frontiers | Frontiers in Applied Mathematics and Statistics

TYPE Original Research
PUBLISHED 11 December 2023
DOI 10.3389/fams.2023.1205959

OPEN ACCESS

EDITED BY
Holger Rauhut,
RWTH Aachen University, Germany

REVIEWED BY
Junhong Lin,
Zhejiang University, China
Stefan Kunis,
Osnabrück University, Germany

*CORRESPONDENCE
Jan Christian Hauffen
✉ j.hauffen@tu-berlin.de

RECEIVED 14 April 2023
ACCEPTED 17 October 2023
PUBLISHED 11 December 2023

CITATION
Hauffen JC, Jung P and Mücke N (2023) Algorithm unfolding for block-sparse and MMV problems with reduced training overhead. *Front. Appl. Math. Stat.* 9:1205959. doi: 10.3389/fams.2023.1205959

Algorithm unfolding for block-sparse and MMV problems with reduced training overhead

Jan Christian Hauffen[1]*, Peter Jung[1,2] and Nicole Mücke[3]

[1]Electrical Engineering and Computer Science, Communications and Information Theory, Technical University Berlin, Berlin, Germany, [2]Institute of Optical Sensor Systems, German Aerospace Center (DLR), Berlin, Germany, [3]Institute for Mathematical Stochastics, Carl-Friedrich-Gauß-Faculty, Technical University Brunswick, Brunswick, Germany

In this study, we consider algorithm unfolding for the multiple measurement vector (MMV) problem in the case where only few training samples are available. Algorithm unfolding has been shown to empirically speed-up in a data-driven way the convergence of various classical iterative algorithms, but for supervised learning, it is important to achieve this with minimal training data. For this, we consider learned block iterative shrinkage thresholding algorithm (LBISTA) under different training strategies. To approach almost data-free optimization at minimal training overhead, the number of trainable parameters for algorithm unfolding has to be substantially reduced. We therefore explicitly propose a reduced-size network architecture based on the Kronecker structure imposed by the MMV observation model and present the corresponding theory in this context. To ensure proper generalization, we then extend the analytic weight approach by Liu and Chen to LBISTA and the MMV setting. Rigorous theoretical guarantees and convergence results are stated for this case. We show that the network weights can be computed by solving an explicit equation at the reduced MMV dimensions which also admits a closed-form solution. Toward more practical problems, we then considered convolutional observation models and show that the proposed architecture and the analytical weight computation can be further simplified and thus open new directions for convolutional neural networks. Finally, we evaluate the unfolded algorithms in numerical experiments and discuss connections to other sparse recovering algorithms.

KEYWORDS

unfolding, compressed sensing, multiple measurement vector problem, deep learning, block-sparsity

1 Introduction

This study connects the multiple measurement vector (MMV) problem, block- or joint-sparsity and recent results of deep unfolding of the iterative shrinkage thresholding algorithm (ISTA) to reconstruct unknown joint-sparse vectors from given linear observations. Such vectors could be, for example, signals received at the different antennas in a wireless communication problem or, in a computational imaging setup, discrete images observed at different detectors or aggregation stages. Compressed sensing is a way to reconstruct compressive measurements from their underdetermined systems and first theoretical breakthroughs were achieved by Candés et al. [1] and Donoho [2], leading to an approach where fewer samples can be used, than stated within the Nyquist–Shannon sampling theorem [3]. They were able to show that unknown vectors can be reconstructed using convex optimization if the linear mapping fulfilled certain assumptions [1, 4]. These

idea rely on minimizing ℓ_1-norm to promote sparsity and the approach of basis pursuit [5]. These convex optimization problems could then be solved with iterative algorithms, in Figueiredo et al. [6] gradient projection approaches are presented and in Fornasier and Rauhut [7] the idea of thresholding algorithms, which will be also discussed in this study. Although this is already a well researched field, in practice leading to high computational effort because of many iterations and large underlying systems and is thus not suitable for real-world applications. Thus, Karol Gregor and Yann LeCun proposed to use the iterative structure of these algorithms for a neural network and to train each iteration step [8], which is also referred to as deep unfolding and will be discussed in Section 2. Convergence for deep unfolding of the iterative thresholding algorithm has also been studied by Chen et al. [9]. Liu and Chen [10] proposed in to exclude the weight matrix from the data-driven optimization approach and pre-compute this by data-free optimization and thus presented Analytical LISTA (ALISTA). In recent results, Chen et al. could reduce the training procedure even more, by showing that only tuning three hyperparameters is sufficient, proposing HyperLISTA [11]. Creating large sets of training data is often difficult in practice, and thus, it is important to reduce the trainable parameters. Therefore, we will extend the already stated concepts for the block-sparse setting and especially for the multiple measurement vector problem, developing suitable learned algorithms, with only a few trainable parameters and similar theoretical guarantees as ALISTA.

1.1 Multiple measurement vector problems

Multiple measurement vector (MMV) problems occur in many applications, for example, in tomography [12], communication [13], blurred image reconstruction, or superresolution [14]. In the following, we derive the connection to block-sparsity. In an MMV problem, we assume that we derived $d \in \mathbb{N}$ measurements $y^l \in \mathbb{R}^m$, $l = 1, \dots, d$ from d sparse signal vectors $x^l \in \mathbb{R}^n$ sharing the same support $supp(x^l) = \{i : |x_i^l| \neq 0\}$, which is referred to as joint-sparsity [15, 16]. The MMV problem can then be presented as solving the following d equations

$$y^l = Kx^l + \tilde{\epsilon} \tag{1}$$

for $l = 1, \dots, d$ and with $K \in \mathbb{R}^{m \times n}$. This can be rewritten in the following matrix equation form

$$Y = KX + \tilde{E}, \tag{2}$$

where $X = (x^1, \dots, x^d) \in \mathbb{R}^{n \times d}$, $Y = (y^1, \dots, y^d) \in \mathbb{R}^{m \times d}$. With the vectorizing operator $\mathrm{vec}(\cdot)$, stacking each column of a matrix on each other, we can cast Equation (2) into a block-sparse problem. We have

$$\begin{aligned} & Y = KX \\ \Leftrightarrow\ & Y^T = X^T K^T \\ \Leftrightarrow\ & \mathrm{vec}(Y^T) = (K \otimes I_d)\mathrm{vec}(X^T), \end{aligned}$$

where we used the well-known vectorization property of matrix equations, see for example Schacke [17]. Here, $\otimes$ is the Kronecker product. The vector $x = \mathrm{vec}(X^T)$ is block-sparse with n blocks of length d, if the signals x^l are jointly sparse. With $D = (K \otimes I_d) \in \mathbb{R}^{n_y \times n_x}$, where $n_y = m \cdot d$ and $n_x = n \cdot d$, we obtain the block-sparse setting considered in this work.

1.2 Block sparsity

In the more general setting, we want to reconstruct an unknown vector $x \in \mathbb{R}^{n_x}$ from a given matrix $D \in \mathbb{R}^{n_y \times n_x}$ and given $y \in \mathbb{R}^{n_y}$

$$y = Dx + \epsilon\,, \tag{3}$$

with $n_x = nd, n_y = md$ for some $n, m, d \in \mathbb{N}$. We assume that noise $\epsilon \in \mathbb{R}^{n_y}$ is added to Dx. In applications, we often have this problem is ill-posed, i.e., $n_y < n_x$ or D is not invertible. We assume that x is the concatenation of n "smaller" vectors of length d, called blocks, i.e., $x[i] \in \mathbb{R}^d$,

$$x^T = [\underbrace{x_1 \dots x_d}_{=x[1]} \underbrace{x_{d+1} \dots x_{2d}}_{=x[2]} \dots \underbrace{x_{n_x-d+1} \dots x_{n_x}}_{=x[n]}]^T\,. \tag{4}$$

Following the notation in Yonina and Eldar [18], we define

$$\|x\|_{2,0} = \sum_{i=1}^{n} I(\|x[i]\|_2 > 0)\,,$$

where $I(\|x[i]\|_2 > 0) = 1$ if $\|x[i]\|_2 > 0$ and equal to zero otherwise and the ℓ_2-norm is defined as $\|x\|_2^2 = \sum_{i=1}^n |x_i|^2$. We call $x \in \mathbb{R}^{n_x}$ s-block-sparse if $\|x\|_{2,0} \leq s$. Similar to Equation (4), we can construct the matrix D in Equation (3) from n "smaller" matrices $D[i] \in \mathbb{R}^{n_y \times d}, i = 1, \dots, n$

$$D = (D[1]\, D[2]\, \dots\, D[n])\,.$$

Without loss of generality, we can assume that these blocks are orthonormal, see Yonina and Eldar [18], i.e., $D[i]^T D[i] = I_d$, where I_d is the $d \times d$ identity matrix. This assumption simplifies the presentation of several statements below.

Definition 1. The block-coherence of a matrix $D \in \mathbb{R}^{n_y \times n_x}$ is defined as

$$\mu_b(D) = \max_{i \neq j} \frac{1}{d} \|D[i]^T D[j]\|_2\,. \tag{5}$$

Here, we use $\|A\|_2 = \sqrt{\lambda_{max}(A^T A)}$, where λ_{max} denotes the largest eigenvalue of matrix $A^T A$. Note that the block coherence can also be introduced with a normalization factor $1/\left(\|D[i]\|_2 \|D[j]\|_2\right)$, but since we assume orthonormal blocks, we can neglect this. This reduces to the already known coherence μ for $d = 1$

$$\mu(D) = \max_{i \neq j} |D_{:,i}^T D_{:,j}|\,,$$

where $D_{:,i}$ denotes the ith column of D, see Donoho et al. [19]. In Yonina and Eldar [18], it is shown that $0 \leq \mu_b(D) \leq \mu(D) \leq 1$, and it is possible to derive recovery statements for small μ_b similar to μ, for details see Yonina and Eldar [18]. We can consider also the cross block coherence, which compares two matrices and will be important in the following.

Definition 2. For $B, D \in \mathbb{R}^{n_y \times n_x}$ with $B[i]^T D[i] = I_d$ the cross block coherence is defined as

$$\mu_b(B, D) = \max_{i \neq j} \frac{1}{d} \|B[i]^T D[j]\|_2 . \tag{6}$$

Similar to ordinary basis pursuit and LASSO [5, 20–22], we use the following $\ell_{2,1}$-LASSO to solve Equation (3)

$$\min_{x \in \mathbb{R}^{n_x}} \frac{1}{2} \|Dx - y\|_2^2 + \alpha \|x\|_{2,1} \tag{7}$$

where $\|x\|_{2,1} = \sum_{i=1}^{n} \|x[i]\|_2$ is the $\ell_{2,1}$-norm of x, which will promote block-sparsity for the solution of Equation (7). This convex programm can be solved by the fixed point iteration known as the block iterative shrinkage thresholding algroithm (Block-ISTA/BISTA):

$$x^{(k)} = \eta_{\alpha\gamma} \left(x^{(k-1)} - \gamma \left[D^T \left(D x^{(k-1)} - y \right) \right] \right) , \tag{8}$$

where η_α is the block-soft-thresholding operator, given as

$$\eta_\alpha (x)[i] = \max \left\{ 0, 1 - \frac{\alpha}{\|x[i]\|_2} \right\} x[i]. \tag{9}$$

BISTA is an already well-studied proximal gradient algorithm based on the functional in Equation (7). It is known that the fixed point iteration in Equation (8) converges for $\gamma \in (0, \frac{1}{L})$, where $L = \|D\|_2^2$ is the Lipschitz constant of the least squares term in Equation (7) w.r.t to x, to a solution of Equation (7), if at least one solution exists, see for example Byrne [23], Bauschke et al. [24], and Beck [25]. On the other hand, the choice of the regularization parameter α has to be done empirically and is very crucial for a "good" recovery. If α is set too large, this can lead to too much damping, possibly setting blocks to 0 that actually have a non-zero norm. If α is too small, we get reverse effects. In practice, this leads to problems since computing the iterations require high computational effort. Deep unfolding is a way to tackle these problems, i.e., reduce the number of iterations by learning optimal regularization parameters and step-sizes. There are already classical concepts in increasing the convergence speed by using an additional step in updating the current iterate $x^{(k)}$, by using the previous $x^{(k-1)}$, resulting in Block Fast ISTA [25, 26]. On the other hand, the choice of optimal parameters is still solved empirically.

2 Deep unfolding and learned BISTA

Recently, the idea of deep unfolding has been developed, where the goal is to optimize these parameters of such an iterative algorithm [8, 10, 27]. This in turn gives us an iterative algorithm with optimal chosen step-size γ and regularization parameters, but we will see that we do not have to restrict our self only to those parameters. Recently, Fu et al. proposed Ada-BlockLISTA by applying deep unfolding to block-sparse recovery [28]. They show an increase in the convergence speed with numerical examples but do not cover theoretical studies.

In the following, we are going to present the idea of deep unfolding, then we are going to derive Learned BISTA (LBISTA).

2.1 Deep unfolding

We will now formalize the concept of deep unfolding for an arbitrary operator that depends on a certain set of parameters, before we apply this to the previously presented fixed point iteration. To this end, we define an operator

$$T(\cdot\,;\, \theta,\, y) : X \to X \tag{10}$$

which depends on a set of parameters $\theta \in \Theta$, such as the stepsize of a gradient descent operator and an input y. For example, for BISTA, this would be $\theta = (\alpha, \gamma)$ with

$$T(x\,;\, \theta,\, y) = \eta_{\alpha\gamma} \left(x - \gamma \left[D^T \left(Dx - y \right) \right] \right) .$$

We assume that $\mathrm{Fix}(T(\cdot\,;\, \theta,\, y)) \neq \emptyset$ and that we have convergence for the fixed point iteration

$$T^k \left(x^{(0)}\,;\, \theta,\, y \right) = x^{(k)} \tag{11}$$

for an arbitrary $x^{(0)} \in X$. Deep unfolding now interprets each iteration step as the layer of a neural network and uses the parameters $\theta \in \Theta$ as trainable variables. In more detail, this means we look at the Kth iteration of Equation (11), i.e., the composition

$$\underbrace{(T \circ \cdots \circ T)}_{K \text{ times}}(x^{(0)}\,;\, \theta,\, y) = x^{(K)}$$

for some $K \in \mathbb{N}$. By unfolding this iterative scheme, we define

$$T_{\theta^{(k-1)}} (\cdot\,;\, y) \coloneqq T \left(\cdot\,;\, \theta^{(k)}, y \right)$$

for $k = 1, \ldots, K$ and set $\theta^{(k)}$ as the set of trainable variables in this operator, so they can vary in each iteration step. For example, in LBISTA, we will get $\theta^{(k)} = (\alpha^{(k)}, \gamma^{(k)})$ in the later called tied case. With this we get the following composition

$$\left(T_{\theta^{(K-1)}} \circ \cdots \circ T_{\theta^{(0)}} \right) \left(x^{(0)}\,;\, y \right) = x^{(K)} . \tag{12}$$

and define the operator

$$\mathcal{T}_{\tilde{\theta}} \coloneqq T_{\theta^{(K-1)}} \circ \cdots \circ T_{\theta^{(0)}}, \tag{13}$$

where we get the full parameter space $\tilde{\Theta} = \Theta \times \cdots \times \Theta$, with trainable variables $\tilde{\theta} = \bigcup_{i=1}^{K} \theta^{(i)}$. $\mathcal{T}_{\tilde{\Theta}}$ will then be the neural network which will be trained with respect to $\tilde{\theta}$. So in the end, after training, we have an iterative algorithm with fixed but optimized parameters. It seems thus that deep unfolding can be applied to any iterative algorithm and help us to estimate the best choice of parameters, but we will only present deep unfolding for BISTA and consider deep unfolding for arbitrary operators in future studies.

2.2 Learning

In this section, we give an overview for the training procedure used in this study. The general idea of supervised learning is to choose model parameters such that the predictions are close, in

for $k \leq K$ **do**
  while $\neg(NMSE(x^{(k)}, x^*_{i,\text{validation}}) < tol \text{ for 5,000 Iterations})$ **do**
    $Adam\left(t_r, l(\hat{x}^{(k)} - x^*_{i,\texttt{train}}), \theta^{(k-1)}\right)$
    with $x^{(k)} = \left(T_{\theta^{k-1}} \circ \cdots \circ T_{\theta^0}\right)(x_0; y_{i,\texttt{train}})$ for all $i = 1, \ldots, n_{train}$

Algorithm 1. Training.

some sense, to the unknown target, i.e., in our case, the unknown vector x in Equation (3) generating the measurement y. Hence, we aim to minimize an expected loss over an unknown distribution $\mathcal{D}$:

$$\min_{\theta}\left[R(\theta) := \mathbb{E}_{x^*\sim\mathcal{D}}\left[\ell(\hat{x} - x^*)\right]\right] \tag{14}$$

where $\ell(\cdot)$ is a given loss function, $\hat{x} = \mathcal{T}_{\tilde{\theta}}\left(x^{(0)}; y\right)$ is the output of the model, and x^* is the ground-truth. Here, we will use the squared ℓ_2-loss $\ell(x) = \frac{1}{2}\|x\|_2^2$. The objective functional $R(\theta) = \mathbb{E}_{x^*\sim\mathcal{D}}\left[\ell(\hat{x} - x^*)\right]$ is also called risk of the model. Since the underlying distribution $\mathcal{D}$ is unknown, we take a batch of S independently drawn samples of input and output data (x_j^*, y_j) for $j = 1, \ldots, n_{train}$ according to Equation (3) and minimize instead data-driven the empirical risk

$$R_S(\theta) = \frac{1}{n_{train}} \sum_{j=1}^{n_{train}} \ell(\hat{x}_j - x_j^*). \tag{15}$$

Proceeding in this way for all layer at once is sometimes referred as end-to-end learning. Because of the special structure of our deep unfolding models, and inspired by Musa et al. [29], we instead train the network layer-wise, by optimizing only $\theta_{case}^{(k-1)}$ for layer k yielding the following training procedure: Let $k \in \{1, \ldots, K\}$,

$$\min_{\theta_{case}^{(k-1)}} \mathbb{E}_{x^*\sim\mathcal{D}}\left[\ell(\hat{x}^{(k)} - x^*)\right],$$

where $\hat{x}^{(k)}$ is the output of the kth layer. We realized this training as follows, we generate a validation set $\left(x^*_{i,\text{validation}}, y_{i,\text{validation}}\right)$, used to evaluate the model while training and a training set $\left(x^*_{i,\text{train}}, y_{i,\text{train}}\right)$, $i = 1, \ldots, n_{train}$, used to calculate (15). This objective is locally minimized by gradient descent methods. As a stopping criteria, we evaluate the normalized mean square error, defined as

$$NMSE(x, \hat{x}^{(k)}) = \frac{\|\hat{x}^{(k)} - x^*\|_2^2}{\|x^*\|_2^2},$$

depending on the validation set, and stop if the maximum of all evaluated *NMSE* stays the same for a given number of iterations. See Algorithm 1, where *Adam* is the ADAM Optimizer [30] depending on an training rate t_r and the functional which should be minimized with respect to given variables, here the loss function $\ell(\cdot)$ with respect to θ^{k-1}.

2.3 Learned BISTA

In the following, we present four different unfolding techniques for BISTA. We present a tied (weights are shared between different layers) and untied (individual weights per layer) case, which refers to different training approaches

$$S = I - B^T D$$
$$B = \gamma D.$$

Tied LBISTA: The idea of LBISTA is now to fix the matrices S and B for all layers but also include them in our set of trainable variables:

$$x^{(k)} = \eta_{\alpha^{(k-1)}}\left(Sx^{(k-1)} + B^T y\right). \tag{16}$$

For LBISTA (Equation 16), we get trainable variables

$$\theta = \left(\left(\alpha^{(k)}\right)_{k=0}^{K-1}, S, B\right),$$

where we initialize $S = I - B^T D$ and $B = \gamma D$. Algorithm (16) is also referred to as vanilla LISTA in the sparse case. Inspired by the LISTA-CP model, i.e., LISTA with coupled parameters, proposed in Liu and Chen [10], we will also consider LBISTA-CP

$$x^{(k)} = \eta_{\alpha^{(k-1)}}\left(x^{(k-1)} - \gamma^{(k-1)} B^T\left(Dx^{(k-1)} - y\right)\right). \tag{17}$$

For LBISTA-CP (Equation 17), we get

$$\theta = \left(\left(\alpha^{(k)}\right)_{k=0}^{K-1}, \left(\gamma^{(k)}\right)_{k=0}^{K-1}, B\right),$$

where we initialize $B = D$.

Untied LBISTA: The idea of untied LBISTA is then to use in each layer different matrices S and B to train, i.e.,

$$x^{(k)} = \eta_{\alpha^{(k-1)}}\left(S^{(k-1)}x^{(k-1)} + \left(B^{(k-1)}\right)^T y\right). \tag{18}$$

For LBISTA (untied) Equation (18), we get trainable variables

$$\theta = \left(\left(\alpha^{(k)}\right)_{k=0}^{K-1}, \left(S^{(k)}\right)_{k=0}^{K-1}, \left(B^{(k)}\right)_{k=0}^{K-1}\right),$$

where we initialize $S^{(k)} = I - B^T D$ and $B^{(k)} = \gamma D$ for every $k = 0, \ldots, K-1$. Inspired by Algorithm (17), we will also consider Algorithm (19), which will be referred to as LBISTA-CP (untied), [9, 10]:

$$x^{(k)} = \eta_{\alpha^{(k-1)}}\left(x^{(k-1)} - \gamma^{(k-1)}\left(B^{(k-1)}\right)^T\left(Dx^{(k-1)} - y\right)\right). \tag{19}$$

For LBISTA-CP (untied) (Equation 19), we get

$$\theta = \left(\left(\alpha^{(k)}\right)_{k=0}^{K-1}, \left(B^{(k)}\right)_{k=0}^{K-1}\right),$$

where we initialize $B^{(k)} = D$ for every $k = 0, \ldots, K-1$. Hence, compared to $\mathcal{O}(n_y n_x + K)$ parameters in the tied case, now more training data and longer training time is required to train now $\mathcal{O}(K n_y n_x + K)$ parameters.

We initialize the trainable variables with values from original BISTA. In Chen et al. [9], it has been shown that convergence of LISTA-CP (untied) can be guaranteed if the matrices $B^{(k)}$ belong to a certain set and their proof can be extended to block-sparsity. The steps are very similar to the convergence proof for learned block analytical ISTA given in the next section.

3 Analytical LBISTA

In the previous section, we presented several approaches for learned BISTA where optimal weights are optimized in a data-driven fashion. In Liu and Chen [10] instead proposed to analytically pre-compute the weights and only train step-size and threshold parameters. It turns out that this Analytic LISTA (ALISTA) with the so called analytical weight matrix is as good as the learned weights. In the following, we are going to extend and improving the theoretical statements for ALISTA to the block-sparse case and propose analytical LBISTA. In contrast to Liu and Chen [10], we will provide a direct solution and also show different ways to calculate the analytical weight matrix in different settings.

3.1 Upper and lower bound

This part of the study will focus on combining and extending several theoretical statements from Chen et al. [9] and Liu and Chen [10] and applying these for the block-sparse case. With the following two theorems, we are then going to present analytical LBISTA, by showing that this is as good as LBISTA-CP (untied) with a pre-computed B.

3.1.1 Upper bound

In this section, we start with an upper bound for the error of the approximation generated by Equation (19), i.e., LBISTA-CP (untied), and the exact solution x^* for given parameters. For this, we modify Assumption 1 from Liu and Chen [10] to be consistent with the block-sparse setting.

Assumption 1. We assume $(x, \epsilon) \in \mathcal{X}_b(M, s, \sigma)$ with

$$\begin{aligned}(x,\epsilon) &\in \mathcal{X}_b(M,s,\sigma)\\ &= \{x : \|x[i]\|_2 \le M \,\forall i = 1,\ldots,n, \quad \|x\|_{2,0} \le s, \|\epsilon\|_2 \le \sigma\}.\end{aligned} \tag{20}$$

As already mentioned, a matrix with small block-coherence has good recovery conditions. In Liu and Chen [10] proposed analytical LISTA, where the pre-computed matrix B is minimizing the mutual cross-coherence. This motivates the following definition.

Definition 3. With $D \in \mathbb{R}^{n_y \times n_x}$, we define the generalized mutual block-coherence

$$\begin{aligned}\tilde{\mu}_b(D) &= \inf_{\substack{B\in\mathbb{R}^{n_y\times n_x}\\ B^T[l]D[l]=I_d\\ \forall 1\le l\le n}} \left\{ \max_{\substack{i\neq j\\ 1\le i,j\le n}} \frac{1}{d}\|B^T[i]D[j]\|_2 \right\}\\ &= \inf_{\substack{B\in\mathbb{R}^{n_y\times n_x}\\ B^T[l]D[l]=I_d\\ \forall 1\le l\le n}} \{\mu_b(B,D)\}\,.\end{aligned} \tag{21}$$

We define, analogously to Liu and Chen [10] with $\mathcal{W}_b(D)$ the set of all $B \in \mathbb{R}^{n_y \times n_x}$ which attain the infimum in Equation (21), i.e., $\mathcal{W}_b(D) = \left\{B \in \mathbb{R}^{n_y \times n_x} : \mu_b(B, D) = \tilde{\mu}_b(D)\right\}$.

Note that the set $\mathcal{W}_b(D)$ is non-empty because the set of feasible matrices $\{B \in \mathbb{R}^{n_y \times n_x} : B^T[i]D[i] = I_d, 1 \le i \le n\}$ contains at least D because we assume $D^T[i]D[i] = I_d$, also $0 \le \tilde{\mu}_b(D) \le \mu_b(D)$ and therefore, Equation (21) is a feasible and bounded program, see Supplementary material to [9]. We will call matrices from $\mathcal{W}_b(D)$ analytical weight matrices.

Definition 4. The block-support of a block-sparse-vector $x \in \mathcal{X}_B(b, s)$ is defined as

$$\mathrm{supp}_b(x) = \{i : \|x[i]\|_2 \neq 0 \,\forall i = 1, \ldots n\}\,. \tag{22}$$

We will now derive an upper bound for the ℓ_2-error and thus showing convergence of LBISTA-CP for a special matrix B and given parameters $\alpha^{(k)}$ and $\gamma^{(k)}$. In [9], Liu et al. showed linear convergence for unfolded ISTA with additional noise, more precisely for LISTA-CP (untied), if the matrices $B^{(k)}$ belonged to a certain set. In Liu and Chen [10], it was shown that we can pre-compute such a matrix B, chose $B^{(k)} = B$, chosen by a data-free optimization problem and still have the same performance. For this, new proposed unfolded algorithm linear convergence was also shown, without additional noise. In Liu and Chen [10], convergence only in the noiseless case was shown, but the results derived in Chen et al. [9] were derived with bounded noise. Thus, we are going to combine these two proofs and extend it to block-sparsity:

For given (x, ϵ), $y = Dx + \epsilon$ and parameters $\{\theta^{(k)}\}_{k=1}^K$, we abbreviate with $\{x^{(k)}(x, \epsilon)\}_{k=1}^K$ the sequence generated by (19) with $x^{(0)} = 0$. Further, we define

$$\begin{aligned}C_{\mathcal{X}}^{(k)} &= \sup_{(x,\epsilon)\in\mathcal{X}_b(M,s,\sigma)} \{\|x^{(k)}(x,\epsilon) - x\|_{2,1}\}\\ C &= \sup_k \max_{j=1,\ldots,n} |\gamma^{(k)}| \|B[j]\|_2\end{aligned} \tag{23}$$

Theorem 1. For any $B \in \mathcal{W}_b(D)$ and any sequence $\gamma^{(k)} \in \left(0, \frac{2}{\mu(2s-1)+1}\right)$ and parameters $\left(B^{(k)} := B, \alpha^{(k)}, \gamma^{(k)}\right)$, for $k \le K$, with

$$\frac{\alpha^{(k)} - C\sigma}{\gamma^{(k)}\mu(D)C_{\mathcal{X}}^{(k)}} \in [1, \kappa] \tag{24}$$

for some $\kappa \geq 1$, with $\mu = d\tilde{\mu}_b(D)$. With $M > 0$ and $s < (\mu^{-1}+1)/2$, we have

$$\operatorname{supp}_b\left(x^{(k)}(x^*)\right) \subset \operatorname{supp}_b\left(x^*\right) =: \mathbb{S}\,, \tag{25}$$

$$\|x^{(k)} - x^*\|_2 \leq \exp\left(-\sum_{\tau=0}^{k-1}\tilde{a}(\tau)\right) sM \tag{26}$$
$$+ C\sigma\left(1 + \sum_{\tau=0}^{k-1}\exp\left(-\sum_{s=\tau}^{k-\tau}\tilde{a}(s)\right)\right).$$

where

$$\tilde{a}(\tau) = -\log\left(\gamma^{(\tau)}\mu\left((\kappa+1)s-1\right) + |1-\gamma^{(\tau)}|\right) > 0\,.$$

Note that in the noiseless case and with $\kappa = 1$, we obtain the results in Liu and Chen [10]. However, from the latter theorem, it follows that one can relax this condition to $\kappa > 1$. We also see from the proof of Theorem 1 that one needs at least $\alpha^{(k)} \geq d\gamma^{(k)}\tilde{\mu}_b(D)C_{\mathcal{X}}^{k()} + C\sigma$ to have Equation (25). On the other hand, one can always find such a κ from the trained (and then fixed) parameters and thus use therefore the theorem afterward. Obviously, a worse α effects the upper bound of the ℓ_2-error and thus appears in $\tilde{a}(\tau)$. Summarizing, above theorem shows now convergence on the training set even if $\kappa \neq 1$.

3.1.2 Lower bound

This section states the lower bound for the $\ell_{2,1}$-error, showing that for convergence in the $\ell_{2,1}$-norm the defined parameters in Theorem 1 are optimally chosen. We now modify Assumption 2 from Liu and Chen [10] to be consistent with the block-sparse setting.

Assumption 2. x^* is sampled from P_X. P_X satisfies: $2 \leq \mathbb{S} \leq s$ and $\mathbb{S}$ is uniformly distributed over the whole index set. The non-zero blocks of x^* satisfy the uniform distribution and $\|x[i]\|_2 \leq M$ for all $i \in \mathbb{S}$. And we assume, $\epsilon = 0$.

The latter theorem states that the analytical weight matrix should minimize the generalized mutual block coherence. Therefore, for a lower bound, we will only consider matrices that are bounded away from the identity.

Definition 5. With $D \in \mathbb{R}^{n_y \times n_x}$, $s \leq 2$, $\bar{\sigma}_{\min} > 0$ we set

$$\bar{\mathcal{W}}(D, s, \bar{\sigma}_{\min}) := \{B \in \mathbb{R}^{n_y \times n_x} : \sigma_{\min}\left(I - \left(B[(j \in \mathbb{S})]\right)^T D[(j \in \mathbb{S})]\right) \geq \bar{\sigma}_{\min}\}\,. \tag{27}$$

The parameters are chosen from the following set.

Definition 6. Let $\{x^{(k)}\}_{k=1}^{\infty}$ be generated by $x^{(k+1)} = \eta_{\alpha^{(k)}}\left(x^{(k)} - (B^{(k)})^T(Dx^{(k)} - y)\right)$ with parameters $\{B^{(k)}, \alpha^{(k)}\}_{k=0}^{\infty}$ and $x^{(0)} = 0$. We define the set of all parameters guaranteeing no false positive blocks in $x^{(k)}$ by

$$\mathcal{T} = \{\{B^{(k)} \in \bar{\mathcal{W}}(D, s, \bar{\sigma}_{\min}), \theta^{(k)}\}_{k=0}^{\infty} : \operatorname{supp}_B(x) \subset \mathbb{S},\ \forall x^* \in \mathcal{X}_b(M, s, 0), \forall k\}\,. \tag{28}$$

This set is non-empty since Equation (25) holds true if $\alpha^{(k)}$ are chosen large enough. Following mainly the proof in Liu and Chen [10] by extending the setting from sparsity to block-sparsity, the lower bound for the $\ell_{2,1}$-norm can be stated as follows.

Theorem 2. Let $\{x^{(k)}\}_{k=1}^{\infty}$ be generated by $x^{(k+1)} = \eta_{\alpha^{(k)}}\left(x^{(k)} - (B^{(k)})^T(Dx^{(k)} - y)\right)$. Under Assumptions 2, $\{W^{(k)} \in \bar{\mathcal{W}}(D, s, \bar{\sigma}_{\min}), \theta^{(k)}\}_{k=0}^{\infty} \in \mathcal{T}$ and $\epsilon > 0$, we have

$$\|x^{(k)} - x^*\|_{2,1} \geq \epsilon\|x^*\|_2 \exp(-ck)\,, \tag{29}$$

with probability $1 - \epsilon s^{\frac{3}{2}} - \epsilon^2$ and $c = \log 3 - \log\bar{\sigma}_{min}$.

3.2 Analytical LBISTA

Analogously to Liu and Chen [10] and following the previous two theorems, decompose LBISTA-CP (untied), Algorithm 17, into two steps:

$$x^{(k)} = \eta_{\alpha^{(k-1)}}\left(x^{(k-1)} - \gamma^{(k-1)}\tilde{B}^T\left(Dx^{(k-1)} - y\right)\right), \tag{30}$$

where, in the first step, $\tilde{B}$ is pre-computed, such that

$$\mu_b(\tilde{B}, D) = \tilde{\mu}_b(D)\,.$$

In the second step, the parameters $\theta = \left(\left(\alpha^{(k)}\right)_{k=0}^{K-1}, \left(\gamma^{(k)}\right)_{k=0}^{K-1}\right)$ are trained layer wise, as discussed in the previous section. This results in a comparable method, with only $\mathcal{O}(K)$ trainable parameters, instead of $\mathcal{O}(n_y n_x + K)$ for LBISTA-CP (Equation 17) or even $\mathcal{O}(Kn_y n_x + K)$ for LBISTA (untied) Equation 18.

4 Computing the analytical weight matrix

ALBISTA relies on the analytical weight matrix, deriving this matrix can be challenging in practice, thus this section focuses on computing this matrix. We follow the procedure in Liu and Chen [10] by estimating Equation (21) with an upper bound. But in addition to Liu and Chen [10], we state a closed form for the upper bound, and currently, this is done by a projected gradient descent approach.

4.1 Solving an upper bound

Since the objective in Equation (21) is not differentiable, one solves the following upper bound problem

$$\min_{B \in \mathbb{R}^{n_y \times n_x}} \frac{1}{d}\|B^T D\|_F^2 \quad \text{s.t. } B^T[i]D[i] = I_d \text{ for } i = 1, \ldots, n\,. \tag{31}$$

This is derived from the following inequality

$$\begin{aligned}\max_{i\neq j}\frac{1}{d}\|B^T[i]D[j]\|_2^2 &\leq \max_{i\neq j}\frac{1}{d}\|B^T[i]D[j]\|_F^2\\ &\leq \frac{1}{d}\sum_{i,j}\|B^T[i]D[j]\|_F^2\\ &= \frac{1}{d}\|B^TD\|_F^2\,.\end{aligned}$$

In Liu and Chen [10], this is solved by a projected gradient method, but since this is a constrained linear least squares problem in the vector space of matrices with Frobenius inner product the following Theorem states a closed form of the solution of Equation (31).

Theorem 3. The minimizer $B \in \mathbb{R}^{n_y\times n_x}$ of Equation (31) is given as the concatenation

$$B = (B[1]\,, B[2]\,, \ldots,\, B[n])\ ,$$

where the n blocks are given as

$$B[i] = K_i^+\ (D[i] - E_iH_i)$$

with

$$\begin{aligned}K_i &= \left(2DD^T\right)^2 + D[i]D[i]^T\,,\\ E_i &= 2DD^TD[i]\,,\\ R_i &= D[i] - 2DD^TK_i^+E_i\,,\\ S_i &= -D[i]^TK_i^+E_i\,,\\ L_i &= R_i^TR_i + S_i^TS_i\,,\\ M_i &= K_i^+E_i(I - L_i^+L_i)\,,\\ H_i &= L_i^+S_i^T + (I - L_i^+L_i)(I + M_i^TM_i)^{-1}\\ &\quad (K_i^+E_i)^TK_i^+(D[i] - E_iL_i^+S_i^T)\,,\end{aligned}$$

for $i = 1, \ldots, n$.

The proof can be found in Appendix C.

Let $d = 1$ and the singular value decomposition of D given as $D = V\Sigma U^T$ and assume $B = V\tilde{\Sigma}U^T$. Then, the solution of Equation (31) is given in an even simpler form since

$$\begin{aligned}\|B^TD\|_F^2 &= \|U\tilde{\Sigma}V^TV\Sigma U^T\|_F^2\\ &= \|U\tilde{\Sigma}\Sigma U^T\|_F^2\\ &= \|\Sigma^+\Sigma\|_F^2.\end{aligned}$$

Choosing $B = D^{+,T}diag(\tilde{d})^{-1}$, where $\tilde{d} = diag(D^+D)$, i.e., a normalized pseudo-inverse, yields also a solution for Equation (31). Here, *diag* follows the matlab/python notation, where *diag* of a matrix gives the vector of the main diagonal and gives a diagonal matrix with a given vector on its main diagonal. For $d \geq 2$, the orthonormal block constraints in Equation (31) would not be met since with this construction we can only guarantee that the diagonal elements of B^TD are equal to one but cannot control what happens on the off-diagonal, thus not yielding a feasible solution.

4.2 Computing the analytical weight matrix in a MMV problem

In practice, often large data sets are obtained, i.e., by a large amount of measurements or measurements y^l, x^l representing pictures. For instance, 1,000 pixels and 200 measurements lead to a matrix D with $(1,000 \cdot 200)^2$ elements. Applying the theory for analytical LBISTA could thus be difficult in practice. Although D is sparse, it can take a long time to calculate the analytical weight matrix. The following Theorem states the connection between the MMV setting and the block sparse setting for ALBISTA, showing that it is sufficient to minimize the generalized mutual coherence Equation (32), see Liu and Chen [10], for K instead of minimizing the generalized mutual block-coherence (Equation 21) for $D = K \otimes I_d$. Moreover, in many applications the measurement model involves convolutions, see for example Ahmadi et al. [14], so we conclude this section by considering the cases where K is a circular or Toeplitz matrix.

Theorem 4. Let $\tilde{B}$ attain the minimum in

$$\inf_{\substack{\tilde{B}\in\mathbb{R}^{n\times n}\\ \tilde{B}_{:,i}^TK_{:,i}=1\\ \forall 1\leq i\leq n}} \max_{i\neq j} |\tilde{B}_{:,i}^TK_{:,j}|\,, \tag{32}$$

where $\tilde{B}_{:,i}$ refers to the ith column. Then the minimum of

$$\inf_{\substack{B\in\mathbb{R}^{n_y\times n_x}\\ B^T[l]D[l]=1\\ \forall 1\leq l\leq n}} \left\{ \max_{\substack{i\neq j\\ 1\leq i,j\leq n}} \frac{1}{d}\|B^T[i]D[j]\|_2 \right\}\,,$$

is given as $B = \tilde{B} \otimes I_d$, if $D = K \otimes I_d$.

The proof can be found in Appendix D. Moreover, the following relation holds, if $D = K \otimes I_d$,

$$\mu_b(D) = \frac{1}{d}\mu(K),$$

thus, the block-coherence of D can be enhanced by increasing the number of measurements d. It is feasible to solve Equation (31) by the pseudo inverse in the MMV setting, since this is solved for K, i.e., $d = 1$

4.3 Circular matrix case

Consider now the following setting, where the measurements $y^l \in \mathbb{R}^n$ are obtained by a circular convolution of x^l with vector k, i.e.,

$$\begin{aligned}y^l &= k \circledast x^l\\ &= Kx^l,\ l = 1, \ldots, d\,,\end{aligned} \tag{33}$$

where K is a circular matrix generated by a vector $k \in \mathbb{R}^n$, i.e., $K = circ(k) \in \mathbb{R}^{n\times n}$. Applying Theorem 3 to the circular case yields the following lemma.

Lemma 1. Let $k \in \mathbb{R}^n$ and let $B = circ(b) \in \mathbb{R}^{n\times n}$ where $b \in \mathbb{R}^n$ is given by

$$\begin{pmatrix} 2KK^T & k \\ k^T & 0 \end{pmatrix} \begin{pmatrix} b \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{34}$$

$\lambda \in \mathbb{R}$. Then, B attains the minimum in Equation (31).

The latter statement implies a simpler way to compute $b \in \mathbb{R}^n$ by using singular value decomposition

$$\begin{aligned} B^*K &= U^* diag\left(\sigma(B)\right) UU^* diag\left(\sigma(K)\right) U \\ &= U^* diag\left(\sigma(B) \odot \sigma(K)\right) U, \end{aligned}$$

where $\sigma(B) \in \mathbb{R}^n$ is the vector of singular values of B, respectively K, filled with zeros and U an unitary matrix. With $U = 1/\sqrt{n}F$, where F is the Discrete Fourier Transform (DFT) matrix, this leads to the conclusion $\sigma(B) = Fb = \hat{b}$, and thus

$$b = F^{-1}\left(\overline{1/\hat{k}}\right), \tag{35}$$

where $\hat{k} = Fk$. The expression $1/\hat{k}$ should be interpreted point wise and to be zero if $\hat{k}_i = 0$. This also concludes that the computation of B tends to be difficult in practice if K has not full rank, since this means that $\hat{k}$ has at least one zero entry. In this case, b has to be scaled with $\frac{n}{rank(K)}$ since

$$\begin{aligned} b^T k &= \frac{1}{n}\hat{b}^T\hat{k} = \frac{1}{n}\|\hat{k}\|_0 \\ &= \frac{rank(K)}{n} \overset{!}{=} 1. \end{aligned}$$

4.4 Toeplitz matrix case

Considering the more general convolutional setting,

$$y = k * x, \tag{36}$$

where $k \in \mathbb{R}^{\tilde{m}}$ and $x \in \mathbb{R}^n$ with $\tilde{m} < n$. This results in a Toeplitz matrix K

$$K = \begin{pmatrix} k_1 & 0 & \dots & 0 \\ k_2 & k_1 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ k_{\tilde{m}} & k_{\tilde{m}} & & \vdots \\ 0 & k_{\tilde{m}} & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_{\tilde{m}} \end{pmatrix} \in \mathbb{R}^{m\times n}.$$

The reasoning of the previous section cannot be applied to show that the solution of Equation (31) must be a Toeplitz matrix. But the following can be observed: Let b be constructed as discussed for $\tilde{K} = circ(\underline{k})$, where $\underline{k}$ is the concatenation of k and a zero vector of suitable dimensions, i.e., the first column of K. The analytical weight matrix B w.r.t. K can be constructed as

$$B_{:,i} = T^{i-1}b,\ i = 1, \dots, n,$$

where T is the $m \times m$ cyclic shift matrix, i.e., only the $m \times n$ submatrix of $\tilde{B} = circ(b) \in \mathbb{R}^{m\times m}$ is used. The columns of K can also be expressed through the cyclic shift of $\underline{k}$. Hence,

$$B_{:,i}^T K_{:,i} = b^T T^{-(i-1)} T^{i-1} k = b^T k = 1,$$

i.e., B is a feasible solution of Equation (31) for K. On the other hand, the cross coherence is bounded since

$$\begin{aligned} \max_{i,j\le n.\, i\neq j} |B_{:,i}^T K_{:,j}| &= \max_{i,j\le n,\, i\neq j} |b^T T^{-(i-1)} T^{j-1} k| \\ &= \max_{i,j\le n,\, i\neq j} |b^T T^{j-i} k| \\ &\le \max_{i,j\le m,\, i\neq j} |b^T T^{j-i} k| \\ &= \max_{i,j\le m,\, i\neq j} |\tilde{B}_{:,i}^T \tilde{K}_{:,j}| \\ &= \|\tilde{B}^T\tilde{K} - I_m\|_{\max} \\ &= \|\sqrt{n}F^{-1} diag\left(\sigma(B) \odot \sigma(K)\right) \frac{1}{\sqrt{n}} F - I_m\|_{\max} \\ &= \|F^{-1} I_m F - I_m\|_{\max} = 0, \end{aligned}$$

where $\|D\|_{\max} = \max_{i,j}|d_{ij}|$ is the maximum norm and $diag\left(\sigma(B) \odot \sigma(K)\right) = I_m$ derives from Equation (35). Note: To have this upper bound $\tilde{K}$ needs to have full rank, which is the case if the full time continuous FT of k has no zero points. Or one has to adjust the discrete grid. Thus, constructing B by extending K to a circular matrix is a feasible approach.

4.5 Connection to CNNs

It is known that using the Fast Fourier Transform (FFT) in CNNs can decrease the computation time if the convolutional filter is big [31, 32]. In Pratt et al. [31] showed that training weights in the Fourier domain can reduce training time while maintaining efficiency. By using this, the costs of $\mathcal{O}(n^2)$ operations could be reduced to $\mathcal{O}(n\log(n))$ operations. To connect the theory of FFT-CNNs and of unrolling ISTA in the context of deconvolution (Equations 33 or 36), the gradient step can be viewed as follows:

$$\begin{aligned} & x - \gamma b * \left(k * x - y\right) \\ &= x - \gamma b * k * x + \gamma b * y \\ &= \left(e - \gamma b * k\right) * x + \gamma b * y \\ &= f(\gamma) * x + \gamma\tilde{b}. \end{aligned} \tag{37}$$

This can be interpreted as a convolutional layer with kernel $f(\gamma) = \left(e - \gamma b * k\right)$ and bias $\tilde{b} = b*y$, where $e = [1, 0, \dots, 0]$. This means that ALISTA, with a Toeplitz matrix, can be interpreted as a CNN only two trainable parameters per layer, γ, λ. In the setting of FFT-CNN, the update rule can be formulated as

$$\begin{aligned} & x - \gamma b * \left(k * x - y\right) \\ &= F^{-1}\left(\left(\hat{e} - \gamma\hat{b} \odot \hat{k}\right) \odot \hat{x}\right) + \gamma\tilde{b}. \end{aligned}$$

On the other hand, using FFT CNNs shows only a speed up if we deal with large data sets, i.e. by evaluating high-resolution images, or large filters, i.e., if $\tilde{m}$ is greater than $\log(n)$.

TABLE 1 Properties of matrix K in both scenarios.

Case	Dimensions	$rank(K)$	$\mu(K)$
Gauss	$K \in \mathbb{R}^{32\times128}$	32	0.6268
Circ.	$K \in \mathbb{R}^{128\times128}$	32	0.6833

5 Numerical examples

In the following, we are going to present numerical results achieved by the presented algorithms.[1] We will investigate two MMV scenarios. First, the measurements y^l are obtained with a random Gaussian matrix, and second obtained by a redcued-rank random circular convolution. In each scenario, we will enforce the Kronecker structure and thus reducing the training cost by training only low dimensional $m \times n$ matrices. Furthermore, in the convolutional setting, the circular structure will be also enforced, thus reducing the training costs even more. To have a fair comparison, all algorithms are initialized with the analytical weight matrix.

5.1 MMV setting

The training data are sampled from an unknown distribution $\mathcal{X}$ and generated as follows. The signals x are generated for a given number of blocks n, given block length d and a possibility if a block $x[i]$ is active or not, i.e., if $\|x[i]\|_2 \neq 0$ or $\|x[i]\|_2 = 0$, called pnz (probability of non-zeros). If a block is active, the elements of this blocks are given by a normal Gaussian distribution with variance $\sigma^2 = 1$. The measurements y are obtained by Equation (3), where the elements of ϵ are given from a normal Gaussian distribution with variance $\sigma^2 = pnz \cdot n_x/n_y \cdot 10^{-SNR_{dB}/10}$. SNR_{dB} is the signal-to-noise ratio given in decibel. We consider the following cases. In each case, we generate x with $d = 15$, $n = 128, m = 32$, and $pnz = 10\%$.

Gaussian measurement matrix: In the Gaussian setting, we sample a $m \times n$ matrix K iid from a Gaussian distribution with variance $\sigma^2 = 1$. We normalize the columns, s.t. D has orthonormal blocks, as assumed in the beginning.

Circular convolution matrix: We construct the circular matrix as follows. At first, we generate a random iid sampled vector $\tilde{a}$ but set a certain amount of elements to zero. We define $k = \Re\left(F^{-1}\tilde{a}\right)$, and thus we can generate a rank deficient Matrix $D = K \otimes I_d$, where $K = circ(k)$. Thus, y^l is obtained through a circular convolution with symmetric $\hat{k} = Fk$. It is important for this section to generate a rank deficient matrix to have compressive observations. Otherwise, we would get a trivial problem if K, respectively D, has full rank. Computing $D = K \otimes I_d$ yields the desired matrix. The properties of these two matrices can be found in Table 1.

1 Code: https://github.com/janhauffen/Block-ALISTA.

5.2 Discussion

The results of the proposed methods can be found in Figure 1A for the Gaussian Problem case in Figure 1B for the circular case. We also consider the performance of a version of AMP [33]. AMP can be viewed as an Bayesian extension of ISTA with an additional Onsager correction term, before applying the thresholding operator, i.e.,

$$v^{(k)} = y - Dx^{(k)} + b^{(k)}v^{(k-1)}$$
$$x^{(k+1)} = \eta_\alpha\left(x^{(k)} + \gamma D^T v^{(k)}\right)$$

where $b^{(k)} = \mathbb{E}\left[\eta'(x^{(k)})\right]$. Here, we will train only γ, α, with the same procedure as already discussed and choose $D^T = B^T$. By using B^T instead of D^T we are resembling Orthogonal AMP (OAMP) [34], which follows a similar idea as ALISTA. In Ma and Ping [34], B is chosen to be de-correlated with respect to K, i.e.,

$$\mathrm{tr}(I_{n_y} - B^T D) = 0.$$

The analytical weight matrix B satisfies also this condition. Moreover, in Ma and Ping [34], the choice of different matrices is also discussed. Thus, untrained ALISTA with correction term can be viewed as a special case of OAMP. Note also, that the structure of Trainable ISTA, proposed in Ito et al. [35], is also based on OAMP and thus there are interrelations between AMP, unfolding ISTA, and analytical ISTA. We refer to learned AMP with analytical matrix as ALAMP. Different from Ma and Ping [34], we use the $\ell_{2,1}$-regularizer, instead of only ℓ_1-regularization, since we consider the MMV setting. This is also discussed in [36, 37]. Every proposed algorithm performs better, in terms of *NMSE*, as their untrained original. Interestingly learned AMP, with analytical weight matrix B, has a performance almost as good as LBISTA (untied) with only a fraction of trainable parameters. This may come from the fact, that block-soft thresholding is not the correct MMSE estimator for the generated signals x and thus the correction of $b^{(k)}$ yields the better estimation for x. As expected, we get almost the same performance of ALBISTA and LBISTA CP (untied). In Figure 2A, we show similar plots for the justification of Theorem 1, as also seen in Liu and Chen [10]. In particular, Figure 2A shows that $\frac{\alpha^{(k)}}{\gamma^{(k)}}$ is proportional to the maximal $\ell_{2,1}$-error over all training signals. An interesting behavior, which carries over from the sparse case, is that the learned $\ell_{2,1}$-regularization parameters approach zero, as k increases. If α is close to zero, we approach a least-squares problem. This means that after LBISTA found the support of the unknown signal x^* the algorithm consist only of the least squares fitting. Figure 2B shows that the trained $\gamma^{(k)}$ are bound in an interval. Note that, in contrast to Liu and Chen [10], Theorem 1 is based on a more general assumption, onto the thresholding parameters. One can take a suitable κ and obtains the upper bound for the ℓ_2-error and thus have convergence on the training set if the sparsity assumptions are met. Figure 3 shows the training loss over the training iterations for the results presented in Figure 1. One can see that ALBISTA needs less training iterations as LBISTA CP (untied) or LBISTA (untied) and thus less training data. The observed jumps occur when moving from one layer to the next due to the layer-wise

training, Algorithm 1. A layer is defined to be optimized if NMSE converged within $1e-5$.

6 Conclusion

We proposed ALISTA for the block-sparse and MMV case, important for many real-world applications, and derived corresponding theoretical convergence and recovery results. We relaxed the conditions for the regularization parameter and thus obtained a more precise upper-bound after ALISTA is trained. Nevertheless, this is still dependent on a sharp sparsity assumption on the unknown signals. We investigated and derived a direct solution for the analytical weight matrix in the general block sparse setting as well for one convolutional scenarios. The last

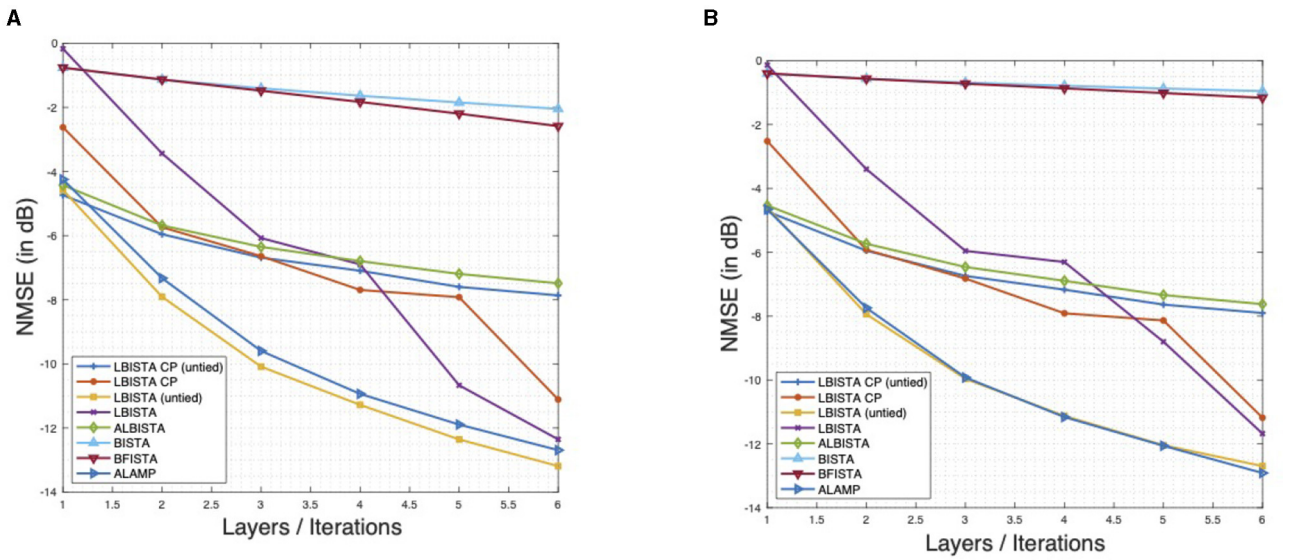

FIGURE 1
NMSE in dB over layers/iterations, pnz = 10%, without noise. BISTA, and FastBISTA are evaluated with $\alpha = 1$ and $\gamma = 1/(1.01\|D\|^2)$. **(A)** Gaussian measurement problem. **(B)** Circular convolution problem.

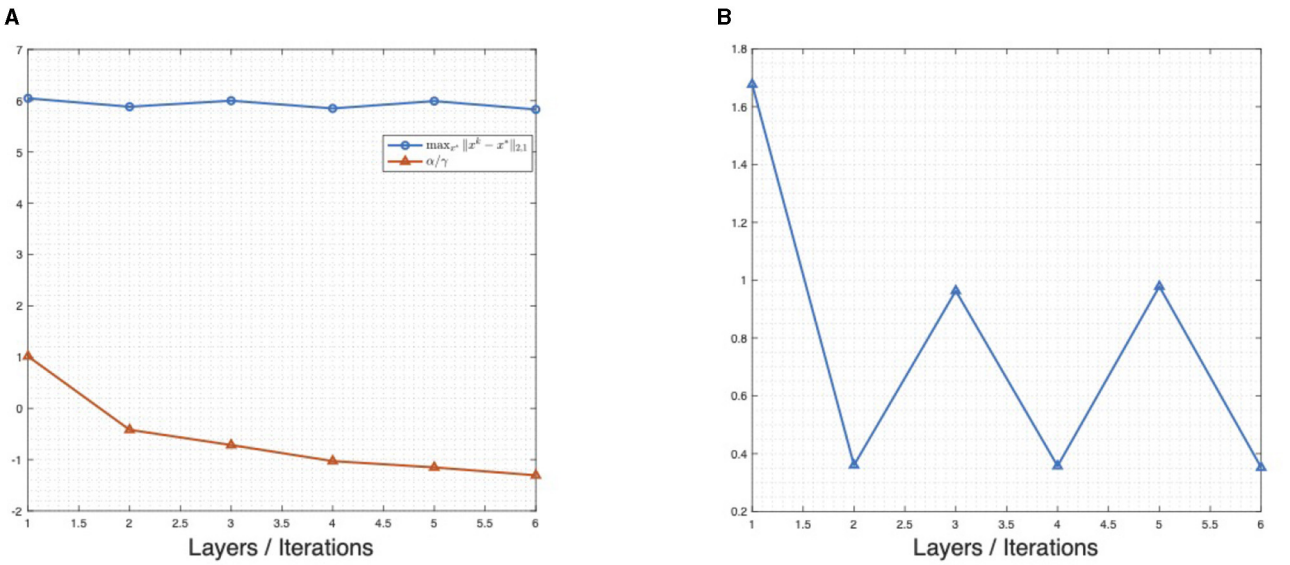

FIGURE 2
Plots justifying results in Theorem 1. **(A)** Parameters and maximal $\ell_{2,1}$-error in the noiseless case on the problem with circular convolution matrix. **(B)** Trained γ over layers/iterations on the problem with Gaussian measurement matrix.

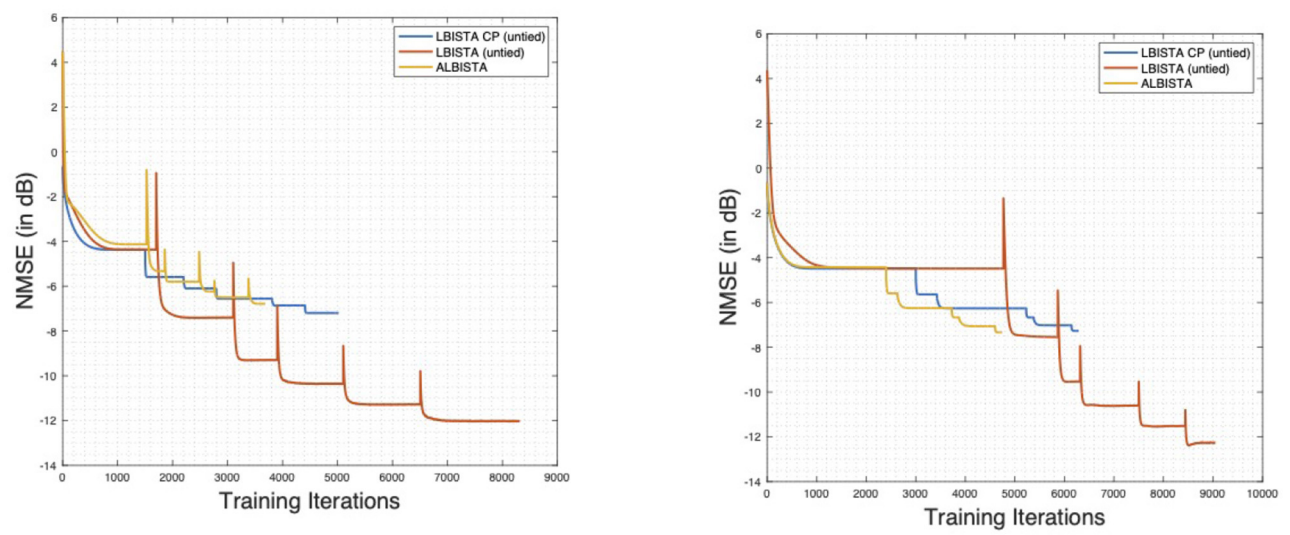

FIGURE 3
Training history for results shown in Figure 1A respectively Figure 1B. One can see that ALBISTA needs less training iterations than LBISTA CP (untied) or LBISTA (untied). **(A)** Gaussian measurement problem. **(B)** Circular convolution problem.

section provides numerical results and includes interrelations to AMP.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/janhauffen/Block-ALISTA.

Author contributions

JH, NM, and PJ contributed to the concept of the work. JH wrote the first draft of the article. NM and PJ supervised the work. All authors contributed to manuscript revision, read, and approved the submitted version.

Acknowledgments

We acknowledge support by the German Research Foundation and the Open Access Publication Fund of TU Berlin.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fams.2023.1205959/full#supplementary-material

References

1. Candès EJ, Romberg J, Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Inf Theory*. (2006) 52:489–509. doi: 10.1109/TIT.2005.862083

2. Donoho DL. Compressed sensing. *IEEE Trans Inf Theory*. (2006) 52:1289–306. doi: 10.1109/TIT.2006.871582

3. Shannon CE. Communication in the presence of noise. *Proc IRE*. (1949) 37:10–21. doi: 10.1109/JRPROC.1949.232969

4. Candes EJ, Tao T. Decoding by linear programming. *IEEE Trans Inf Theory*. (2005) 51:4203–15. doi: 10.1109/TIT.2005.858979

5. Rudelson M, Vershynin R. Geometric approach to error-correcting codes and reconstruction of signals. *Int Mathem Res Notices*. (2005) 2005:4019–41. doi: 10.1155/IMRN.2005.4019

6. Figueiredo MA, Nowak RD, Wright SJ. Gradient projection for sparse reconstruction: application to compressed sensing and other

inverse problems. *IEEE J Sel Top Signal Process*. (2007) 1:586–97. doi: 10.1109/JSTSP.2007.910281

7. Fornasier M, Rauhut H. Iterative thresholding algorithms. *Appl Comput Harmon Anal*. (2008) 25:187–208. doi: 10.1016/j.acha.2007.10.005

8. Gregor K, LeCun Y. Learning fast approximations of sparse coding. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning* (2010). p. 399–406.

9. Chen X, Liu J, Wang Z, Yin W. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In: *Conference on Neural Information Processing Systems (NeurIPS 2018)* (2018).

10. Liu J, Chen X. ALISTA: Analytic weights are as good as learned weights in LISTA. In: *International Conference on Learning Representations (ICLR)* (2019).

11. Chen X, Liu J, Wang Z, Yin W. Hyperparameter tuning is all you need for LISTA. In: *Advances in Neural Information Processing Systems* (2021). p. 34.

12. Gorodnitsky IF, George JS, Rao BD. Neuromagnetic source imaging with FOCUSS: a recursive weighted minimum norm algorithm. *Electroencephalogr Clin Neurophysiol*. (1995) 95:231–51. doi: 10.1016/0013-4694(95)00107-A

13. Fengler A, Musa O, Jung P, Caire G. Pilot-based unsourced random access with a massive MIMO receiver, interference cancellation, and power control. *IEEE J Select Areas Commun*. (2022) 40:1522–34. doi: 10.1109/JSAC.2022.3144748

14. Ahmadi S, Burgholzer P, Mayr G, Jung P, Caire G, Ziegler M. Photothermal super resolution imaging: a comparison of different thermographic reconstruction techniques. *NDT E Int*. (2020) 111:102228. doi: 10.1016/j.ndteint.2020.102228

15. Ziniel J, Schniter P. Efficient high-dimensional inference in the multiple measurement vector problem. *IEEE Trans Signal Proc*. (2012) 61:340–54. doi: 10.1109/TSP.2012.2222382

16. Chen J, Huo X. Theoretical results on sparse representations of multiple-measurement vectors. *IEEE Trans Signal Proc*. (2006) 54:4634–43. doi: 10.1109/TSP.2006.881263

17. Schacke K. On the kronecker product. Master's thesis, University of Waterloo. (2004).

18. Yonina C, Eldar HB. Block-sparsity: coherence and efficient recovery. *arXiv preprint arXiv:08120329*. (2008). doi: 10.48550/arXiv.0812.0329

19. Donoho DL, Elad M, Temlyakov VN. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans Inf Theory*. (2005) 52:6–18. doi: 10.1109/TIT.2005.860430

20. Chen SS, Donoho DL, Saunders MA. Atomic decomposition by basis pursuit. *SIAM Rev*. (2001) 43:129–59. doi: 10.1137/S003614450037906X

21. Kutyniok G. Compressed sensing. *Mitteilungen der Deutschen Mathematiker-Vereinigung*. (2014) 1:24–9. doi: 10.1515/dmvm-2014-0014

22. Foucart S, Rauhut H. A mathematical introduction to compressive sensing. *Bull Am Math*. (2017) 54:151–65. doi: 10.1090/bull/1546

23. Byrne CL. *Applied Iterative Methods*. Wellesley, MA: AK Peters. (2008). doi: 10.1201/9780429295492

24. Bauschke HH, Combettes PL, Bauschke HH, Combettes PL. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY: Springer. (2011). doi: 10.1007/978-1-4419-9467-7

25. Beck A. A fast iterative shrinkage-thresholding algorithm for linear inverse problem. *Soc Ind Appl Mathem*. (2009) 2:183–202. doi: 10.1137/080716542

26. Combettes PL, Pesquet JC. Proximal splitting methods in signal processing. In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer (2011). p. 185–212. doi: 10.1007/978-1-4419-9569-8_10

27. Kim D, Park D. Element-wise adaptive thresholds for learned iterative shrinkage thresholding algorithms. *IEEE Access*. (2020) 8:45874–86. doi: 10.1109/ACCESS.2020.2978237

28. Fu R, Monardo V, Huang T, Liu Y. Deep unfolding network for block-sparse signal recovery. In: *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE (2021). p. 2880–2884. doi: 10.1109/ICASSP39728.2021.9414163

29. Musa O, Jung P, Caire G. Plug-and-play learned gaussian-mixture approximate message passing. In: *ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE (2021). p. 4855–4859. doi: 10.1109/ICASSP39728.2021.9414910

30. Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.*6980 (2015). doi: 10.48550/arXiv.1412.6980

31. Pratt H, Williams B, Coenen F, Zheng Y. FCNN: Fourier convolutional neural networks. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer (2017). p. 786–798. doi: 10.1007/978-3-319-71249-9_47

32. Chitsaz K, Hajabdollahi M, Karimi N, Samavi S, Shirani S. Acceleration of convolutional neural network using FFT-based split convolutions. *arXiv preprint arXiv:200312621*. (2020). doi: 10.48550/arXiv.2003.12621

33. Donoho DL, Maleki A, Montanari A. Message-passing algorithms for compressed sensing. *Proc Nat Acad Sci*. (2009) 106:18914–9. doi: 10.1073/pnas.0909892106

34. Ma J, Ping L. Orthogonal amp. *IEEE Access*. (2017) 5:2020–33. doi: 10.1109/ACCESS.2017.2653119

35. Ito D, Takabe S, Wadayama T. Trainable ISTA for sparse signal recovery. *IEEE Trans Signal Proc*. (2019) 67:3113–25. doi: 10.1109/TSP.2019.2912879

36. Kim J, Chang W, Jung B, Baron D, Ye JC. Belief propagation for joint sparse recovery. *arXiv preprint arXiv:1102.3289*. (2011). doi: 10.48550/arXiv.1102.3289

37. Chen Z, Sohrabi F, Yu W. Sparse activity detection for massive connectivity. *IEEE Trans Signal Proc*. (2018) 66:1890–904. doi: 10.1109/TSP.2018.2795540