UNROLLING PALM FOR SPARSE SEMI-BLIND SOURCE SEPARATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Blind Source Separation (BSS) has become a well established tool for a wide range of applications – for instance, in astrophysics and remote sensing. Classical sparse BSS methods, such as the **P**roximal **A**lternating **L**inearized **M**inimization (PALM) algorithm, nevertheless often suffer from a difficult hyperparameter choice, which undermines their results. To bypass this pitfall, we propose in this work to build on the thriving field of algorithm unfolding/unrolling. Unrolling PALM enables to leverage the data-driven knowledge stemming from realistic simulations or ground-truth data by learning both PALM hyperparameters and variables. In contrast to most existing unrolled algorithms, which assume a fixed known dictionary during the training and testing phases, this article further emphasizes on the ability to deal with variable mixing matrices (a.k.a. dictionaries). The proposed Learned PALM (LPALM) algorithm thus enables to perform semi-blind source separation, which is key to increase the generalization of the learnt model in real-world applications. We illustrate the relevance of LPALM in astrophysical multispectral imaging: the algorithm not only needs up to $10^4 - 10^5$ times less iterations than PALM, but also improves the separation quality, while avoiding the cumbersome hyperparameter and initialization choice of PALM. We further show that LPALM outperforms other unrolled source separation methods in the semi-blind setting.

1 INTRODUCTION

Blind source separation (BSS) : In a wide range of scientific domains, including remote sensing (Dobigeon et al., 2013) and astrophysics (Picquenot et al., 2019), BSS is a very popular paradigm aiming at finding physical decompositions of multi-valued data (Comon & Jutten, 2010). Specifically, the data-set at hand is assumed to have been generated by a mixture of elementary signals, called sources. Assuming a linear mixing model, this mathematically writes as

$$X = A^* S^* + N. \tag{1}$$

The objective of BSS is then to recover the unknown mixing $A^* \in \mathbb{R}^{m \times n}$ and sources $S^* \in \mathbb{R}^{n \times t}$ matrices from the knowledge of the data set $X \in \mathbb{R}^{m \times t}$ only (and often despite the presence of a Gaussian noise $N \in \mathbb{R}^{m \times t}$). Unfortunately, it has long being emphasized that BSS is an ill-posed matrix factorization problem (Comon & Jutten, 2010): merely minimizing the reconstruction error $\|X - AS\|_F^2$ over A and S (with $\|.\|_F$ standing for the Frobenius norm) often leads to spurious estimates which are very different from the true underlying A^* and S^* matrices. As such, most BSS approaches impose additional priors on the sought-after factors. Among them, sparsity has led to some of the state-of-the-art results (Zibulevsky & Pearlmutter, 2001; Bobin et al., 2007; 2015). It amounts to tackling the following minimization problem:

$$\underset{A \in \mathbb{R}^{m \times n},\, S \in \mathbb{R}^{n \times t}}{\arg\min} \quad \underbrace{\frac{1}{2}||X - AS||_F^2}_{\text{data-fidelity term}} + \underbrace{\lambda ||S||_1}_{\text{sparsity constraint}} + \underbrace{\iota_{\{A^i, ||A^i||_2^2 \leq 1\, , i \in [1,n]\}}(A)}_{\text{oblique constraint}}, \tag{2}$$

where $||S||_1 = \sum_{i=1}^n \sum_{j=1}^t |S_{ij}|$, ι_B is the characteristic function of a set B, and A^i denotes the i-th column of A. In the following, we will specifically focus on sparse BSS.
Being a difficult multi-convex problem, the optimization strategy is key to minimize (2). Among

the various iterative optimization algorithms that have been proposed so far, PALM (Attouch et al., 2010; Bolte et al., 2014) is particularly appealing due to its mathematical guarantees (Xu & Yin, 2013) because it has been proved to converge to a critical point of (2). An iteration of PALM for minimizing the cost function (2) consists in alternating the minimization with respect to S and A. Specifically, at each iteration: i) a proximal gradient step of (2) is applied to update S while A is fixed; ii) A is updated thanks to a proximal gradient step of (2) with S fixed. The process is then repeated until the algorithm converges. We refer the reader to Section 2 for more details about PALM.

In the context of sparse BSS though, as highlighted in (Kervazo et al., 2020) PALM suffers from severe limitations, especially due to the high sensitivity of the estimated factors with respect to the choice of the regularization parameters. It is also emphasized in (Kervazo et al., 2020) that the solution strongly depends on the initialization. Altogether, these flaws, which are confirmed in our numerical experiments, undermine the results of PALM and limit its applicability in real-world applications. Therefore, we propose to investigate how algorithm unrolling methods can bypass these limitations.

Algorithm unrolling : Algorithm unrolling (Monga et al., 2021) was pioneered in (Gregor & LeCun, 2010) for sparse coding. In our context, the sparse coding problem mostly corresponds to *non-blind* source separation, in which $A = A^*$ is known in (2). Quite similarly to PALM, such a cost function is usually minimized through an iterative proximal gradient step algorithm, such as the well known Iterative Shrinkage Thresholding Algorithm (ISTA) (Daubechies et al., 2004; Blumensath & Davies, 2008). Each ISTA iteration writes as:

$$\tilde{S}^{(k+1)} = \text{ST}_{\frac{\lambda}{L_S^{(k)}}} \left(\tilde{S}^{(k)} - \frac{1}{L_S^{(k)}} A^{*T}(A^* \tilde{S}^{(k)} - X) \right), \tag{3}$$

with $\tilde{S}^{(k)}$ being the estimate of S^* at iteration k, $L_S^{(k)}$ the Lipschitz constant of the gradient of (2) data-fidelity term. ST is the soft-thresholding operator [1].

In (Gregor & LeCun, 2010), ISTA iterations are viewed as the layers of a recurrent neural network (RNN) with parameters $\theta = \frac{\lambda}{L}$, $W_1 = \mathbb{I} - \frac{1}{L} A^{*T} A^*$ and $W_2 = \frac{1}{L} A^{*T}$. Instead of being fixed, the parameters θ, W_1, W_2 are however trained using backpropagation. To make the distinction clearer in the following, we will write trainable variables with underlined letters, for instance $\underline{\theta}, \underline{W_1}, \underline{W_2}$. Assuming parameters to be untied across the layers, the kth layer of LISTA is thus given by:

$$\tilde{S}^{(k+1)} = \text{ST}_{\underline{\theta}^{(k)}} \left(\underline{W_1}^{(k)} \tilde{S}^{(k)} + \underline{W_2}^{(k)} X \right), \quad k = 0, 1, ..., K-1. \tag{4}$$

Consequently, the resulting network, named Learned ISTA (LISTA), "unrolls" ISTA and truncates it to K iterations. The appeal of unrolled algorithms is twofold. On the one hand, they yield better estimates than iterative algorithms thanks to their higher representation power (Monga et al., 2021). On the other hand, numerous empirical results (Sprechmann et al., 2015; Wang et al., 2016; Zhang & Ghanem, 2018; Zhou et al., 2018) show that they generalize well to unseen samples, thanks to their interpretable structure.

Nevertheless, when the input distribution changes too much between the training and testing phases (e.g., A^* changes, which is in particular the case in BSS), they often lead to impaired results as they tend to (over-)fit the training set distribution. To make the network more robust with respect to small changes in the dictionary A^*, it has been proposed to introduce small perturbations $\epsilon_A \sim \mathcal{N}(0, \sigma_{\max})$ on the elements of the dictionary A^* during the training (Liu & Chen, 2019). Although such an attempt might be beneficial in some sparse coding problems (Zhao et al., 2011), the dictionaries vary much more in BSS because of the blind nature of the problem (and the variations might not be Gaussian). In contrast, we propose in this work to use a taylored PALM-based unrolling structure to account for the fact that A^* is unknown and might largely change in the training and testing phases.

1.1 Related works

In hyperspectral imaging, remote sensing sensors generally have low spatial resolution. Therefore, each pixel usually contains several materials, calling for unmixing problems that can be recast as source separation (Bioucas-Dias & Plaza, 2011). Recently, (Qian et al., 2019) proposed to use unrolling for non-blind hyperspectral unmixing HSU (A^* known beforehand), mostly by unrolling

[1] $\text{ST}_\theta(v) = \text{sign}(v)\max(0, |v| - \theta)$, which applies component-wise.

ISTA into the same architecture as (Gregor & LeCun, 2010). In the same context, (Zhou & Rodrigues, 2021) unrolled the ADMM algorithm (Bioucas-Dias & Figueiredo, 2010). In an attempt to perform *blind* HSU, (Qian et al., 2020) also unrolled ISTA to estimate S^*, but proposed to add an extra layer in the neural network to learn A^*. The drawback is thus that the estimate $\tilde{A}$ is hard-coded in the network architecture, which is not well suited to the application on new hyperspectral images. Upon preparing this work, we became aware of the very recent work of (Xiong et al., 2021), which proposes an unrolled architecture in order to account for the spectral variabilities in A^*. They use an unsupervised cost function, which is a sum over the K layers of the network of the data set reconstruction error ($X_{\texttt{train}}$ being the training set of size $t_{\texttt{train}}$, *i.e.* here a patch of the hyperspectral image):

$$\text{Loss} = \frac{1}{2t_{\texttt{train}}} \sum_{k=1}^{K} \|X_{\texttt{train}} - \tilde{A}^{(k)} \tilde{S}^{(k)}\|_F^2. \tag{5}$$

Such a loss is quite interesting as it does not require to know the ground truth factors A^* and S^* at training. A huge drawback is that the reconstruction-based error has long been advocated to possibly lead to spurious estimates, which are very different from the ground truth A^* and S^* factors (Comon & Jutten, 2010; Zibulevsky & Pearlmutter, 2001; Gillis, 2020). Consequently, (Nasser et al., 2021) rather added further priors on the sought after-factors in their unrolled algorithm cost function, at the price of adding extra hyperparameters to be tuned by hand. Since they focus on unrolling the multiplicative update (MU) algorithm, their network structure is quite different from both (Xiong et al., 2021) and the algorithm we investigate in this work.

1.2 Motivations and Contributions

As stated above, current source separation state-of-the-art lets the practitioner in an unsatisfactory situation: 1) classical *Blind* source separation methods based on algorithms such as PALM are hindered by a cumbersome hyperparameter choice. Until now, unsupervised unrolling-based methods have felt short of solving such an issue: they often either lead to spurious solution or still require parameters to tune by hand. 2) *Non-blind* source separation (NBSS) is limited by the fact that most methods assume A^* to be known and fixed during both train and test phases, which is often unrealistic in practical experiments.

In this article, we propose to merge the best of both worlds by performing *semi-blind* source separation (SBSS). Specifically, practitioners usually have access to an *imperfect* knowledge of the mixing matrix A^*, for instance by physics-derived simulators they might possess (which is in particular the case in astrophysics) or by approximate ground-truths of signals/images of the same nature as the ones they expect to observe (for instance, databases of spectra in remote sensing). The proposed Learned-PALM (LPALM) algorithm gives a principled and easy way to leverage such an imperfect knowledge to bypass the flaws of conventional sparse BSS algorithm. Specifically, the contributions of this article are:

- We unroll the PALM algorithm into the LPALM network, which is specifically taylored for sparse SBSS and to cope with **imperfectly known mixing matrices** during the training. In particular, in contrast to usual supervised unrolled algorithms such as LISTA (Gregor & LeCun, 2010) and LISTA-CP (Chen et al., 2018), which do not alternate between A and S updates, LPALM better accounts for the variability of the mixing matrix A^* which occurs in BSS problems.
- We experimentally highlight the limitations of current unsupervised unrolled BSS algorithms originating from HSU. By leveraging the information contained within astrophysics simulations, LPALM largely outperforms the networks of (Xiong et al., 2021; Qian et al., 2020; Nasser et al., 2021) in terms of i) separation quality and ii) computation time. Compared to (Nasser et al., 2021), the approach is different as it does not require the mixtures to be nonnegative, in contrast to the MU updates they use. This in particular enables to easily leverage sparsifying transforms such as wavelets.
- We provide an in-depth **comparative study between LPALM and PALM**. Specifically, we show that LPALM i) **overcomes the limitations of PALM**, with a lowered sensitivity to the initialization and better generalization properties and ii) **accelerates the estimation time up to $10^4 - 10^5$ times**. Altogether, these are key properties to make source separation reliable in real applications.

2 UNROLLING FOR SEMI-BLIND SOURCE SEPARATION

Following (Bolte et al., 2014), the PALM algorithm sequentially and alternatively updates the factors S and A using a proximal gradient step. More precisely, this boils down to the following steps:

$$\begin{aligned}\tilde{S}^{(k+1)} &= \text{ST}_{\theta^{(k)}}\left(\tilde{S}^{(k)} - \frac{1}{L_S^{(k)}}\tilde{A}^{(k)T}(\tilde{A}^{(k)}\tilde{S}^{(k)} - X)\right)\\ \tilde{A}^{(k+1)} &= \Pi_{||.||_2\leq 1}\left(\tilde{A}^{(k)} - \frac{1}{{L_A}^{(k)}}(\tilde{A}^{(k)}\tilde{S}^{(k+1)} - X)\tilde{S}^{(k+1)T}\right)\end{aligned} \tag{6}$$

where $\theta^{(k)} = \lambda/{L_S}^{(k)}$ and ${L_S}^{(k)}$ (*resp.* ${L_A}^{(k)}$) is usually chosen as the squared spectral norm of $S^{(k)}$ (*resp.* $A^{(k)}$). A natural unrolling architecture for PALM consists in keeping a similar alternating design, but different models can be used to update S and A, which is discussed below.

Update step for S: In the framework of PALM, the $\tilde{S}$ update is equivalent to a special case of an ISTA, which makes LISTA (equation 4) the most straightforward trainable model to use. A key feature of LISTA is that it leverages a reparameterization of a standard proximal gradient descent update rule through the two trainable linear operators $\underline{W_1^{(k)}}$, $\underline{W_2^{(k)}}$ and $\underline{\theta^{(k)}}$. The advantage is twofold i) learning $\underline{\theta^{(k)}}$ enables to bypass PALM cumbersome hyperparameter choice ; ii) learning $\underline{W_1^{(k)}}$ and $\underline{W_2^{(k)}}$ allows for extra degrees of freedom to efficiently estimate S^*. Nevertheless, in the LISTA framework, A^* is assumed to be fixed across the training and test sets, which significantly deviates from the present SBSS case. This might lead to deteriorated estimates of S^*, since there is in particular every reason to believe $\underline{W_1^{(k)}}$ and $\underline{W_2^{(k)}}$ to be A^*-dependent (recall that initially LISTA reparametrization was motivated by $W_1 = \mathbb{I} - \frac{1}{L}A^{*T}A^*$ and $W_2 = \frac{1}{L}A^{*T}$).
On the other hand, the empirical success of LISTA has motivated several works on its theoretical understanding (Moreau & Bruna, 2019; Giryes et al., 2018; Liu & Chen, 2019; Ablin et al.). Of particular interest to us, (Chen et al., 2018) proved an asymptotic coupling of LISTA weights. Based on this, they derived a new architecture named LISTA-CP. In contrast to LISTA, using (4) as an update, a layer of LISTA-CP is given by:

$$\tilde{S}^{(k+1)} = \text{ST}_{\underline{\theta^{(k)}}}(\tilde{S}^{(k)} + \underline{W^{(k)}}^T(X - A^*\tilde{S}^{(k)})), \quad k = 0, 1, ..., K-1. \tag{7}$$

LISTA-CP explicitly makes use of A^* and is thus in principle better suited to deal with variable mixing matrices, provided A^* to be known or at least decently estimated. In section 3.1, we make a thourough experimental comparison between ISTA, LISTA, LISTA-CP, and ISTA-LLT (defined in the same section) in the presence of mixing matrix variabilities in the training and testing stages. These experiments confirm the insights described above : in a nutshell, LISTA does not accelerate the estimation of S^* compared to ISTA for the same number of layers/iterations, and is also outperformed by LISTA-CP to a quite large extent, even when inaccurate A^* are used in the updates. Consequently, our final choice is to opt for a LISTA-CP structure for $\tilde{S}$ update in the LPALM algorithm.

Update step for A: The update of A merely aims at minimizing a least-squares problem, constraining each column of A to belong to the ℓ_2 ball of radius 1. Several update strategies have been tested, which allowed to opt for an ISTA-LL (ISTA with **L**earned $\mathbf{L}_A$) update where only the step size is trained.

The LPALM algorithm: Putting together the above updates, the LPALM algorithm then reads as[2]:

$$\begin{aligned}\tilde{S}^{(k+1)} &= \text{ST}_{\underline{\theta^{(k)}}}\left(\tilde{S}^{(k)} - \underline{W^{(k)}}^T(\tilde{A}^{(k)}\tilde{S}^{(k)} - X)\right)\\ \tilde{A}^{(k+1)} &= \Pi_{||.||_2\leq 1}\left(\tilde{A}^{(k)} - \frac{1}{\underline{L_A}^{(k)}}(\tilde{A}^{(k)}\tilde{S}^{(k+1)} - X)\tilde{S}^{(k+1)T}\right)\end{aligned} \tag{8}$$

[2]The code will be made available online.

where $\{\underline{\theta}^{(k)}, \underline{W}^{(k)}, \underline{L_A}^{(k)}\}_{k=\{1,\dots,K\}}$ are the $K(m \times n + 2)$ trainable parameters. On top of the above discussion on S and A individual updates, a further interest to use for S-update (*resp.* A-update) a LISTA-CP-like (*resp.* ISTA-LL-like) update is that it enables to keep the alternating structure of PALM. On the contrary, when using LISTA, A^* (resp. S^*) appears only implictly through the $\underline{W_1}^{(k)}$ and $\underline{W_2}^{(k)}$ matrices, which prohibits a direct alternation. In addition, a LISTA-CP-like architecture for A-update would make the network specific to an image/signal size, which is undesirable in real world applications.

LPALM loss function: In contrast to other unrolling methods (Xiong et al., 2021; Nasser et al., 2021), the semi-blind context allows to benefit from the knowledge of S^* and A^* *during the training phase*. It is then possible to define a loss which is sensitive to the quality of the predicted S and A, such as the averaged sum of NMSE of both outputs:

$$\begin{aligned} \text{Loss}_{\text{LPALM}} &= \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(\text{NMSE}(\tilde{S}^i_{\text{pred}}, {S^*}^i) + \text{NMSE}(\tilde{A}^i_{\text{pred}}, {A^*}^i) \right) \\ &= \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(\frac{\|\tilde{S}^i_{\text{pred}} - {S^*}^i\|_F^2}{\|{S^*}^i\|_F^2} + \frac{\|\tilde{A}^i_{\text{pred}} - {A^*}^i\|_F^2}{\|{A^*}^i\|_F^2} \right) \end{aligned} \tag{9}$$

It is important to point out that this loss does not require a fine tuning of parameters as in the unsupervised algorithm of (Nasser et al., 2021). Additionally, since it is based on the predicted factors and not only the reconstruction error, the proposed LPALM will be shown in the experimental section to be more robust with respect to spurious solutions, which is essential to efficiently tackle non-convex problems. This is due to the fact that we leverage the knowledge contained within the training set, leading to some implicit regularization of the original problem. Implementation details are described in Appendix A.

3 Numerical experiments

In this section, we assess the LPALM algorithm on astrophysical data simulations derived from hyperspectral X-ray images (Picquenot et al., 2019); see details in Appendix B. The corresponding spectra in A^* have large variabilities over the data set, as illustrated in Figure 1. In Subsection 3.1, we numerically confirm the unrolling choices done in the previous section using a relatively simple data set with $n = 2$ sources. In Subsection 3.2, we emphasize on the ability of LPALM to bypass PALM classical flaws, based on the realistic data set of Figure 1 (in which $n = 4$). Lastly, in Subsection 3.3, we compare LPALM to state-of-the art methods on a data set including real sources.

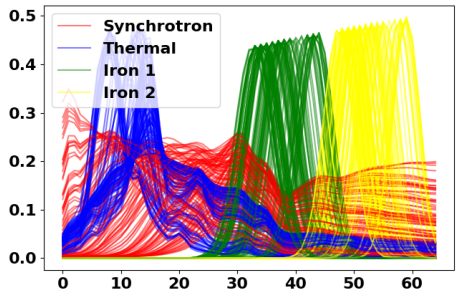


Figure 1: Illustration of the variability of the $A^* \in \mathbb{R}^{65\times 4}$ matrices within the data set of Section 3. 100 examples of A^* are depicted, each column being drawn with a different color.

In the following, the metric used to assess the separation quality will be the NMSE between the output of the network and the ground truth, computed either on the estimated sources $\tilde{S}_{\text{pred}}$ or the estimated mixing matrices $\tilde{A}_{\text{pred}}$:

$$\text{NMSE}(\tilde{S}_{\text{pred}}, S^*) = \frac{||\tilde{S}_{\text{pred}} - S^*||_F^2}{||S^*||_F^2} \text{ and } \text{NMSE}(\tilde{A}_{\text{pred}}, A^*) = \frac{||\tilde{A}_{\text{pred}} - A^*||_F^2}{||A^*||_F^2}. \tag{10}$$

3.1 Which model is empirically the best for updating S in LPALM ?

To experimentally confirm the unrolling choices we did for the S-update in Section 2, we assess the robustness of LISTA and LISTA-CP in the presence of mixing matrix variabilities at the training and testing stages. We also test ISTA-LLT (ISTA with **L**earned $\mathbf{L}_S$ and **T**hreshold), which is

parameterized as follows:

$$\tilde{S}^{(k+1)} = \mathrm{ST}_{\underline{\theta^{(k)}}}\left(\tilde{S}^{(k)} - \frac{1}{\underline{L_S^{(k)}}} A^{*T}(A^*\tilde{S}^{(k)} - X)\right) \tag{11}$$

The goal of this subsection is twofold : 1) in the SBSS context, A^* varies between the samples: we determine which of the tested unrolling methods is more robust in such a setting; 2) while the updates of LISTA-CP (Eq. 7) and ISTA-LLT (Eq. 11) require in principle the perfect knowledge of A^* for each sample, in SBSS A^* is unknown in the test phase and must be iteratively estimated within LPALM. Therefore, we evaluate LISTA-CP and ISTA-LLT when A^* is perfectly estimated but also when it is not.

Data set: There are 10 000 mixing matrices $A^* \in \mathbb{R}^{40\times 2}$, which correspond to the thermal and a Gaussian-shaped line emission, represented in Figure 5, see Appendix B. To mimic mildly sparse sources, the 10 000 matrices $S^* \in \mathbb{R}^{2\times 500}$ are simulated using generalized Gaussian distribution with a shape parameter $\beta = 0.3$. This makes a total of 10 000 samples of $X \in \mathbb{R}^{40\times 500}$, which are further corrupted with white Gaussian noise so that SNR $= 30$ dB.

Implementation: The total number of layers for the unrolled methods is $K = 5$. The weights initialization is described in the Appendix A. To assess the quality of LISTA-CP and ISTA-LLT when A^* is not perfectly known, a corrupted version of the mixing matrix $A^* + N_A$, with N_A a white Gaussian noise such that the SNR with respect to A^* is 10 dB, is used in LISTA-CP (7) and ISTA-LLT (11) updates. The corresponding algorithm results are denoted as LISTA-CP (SNR $= 10$ dB) and ISTA-LLT (SNR $= 10$ dB) in Figure 2.

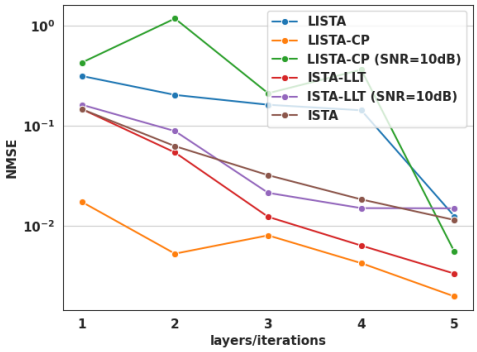


Figure 2: NMSE of several unrolled non-blind architectures, compared to ISTA truncated to 5 layers. The LISTA-CP and ISTA-LLT curves correspond to the updates (7) and (11) where A^* is perfectly known. The LISTA-CP (SNR $= 10$ dB) and ISTA-LLT (SNR $= 10$ dB) curves correspond to updates using a noisy estimate of A^*.

ISTA truncated at 5 layers is used as a benchmark to compare the architectures. Its regularization parameter is $\lambda_{\text{ISTA}} = 2.6 \times 10^{-5}$, which was chosen so as to minimize the average NMSE over the test set.

Discussion: Firstly, LISTA gives similar reconstruction qualities than ISTA truncated to 5 layers, which would not be surprising if A^* were the same for all the data samples. Indeed, LISTA has a larger representation power than ISTA. Using the ground truth S^* in the loss function also enables the network to learn an additional prior on the sources. However, this was not necessarily expected in our experiment as LISTA trainable parameters $\underline{W_1}^{(k)}$ and $\underline{W_2}^{(k)}$ tend to be optimized by marginalizing over the varying mixing matrices A^* in the training set, which could have led to deteriorated results. Such a phenomenon probably explains the superiority of LISTA-CP and ISTA-LLT over LISTA, since they precisely use information about the mixing, which enhances their robustness with respect to the variabilities of A^*. As expected, the results of LISTA-CP and ISTA-LLT both deteriorate when noisy mixing matrices are used within their updates (LISTA-CP (SNR $= 10$ dB) and ISTA-LLT (SNR $= 10$ dB) curves). While the results of ISTA-LLT (SNR $= 10$ dB) become worse than ISTA, LISTA-CP (SNR $= 10$ dB) still gives better estimates than ISTA. The higher ability of the LISTA-CP architecture to deal with noisy A^* estimate might be linked to a higher capacity of the network than ISTA-LLT, as there is a much higher number of parameters. Such a property justifies the choice of a LISTA-CP like update in LPALM.

3.2 PALM AND LPALM, A COMPARATIVE STUDY

In this section, we specifically focus on comparing the PALM and LPALM algorithms to show that the latter enables to bypass PALM flaws. As emphasized in (Kervazo et al., 2020), the main pitfalls of PALM lie in its high sensitivity to the initialization and to the regularization parameter choice.

Table 1: Median NMSE of PALM over the test set, with the best λ chosen from the training set, compared to that of LPALM.

Algorithm	PALM	LPALM
Median NMSE	0.040	0.00085

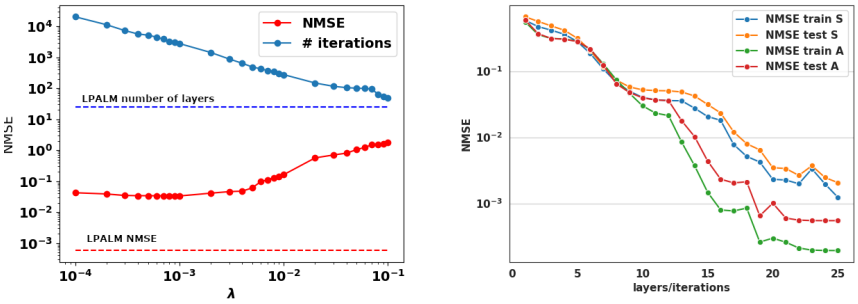
On top of that, both are coupled, which makes it very hard to use PALM in real-world applications.
Data set: There are 900 mixing matrices $A^* \in \mathbb{R}^{65\times 4}$, which are exemplified in Figure 1. As such, this data set is more difficult than the one of the above subsection. The 900 matrices $S^* \in \mathbb{R}^{4\times 500}$ are simulated using generalized Gaussian distribution with a shape parameter $\beta = 0.3$. The mixtures are further corrupted with white Gaussian noise so that SNR $= 30$ dB. The training set consists in 750 samples. The NMSE in Subsections 3.2.1 and 3.2.2 is evaluated on the estimations $\tilde{S}$ of S^*.

3.2.1 Impact of the initialization on LPALM

In this experiment, PALM and LPALM are initialized with the same (bad) first guess estimate for A^* and S^*, which enables to compare the sensitivity of each algorithm to the initialization.
Implementation: Both algorithms are initialized with $\tilde{A}^{(0)} = \Pi_{||.||_2=1}(\underline{1}^{m\times n})$, $\tilde{S}^{(0)} = \underline{0.1}^{n\times t}$. PALM is applied to a single test sample with 10^5 different values of the regularization parameter λ ranging from 10^{-7} to 10^{-2}. It is stopped when consecutive estimates verify both $||\tilde{S}^{(k+1)} - \tilde{S}^{(k)}||_F < 10^{-7}$ and $||\tilde{A}^{(k+1)} - \tilde{A}^{(k)}||_F < 10^{-7}$.
Discussion: With these settings, PALM requires only, over the different tested regularization parameters, a median number of 24 iterations to converge (max = 24, min = 22, std = 0.963). Nevertheless, the median NMSE of the PALM solution is very high: 0.8249 (max $= 0.8297$, min $= 0.8213$, std $= 0.0024$). On the contrary, the LPALM algorithm yields a good NMSE of 0.0010 on the same testing sample, which highlights that LPALM leads to a much more robust and reliable solver than PALM with respect to initialization.

3.2.2 LPALM overcomes the difficult hyperparameter choice in PALM

To try making PALM better work on the test set, we now infer both its initialization and regularization parameter from the training set. We then compare its results to the one of LPALM.
Implementation: PALM is now initialized with a mixing matrix $\tilde{A}^{(0)}$ taken at random from the training set, and $\tilde{S}^{(0)} = A^{\dagger(0)}X$, where X is a data matrix from the test set to be factorized. The λ hyperparameter is chosen as the one minimizing the median NMSE between PALM source estimate $\tilde{S}_{\text{pred}}$ and S^* over the training set. It is thus the best regularization parameter, among the 28 values we tried, that can be chosen from the training set.
Discussion: Figure 3 shows the results of PALM over the training set. Specifically, we see that the best regularization parameter is $\lambda = 1\times 10^{-3}$, for which PALM reaches a median NMSE of 0.033 in 2794.5 iterations. Thus, when leveraging the training set to use a better initialization, PALM is able to find much better results than in the previous subsection. On the training set still, LPALM reaches better separation qualities than PALM, which is expected as it has more capacity to benefit from the learning. Table 1 displays the average results of both algorithms on the test set. Interestingly, LPALM generalizes well to unseen samples and maintains much better results than PALM, with a difference of more than one order of magnitude. As such, unrolling PALM enables to better leverage the knowledge of the training set than applying PALM with a good initialization and the best λ value from the training set. Furthermore, we highlight that LPALM requires about 100 times less iterations than PALM (for the best λ value, otherwise the ratio can rise even much more), making it much more tractable for large-scale data sets: on the test set, PALM required a median number of 2743 iterations, to be compared with the 25 layers of LPALM. For the sake of exhaustiveness, we furthermore plot the evolution of the values of the NMSE across the layers of LPALM. It is interesting to note that in this experiment, LPALM seems to exhibit a quite monotonic improvement of its estimates over the layers, although this is not guaranteed and might not be the case for other data sets.



(a) (b)

Figure 3: (a) PALM median NMSE and required number of iterations *w.r.t* λ over 45 samples of the train set; (b) LPALM NMSEs per layer on both train and test sets.

3.3 Application to realistic data

In this subsection, we compare LPALM to three unrolled HSU methods: MNNBU (Qian et al., 2020), SNMF-net (Xiong et al., 2021) and DNMF (Nasser et al., 2021). These unsupervised methods are all designed to be applied to a single image X: a part of the pixels of the image is chosen to be the training set, and the model is tested on the same image. Concerning LPALM, the training set is created in a similar way as in the above Subsection; see Appendix B. The methods are compared on a data set made of the same *testing* mixing matrices as in the above Subsection, but this time realistic astrophysical sources are used; see Appendix B. In addition, we had to slightly modify the data set to be able to perform fair comparisons with other unrolled architectures: indeed, MNNBU, SNMF-net and DNMF use a sum-to-one constraint on S^* columns. To use them in the proper way, we therefore normalized the S^* matrix before applying these methods. We furthermore point out that these unrolled networks assume the nonnegativity and sparsity of the sources in the *same* domain. The data are mildly sparse in the pixel domain. In LPALM, a wavelet-based preprocessing, which is customary for SBSS, is performed (see Appendix B for more details). Due to the non-negativity constraints for the other methods, it was impossible to transform the X matrix in the wavelet domain when using them; they are applied in the pixel domain.

The results of LPALM applied to realistic data are shown in Figure 4. Qualitatively, all the four estimated spectra are in very good agreement with the ground truth. Similarly, the estimated sources are well estimated. The corresponding results of the other methods are shown in Appendix D (Figures 8, 9 and 10). Quantitatively, the comparison is performed in terms of the NMSE on the estimated mixing matrix A because the ground truth sources are not the same for LPALM and other unrolled algorithms due to the column normalization. The results are shown in Table 2. The first line displays the NMSE for the example of Figure 4, which clearly confirms LPALM superiority over other state-of-art methods. This is due to the fact that LPALM enables to leverage the knowledge contained within the whole training set (in particular, in the ground-truth), while the other methods only deal with single images. Furthermore, DNMF is better than MNNBU and SNMF-net: this is probably related to the fact that it uses a regularized cost function and not only the reconstruction error, which is less likely to lead to spurious solutions.

We further assessed the generalization capacity of the four methods. The second line of the table shows the median NMSE when the same learned models are applied on the other 149 examples of the test set. These results show that the three tested state-of-the-art methods do not generalize well to unseen samples. This is expected, as they are here trained on a single image and thus cannot learn the A^* variabilities over the different images. This therefore demonstrates that the proper way to deal with A^* variabilities with these networks is to re-train them for each new image: this might be impractical for real-life applications, due to the computational cost and the difficult hyperparameter setting of some of them (learning rate schedule in particular). Lastly, the third line of Table 2 shows, over 20 samples, the LPALM three competitors' results when they are retrained for each new input matrix X. Interestingly, even in this setting LPALM largely outperforms the other methods, by almost two orders of magnitude. Therefore, LPALM does not need a retraining for each new image

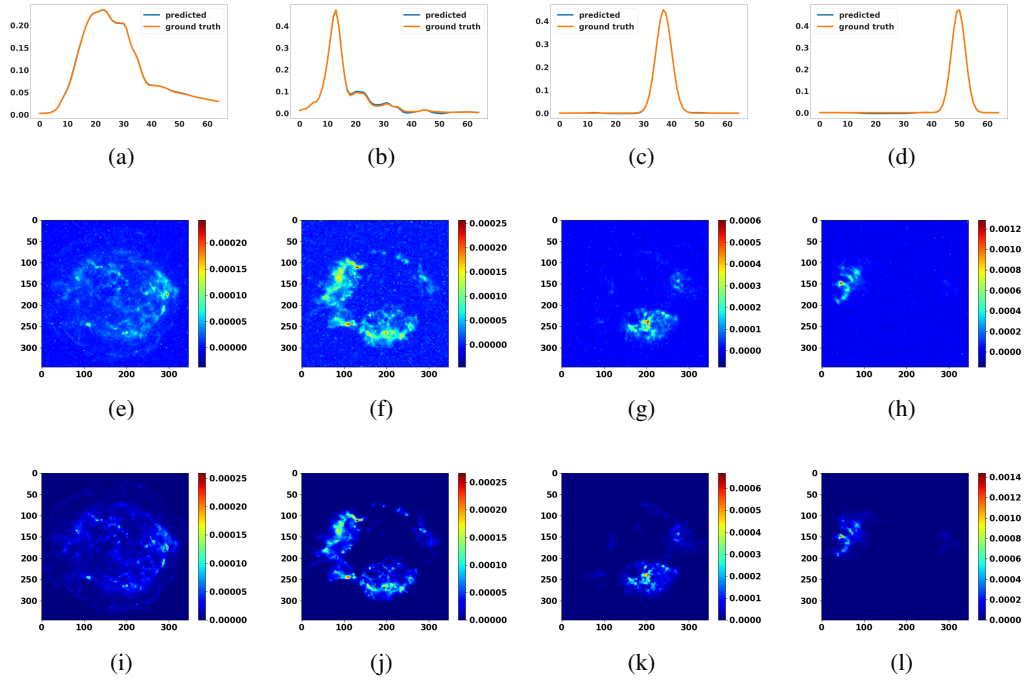


Figure 4: LPALM prediction vs ground truth of: from (a) to (d) the mixing matrix, from (e) to (h) the sources (the ground truth sources are shown from (i) to (l))

Table 2: Comparison of LPALM with state-of-the-art methods in terms of NMSE over A

Algorithm	MNNBU	SNMF-net	DNMF	**LPALM**
NMSE(A_{pred}, A^*)	0.0962	0.2700	0.0466	**0.0002**
Generalization (model evaluation)	0.4785	0.3053	0.5469	**0.0009**
Model retraining	0.3181	0.3073	0.0922	**0.0009** (no retraining)

(as it is not trained on a single image), but it furthermore gives much better results than the other methods needing a retraining.

4 Conclusion

In this article, we introduce LPALM, an unrolled version of the PALM algorithm taylored for solving sparse SBSS problems **in the presence of variabilities of both generative factors** A^* and S^*. This is key to apply unrolled sparse SBSS algorithms to real-world, where the mixing matrix is generally unknown but well constrained by the underlying physical phenomena. More precisely, the chosen unrolling architecture is thoroughly justified by both methodological considerations and experimental comparisons with state-of-the-art methods in the field. In contrast to available unrolled BSS methods that work in a purely unsupervised way without knowledge of the ground truth factors at training, the proposed LPALM algorithm provides significantly **more efficient and robust solutions with respect to the initialization and spurious stationary points** that originate from the non-convexity of the underlying problem. Moreover, we illustrate that LPALM yields very **good generalization performances**, enabling to bypass PALM difficult regularization hyperparameter choice. Altogether, this makes LPALM a good unrolling-based candidate to analyze real-world data. To the best of our knowledge, on the "learning representations" level, this is **the first work using unrolling techniques for astronomical data**, and we believe that it would open much more perspectives to the introduction of such techniques in astronomy. Further research includes the application of LPALM to other domains, such as remote sensing.

REFERENCES

Pierre Ablin, Thomas Moreau, Mathurin Massias, and Alexandre Gramfort. Learning step sizes for unfolded sparse coding. *sign (x)*, 50:2.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of operations research*, 35(2):438–457, 2010.

José M Bioucas-Dias and Mário AT Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pp. 1–4. IEEE, 2010.

José M Bioucas-Dias and Antonio Plaza. An overview on hyperspectral unmixing: Geometrical, statistical, and sparse regression based approaches. In *2011 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1135–1138. IEEE, 2011.

Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.

Jérôme Bobin, Jean-Luc Starck, Jalal Fadili, and Yassir Moudden. Sparsity and morphological diversity in blind source separation. *IEEE Transactions on Image Processing*, 16(11):2662–2674, 2007.

Jerome Bobin, Jeremy Rapin, Anthony Larue, and Jean-Luc Starck. Sparsity and adaptivity for the blind separation of partially correlated sources. *IEEE Transactions on Signal Processing*, 63(5): 1199–1213, 2015.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1):459–494, 2014.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. 2018.

Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Nicolas Dobigeon, Jean-Yves Tourneret, Cédric Richard, José Carlos M Bermudez, Stephen McLaughlin, and Alfred O Hero. Nonlinear unmixing of hyperspectral images: Models and algorithms. *IEEE Signal processing magazine*, 31(1):82–94, 2013.

Nicolas Gillis. *Nonnegative Matrix Factorization*. SIAM, 2020.

Raja Giryes, Yonina C Eldar, Alex M Bronstein, and Guillermo Sapiro. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transactions on Signal Processing*, 66(7):1676–1690, 2018.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

Daniel C Heinz et al. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE transactions on geoscience and remote sensing*, 39(3):529–545, 2001.

Christophe Kervazo, Jérôme Bobin, Cécile Chenot, and Florent Sureau. Use of palm for $\ell 1$ sparse matrix factorization: Difficulty and rationalization of a two-step approach. *Digital Signal Processing*, 97:102611, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jialin Liu and Xiaohan Chen. Alista: Analytic weights are as good as learned weights in lista. In *International Conference on Learning Representations (ICLR)*, 2019.

Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

Thomas Moreau and Joan Bruna. Understanding trainable sparse coding via matrix factorization. 2019.

José MP Nascimento and José M Bioucas Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, 43(4), 2005.

Rami Nasser, Yonina C Eldar, and Roded Sharan. deep unfolding for non-negative matrix factorization with application to mutational signature analysis. *arXiv preprint arXiv:2108.09138*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Adrien Picquenot, Fabio Acero, Jérôme Bobin, Pierre Maggi, Jean Ballet, and Gabriel W Pratt. Novel method for component separation of extended sources in x-ray astronomy. *Astronomy & Astrophysics*, 627:A139, 2019.

Qipeng Qian, Fengchao Xiong, and Jun Zhou. Deep unfolded iterative shrinkage-thresholding model for hyperspectral unmixing. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 2151–2154. IEEE, 2019.

Yuntao Qian, Fengchao Xiong, Qipeng Qian, and Jun Zhou. Spectral mixture model inspired network architectures for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10):7418–7434, 2020.

Pablo Sprechmann, Alexander M Bronstein, and Guillermo Sapiro. Learning efficient sparse and low rank models. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1821–1833, 2015.

Jean-Luc Starck, Fionn Murtagh, and Jalal Fadili. *Sparse Image and Signal Processing - Wavelets, Curvelets, Morphological Diversity*. Cambridge University Press, 2010. ISBN 978-0-521-11913-9.

Zhangyang Wang, Qing Ling, and Thomas Huang. Learning deep l0 encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Fengchao Xiong, Jun Zhou, Shuyin Tao, Jianfeng Lu, and Yuntao Qian. Snmf-net: Learning a deep alternating neural network for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on imaging sciences*, 6(3):1758–1789, 2013.

Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1828–1837, 2018.

Bin Zhao, Li Fei-Fei, and Eric P Xing. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR 2011*, pp. 3313–3320. IEEE, 2011.

Chao Zhou and Miguel RD Rodrigues. An admm based network for hyperspectral unmixing tasks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1870–1874. IEEE, 2021.

Joey Tianyi Zhou, Kai Di, Jiawei Du, Xi Peng, Hao Yang, Sinno Jialin Pan, Ivor Tsang, Yong Liu, Zheng Qin, and Rick Siow Mong Goh. Sc2net: Sparse lstms for sparse coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Michael Zibulevsky and Barak A Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural computation*, 13(4):863–882, 2001.

APPENDIX

A IMPLEMENTATION DETAILS

For the training, Adam optimizer (Kingma & Ba, 2014) is used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For the implementation, Pytorch library (Paszke et al., 2019) is used.
The training is done on 100 epochs with a learning rate LR=0.0001. The batch size is 1. We note that no fancy training is needed (in contrast to some of the other unrolled architectures we compared LPALM to).
Non-Blind Setting: The number of layers is $K = 5$. The parameters are initialized as follows:

- LISTA and LISTA-CP: 1) An A is selected from the training set, 2) L_S is taken as the largest eigenvalue of A^TA , 3) $\theta^{(0)} = \frac{\lambda}{L_S}$, $\lambda = 10^{-5}$, 4) $W_1^{(0)} = \mathbb{I} - \frac{1}{L}A^TA$ (only for LISTA), 5) $W_2^{(0)} = \frac{1}{L}A^T$. $\tilde{S}^{(0)} = (0)^{2\times 500}$.
- ISTA-LLT: 1) $L_S^{(k)}$ is initialized at 1, for all $k \in [0, K-1]$, 2) $\theta^{(k)}$ is initialized at 10^{-3}, for all $k \in [0, K-1]$. $\tilde{S}^{(0)} = (0)^{2\times 500}$.

LPALM: The number of layers is $K = 25$. The parameters are initialized as follows: 1) 2 matrices A and S are selected from the training set, 2) $L_A^{k(0)}$ is taken as the largest eigenvalue of SS^T, for all $k \in [0, K-1]$, 3) $\theta^{k(0)} = \frac{\lambda}{L_S}$, for all $k \in [0, K-1]$, where $\lambda = 10^{-5}$, and L_S is taken as the largest eigenvalue of A^TA, 4) $W^{k(0)} = \frac{1}{L_S}A$, for all $k \in [0, K-1]$, 5) $\tilde{S}^{(0)} = (0)^{n\times t}$, 6) $\tilde{A}^{(0)} = \Pi_{||.||=1}((1)^{m\times n})$.

B DETAILS ABOUT THE ASTROPHYSICAL DATA

In this article, the mixing matrices are derived from astrophysical simulations, described in (Picquenot et al., 2019), which have been derived from real astrophysical data: the Cassiopea A supernovae remnant as observed by the X-ray space telescope Chandra `chandra.harvard.edu`.

SYNTHETIC DATA IN SECTION 3.1

There are 10 000 mixing matrices A^* of size 40×2. The first column corresponds to a specific thermal emission spectrum and the second one to a Gaussian-shaped line emission, whose variabilities across the training set are displayed in figure 5. To mimic mildly sparse sources, 10 000 matrices S^* of size 2×500 are generated using a generalized Gaussian distribution with a shape parameter $\beta = 0.3$ (which is usually a good proxy to mimic the distribution of wavelet coefficients of natural images). The whole data set then consists in 10 000 samples of $X = AS + N \in \mathbb{R}^{40\times 500}$, with N an additive white Gaussian noise such that SNR $= 30$ dB. This data set is split into 8 000 samples constituting the training set and 2 000 for the test set.

SYNTHETIC DATA IN SECTION 3.2

The A^* matrices are composed of 4 different spectra of size 65 each: i) a synchrotron emission spectrum, ii) a thermal emission spectrum that is composed of various emission lines, which variabilities are similar and iii) two line emission spectra that are related to a single atomic component (e.g. iron) but with different redshifts due to the Doppler effect. The columns of the mixing matrices and their variabilities are displayed in Figure 1.

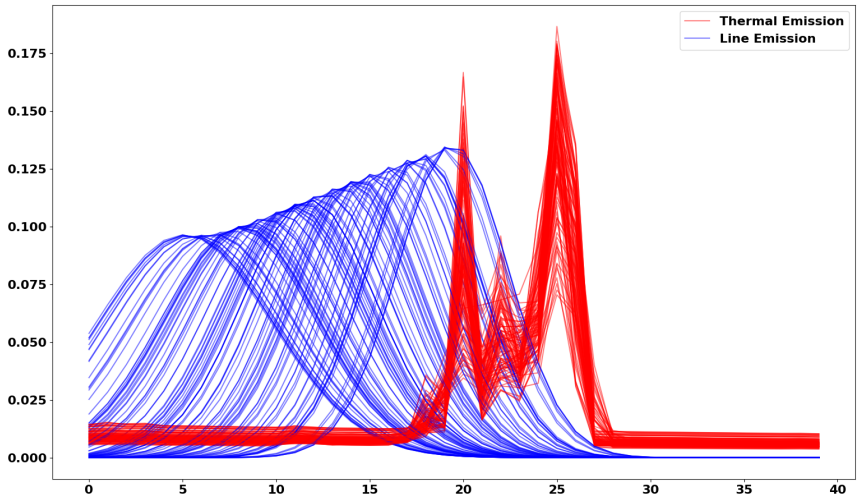


Figure 5: Illustration of the variability of the $A^* \in \mathbb{R}^{40\times 2}$ matrices within the data set of Section 3.1. 100 examples of A^* are depicted, each column being drawn with a different color.

The kind of sources we focused on are remnants of supernovae; unfortunately, only few samples with a good estimate of the "ground truth" sources are actually available. We only have access to a single hyperspectral image of such an astrophysical object, which we use for testing in Section 3.3. We therefore resorted to the same stochastic model for the sources as in Section 3.1: the source coefficients in the wavelet domain are mimicked using a generalized Gaussian distribution with a shape parameter $\beta = 0.3$. The data set is then made of 900 $X \in \mathbb{R}^{65\times 500}$ matrices such that $X = A^* S^* + N$, with N. The training set is constituted of 750 samples of these matrices.

Realistic data in Section 3.3

For the whole data set, the A^* matrices are the same as in Section 3.2. Training and test set however differ concerning the sources:

- **Test set**: In contrast to Section 3.2, we here assess LPALM when S^* is a real single hyperspectral image of a supernovae remnant. Dealing with these data is rather challenging since these sources exhibit partial correlations (pixels where more than one sources are active with a large amplitude), which largely hampers the performances of most BSS methods (Bobin et al., 2015). To limit the impact of multiple active sources in X, we apply a randomization on the pixels of each source. Properly tackling the partial correlation problem using unrolled architectures is left for future research, but we still present LPALM results when such a randomization is not performed in Appendix C.
  In addition, it is customary to pre-process astrophysical data by first representing them in the wavelet domain (isotropic undecimated wavelets Starck et al. (2010)). We denote by Φ the transform from the direct to wavelet domains, Φ^{-1} is its inverse. When a single wavelet scale is computed, the index 0 refers to the so-called coarse scale (i.e. large-scale approximation) and 1 refers to the fine scale. Once the model is trained, it is tested on the realistic mixture X as follows: i) wavelet-based pre-processing: $X \xrightarrow{\Phi} \{{}^0X, {}^1X\}$, ii) application of the learnt model: LPALM$({}^1X) \longrightarrow A_{\text{pred}}, {}^1S_{\text{pred}}$ to derive the estimated wavelet fine scales and the mixing matrix, iii) reconstruction of the coarse scale by applying the pseudo-inverse of the estimated mixing matrix to the coarse scale of the data: ${}^0S_{\text{pred}} \longleftarrow A_{\text{pred}}^{\dagger}\, {}^0X$, iv) inverse wavelet transform: $S_{\text{pred}} \longleftarrow \Phi^{-1}({}^0S_{\text{pred}}, {}^1S_{\text{pred}})$.

- **Train set**: in the absence of a large amount of training data for the sources S^*, we generated synthetic sources. More specifically, we generated the fine scales of the sources according to a generalized Gaussian distribution (GGD) with parameters m, σ, and β (resp. mean, standard deviation and shape parameter). These parameters are fitted to the tp non-zero elements of each true reference source ($t = 3000$ is the total number of pixels and p is the proportion of non-zero elements). Then a realization of tp samples of a GGD with the estimated parameters is simulated, the resulting vector is concatenated with $(1-p)t$ zero elements. Finally, a random shuffling is applied to the vector. The GGD parameters are estimated from the histograms of non-zero elements of each source. This allows to simulate random sources that have approximately the same statistics as the original astrophysical sources in the wavelet domain. Such a simple process already enables to obtain a training set that is decent enough to enable LPALM to have good results in the test phase.

For both training and test sets, the data generation of X was completed by adding a white Gaussian noise N such that SNR $= 30$ dB.

C LPALM SEPARATION RESULTS IN PRESENCE OF PARTIAL CORRELATIONS

The original astrophysical data exhibit partial correlations, which are known to dramatically hamper the performances of most SBSS methods (Bobin et al., 2015). For that purpose, we previously presented results where the sources are randomized, to avoid such partial correlations. Figure 6 shows the results of LPALM when applied to the raw astrophysical data. These results are of reasonable quality but show leakages that directly originate from the presence of these partial correlations. Since the training set used to train LPALM does not contain such partial correlations, it cannot learn an unmixing procedure that is robust to this type of artifacts. An in-depth comparison with other unrolled algorithms in the presence of partial correlations, as well as properly tackling this issue within LPALM, is left for future work.

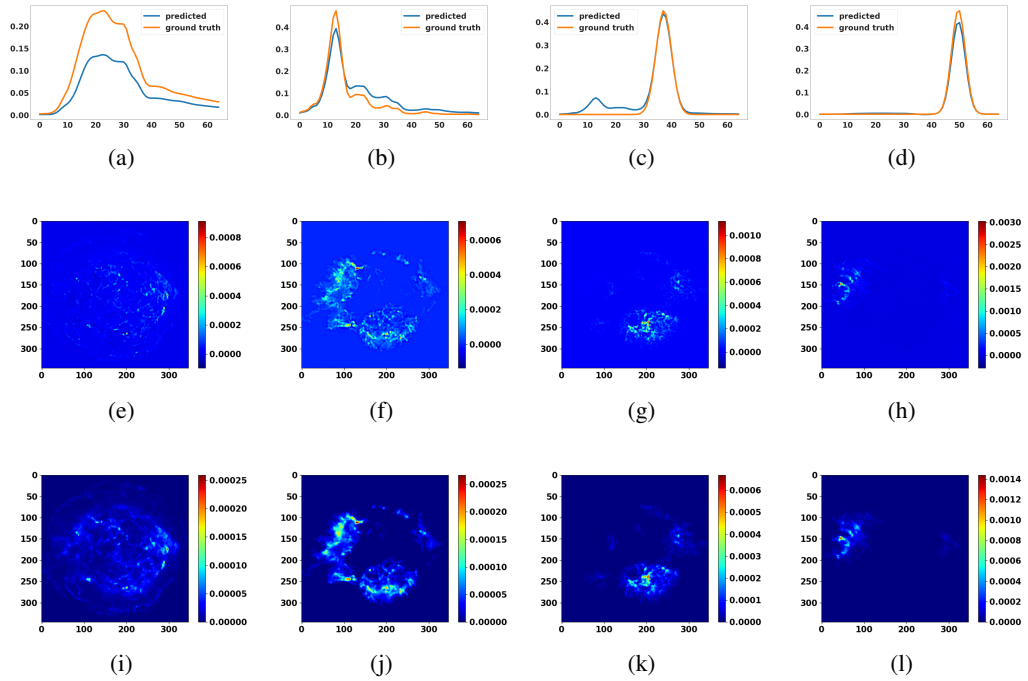


Figure 6: Prediction vs ground truth of: From (a) to (d) the mixing matrix, from (e) to (h) the sources (the ground truth sources are shown from (i) to (l))

D DETAILS ON SECTION 3.3 EXPERIMENTS

All the three methods which we compare with LPALM are trained on a single multi-spectral image X. The training set consists in all the pixels of X, which follows the observation of (Xiong et al., 2021) and (Qian et al., 2020) that the more pixels used, the better the separation quality. Several hyperparameters (learning rate, number of epochs...) were tested in order to have the best performance *w.r.t.* the corresponding loss function.

MNNBU: The VCA (Nascimento & Dias, 2005) and FCLS (Heinz et al., 2001) algorithms are respectively used to initialize the A and S matrices, following the corresponding article recommendations. The initialization is shown in Figure 7. We see that the columns of A^* are well extracted by the algorithm except for the synchrotron emission.
A learning rate schedule is used: lr $= 0.002$ at the first 99 epochs, and it is divided by 1.2 each 50 epochs starting from epoch 100. The total number of epochs for training is 500 epochs.

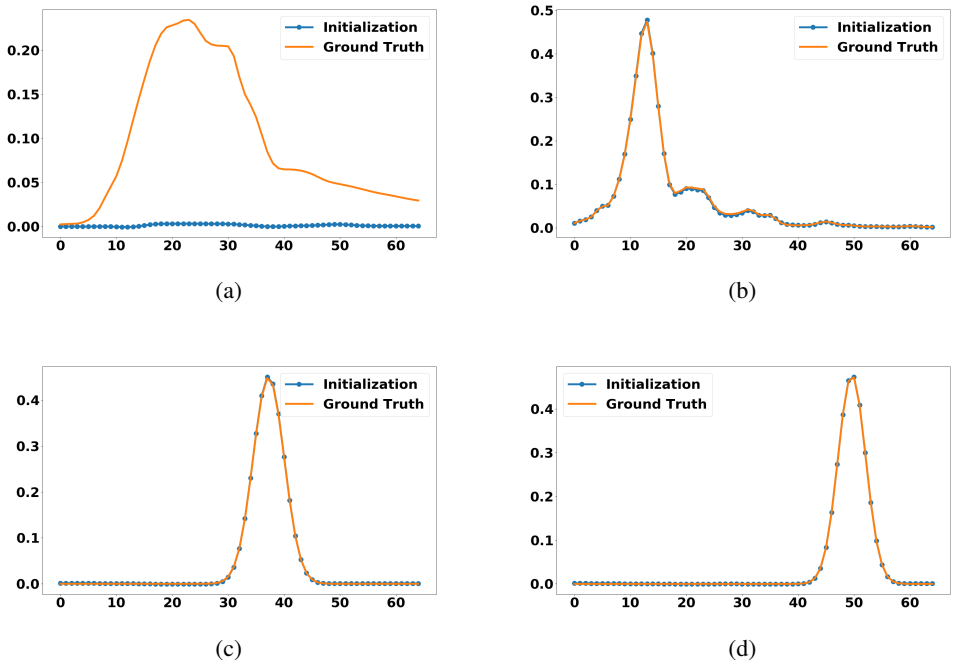


Figure 7: VCA results when applied to the data of Figure 4.

SNMF-net: The VCA (Nascimento & Dias, 2005) and FCLS (Heinz et al., 2001) algorithms are respectively used to initialize the A and S matrices, following the corresponding article recommendations; see Figure 7. The model hyperparameters are kept the same as the original paper code (learning rate schedules, number of layers...), except reducing the initial learning rate which was tuned in order to have the most possible decrease of the loss function.

DNMF: DNMF is initialized with matrices of ones (Nasser et al., 2021). The learning rate is 0.0008, there are 1500 epochs, and we took the best regularization parameters among the ones tested in (Nasser et al., 2021). As such, DNMF might be slightly disadvantaged compared to MNNBU and SNMF-net, since these benefit from a much better initialization (note that however LPALM does not benefit from a better initialization than DNMF).

Computation of the second line of Table 2: The loss of MNNBU is trivially calculated using the ground truth A^* and the predicted matrix A_{pred}, which is a hard-coded weight matrix in the network. For SNMF-net, the weights of the model are applied while initializing for each example with VCA and FCLS. For DNMF, the learned weights are used to predict S from X, then a nonnegative least square algorithm is applied to predict A from X and DNMF output.

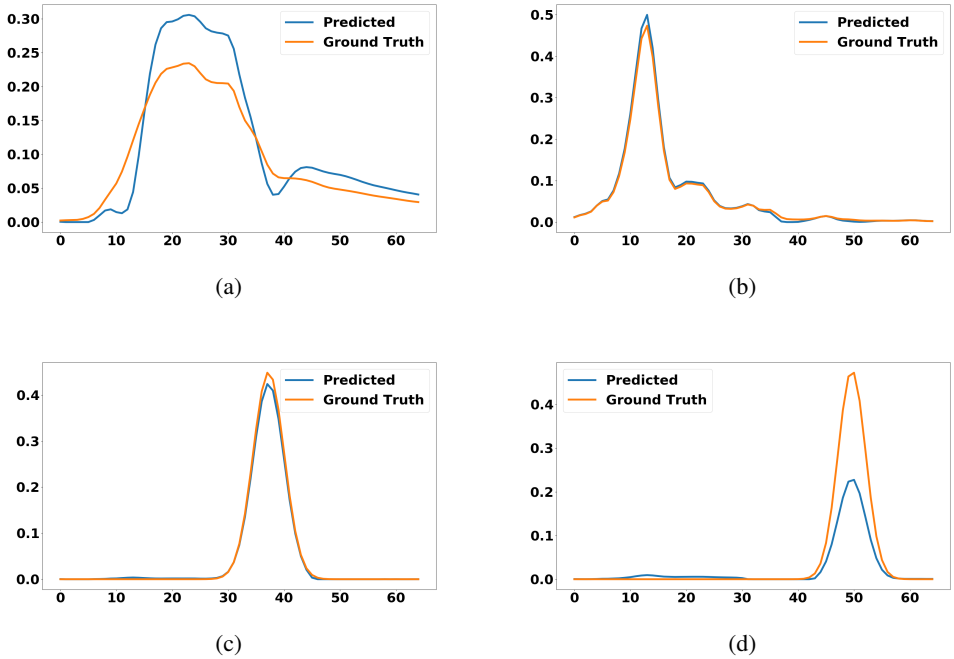


Figure 8: MNNBU predicted spectra from the data set of Subsection 3.3

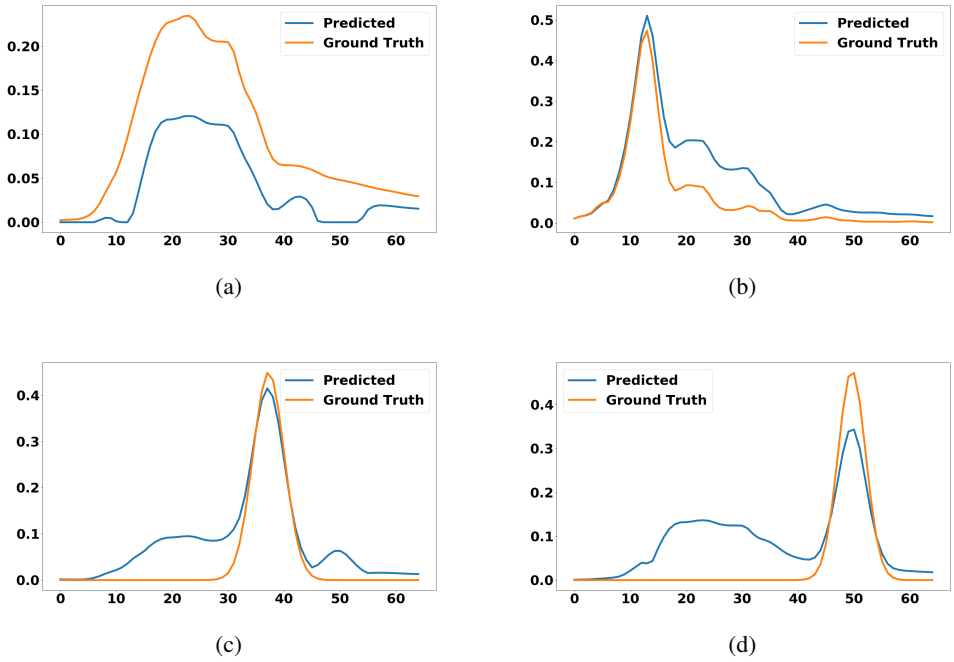


Figure 9: SNMF predicted spectra from the data set of Subsection 3.3

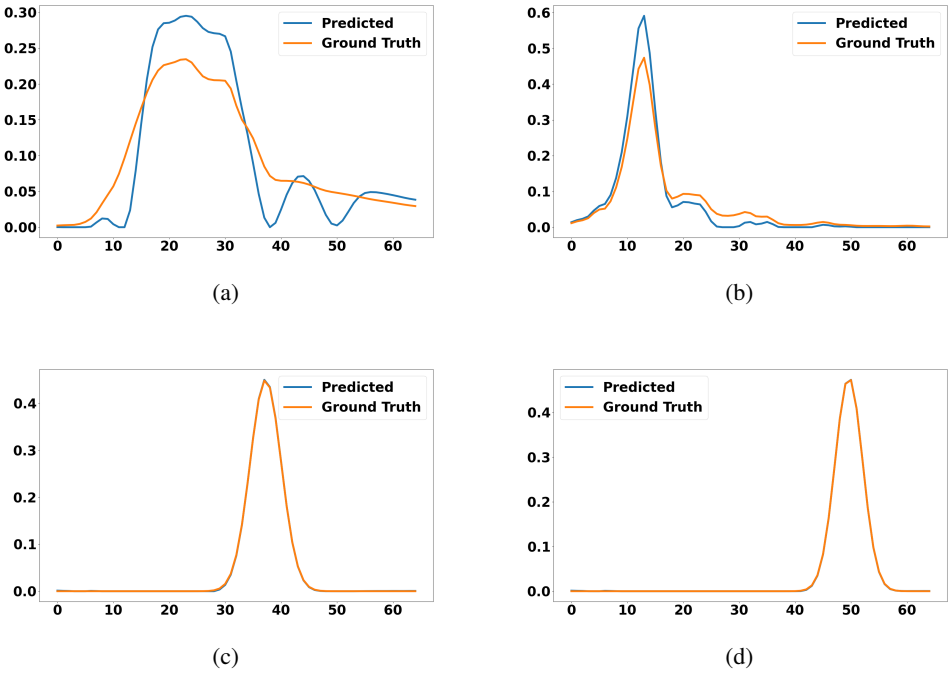


Figure 10: DNMF predicted spectra from the data set of Subsection 3.3