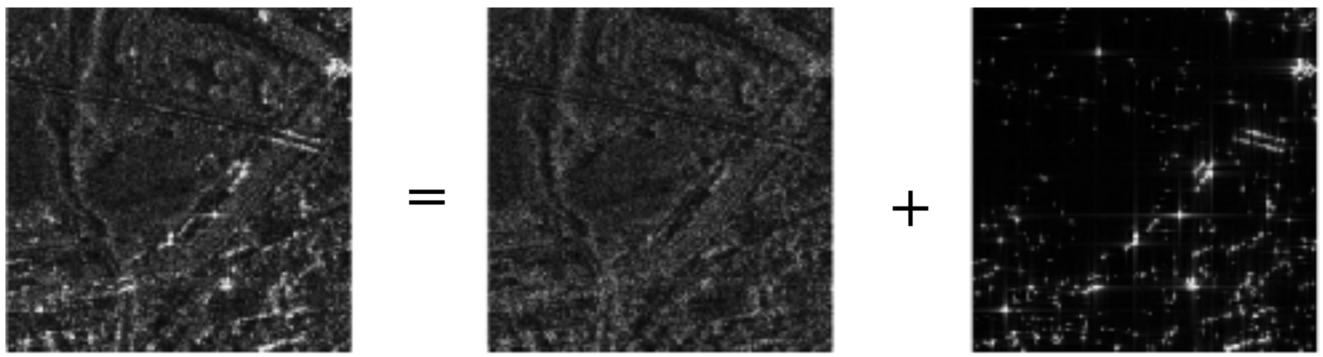
Sparse view tomosynthesis



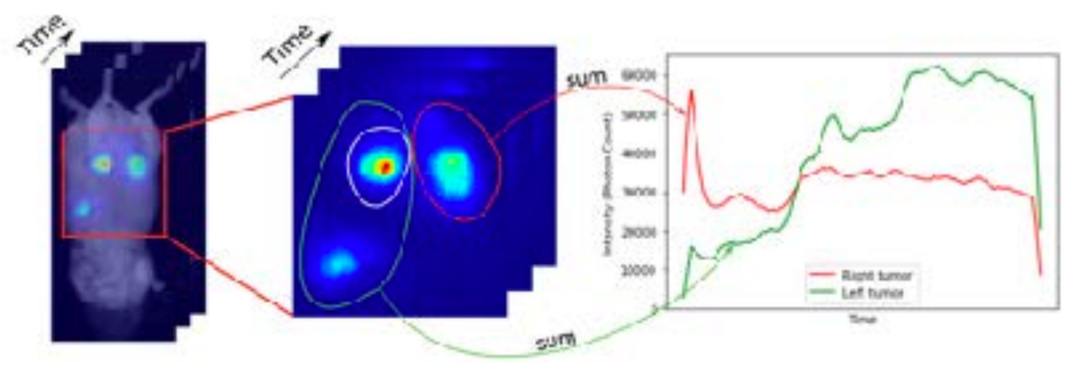


Inverse problems (2/2): source separation

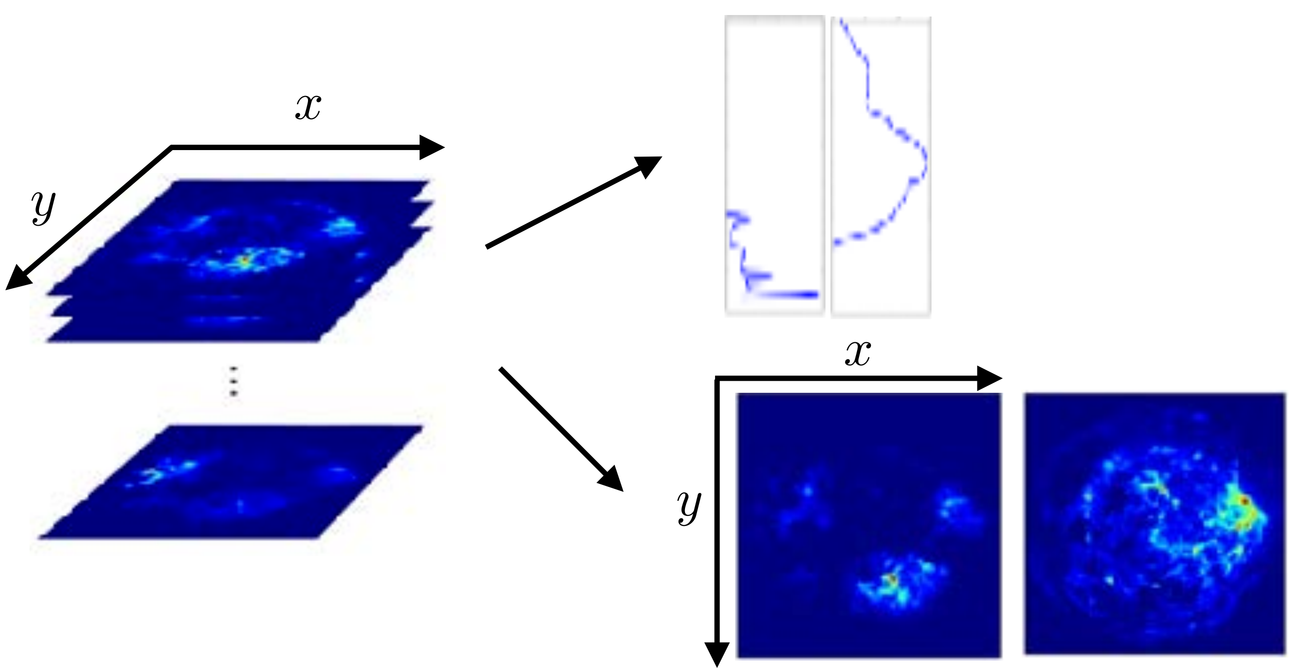
SAR target extraction



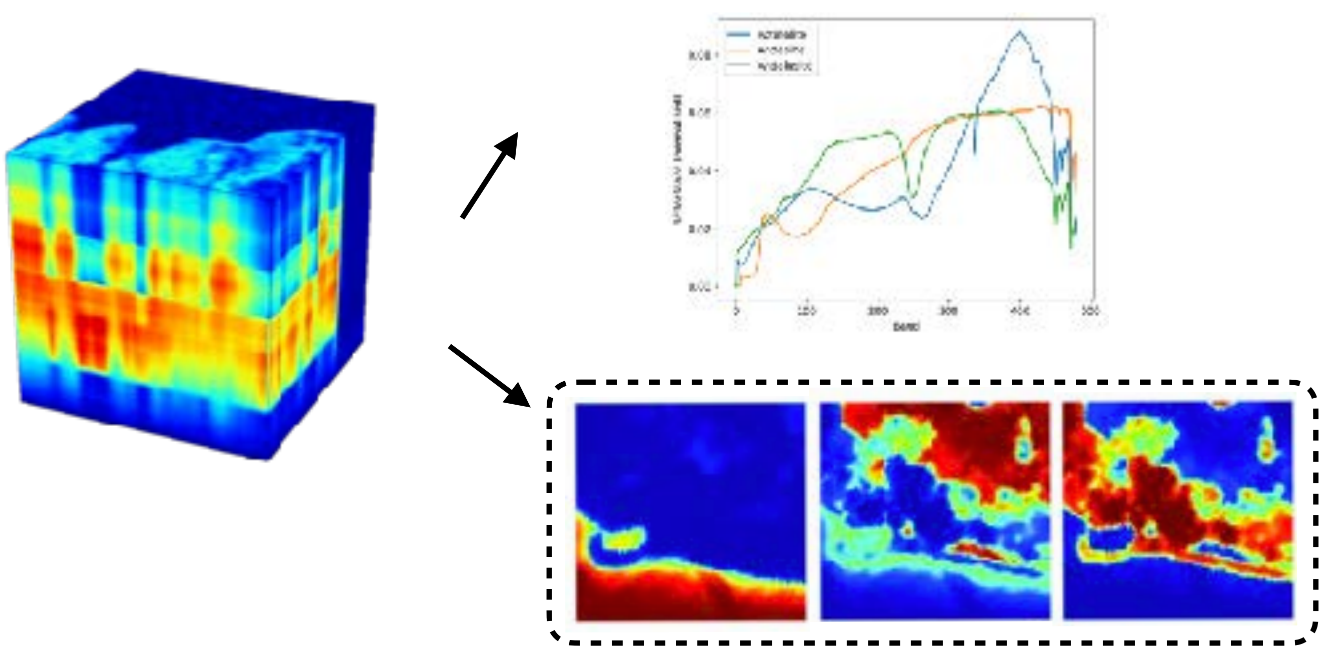
Bioluminescence imaging



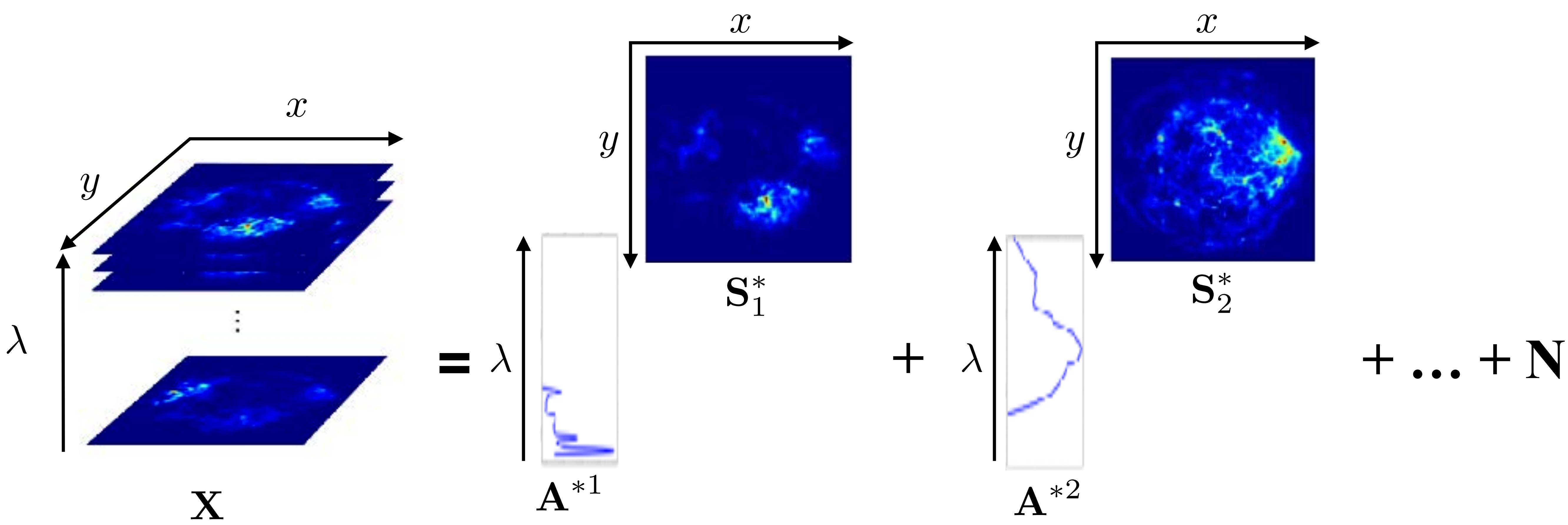
Astrophysics



Hyperspectral unmixing (earth monitoring)



BSS: Linear model [Comon10]



$$\boxed{\mathbf{X} = \mathbf{A}^*\mathbf{S}^* + \mathbf{N}}$$

X : m rows observations and t samples columns (m x t)
A* : mixing function (m x n)
$s_{ik}^* \in \mathbb{R}$ **:** abundance of the k*th* material in the i*th* pixel
N : noise and model imperfections (m x t)

Goal of BSS : estimate A* and S* from X (up to limited indeterminacies)

Classical methods: sparse source separation [Zibulevsky01]

$$\mathbf{X} = \mathbf{A}^*\mathbf{S}^* + \mathbf{N} = \mathbf{A}^*\mathbf{P}^{-1}\mathbf{P}\mathbf{S}^* + \mathbf{N} = \tilde{\mathbf{A}}\tilde{\mathbf{S}} + \mathbf{N} \quad \text{with } \tilde{\mathbf{A}} = \mathbf{A}^*\mathbf{P}^{-1} \text{ and } \tilde{\mathbf{S}} = \mathbf{P}\mathbf{S}^*$$

Infinite number of possible (non-physical) solutions

=> ill-posed problem requiring to introduce additional priors: ICA [Comon10], NMF [Gillis14], **sparsity**… + deep-learning extensions

Sparse source separation as an optimization problem [Zibulevsky01]:

$$\underset{\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{S}\in\mathbb{R}^{n\times t}}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|\mathbf{X}-\mathbf{A}\mathbf{S}\|_F^2}_{\textbf{Data-fidelity}} + \underbrace{\sum_{i=1}^{n}\lambda_i\|\mathbf{S}^i\|_1}_{\textbf{Sparsity}} + \underbrace{\iota_{\{i\in[1..n];\|\mathbf{A}^i\|_2^2\leq 1\}}(\mathbf{A})}_{\textbf{Oblique}}$$

$\lambda_1,\ldots,\lambda_n$: regularization hyper-parameters

Challenges:

- **Non-smooth** (use proximal operators [Parikh14])
- **Non-convex** (non-unique minima)
- **Difficult hyperparameter** / prior choice

Classical methods: optimization + handcrafted « automatic » parameter choice

$$\underset{\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{S}\in\mathbb{R}^{n\times t}}{\text{argmin}} \; \underbrace{\frac{1}{2}\|\mathbf{X}-\mathbf{A}\mathbf{S}\|_F^2}_{\textbf{Data-fidelity}} + \underbrace{\sum_{i=1}^{n}\lambda_i\|\mathbf{S}^i\|_1}_{\textbf{Sparsity}} + \iota_{\{i\in[1..n];\|\mathbf{A}^i\|_2^2\leq 1\}}(\mathbf{A})$$

$\lambda_1,\ldots,\lambda_n$: regularization hyper-parameters

PALM [Bolte14]

Initialize $\mathbf{A}$ and $\mathbf{S}$

While not converged over $\mathbf{A}$ and $\mathbf{S}$ do:

for $i = 1..n$:

$$\mathbf{S}^i \leftarrow \mathcal{S}_{\eta\lambda_i}\left(\mathbf{S} - \eta\left(\mathbf{A}\mathbf{S}^i - \mathbf{X}^i\right)\right)$$

$$\mathbf{A} \leftarrow \Pi_{\|.\|_2\leq 1}\left(\mathbf{A} - \xi\left(\mathbf{A}\mathbf{S}-\mathbf{X}\right)\right)$$

with:

- $\mathcal{S}$ the soft-thresholding
- $\Pi_{\|.\|_2\leq 1}(\,.\,)$ the projection on the unit ℓ_2 sphere
- η, ξ some gradient step-sizes

Limitations

- The hyperparameter choice is often handcrafted
- PALM takes several thousand iterations to converge => slow for large datasets

Deep learning alternative approach

- If we have access to a data base with examples of mixtures and the corresponding factors **A*** and **S***, can we obtain better separation results by introducing some learnt components within PALM?

- It corresponds to **algorithm unrolling**

  => enables to bypass the cumbersome hyper-parameter choice
  => much more computationally efficient than PALM
  => yield interpretable neural networks

- We first apply it in astrophysics, and then to earth monitoring

Algorithm unrolling: methodology

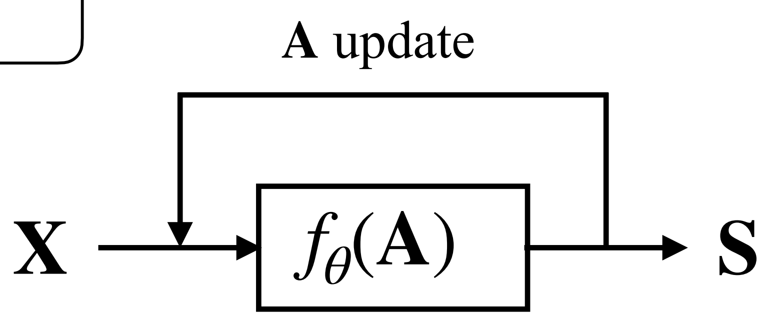
- **Going back to PALM**

while not converged do:

$$\mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{L_S}}\left(\mathbf{S} - \frac{1}{L_S}\mathbf{A}^T(\mathbf{A}*\mathbf{S} - \mathbf{X})\right)$$

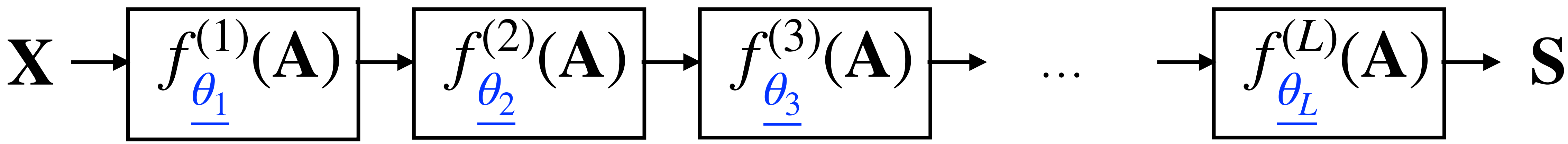
Update $\mathbf{A}$

It can be sketched as:



with θ the algorithm parameters (gradient step sizes…)

- **Algorithm unrolling** truncates this scheme to rewrite it in the form of a neural network with a small number of layers (iterations):



- The algorithms parameters $\theta_{(k)}$ becomes ***trainable*** on a ***training set*** (*i.e.* they becomes the weights of the neural network)
- The number of iterations L is usually much smaller than in the original algorithm

Being more specific: LISTA algorithm

- Further possible to learn a reparametrization of the update [Gregor, Lecun 10]

$$\mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{L}}\left(\mathbf{S} - \frac{1}{L}\mathbf{A}^T(\mathbf{AS}-\mathbf{X})\right) \Leftrightarrow \mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{L}}\left(\underbrace{\left(\mathbf{I} - \frac{1}{L}\mathbf{A}^T\mathbf{A}\right)}_{\mathbf{W}_1}\mathbf{S} + \underbrace{\frac{1}{L}\mathbf{A}^T}_{\mathbf{W}_2}\mathbf{X}\right)$$

learning some parts of the update $\longrightarrow$

$$\mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{L_S}}\left(\mathbf{W}_1\mathbf{S} + \mathbf{W}_2\mathbf{X}\right)$$

$$\Theta = \{\lambda/L_S, \mathbf{W}_1, \mathbf{W}_2\}$$

LISTA update

☺ Much faster than merely learning the hyperparameters

☹ No use of the theoretical link between $\mathbf{W}_1$ and $\mathbf{W}_2$

- In LISTA-CP [chen18], the theoretical coupling of $\mathbf{W}_1$ and $\mathbf{W}_2$, leading to the update:

$$\mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{L_S}}\left(\mathbf{W}_1\mathbf{S} + \mathbf{W}_2\mathbf{X}\right) \longrightarrow \boxed{\mathbf{S} \leftarrow \mathcal{S}_{\frac{\lambda}{L_S}}\left(\mathbf{S} - \mathbf{W}^T(\mathbf{AS}-\mathbf{X})\right)}$$

(LISTA update) (LISTA-CP update)

☺ LISTA-CP takes into account the relationship between $\mathbf{W}_1$ and $\mathbf{W}_2$ => better empirical results

☹ LISTA-CP requires an estimate of $\mathbf{A}$

Unrolling PALM

- The way to unroll PALM was chosen according to the previous remarks and experimental trials :

Learned-PALM (LPALM) [Fahes22]

initialize A and **S** with a very generic initialization

for k from 1 to L **do** :

$$\mathbf{S} \leftarrow \mathcal{S}_{\underline{\gamma}}\left(\mathbf{S} - \underline{\mathbf{W}}^T(\mathbf{AS} - \mathbf{X})\right) \qquad \text{(LISTA-CP update)}$$

$$\mathbf{A} \leftarrow \Pi_{\|.\| \leq 1}\left(\mathbf{A} + \frac{1}{\underline{L_A}}(\mathbf{X} - \mathbf{AS})\mathbf{S}^T\right) \qquad \text{(learning step-size)}$$

end for

return A, S

- The loss function is chosen as

$$NMSE(\mathbf{A}, \mathbf{A}^*) + NMSE(\mathbf{S}, \mathbf{S}^*)$$

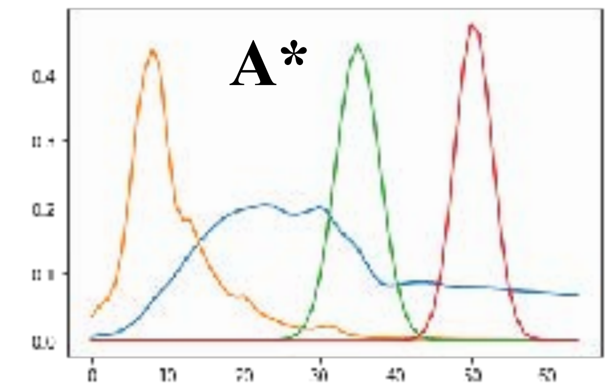
over the training set.

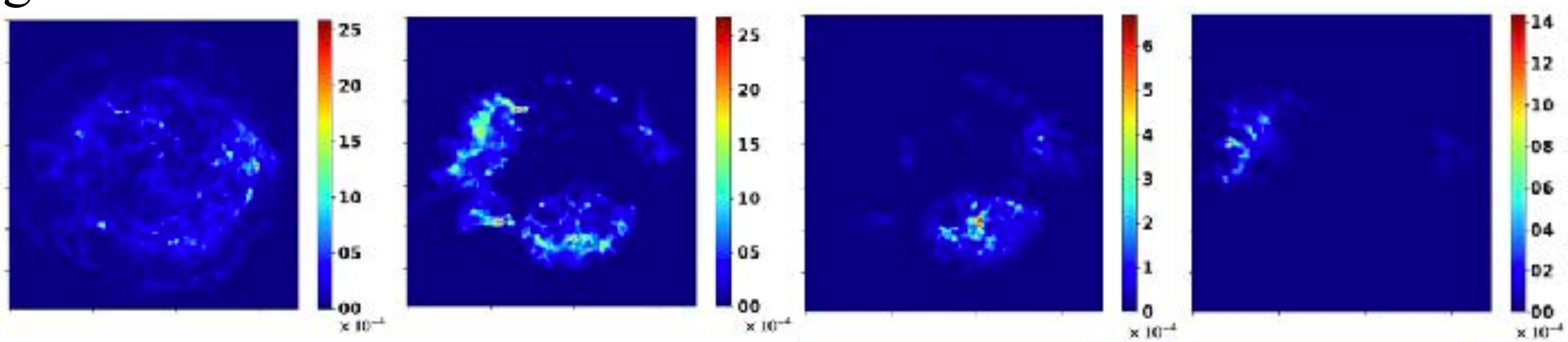
Numerical experiments: datasets

LPALM is tested on astrophysical simulations of the Cassiopea A supernovae remnant as observed by the X-ray telescope Chandra. There are $n = 4$ emissions: synchrotron, thermal and 2 red-shifted irons

- **Test set:**

  $\mathbf{X} = \mathbf{A}^*\mathbf{S}^* + \mathbf{N}$ with:
  - $\mathbf{A}^*$ coming from realistic simulations
  - $\mathbf{S}^*$ being real supernovae maps (reshaped into vectors)
  - $\mathbf{N}$ generated Gaussian noise

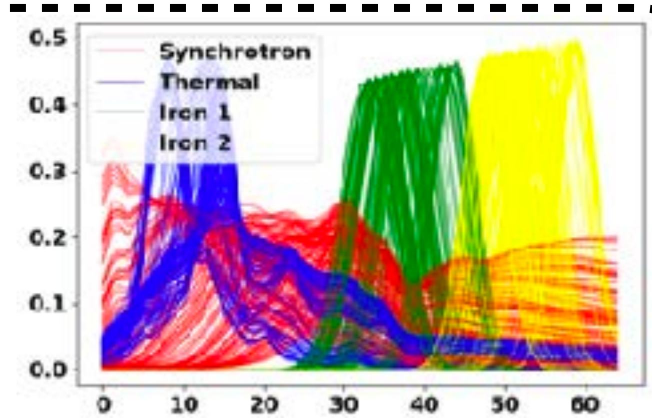




S*

- **Train set:**

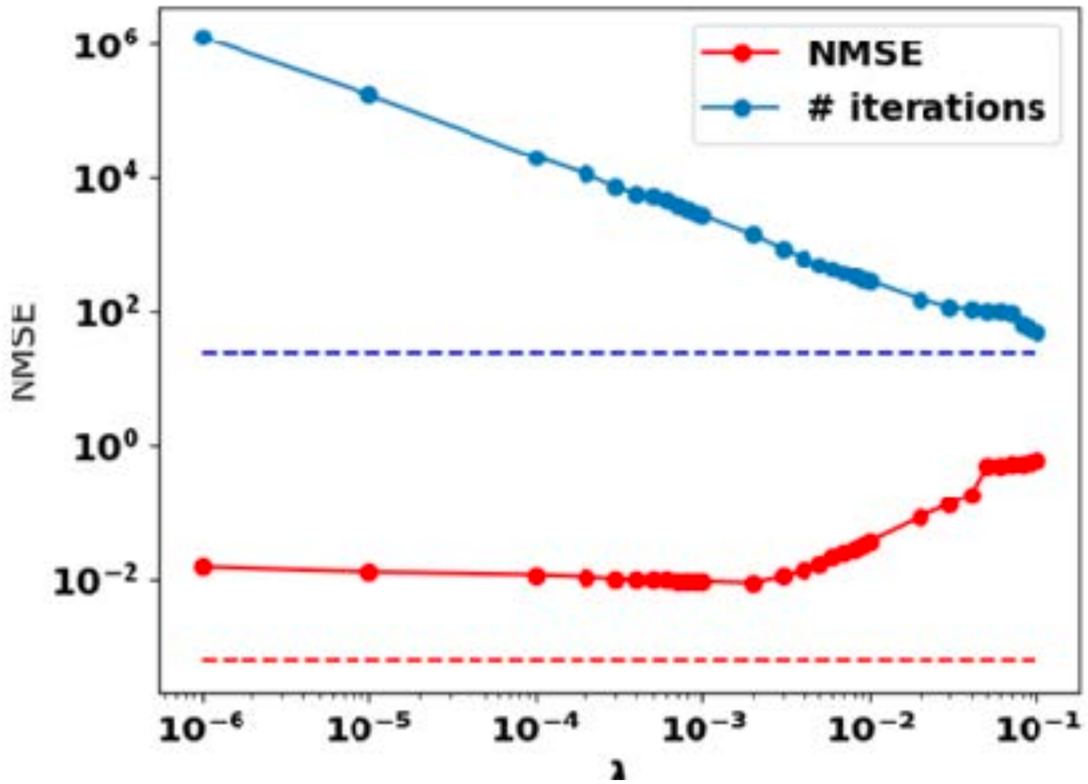
  750 samples of ${}^i\mathbf{X} = {}^i\mathbf{A}^*{}^i\mathbf{S}^* + {}^i\mathbf{N}$ with:
  - ${}^i\mathbf{A}^*$ coming from realistic simulations (quite different from the test set)
  - ${}^i\mathbf{S}^*$ generated using a Bernouilli Generalized-Gaussian distribution
  - ${}^i\mathbf{N}$ generated Gaussian noise



${}^i\mathbf{A}^*$ for $i = 1..50$. Each column is represented with a different color

Numerical results: comparison with PALM

LPALM is compared with PALM, by optimizing PALM parameters over the train set:



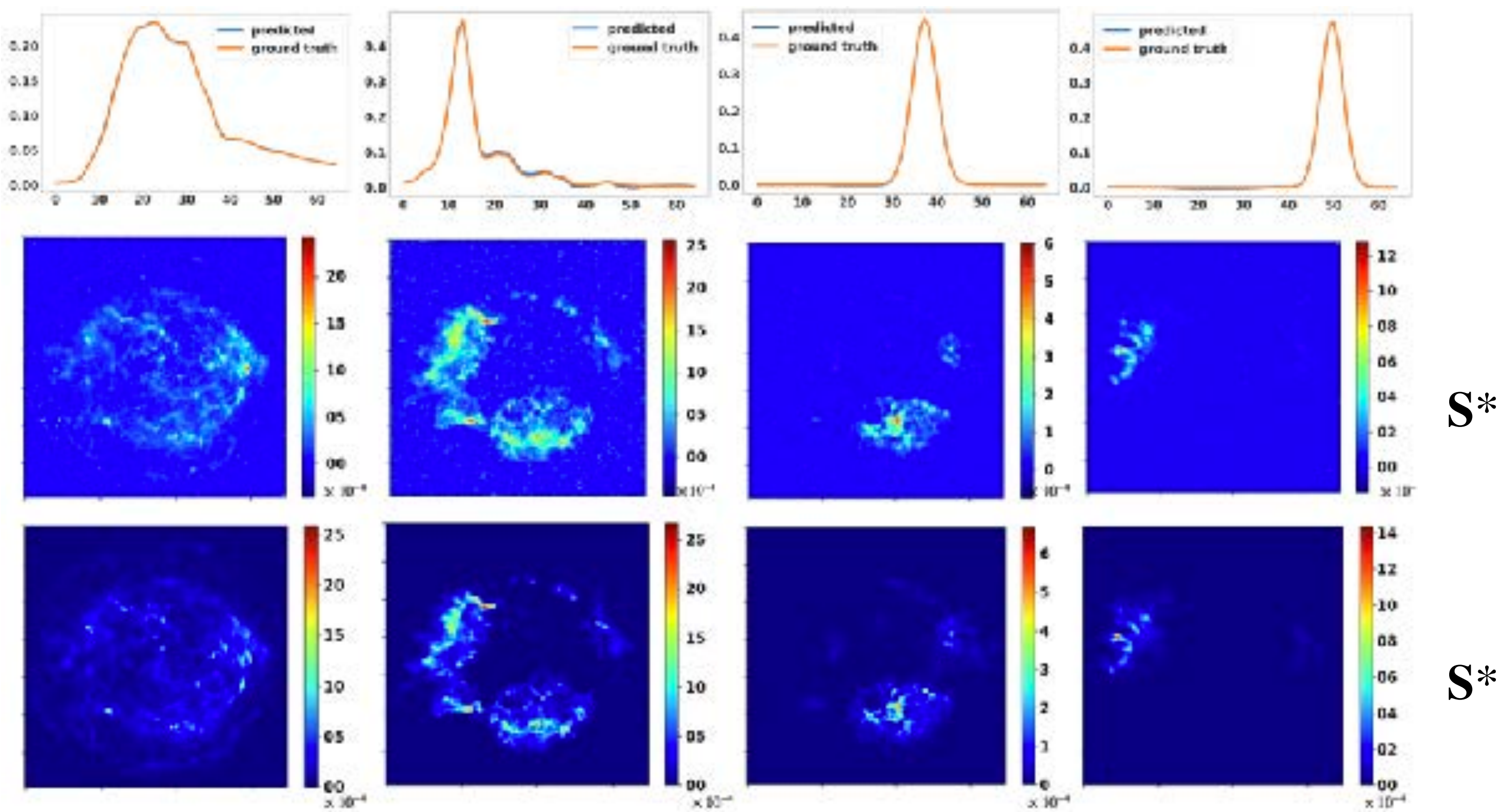
Blue, plain and dashed lines: median number of iterations for PALM and LPALM, respectively

Red, plain and dashed lines: median NMSE for PALM and LPALM, respectively

LPALM largely outperforms PALM, both:

- in terms of separation quality
- in terms of number of iterations

Numerical results: comparison with other unrolled methods



Algorithm	MNNBU	SNMF-net	DNMF	LISTA	**LPALM**
NMSE(A_{pred}, A^*)	0.0962	0.2700	0.0466	0.0067	**0.0002**
Generalization (model evaluation)	0.4785	0.3053	0.5469	0.0088	**0.0009**
Model retraining	0.3181	0.3073	0.0922	-	-

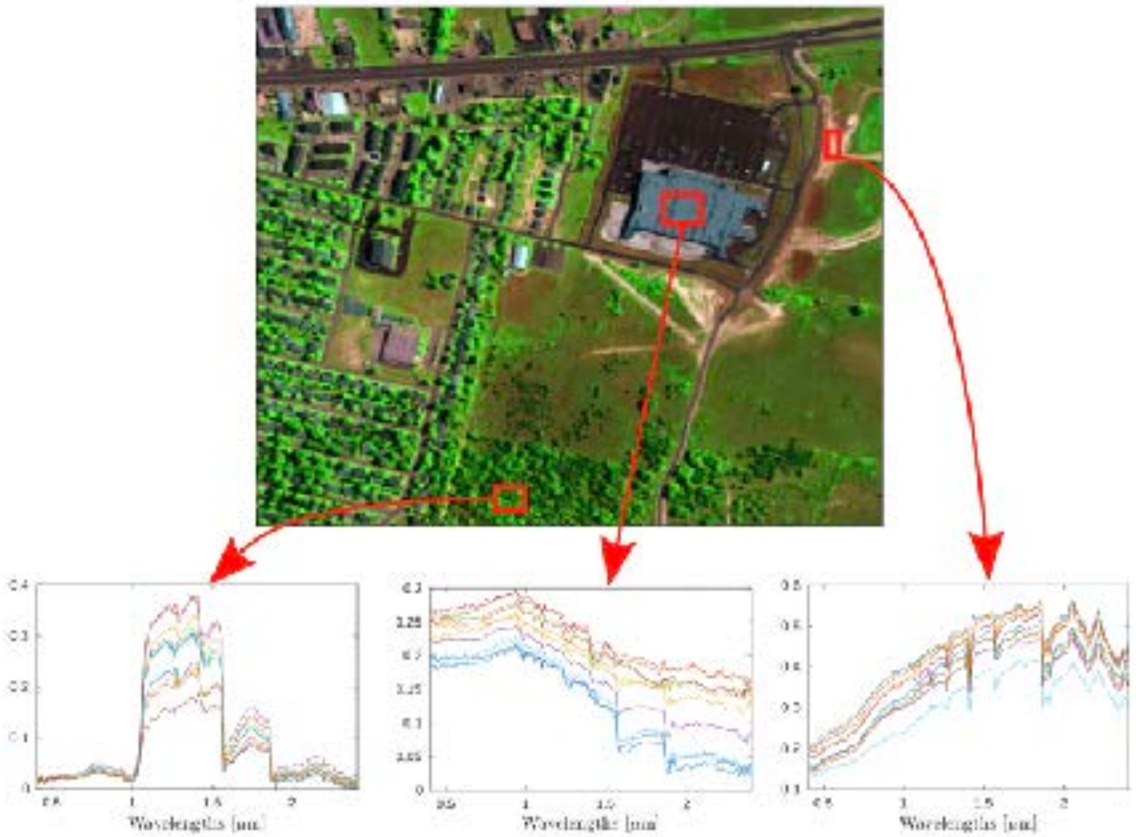
LPALM largely outperforms its competitors:

- LISTA lacks of flexibility to handle varying ${}^{i}\mathbf{A}^*$ matrices
- DNMF suffers from a training using the reconstruction error only: $\|{}^{i}\mathbf{X} - {}^{i}\mathbf{A}{}^{i}\mathbf{S}\|_2^2$, which is well-known to lead to spurious solutions

LPALM for earth observation: a self-supervised approach

- A major limitation of LPALM for earth observation is that it requires some datasets with ground-truths $\mathbf{A}^*$ and $\mathbf{S}^*$ for training, which is difficult to obtain in earth observation

- In addition, in earth observation, $\mathbf{A}^*$ is non-stationnary over the image (so-called spectral variabilities)

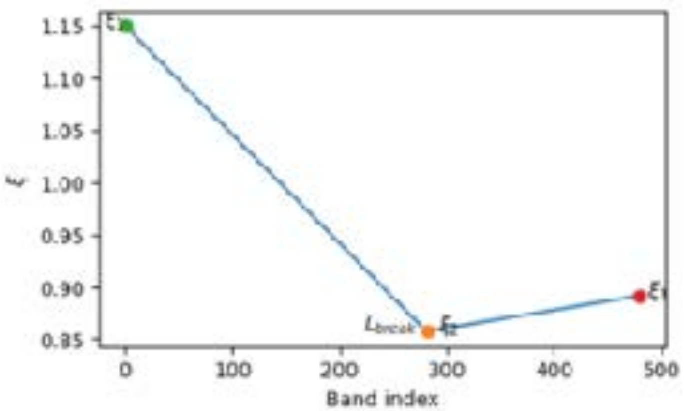


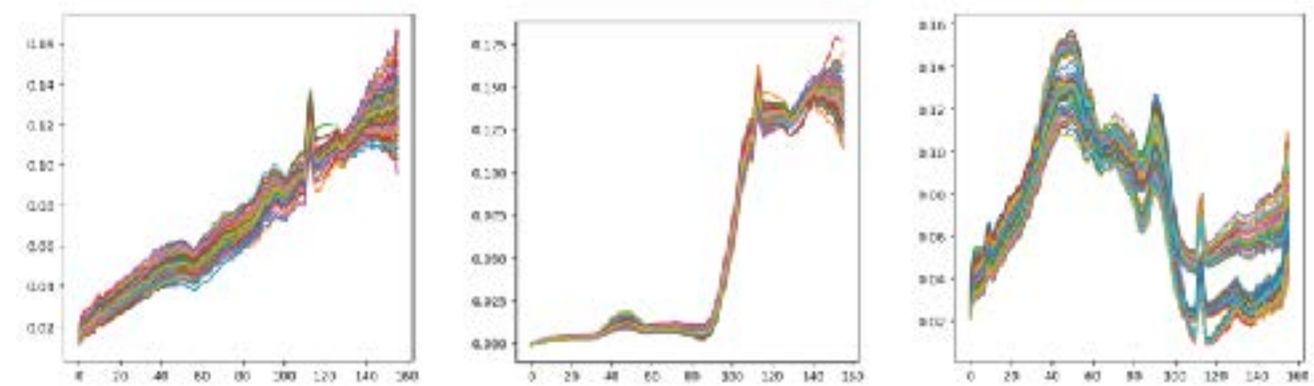
Proposed approach [Hadjeres24]: from the considered hyperspectral image to unmix, we <u>generate several synthetic images with ground-truths</u> to train LPALM. Furthermore, we <u>leverage spectral variability to increase the diversity</u> in the synthetic training set.

Training LPALM in a fully unsupervised way

Synthetic spectra ${}^{(i)}\mathbf{A}^*$ generation :

- Launch several time a randomized model-based unsupervised spectra extraction algorithm, VCA, to extract different examples of pure material spectra.
- Spectral data-augmentation: a piecewise-linear model-based based perturbation [Thouvenin15] is applied to augment the extracted endmember library. We obtain several ${}^{(i)}\mathbf{A}^*, i = 1..N_{train}$





Synthetic abundances ${}^{(i)}\mathbf{S}^*$ generation :

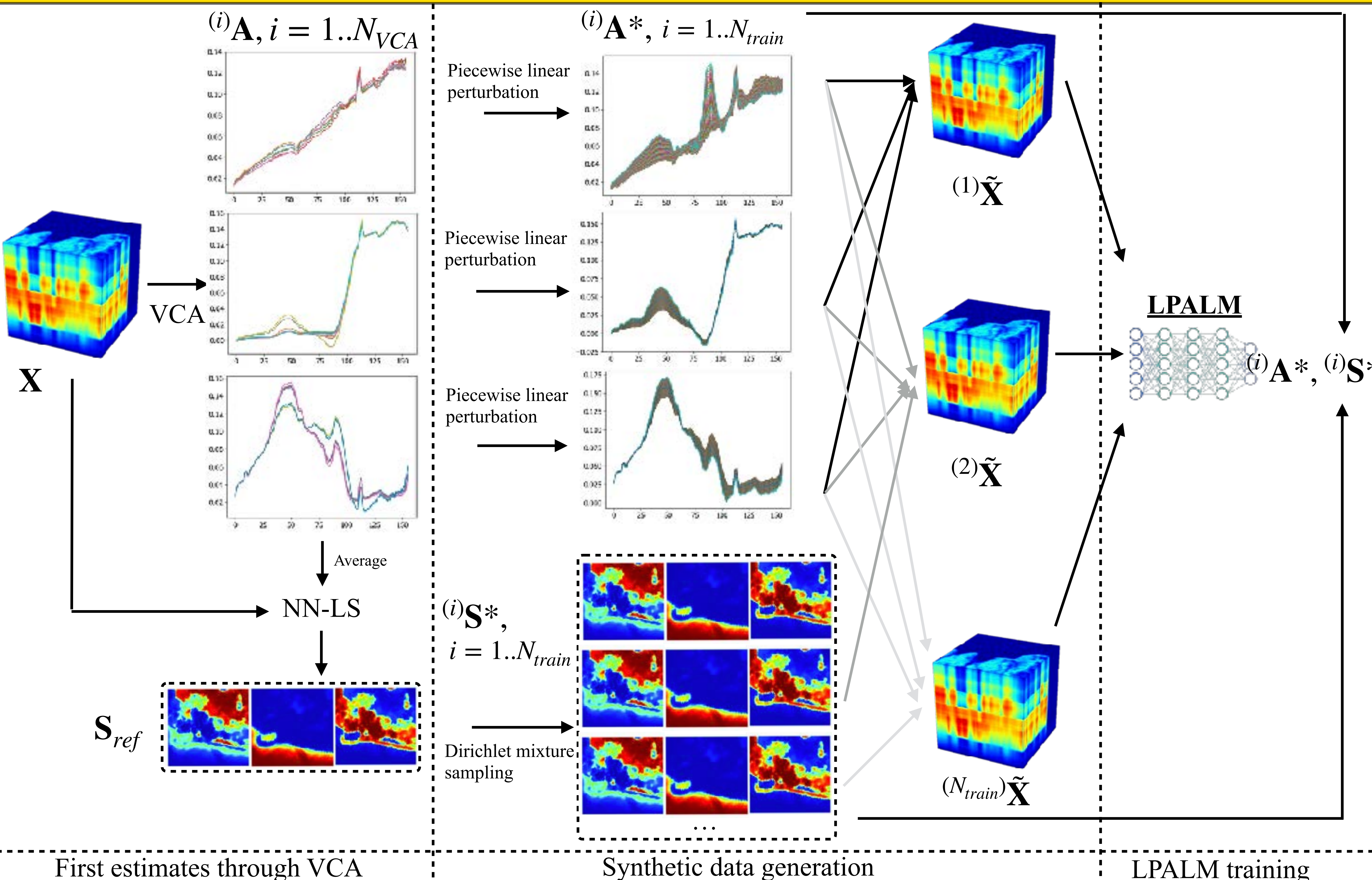
- Compute $\mathbf{A}_{ref} = \underset{i=1..N_{train}}{\text{MEAN}}\left({}^{(i)}\mathbf{A}^*\right)$, estimate $\mathbf{S}_{ref}$ by nonnegative least squares: $\underset{\mathbf{S}_{ref}\geq 0}{\text{argmin}} \frac{1}{2}\|\mathbf{X} - \mathbf{A}_{ref}\mathbf{S}_{ref}\|_F^2$
- Model $\mathbf{S}_{ref}$ by a mixture of Dirichlet
- Draw new samples ${}^{(i)}\mathbf{S}^*, i = 1..N_{train}$ following the mixture of Dirichlet distribution.

Data generation : using the ${}^{(i)}\mathbf{A}^*$ and ${}^{(i)}\mathbf{S}^*$ generated above, compute new datasets:

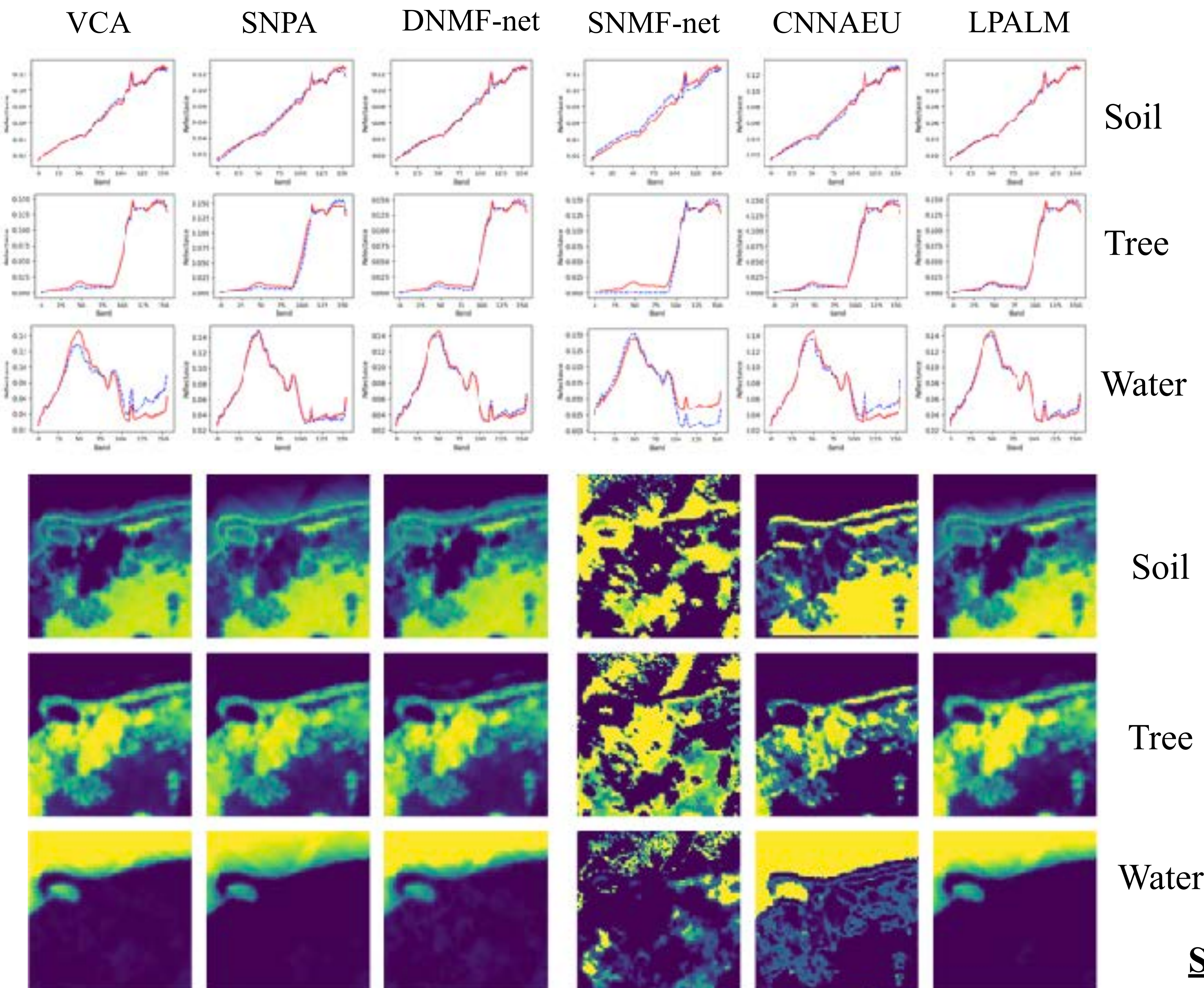
$$ {}^{(i)}\tilde{\mathbf{X}} = {}^{(i)}\mathbf{A}^{*(i)}\mathbf{S}^* + {}^{(i)}\mathbf{N} $$

=> for the ${}^{(i)}\tilde{\mathbf{X}}$ dataset, the groundtruth ${}^{(i)}\mathbf{A}^*$ and ${}^{(i)}\mathbf{S}^*$ is known!

Training LPALM in a fully unsupervised way



Results on the Samson dataset



Spectral Angular Distance (SAD)

Algorithm	VCA	SNPA	DNMF	SNMF	CNNAEU	LPALM
Soil	0.0236	0.0404	0.0233	0.0713	0.0323	**0.0152**
Tree	0.0417	0.0753	0.0494	0.1112	0.0418	**0.0356**
Water	0.1655	**0.0326**	0.0368	0.2164	0.0959	0.0361
Mean	0.0769	0.0494	0.0365	0.1330	0.0567	**0.0290**

Unrolled Nonnegative Matrix Factorization

Limitation of LPALM

Learned-PALM (LPALM) [Fahes22]

initialize A and **S** with a very generic initialization

for k from 1 to L **do** :

$$\mathbf{S} \leftarrow \mathcal{S}_{\underline{\gamma}}\left(\mathbf{S} - \underline{\mathbf{W}}^T(\mathbf{AS} - \mathbf{X})\right) \quad \text{(LISTA-CP update)}$$

$$\mathbf{A} \leftarrow \Pi_{\|.\|\leq 1}\left(\mathbf{A} + \frac{1}{\underline{L_A}}(\mathbf{X} - \mathbf{AS})\mathbf{S}^T\right) \quad \text{(learning step-size)}$$

end for

return A, S

A limitation of LPALM:

- the learnt parameters are the same for all the training samples
- once the parameters are learnt on the training set, they are fixed

=> LPALM is not very adaptative to diversity in the training and test set

Nonnegative matrix factorization (NMF) and Multiplicative Updates

In the following, we will consider another type of regularization than sparsity:

$$\underset{\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{S}\in\mathbb{R}^{n\times t}}{\text{argmin}} \; \underbrace{\frac{1}{2}\|\mathbf{X}-\mathbf{A}\mathbf{S}\|_F^2}_{\textbf{Data-fidelity}} + \underbrace{\iota_{.\geq 0}(\mathbf{A}) + \iota_{.\geq 0}(\mathbf{S})}_{\textbf{Nonnegativity}}$$

A classical algorithm to minimize it is the Multiplicative Update (MU) [Lee,Seung1999]

while not converged:

$$\mathbf{A}^{(l+1)} \leftarrow \mathbf{A}^{(l)} \odot \frac{\mathbf{X}{\mathbf{S}^{(l)}}^T}{\mathbf{A}^{(l)}\mathbf{S}^{(l)}{\mathbf{S}^{(l)}}^T}$$

$$\mathbf{S}^{(l+1)} \leftarrow \mathbf{S}^{(l)} \odot \frac{{\mathbf{A}^{(l+1)}}^T\mathbf{X}}{{\mathbf{A}^{(l+1)}}^T\mathbf{A}^{(l+1)}\mathbf{S}^{(l)}}.$$

Nonnegative matrix factorization (NMF) and Multiplicative Updates

To speed up MU, we can unroll it => Non Adaptative Learned Multiplicative Update (NALMU)

Algorithm 1 NALMU

Require: $\mathbf{X}, L_{NALMU}$

Initialize $\mathbf{A}^{(1)}$ and $\mathbf{S}^{(1)}$ with positive coefficients

for $l \in \{1..L_{NALMU}\}$ **do**

$$\mathbf{A}^{(l+1)} \leftarrow \mathbf{A}^{(l)} \odot \underline{\mathbf{W_A}}^{(l)} \odot \frac{\mathbf{X}\mathbf{S}^{(l)^T}}{\mathbf{A}^{(l)}\mathbf{S}^{(l)}\mathbf{S}^{(l)^T}}$$

$$\mathbf{S}^{(l+1)} \leftarrow \mathbf{S}^{(l)} \odot \frac{\mathbf{A}^{(l+1)^T}\mathbf{X}}{\mathbf{A}^{(l+1)^T}\mathbf{A}^{(l+1)}\mathbf{S}^{(l)}}.$$

end for

return $\mathbf{A}^{(L+1)}$ and $\mathbf{S}^{(L+1)}$

Motivation of the new update:

- $\underline{\mathbf{W_A}}$ acts as a mask
- Its is easier to perform learning on **A**

But at this stage, NALMU suffers from the same flaw as LPALM: $\mathbf{W_A}$ is fixed once for all

Nonnegative matrix factorization (NMF) and Multiplicative Updates

To make learned MU more adaptative, we rather proposed the Adaptative Learned Multiplicative Updates (ALMU)

Algorithm 2 ALMU

Require: $\mathbf{X}, L_{NALMU}, L_{ALMU}$

$\mathbf{A}_{NALMU}, \mathbf{S}_{NALMU} = \text{NALMU}(\mathbf{X}, L_{NALMU})$

$\mathbf{A}^{(1)} = \mathbf{A}_{NALMU},\ \mathbf{S}^{(1)} = \mathbf{S}_{NALMU}$

for $l \in \{1..L_{ALMU}\}$ **do**

$\mathbf{A}^{(l+1)} \leftarrow \mathbf{A}^{(l)} \odot \underline{\mathbf{W}_{\mathbf{A}}}^{(l)}(\mathbf{A}_{NALMU}) \odot \frac{\mathbf{X}\mathbf{S}^{(l)T}}{\mathbf{A}^{(l)}\mathbf{S}^{(l)}\mathbf{S}^{(l)T}}$

$\mathbf{S}^{(l+1)} \leftarrow \mathbf{S}^{(l)} \odot \frac{\mathbf{A}^{(l+1)T}\mathbf{X}}{\mathbf{A}^{(l+1)T}\mathbf{A}^{(l+1)}\mathbf{S}^{(l)}}.$

end for

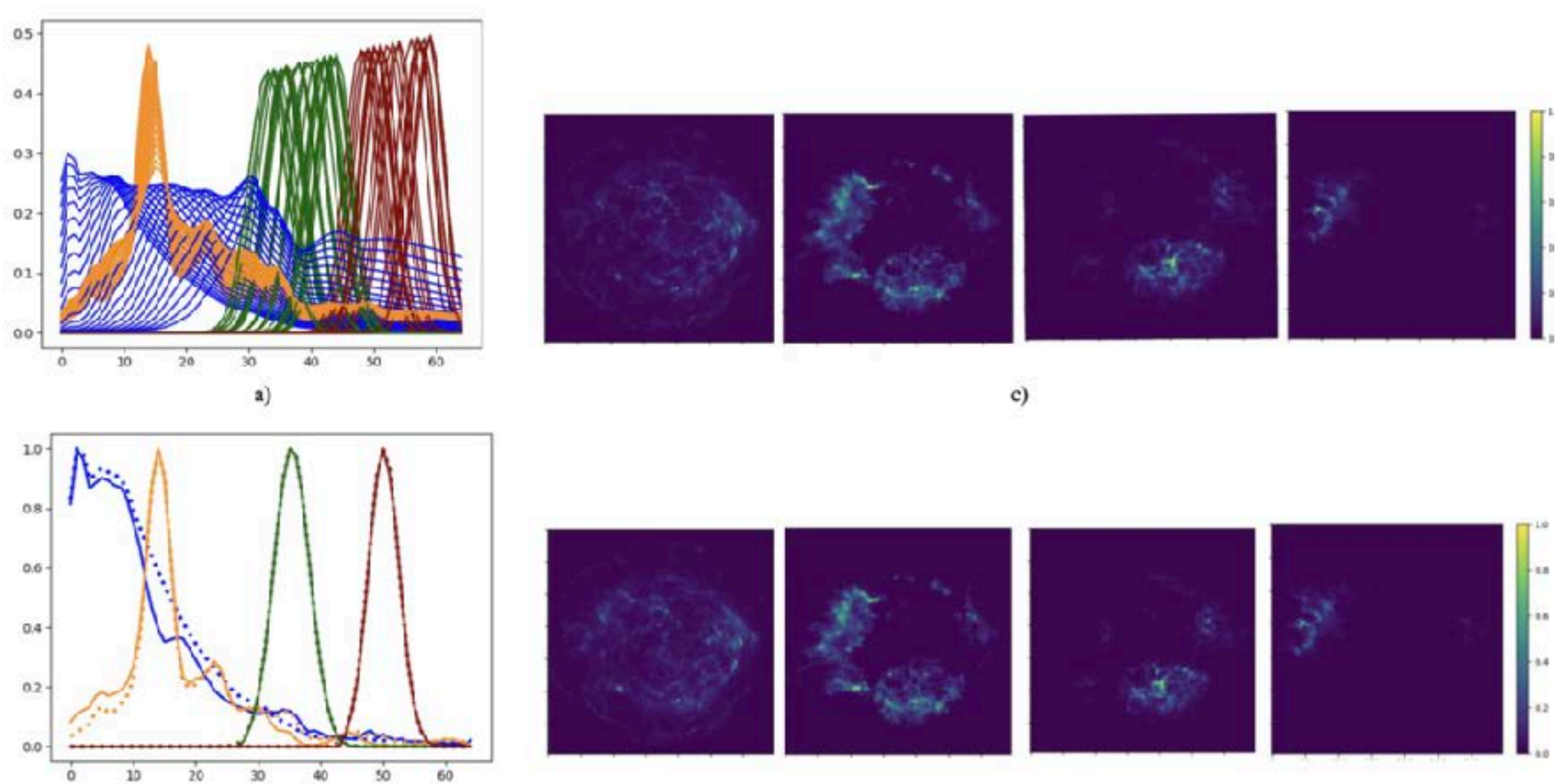
return $\mathbf{A}^{(L+1)}$ and $\mathbf{S}^{(L+1)}$

Motivation of the new update:

- $\mathbf{W}_{\mathbf{A}}$ is now specific for each new entry $\mathbf{X}$
- In practice, it is parametrized with a small MLP
- Predicting it from $\mathbf{A}_{NALMU}$ enables to reduce the computational burden

Results

	MU	DNMF ($\lambda = 0$)	SNMF	CD	NALMU	ALMU (averaged)
SAD($\mathbf{A}^*, \mathbf{A}$)	0.9650	0.7008	0.7767	0.9815	0.9233	0.9878



Kervazo, C., Chetoui, A., & Cohen, J. E. Deep unrolling of the multiplicative updates algorithm for blind source separation, with application to hyperspectral unmixing.

Conclusion

Take-home messages:

- Unrolling is a very flexible tool for inverse problems
- It has a much smaller of parameters to train than black-box neural networks and is much **more scalable** than iterative algorithms
- It is more interpretable than black-box neural networks
- We proposed adatative-to-the-dataset schemes

References

[Zibulevsky01] Zibulevsky, M., & Pearlmutter, B. A. (2001). Blind source separation by sparse decomposition in a signal dictionary. Neural computation, 13(4), 863-882.

[Comon10] Comon, P., & Jutten, C. (Eds.). (2010). Handbook of Blind Source Separation: Independent component analysis and applications. Academic press.

[Gillis14] Gillis, N. (2014). The why and how of nonnegative matrix factorization. Regularization, optimization, kernels, and support vector machines, 12(257), 257-291.

[Dereure23] Dereure, E., Kervazo, C., Seguin, J., Garofalakis, A., Mignet, N., Angelini, E., & Olivo-Marin, J. C. (2023, April). Sparse Non-Negative Matrix Factorization for Preclinical Bioluminescent Imaging. In 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI) (pp. 1-5). IEEE.

[Kervazo20] Kervazo, C., Bobin, J., Chenot, C., & Sureau, F. (2020). Use of palm for $\ell 1$ sparse matrix factorization: Difficulty and rationalization of a two-step approach. Digital Signal Processing, 97, 102611.

[Gregor, Lecun 10] Gregor, K., & LeCun, Y. (2010, June). Learning fast approximations of sparse coding. In Proceedings of the 27th international conference on international conference on machine learning (pp. 399-406).

[Chen18] Chen, X., Liu, J., Wang, Z., & Yin, W. (2018). Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. Advances in Neural Information Processing Systems, 31.

[Fahes22] Fahes, M., Kervazo, C., Bobin, J., & Tupin, F. (2022, April). Unrolling PALM for sparse semi-blind source separation. In International Conference on Learning Representations.

[Hadjeres24] Hadjeres R., Kervazo C., Tupin F., Generating synthetic data to train a deep unrolled network for Hyperspectral Unmixing, accepted to EUSIPCO 2024

[Kern24] Kern J., Bobin J., Kervazo C., EVIL-Deconv: Efficient Variability-Informed Learned Deconvolution using Algorithm Unrolling, submitted to Neurips 2024