Tensor Convolutional Dictionary Learning with CP Low-Rank activations

Pierre Humbert, Laurent Oudre, Nicolas Vayatis, Julien Audiffren

Tensor Convolutional Dictionary Learning with CP Low-Rank activations

Pierre Humbert, *Member, IEEE,* Laurent Oudre, Nicolas Vayatis, and Julien Audiffren

***Abstract*—In this paper, we propose an extension of the standard CDL problem with tensor representation, where each activation is constrained to be "low-rank" through a Canonical Polyadic decomposition. We show that this important additional constraint increases the robustness of the CDL with respect to strong noise and improve the interpretability of the results. Additionally, we discuss in details the benefits of this representation. Then, we propose two new algorithms, based on respectively ADMM or FISTA, that efficiently solve this problem, by leveraging the low-rank property and achieve a lower complexity than the leading CDL algorithms. Finally, we evaluate our approach on a wide range of experiments, highlighting the modularity and the important advantages of this tensorial low-rank formulation.**

***Index Terms*—Convolutional dictionary learning, convolutional sparse coding, tensor, canonical polyadic decomposition**

I. Introduction

The linear decomposition of a signal into few atoms of a learned dictionary (instead of a predefined one), has led to state-of-the-art results in a wide range of topics, including image classification [1], [2], image restoration [3], and signal processing [4]. Recently, its convolutional counterpart known as Convolutional Dictionary Learning (CDL) or Convolutional Sparse Coding (CSC), has gained renewed interest. The central idea behind CDL is to replace the traditional patch based representation with a global shift-invariant one. Various algorithms built around the Alternating Direction Method of Multipliers (ADMM) or the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) have been suggested to efficiently handle the associated CDL problem. However, these solvers are often designed to deal with univariate signals or images [5], although multivariate data with a natural tensor structure are encountered in many scientific areas such as audio signals [6].

To apply CDL techniques to tensor data, one strategy consists in vectorizing the data and to use standard univariate algorithms. However, this naive procedure ignores the multidimensional structure of the data and is frequently sub-optimal. One powerful idea to effectively exploit the structural information is to use multilinear analysis and tensor factorization techniques such as low-rank assumptions [7]. Furthermore, by providing tools for handling multivariate data, these approaches naturally simplify the adaptation of machine learning and statistical methods to tensors. Thus, even recently, many works have consider with great success the tensor framework e.g. in regression [8], [9], [10], [11], image completion [12], processing of audio signals [6], and decomposition of spectrograms or scalograms of ElectroEncephaloGram (EEG) data [13].

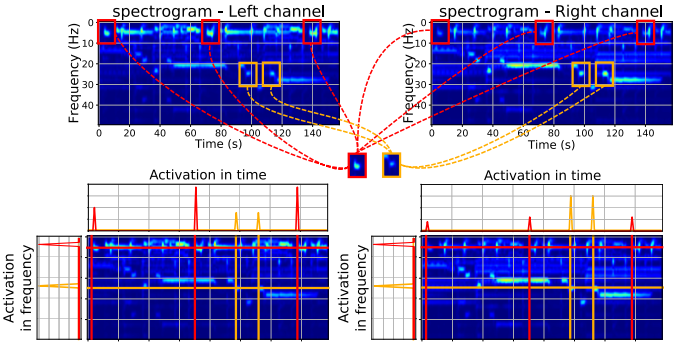


Fig. 1: Spectrograms of a stereo audio signal.

In this article, we introduce two algorithms (one based on ADMM, the other on FISTA) to solve CDL problems that takes into account the underlying structure of the multivariate/tensor signals. Unlike most previous works, we do not rely on a low-rank constraint on the atoms. Instead, we extend the standard minimization CDL problem to the tensor setting by adding low-rank CP decomposition constraints on each activation map. The idea of enforcing low-rank constraints for CDL is not novel: Tigamonti et al. [14] and Sironi et al. [15] used the idea of separable filters for learning low-rank atoms in order to improve computational runtime. More recent publications, including [16], [17], [18] have also successfully used low-rank (or even rank-1) constraints on the dictionary to introduce efficient solvers. Yet, in all these approaches, the low-rank constraints have been enforced on the dictionary/atoms. However, in several application contexts, the low-rank structure naturally appears in the activations rather than in the atoms/dictionary. To illustrate the relevance of our new approach, we display in Figure 1 an example of two spectrograms obtained from a stereo audio recording. Some repetitive patterns/atoms (highlighted in red and orange) are visible on the spectrograms which suggests that a CDL model may appear as natural for such data. Indeed, in the figure, two important properties can be highlighted. First, the activations of these atoms are sparse i.e. atoms are only present in a few places. Second, given a specific atom, we see that it is always present at the same frequency and in both spectrograms i.e. atoms are activated on a grid: for each atom, we can draw a grid where it is activated at each line intersection.

P. Humbert, L. Oudre and N. Vayatis are with Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, F-91190 Gif-sur-Yvette, France.

J. Audiffren is with the Cognition and Perception laboratory, University of Fribourg, Switzerland

Manuscript received –; revised –

This important property is exactly the low CP-rank of the activations. Note that, in this case, the low-rank structure of the data is displayed by the tensors of activations rather than by the observed patterns. In other words, although the time-frequency atoms may be complex (and not following the low-rank assumption), the activations (i.e. the time/frequency/channel positions where these atoms appear) clearly exhibit a low-rank structure. Here, this phenomenon may be explained by the harmonic structure of the audio signals, the tempo grid used by the instruments or the fact that both channels approximately capture the same audio scene.

The organization of this paper is as follows. We first recall in Section II the univariate CDL problem and the notations on tensor algebra. Then, we introduce in Section III our low-rank penalized multivariate CDL problem, referred to as Kruskal Convolutional Dictionary Learning (K-CDL). We propose two algorithms to solve it: one based on ADMM, the other on FISTA, whose properties and derivations are discussed in Section IV. Finally, we conduct in Section V multiple empirical analysis on synthetic and real data to highlight the performances of our approach.

II. Background

This section describes the prerequisite notions for introducing our model, by recalling the standard resolution of the univariate CDL problem as well as important definitions on tensor algebra.

A. Univariate convolutional dictionary learning

Problem. Given a finite set of N signals $\boldsymbol{y}_1, \dots, \boldsymbol{y}_N$ in $\mathbb{R}^M$ and a scalar $\lambda > 0$, the ℓ_1-regularized CDL problem is defined as

$$\min_{\boldsymbol{d}_k, \boldsymbol{z}_{n,k}} \frac{1}{2} \sum_{n=1}^{N} \left(\|\boldsymbol{y}_n - \sum_{k=1}^{K} \boldsymbol{d}_k \circledast \boldsymbol{z}_{n,k}\|_2^2 + \lambda \sum_{k=1}^{K} \|\boldsymbol{z}_{n,k}\|_1 \right) , \quad (1)$$
$$\text{s.t.} \quad \|\boldsymbol{d}_k\|_2 \leq 1 \quad \forall k = 1, \dots, K$$

where the $\boldsymbol{d}_k \in \mathbb{R}^W$ are called the *atoms*, the $\boldsymbol{z}_{n,k} \in \mathbb{R}^M$ the *activation maps*, and $\circledast$ denotes the circular convolutional operator.

Resolution. Even though the CDL problem is not jointly convex in $\left(\{\boldsymbol{d}_k\}_{k=1}^K, \{\boldsymbol{z}_{n,k}\}_{n,k=1}^{N,K} \right)$ [1], it is convex with respect to each variable when the other one is fixed. A natural optimization scheme for minimizing the objective function is therefore to alternate between the minimization with respect to the atoms $\{\boldsymbol{d}_k\}$ when the activation maps $\{\boldsymbol{z}_{n,k}\}$ are fixed and vice versa. This strategy, known as *alternating minimization* or *block coordinate descent* [19], [20], [21], has proven to be very effective for such problems, even though the convergence to a global minimum is not guaranteed in general. Solving Problem (1) therefore consists in solving two subproblems: the Convolutional Dictionary Learning (CDL) which searches for $\{\boldsymbol{d}_k\}$ with fixed $\{\boldsymbol{z}_{n,k}\}$, and a more difficult subproblem which is the core of this article, the Convolutional Sparse Coding (CSC) that updates $\{\boldsymbol{z}_{n,k}\}$ with fixed $\{\boldsymbol{d}_k\}$.

[1] To ease the notation, in the following, we will drop the index k or n when it is clear from the context, e.g. $\{\boldsymbol{d}_k\}_{k=1}^K$ will be denoted by $\{\boldsymbol{d}_k\}$.

ADMM solvers. Zeiler et al. [22] were the first to propose an efficient algorithm for the CSC problem by introducing an auxiliary variable to separate the convolution from the ℓ_1-regularization. This important idea of separating the fidelity term from the sparsity term is commonly used in contemporary methods. To do so, solvers often rely on ADMM [23], [24], that uses the Fourier domain for the computational convenience of convolutions [25], [26], [27]. The algorithm who popularized ADMM for both the CSC and CDL is called Fast Convolutional Sparse Coding (FCSC) [25]. In their paper, authors have shown remarkable improvements in efficiency by exploiting the Parseval's equality and the convolutional theorem for solving the CSC problem. The ADMM algorithm has been proven to converge to the optimal solution [28]. Furthermore, in practice, this algorithm often gives an estimate with sufficient accuracy within tens of iterations. Indeed, with alternate minimization, each iteration does not need to find an optimal point, but a point with medium accuracy. Unfortunately, simple examples show that ADMM can be very slow to converge to high accuracy [29].

FISTA solvers. Using FISTA [30] to solve the CSC problem was first proposed by Chalasani et al. [31]. Based on the Iterative Soft Thresholding Algorithm (ISTA) [32], this popular proximal method has the advantage of being a gradient-based algorithm involving very simple computations. Furthermore, compared to ISTA, FISTA performs an extra step known as the *Nesterov's momentum* which accelerates its convergence. At iteration t, FISTA has thus an optimal theoretical convergence rate guarantee of $\mathcal{O}(1/t^2)$ [30], which makes it very efficient to solve the CSC problem. The proof of convergence and the convergence rates do not depend on the particular structure of the CSC problem and can also be proven.

B. Tensor algebra

A tensor is a multidimensional array which extends the notion of vectors and matrices. Formally, a p-th order tensor is an element of the tensor product of $p \in \mathbb{N}_*$ vector spaces, denoted $\boldsymbol{\mathcal{X}} \in \mathbb{X} \triangleq \mathbb{R}^{n_1 \times \cdots \times n_p}$ and addressed by p indexes. In the following we denote $\boldsymbol{x}^{(q)}$ the vectorization of the folding of $\boldsymbol{\mathcal{X}}$ along the dimension q.

Tensor products. Tensor algebra relies on several products such as Kronecker or Khatri-Rao. The Kronecker product between $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{R}^{k \times \ell}$ is denoted $\boldsymbol{A} \otimes \boldsymbol{B}$. The result is a matrix of size $(mk) \times (n\ell)$ such that

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{1,1}\boldsymbol{B} & \cdots & a_{1,n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m,1}\boldsymbol{B} & \cdots & a_{m,n}\boldsymbol{B} \end{pmatrix} .$$

The Khatri-Rao product [33] between $\boldsymbol{A} \in \mathbb{R}^{m \times k}$ and $\boldsymbol{B} \in \mathbb{R}^{n \times k}$ is denoted $\boldsymbol{A} \odot \boldsymbol{B}$. The result is a matrix of size $(mn) \times (k)$ such that

$$\boldsymbol{A} \odot \boldsymbol{B} = [\boldsymbol{a}_{:,1} \otimes \boldsymbol{b}_{:,1}, \cdots, \boldsymbol{a}_{:,k} \otimes \boldsymbol{b}_{:,k}] .$$

In the following we denote as $\overset{\leftarrow p}{\odot}_{i=1}$ the product of p Khatri–Rao products in reverse order.

CP-rank and Kruskal operator. Given a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{X}$, the *Canonical Polyadic rank* of $\boldsymbol{\mathcal{X}}$ (CP-rank($\boldsymbol{\mathcal{X}}$) is the smallest

$R > 0$ such that there exist $\boldsymbol{x}_i^r \in \mathbb{R}^{n_i}$, $1 \leq i \leq p$, $1 \leq r \leq R$ satisfying

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{x}_1^r \circ \cdots \circ \boldsymbol{x}_p^r \,, \tag{2}$$

where $\circ$ is the outer product. In this case (2) is referred to as the CP decomposition of $\boldsymbol{\mathcal{X}}$. The *Kruskal operator* $[\![\cdot]\!]$ is defined as

$$[\![\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p]\!] \triangleq \sum_{r=1}^{R} \boldsymbol{x}_1^r \circ \cdots \circ \boldsymbol{x}_p^r \,, \tag{3}$$

where $\boldsymbol{X}_i = \left[\boldsymbol{x}_i^1 \mid \ldots \mid \boldsymbol{x}_i^R \right] \in \mathbb{R}^{n_i \times R}$, $1 \leq i \leq p$.

III. Tensor Convolutional Dictionary Learning with CP Low-Rank Activations

A. The K-CDL model

Let $\boldsymbol{\mathcal{Y}}_1, \ldots, \boldsymbol{\mathcal{Y}}_N \in \mathbb{Y} \triangleq \mathbb{R}^{n_1 \times \cdots \times n_p}$ be N tensor inputs of order $p > 0$ i.e. multidimensional signals. We define the regularized *Kruskal Convolutional Dictionary Learning problem* (K-CDL) as

$$\begin{aligned} \min_{\boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{Z}}_{n,k}} & \frac{1}{2} \sum_{n=1}^{N} \left(\left\| \boldsymbol{\mathcal{Y}}_n - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast \boldsymbol{\mathcal{Z}}_{n,k} \right\|_F^2 \right. \\ & \left. + \sum_{q=1}^{p} \alpha_q \sum_{k=1}^{K} \| \boldsymbol{Z}_{n,k,q} \|_1 + \sum_{q=1}^{p} \beta_q \sum_{k=1}^{K} \| \boldsymbol{Z}_{n,k,q} \|_F^2 \right) \\ \text{s.t.} & \begin{cases} \boldsymbol{\mathcal{Z}}_{n,k} = [\![\boldsymbol{Z}_{n,k,1}, \cdots, \boldsymbol{Z}_{n,k,p}]\!] \quad \forall n \,, \\ \boldsymbol{\mathcal{D}}_k \in \mathbb{D}, \| \boldsymbol{\mathcal{D}}_k \|_F \leq 1 \quad \forall k \,, \end{cases} \end{aligned} \tag{4}$$

with $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p), \boldsymbol{\beta} = (\beta_1, \ldots, \beta_p) \succeq 0$ two vectors of hyperparameters, and matrices $\{\boldsymbol{Z}_{n,k,q}\}$ in $\mathbb{R}^{n_q \times R}$.
In this formulation, the $\{\boldsymbol{\mathcal{Z}}_{n,k}\} \in \mathbb{Y}$ are multidimensional sparse activation maps which specify where the multidimensional atoms $\{\boldsymbol{\mathcal{D}}_k\}$ in $\mathbb{D} \triangleq \mathbb{R}^{w_1 \times \cdots \times w_p}$, $(w_1 \leq n_1, \cdots, w_p \leq n_p)$, are placed in the input signals. To take advantage of the tensor structure, we add a CP low-rank constraint on the activation maps via the Kruskal operator. The formulation of the K-CDL problem therefore relies on four important constraints and regularizations explained below.

The low CP-rank constraint on the activations. This constraint controls the linear link between the different modes of the activations maps. Instead of implicitly constraint the CP-rank via the constraint $\{\text{CP-rank}(\boldsymbol{\mathcal{Z}}_{n,k}) \leq R \quad \forall n, k\}$, we choose to embed it using the Kruskal operator $[\![\cdot]\!]$ (3). This is an approach also used in the recent works of [34] and [35]. Hence, an activation $\boldsymbol{\mathcal{Z}}_{n,k}$ is replaced by $[\![\boldsymbol{Z}_{n,k,1}, \cdots, \boldsymbol{Z}_{n,k,p}]\!]$ where the matrices $\{\boldsymbol{Z}_{n,k,q}\}$ are in $\mathbb{R}^{n_q \times R}$.

The unit-ball constraint. The constraint on the $\{\boldsymbol{\mathcal{D}}_k\}$ prevents the scaling indeterminacy between the atoms and the activations as in the standard CDL.

The sparsity regularization. The regularization on the activation tensors. Here, we use a *mode sparsity constraint*, which induces the sparsity of each element of the CP-decomposition for every activation tensors independently. In tensor regression, this regularization is also used in [36] for instance.

The ridge regularization. The CP decomposition is known to be unique when it satisfies the Kruskal condition [37], but only up to permutation of the normalized factor matrices. In other words, the CP decomposition is unchanged by scaling or permutation and the factors of this decomposition may not be unique. In CP decomposition and tensor regression, this property seems to make difficult for optimization methods to find "the" solution because there is not just one. Generally, this is handled by adding ridge penalizations (see [38], [39], and [40]).

Sparsity and ridge regularizations. It should be noted that both the ℓ_1 and ℓ_2 regularizations play important roles in the objective function (4). Recently, several papers in low-rank tensor regression (the closest model to ours but without convolution) have considered objective functions with both ℓ_1 and ℓ_2 penalties [41], [42], [36], [43]. In particular, our problem can be seen as close to the equation (2.4) of [36], where the authors show that using both penalties is essential to obtain a sample complexity bound for tensor recovery (equation (4.1) in [36]). Similarly, [44], [45] were only able to prove the identifiability of their models under the condition that the latent factors of the CP decomposition have same ℓ_2 norm. Importantly, this condition requires the addition of a ℓ_2 penalty to the objective function to be enforced, and cannot be achieved with the ℓ_1 norm only. Finally, in [8] and [45], the authors show that higher levels of ridge regularization tend to convexify the objective and facilitate convergence to a global optimum, which is in line with our previous remarks on the ridge regularization. The ℓ_1 regularization has a different but equally important role. Indeed, the aim of our methods is to solve the complete CDL problem (which consists of learning both the dictionary and the activations in an alternate procedure). This entails a problem with many more unknown variables than known ones. For an extreme example, in the 1-D case, without the sparsity constraint, a single dirac atom and activation signals identical to the input signals would lead to a perfect reconstruction (except at the edges). Sparsity is thus a key element to avoid such trivial solutions.Finally, during our experiments, we observed that with no sparsity imposed, the learned dictionary is of lower quality compared to the one with sparsity. This may be explained by the fact that the non-zeros entries of the activations are indicative where the atoms of the dictionary need to be learned.

In the following we are mostly interested in solving the K-CDL problem with atoms fixed i.e. the Kruskal-CSC (K-CSC) problem. Given one signal $\boldsymbol{\mathcal{Y}}$, and with regard to the previous remarks, the *elastic-net* K-CSC problem is

$$\begin{aligned} \min_{\{[\![\boldsymbol{Z}_{k,1}, \cdots, \boldsymbol{Z}_{k,p}]\!]\}_k} & \frac{1}{2} \left\| \boldsymbol{\mathcal{Y}} - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast [\![\boldsymbol{Z}_{k,1}, \cdots, \boldsymbol{Z}_{k,p}]\!] \right\|_F^2 \\ & + \sum_{q=1}^{p} \alpha_q \sum_{k=1}^{K} \| \boldsymbol{Z}_{k,q} \|_1 + \sum_{q=1}^{p} \beta_q \sum_{k=1}^{K} \| \boldsymbol{Z}_{k,q} \|_F^2 \,, \end{aligned} \tag{5}$$

where the $\{\boldsymbol{Z}_{k,q}\}$ are in $\mathbb{R}^{n_q \times R}$ and the $\| \cdot \|_F^2$ is added to improve the minimization process, as previously discussed.

Note that, contrary to the K-CDL, index n has been removed for sake of readability.

B. Links and differences with state-of-the-art models

Using tensor algebra in dictionary learning and CDL is not novel (see related work in [46]). For instance, authors of [47] have proposed to use tensor factorization techniques to build a more scalable algorithm to solve the univariate CDL problem (1). More recently, [48] introduced a multivariate linear system solver which is also able to solve the CDL problem efficiently.

For the multivariate CDL problem, different lines of research have been investigated. For instance, [46] introduced a convolutional method for visual tensor data that finds atoms efficiently sparifying an input tensor signal. Earlier works have focused on methods to learn separable atoms, an idea introduced in tensorial computer vision by [14] and [15]. In their papers, they proposed two methods to learn high-order CP low-rank dictionary and empirically showed that using separable atoms as dictionaries in CSC provides significant improvements in computational performance with respect to non-separable implementations, while giving little loss in accuracy or reconstruction quality. From this observation, very recently, some papers have re-focused on the 2-D multivariate CDL problem and assumed or learned separable/low-rank 2-D filter banks [49], [16], [17]. The first one, [49], introduced a computationally efficient algorithm when the dictionary atoms are given and already separable. The two others, [16], [17], proposed to directly learn the separable 2-D atoms. A slight modification of this separable CDL problem is proposed by [18] where they empirically showed that this alternative formulation provides a reduction in computation time over the stantard CSC and CDL algorithms.

Instead of trying to strictly extend the multivariate CDL to tensor via the multivariate convolution operator, another approach is to replace the convolution by specific tensor products. In [50] for instance, authors used the t-product (see definition in [51]) to provide another tensor CDL formulation that has the potential to uncover high dimensional correlation among channels, but is also computationally expensive. In [52] and [53] exploit other products such as the t-linear combination.

It is worth mentioning the two closest papers to this article, [34] and [35], which introduce very similar formulations as (4). Nevertheless, note that, [34] only considered order-three tensor with CP-rank equals to one. Furthermore, their algorithm is significantly more time-consuming as its cost is proportional to the size of the signal. In [35], the algorithm relies on the reformulation of the main problem into another one called *CSC with multichannel dictionary filters and single-channel activation maps* [54]. However, as for [34], by not fully tacking into account the properties of the multidimensional convolution, the algorithm has a much higher complexity cost as the ones introduced in the following.

IV. Resolution of the problem

Even though the problem (5) is not convex, it is convex with respect to each of the Z-blocks $\{(\boldsymbol{Z}_{1,q}, \cdots, \boldsymbol{Z}_{K,q})\}_{q=1}^{p}$, or D-block $(\boldsymbol{\mathcal{D}}_1, \cdots, \boldsymbol{\mathcal{D}}_K)$ when the other ones are fixed. Furthermore, the two regularizations are separable with respect to these blocks. A natural optimization scheme for minimizing the objective function is therefore to use a *block-coordinate strategy* or *alternating minimization*. The main idea is to split the main non-convex problem into several convex subproblems; 1) by freezing the D-block and all except one Z-block at a time (referred as $\boldsymbol{\mathcal{Z}}$-step) 2) by only freezing all the Z-blocks (referred as $\boldsymbol{\mathcal{D}}$-step). Although this algorithm monotonically decreases the objective function, a stationary point is not guaranteed to be a local minimum (it can be a saddle point). Fortunately, we will see that in practice the block relaxation algorithm almost always converges to at least a local minimum. All proofs of the following results are deferred to the Appendix.

Sub-problems to learn activations. To solve (5), we also use an iterative strategy. For q varying from 1 to p, we consider

$$\min_{\boldsymbol{Z}_{1,q},\cdots,\boldsymbol{Z}_{K,q}} \frac{1}{2}\left\|\boldsymbol{\mathcal{Y}} - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast [\![\boldsymbol{Z}_{k,1}, \cdots, \boldsymbol{Z}_{k,q}, \cdots, \boldsymbol{Z}_{k,p}]\!]\right\|_F^2 + \alpha_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_1 + \beta_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_F^2 \,. \tag{6}$$

One naive solution is to rewrite the problem as a regression one (without the convolution) and to use tensor regression solvers [8], [10], [11]. However, it requires the construction of a very large circulant tensor which is not tractable in practice due to memory limitation. In the following, we propose two efficient algorithms based on either ADMM or FISTA to solve (6).

A. Preliminary results and lemmas

Let first introduce important properties which will be useful in the following for deriving our ADMM and FISTA solvers.

We first introduce the following notations:

$$f\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^{K}\right) = \frac{1}{2}\left\|\boldsymbol{\mathcal{Y}} - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast [\![\boldsymbol{Z}_{k,1}, \cdots, \boldsymbol{Z}_{k,p}]\!]\right\|_F^2$$

$$g\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^{K}\right) = \underbrace{\alpha_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_1}_{\varphi_{\alpha_q}(\cdot)} + \underbrace{\beta_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_F^2}_{\psi_{\beta_q}(\cdot)} \,.$$

In this equation, f is *the fidelity term* (i.e. the data fitting term) that controls the difference between the input and its reconstruction, and g is the summation of the regularizations φ_{α_q} and ψ_{β_q}.

Lemma 1. (Mode-wise DFT) – *Given the CP-decomposition of a tensor $\boldsymbol{\mathcal{X}} = [\![\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p]\!]$, the Discrete Fourier Transform (DFT) can be performed mode-wise, i.e.*

$$\widehat{\boldsymbol{\mathcal{X}}} = \sum_{r=1}^{R} \widehat{\boldsymbol{x}}_1^r \circ \cdots \circ \widehat{\boldsymbol{x}}_p^r \triangleq [\![\widehat{\boldsymbol{X}}_1, \cdots, \widehat{\boldsymbol{X}}_p]\!] \,, \tag{7}$$

where $\widehat{\cdot}$ denotes the frequency representation of a signal. The complexity of the computation of $\widehat{\boldsymbol{\mathcal{X}}}$ using the FFT goes from $\boldsymbol{\mathcal{O}}(\prod_{i=1}^{p} n_i \log(\prod_{i=1}^{p} n_i))$ to $\boldsymbol{\mathcal{O}}(R\sum_{i=1}^{p} n_i \log(n_i)))$. Notice that the DFT is only performed on the second dimension of each factor matrix, i.e. $\widehat{\boldsymbol{X}}_q = [\widehat{\boldsymbol{X}_q(:,1)} \mid \ldots \mid \widehat{\boldsymbol{X}_q(:,R)}]$.

We see from this lemma the important advantage of separable signals over non-separable ones in term of complexity.

Lemma 2. (Equality in the Fourier domain) – *The orthogonality of the Fourier basis implies a Plancherel formula. Therefore the fidelity term $f(\cdot)$ is equal in the Fourier domain to*

$$f\left(\boldsymbol{Z}_{k,q}\right) = \frac{1}{2\prod_{i=1}^{p} n_i} \left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_{k,1}, \cdots, \widehat{\boldsymbol{Z}}_{k,p}]\!] \right\|_F^2 \triangleq \frac{1}{\prod_{i=1}^{p} n_i} \widehat{f}\left(\widehat{\boldsymbol{Z}}_{k,q}\right) ,$$

where $*$ is the component-wise product and $\widehat{f}$ denotes the fidelity term in the Fourier domain up to the factor $1/\prod_{i=1}^{p} n_i$.

Lemma 3. (A compact vectorized formulation) – *The following equality holds*

$$\widehat{f}\left(\widehat{\boldsymbol{Z}}_{k,q}\right) = \frac{1}{2} \left\| \widehat{\boldsymbol{y}}^{(q)} - \widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I}) \widehat{\boldsymbol{z}}^{(q)} \right\|_F^2 , \tag{8}$$

where $\widehat{\boldsymbol{y}}^{(q)}$ and is the vectorization of the folding of $\widehat{\boldsymbol{\mathcal{Y}}}$ along the dimension q, $\widehat{\boldsymbol{z}}^{(q)} = [\widehat{\boldsymbol{z}}_1^{(q)\intercal}, \ldots, \widehat{\boldsymbol{z}}_K^{(q)\intercal}]^\intercal$ where $\forall k, \widehat{\boldsymbol{z}}_k^{(q)}$ is the vectorization of the matrix $\widehat{\boldsymbol{Z}}_{k,q}$, $\widehat{\boldsymbol{\Gamma}} = [diag(\widehat{\boldsymbol{d}_1}^{(q)}), \ldots, diag(\widehat{\boldsymbol{d}_K}^{(q)})]$ with $\boldsymbol{d}_k^{(q)}$ the vectorization of the folding of $\widehat{\boldsymbol{\mathcal{D}}}_k$ along the dimension q, and

$$\widehat{\boldsymbol{A}} = \begin{pmatrix} \widehat{\boldsymbol{B}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K \end{pmatrix} \quad \text{where} \quad \widehat{\boldsymbol{B}}_k = (\overleftarrow{\odot}_{i=1, i\neq q}^{p} \widehat{\boldsymbol{Z}}_{k,i}) . \tag{9}$$

Here, $\widehat{\boldsymbol{\Gamma}} \in \mathbb{C}^{n_1 \cdots n_p \times K n_1 \cdots n_p}$, $\widehat{\boldsymbol{A}} \in \mathbb{C}^{K \prod_{1,i\neq q}^{p} n_i \times KR}$, $\boldsymbol{I} \in \mathbb{R}^{n_q \times n_q}$, and $\widehat{\boldsymbol{z}}^{(q)} \in \mathbb{C}^{KRn_q}$. Thus, the design matrix $\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ is in $\mathbb{C}^{n_1 \cdots n_p \times KRn_q}$.

B. T-ConvADMM: $\boldsymbol{\mathcal{Z}}$-step.

We now introduce T-ConvADMM, an ADMM-based solver for Problem (6). Considering the previous splitting of the objective function, the iterations of the ADMM algorithm with a scalar $\rho > 0$ and $\{\boldsymbol{U}_k\}$ as dual variables are given by

$$\{\boldsymbol{Z}_{k,q}^{(s+1)}\} = \arg\min_{\{\boldsymbol{Z}_{k,q}\}} \; f\left(\{\boldsymbol{Z}_{k,q}\}\right) + \frac{\rho}{2} \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q} - \boldsymbol{T}_k^{(s)} + \boldsymbol{U}_k^{(s)}\|_F^2 \tag{10}$$

$$\{\boldsymbol{T}_k^{(s+1)}\} = \arg\min_{\{\boldsymbol{T}_k\}} \; g\left(\{\boldsymbol{T}_k\}\right) + \frac{\rho}{2} \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}^{(s+1)} - \boldsymbol{T}_k + \boldsymbol{U}_k^{(s)}\|_F^2 \tag{11}$$

$$\{\boldsymbol{U}_k^{(s+1)}\} = \boldsymbol{U}_k^{(s)} + \boldsymbol{Z}_{k,q}^{(s+1)} - \boldsymbol{T}_k^{(s+1)} \tag{12}$$

As g is fully separable, $\forall k = 1, \ldots, K$, subproblem (11) admits the closed-form solution

$$\boldsymbol{T}_k^{(s+1)} = \frac{1}{1 + 2\boldsymbol{\beta}_q/\rho} \mathcal{S}_{\alpha_q/\rho}(\boldsymbol{Z}_{k,q}^{(s+1)} + \boldsymbol{U}_k^{(s)}) ,$$

where $\mathcal{S}_\gamma(\cdot)$ is the soft-thresholding operator defined as

$$\mathcal{S}_\gamma(\boldsymbol{z})[i] = \text{sign}(\boldsymbol{z}_i) \max(|\boldsymbol{z}_i| - \gamma, 0) . \tag{13}$$

Solving the linear systems. Subproblem (10) also admits a closed-form solution (with conditions [54]). However, this solution is difficult to compute due to the size of the involved matrices. One way to solve it efficiently is to exploit the Parseval's and convolution theorems in order to take advantage of the convolutional structure of the problem (as in the univariate case). Hence, using the Lemma 3, the solution of (10) in the Fourier domain can be written as the solution in $\widehat{\boldsymbol{z}}$ of

$$\left((\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I}) \widehat{\boldsymbol{\Gamma}}^H \widehat{\boldsymbol{\Gamma}} (\widehat{\boldsymbol{A}} \otimes \boldsymbol{I}) + \rho \boldsymbol{I}\right) \widehat{\boldsymbol{z}} = \left((\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I}) \widehat{\boldsymbol{\Gamma}}^H \widehat{\boldsymbol{y}} + \rho(\widehat{\boldsymbol{t}} - \widehat{\boldsymbol{u}})\right) ,$$

where $(\cdot)^H$ stands for the Hermitian transpose. The matrix $\left((\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I}) \widehat{\boldsymbol{\Gamma}}^H \widehat{\boldsymbol{\Gamma}} (\widehat{\boldsymbol{A}} \otimes \boldsymbol{I}) + \rho \boldsymbol{I}\right)$ is of size $KRn_q \times KRn_q$ which can be expensive to inverse. Fortunately, due to its particular diagonal block structure, we can invert this matrix by only solving n_q independent $KR \times KR$ linear systems.

Complexity of T-ConvADMM The pseudo-code of the $\boldsymbol{\mathcal{Z}}$-step of T-ConvADMM for one mode q can be found in Algorithm 1.

The complexity of T-ConvADMM is obtained by the analysis of each step. The pre-computation of the tensors $\widehat{\boldsymbol{\mathcal{Y}}}$ and $\{\widehat{\boldsymbol{\mathcal{D}}}_k\}$ is of complexity $\mathcal{O}((K+1)(M\log(M)))$ with $M = \prod_{i=1}^{p} n_i$. Then, given a particular mode q, we pre-compute the FFT of the remaining $\widehat{\boldsymbol{Z}}_{k,i}, (i \neq q)$. By Lemma 1, these operations have a complexity of $\mathcal{O}(KR(p-1)\sum_{i=1,i\neq q}^{p} n_i \log(n_i))$. Finally, the final matrix inversion involves n_q linear systems of size KR. By using Gaussian elimination or Cholesky decomposition the complexity is therefore of $\mathcal{O}((KR)^3 n_q)$. However, it is possible to take advantage of iterative methods to reduce the complexity (see [29]). Finally, the soft-threshold part and the dual variable updates are of complexity $\mathcal{O}(Kn_q)$. The overall complexity is therefore of order $\mathcal{O}((KR)^3 \sum_{i=1}^{p} n_i)$.

Algorithm 1 T-ConvADMM for mode q

Input: signal $\boldsymbol{\mathcal{Y}}$, dictionary $\boldsymbol{\mathcal{D}}_1, \cdots, \boldsymbol{\mathcal{D}}_K$, regularization and ADMM parameters λ, ρ, tolerance ε

$\widehat{\boldsymbol{\mathcal{Y}}}, \{\widehat{\boldsymbol{\mathcal{D}}}_k\} \longleftarrow \text{DFT}(\boldsymbol{\mathcal{Y}}), \{\text{DFT}(\boldsymbol{\mathcal{D}}_k)\}$

repeat

  for q in $\{1, \cdots, p\}$ **do**

    $\widehat{\boldsymbol{y}}, \{\widehat{\boldsymbol{d}}_k^{(q)}\} \longleftarrow \text{vec}(\widehat{\boldsymbol{Y}}^{(q)}), \{\text{vec}(\widehat{\boldsymbol{\mathcal{D}}}_k^{(q)})\}$

    $\{\widehat{\boldsymbol{Z}}_{k,i}\}_{k=1,i=1,i\neq q}^{K,p} \longleftarrow \{\text{DFT}(\boldsymbol{Z}_{k,i})\}_{k=1,i=1,i\neq q}^{K,p}$

    $\widehat{\boldsymbol{D}} \longleftarrow \widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$

    repeat

      ▷ Update of $\boldsymbol{Z}$ via equation (10)

      $\widehat{\boldsymbol{Z}}^{(s)}, \widehat{\boldsymbol{T}}^{(s)}, \widehat{\boldsymbol{U}}^{(s)} \longleftarrow \text{DFT}(\boldsymbol{Z}^{(s)}), \text{DFT}(\boldsymbol{T}^{(s)}), \text{DFT}(\boldsymbol{U}^{(s)})$

      $\widehat{\boldsymbol{z}}^{(s)}, \widehat{\boldsymbol{t}}^{(s)}, \widehat{\boldsymbol{u}}^{(s)} \longleftarrow \text{vec}(\widehat{\boldsymbol{Z}}^{(s)}), \text{vec}(\widehat{\boldsymbol{T}}^{(s)}), \text{vec}(\widehat{\boldsymbol{U}}^{(s)})$

      $\widehat{\boldsymbol{z}}^{(s+1)} \longleftarrow$ Solve $\left(\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{D}} + \rho \boldsymbol{I}\right) \widehat{\boldsymbol{z}} = \left(\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{y}} + \rho(\widehat{\boldsymbol{t}}^{(s)} + \widehat{\boldsymbol{u}}^{(s)})\right)$

      $\widehat{\boldsymbol{Z}}^{(s+1)} \longleftarrow$ Matricization of $\widehat{\boldsymbol{z}}^{(s+1)}$

      $\boldsymbol{Z}^{(s+1)} \longleftarrow \text{IDFT}(\widehat{\boldsymbol{Z}}^{(s+1)})$

      ▷ Update of $\boldsymbol{T}$ via equation (11)

      $\boldsymbol{T}^{(s+1)} \longleftarrow \text{prox}_{\rho, \alpha_q, \beta_q}(\boldsymbol{Z}^{(s+1)} + \boldsymbol{U}^{(s)})$

      ▷ Update of $\boldsymbol{u}$ via equation (12)

      $\boldsymbol{U}^{(s+1)} \longleftarrow \boldsymbol{U}^{(s)} + \boldsymbol{Z}^{(s+1)} - \boldsymbol{T}^{(s+1)}$

    until $\|\boldsymbol{Z}^{(s+1)} - \boldsymbol{Z}^{(s)}\|_\infty \leq \varepsilon$

  end for

until $\|\boldsymbol{\mathcal{Z}}^{(s+1)} - \boldsymbol{\mathcal{Z}}^{(s)}\|_\infty \leq \varepsilon$

C. T-ConvFISTA: $\boldsymbol{\mathcal{Z}}$-step.

We now introduce our second algorithm T-ConvFista, a FISTA-based solver for Problem (6). The idea of T-ConvFISTA is to consider the previous splitting and to alternate the minimization of (6) between i) a gradient descent on $f(\cdot) + \psi_{\beta_q}(\cdot)$, ii) the proximal operator over $\varphi_{\alpha_q}(\cdot)$, and iii) the Nesterov's momentum. In order to keep the convention of the FISTA algorithm, the proximal is taken only on the ℓ_1 norm. However, it is also possible to take the proximal from the $\ell_1 + \ell_2$ norm, as in T-ConvADMM.

The only complex part within this process is to compute the gradient to perform step i). Indeed, as $\varphi_{\alpha_q}(\cdot)$ is separable, the proximal operator for each $\boldsymbol{Z}_{k,q}$ is the standard soft-thresholding operator.

Computation of the gradient. The gradient descent step is performed in the Fourier domain. This trick decreases the complexity of computing the gradient. A nice formulation of the gradient of function $f(\cdot)$ in the Fourier domain can be computed as a consequence of Lemmas 1, 2 and 3 as

$$\text{IDFT}\left[\left(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I}\right)\widehat{\boldsymbol{\Gamma}}^H\left(\widehat{\boldsymbol{y}}^{(q)} - \widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{z}}^{(q)}\right)\right] . \quad (14)$$

There are several ways to speed-up the computation of the gradient.

First, we can exploit distributed computation by using a parallel matrix-vector multiplication. For instance, as in our specific case $\prod_{i=1}^p n_i \gg KRn_q$, it is valuable to precompute the Gram matrix

$$\boldsymbol{G} = (\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I}) \quad (15)$$

and $(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{y}}^{(q)}$ involved in Lemma 3 and then use distributed computation. In addition, the Gram matrix $\boldsymbol{G}$ has a particular structure that we can exploit. Indeed, the matrix $\boldsymbol{G}$ is composed of K^2 blocks equal to $\widehat{\boldsymbol{A}}_k^H\widehat{\boldsymbol{A}}_\ell$ with

$$\widehat{\boldsymbol{A}}_\ell = \text{diag}(\widehat{\boldsymbol{d}}_\ell^{(q)})\left((\overset{\leftarrow p}{\odot}_{i=1,i\neq q}\widehat{\boldsymbol{Z}}_{\ell,i}) \otimes I\right) . \quad (16)$$

Each of these blocks can be computed in $\boldsymbol{\mathcal{O}}(R^2\prod_{i=1,i\neq q}^p n_i)$ and the full matrix $\boldsymbol{G}$ can therefore be computed in $\boldsymbol{\mathcal{O}}((KR)^2\prod_{i=1,i\neq q}^p n_i)$ operations (see Appendix). Furthermore, $\boldsymbol{G}$ is a $(KRn_q \times KRn_q)$ banded matrix. Its product with $\widehat{\boldsymbol{z}}^{(q)}$ can therefore be made in only $\boldsymbol{\mathcal{O}}((KR)^2n_q)$ operations. Hence, whenever the rank $R \leq \left(\frac{1}{K}\prod_{i=1,i\neq q}^p n_i \log(\prod_{i=1}^p n_i)\right)^{1/2}$ the complexity is smaller than the one of FCSC.

Complexity of T-ConvFISTA. The pseudo-code of the $\boldsymbol{\mathcal{Z}}$-step of T-ConvFISTA for one mode q can be found in Algorithm 2.

The complexity of T-ConvFISTA is obtained by the analysis of each step. The pre-computation of the tensors $\widehat{\boldsymbol{\mathcal{Y}}}$ and $\{\widehat{\boldsymbol{\mathcal{D}}}_k\}$ is of complexity $\mathcal{O}((K+1)(M\log(M)))$ with $M = \prod_{i=1}^p n_i$. Then, given a particular mode q, we pre-compute the FFT of the remaining $\widehat{\boldsymbol{Z}}_{k,i}, (i \neq q)$. By Lemma 1, these operations have a complexity of $\mathcal{O}(KR(p-1)\sum_{i=1,i\neq q}^p n_i \log(n_i))$. Finally, we perform the gradient step in the Fourier domain. Each computation of the gradient is of complexity $\mathcal{O}((KR)^2n_q)$ if the Gram matrix is precomputed. The overall complexity is therefore dominated by $\mathcal{O}((KR)^2n_q)$ for typical value of parameters. As we do this process for every mode, we obtain an overall complexity of $\mathcal{O}((KR)^2\sum_{i=1}^p n_i)$.

Algorithm 2 T-ConvFISTA for mode q

Input: signal $\boldsymbol{\mathcal{Y}}$, dictionary $\boldsymbol{\mathcal{D}}_1, \cdots, \boldsymbol{\mathcal{D}}_K$, regularization and step parameters α, β, η ($\eta = 1/L$, the inverse of Lipschitz constant if calculate), tolerance ε
Initialization: $\boldsymbol{Z}^{(0)}$
$\widehat{\boldsymbol{\mathcal{Y}}}, \{\widehat{\boldsymbol{\mathcal{D}}}_k\} \longleftarrow \text{DFT}(\boldsymbol{\mathcal{Y}}), \{\text{DFT}(\boldsymbol{\mathcal{D}}_k)\}$
$\{\widehat{\boldsymbol{Z}}_{k,i}\}_{k=1,i=1,i\neq q}^{K,p} \longleftarrow \{\text{DFT}(\boldsymbol{Z}_{k,i})\}_{k=1,i=1,i\neq q}^{K,p}$
$\boldsymbol{G} \longleftarrow (\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\boldsymbol{\Gamma}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$
$\boldsymbol{G}' \longleftarrow (\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{y}}^{(q)}$
$t^{(0)} \longleftarrow 1$
repeat
▷ Update of $\boldsymbol{W}$ via a proximal gradient step (ISTA)
$\widehat{\boldsymbol{Z}}^{(s)} \longleftarrow \text{DFT}(\boldsymbol{Z}^{(s)})$
$\widehat{\boldsymbol{z}}^{(s)} \longleftarrow \text{vec}(\widehat{\boldsymbol{Z}}^{(s)})$
$\widehat{\boldsymbol{w}}^{(s+1/2)} \longleftarrow \widehat{\boldsymbol{z}}^{(s)} - \eta\left(\boldsymbol{G}\widehat{\boldsymbol{z}}^{(s)} - \boldsymbol{G}'\right)$
$\widehat{\boldsymbol{W}}^{(s+1/2)} \longleftarrow$ Matricization of $\widehat{\boldsymbol{w}}^{(s+1/2)}$
$\boldsymbol{W}^{(s+1/2)} \longleftarrow \text{IDFT}(\widehat{\boldsymbol{W}}^{(s+1/2)})$
▷ Update of $\boldsymbol{W}$ via a proximal step (ISTA)
$\boldsymbol{W}^{(s+1)} \longleftarrow \text{prox}_{\eta,\alpha_q,\beta_q}\left(\boldsymbol{W}_k^{(s+1/2)}\right)$
▷ Nesterov momentum step (FISTA)
$t^{(s+1)} \longleftarrow \frac{1+\sqrt{1+4\cdot t^{(s)2}}}{2}$
$\boldsymbol{Z}^{(s+1)} \longleftarrow \boldsymbol{W}^{(s+1)} + \frac{t^{(s)}-1}{t^{(s+1)}+1}(\boldsymbol{W}^{(s+1)} - \boldsymbol{W}^{(s)})$
until $\|\boldsymbol{Z}^{(s+1)} - \boldsymbol{Z}^{(s)}\|_\infty \leq \varepsilon$

D. Comparison with previous solvers

Table I collects the theoretical complexity of T-ConvADMM and T-ConvFISTA as well as the two most popular CDL algorithms: ConvFISTA [31] and FCSC with iterative application of the Sherman-Morrison equation (FCSC-ShM) [27]. With typical values for K and R, the theoretical complexity of our two algorithms is much smaller than the complexity of conventional methods with dominant term $\mathcal{O}((KR)^2\sum_{q=1}^p n_q)$ instead of $\mathcal{O}(K\prod_{q=1}^p n_q \log(\prod_{\ell=1}^p n_\ell))$. As an example, for a signal of size $(n_1 \times n_2 \times n_3) = (128 \times 128 \times 128)$ with $K = 12$ atoms and $R = 3$, $(KR)^2(n_1+n_2+n_3) = 497,664$ while $Kn_1n_2n_3\log(n_1n_2n_3) = 366,316,018$.

E. Dictionary update, $\boldsymbol{\mathcal{D}}$-step.

Given activation tensors $\{\boldsymbol{\mathcal{Z}}_{n,k}\}$, the dictionary update aims at improving how the model reconstructs the inputs $\boldsymbol{\mathcal{Y}}_1, \cdots, \boldsymbol{\mathcal{Y}}_N$ by solving

$$\min_{\forall k, \boldsymbol{\mathcal{D}}_k \in \mathbb{D}, \ \|\boldsymbol{\mathcal{D}}_k\|_F \leq 1} \frac{1}{2}\sum_{n=1}^N \left\|\boldsymbol{\mathcal{Y}}_n - \sum_{k=1}^K \boldsymbol{\mathcal{D}}_k \circledast \boldsymbol{\mathcal{Z}}_{n,k}\right\|_F^2 . \quad (17)$$

This step presents no significant difference with existing methods. This problem is smooth and convex and can be solved using classical algorithms such as ADMM [5].

V. Experiments

To illustrate the effectiveness and efficiency of T-ConvADMM and T-ConvFISTA, we consider in this section a wide range of synthetic and real data. To make comparisons

Algorithm	Time complexity (Z-step)	Dominant term
ConvFISTA [31]	$T(\underbrace{KM}_{\text{Gradient}} + \underbrace{KM\log(M)}_{\text{FFTs}} + \underbrace{KM}_{\text{Shrinkage}})$	$K\left(\prod_{i=1}^{p} n_i\right)\log(\prod_{i=1}^{p} n_i)$
FCSC-SM [27]	$T(\underbrace{KM}_{\text{Linear systems}} + \underbrace{KM\log(M)}_{\text{FFTs}} + \underbrace{KM}_{\text{Shrinkage}})$	$K\left(\prod_{i=1}^{p} n_i\right)\log(\prod_{i=1}^{p} n_i)$
T-ConvADMM	$T(\underbrace{(KR)^3\sum_{q=1}^{p} n_q}_{\text{Linear system}} + \underbrace{KR\sum_{q=1}^{p} n_q\log(M)}_{\text{FFTs}} + \underbrace{KR\sum_{q=1}^{p} n_q}_{\text{Shrinkage}})$	$(KR)^3\sum_{q=1}^{p} n_q$
T-ConvFISTA	$T(\underbrace{(KR)^2\sum_{q=1}^{p} n_q}_{\text{Gradient}} + \underbrace{KR\sum_{q=1}^{p} n_q\log(M)}_{\text{FFTs}} + \underbrace{KR\sum_{q=1}^{p} n_q}_{\text{Shrinkage}})$	$(KR)^2\sum_{q=1}^{p} n_q$

TABLE I: Comparison of the Time Complexity of the $\boldsymbol{\mathcal{Z}}$-steps of several solvers. T is the number of iteration, K the number of atoms, M the size of the signal, and R the rank.

that are as fair as possible, each algorithm is implemented in `Python` using `Tensorly` [55] (for tensor algebra), `Sporco` [56] (a `Python` package for convolutional sparse representations with some `C/C++` modules), and standard Python libraries. Furthermore, to save memory and reduce the time complexity, both methods are implemented with sparse matrix packages. We also compare our methods to the two leading batch CDL algorithms presented in the previous sections: FCSC with iterative application of the Sherman-Morrison equation (denoted as FCSC-ShM) [25], [27], and ConvFISTA in the Fourier domain [31], [27]. They are both implemented in `Sporco`. All subsequent simulations are run on a machine through `Linux/Ubuntu` with 16-core of 2.5GHz Intel CPUs and 64GB of RAM. Implementations of the methods are available at https://github.com/pierreHmbt/Tensor_CDL.

A. Evaluation on synthetic data

Dataset. Small-scale and large-scale experiments are performed by considering two different datasets:

- A *small-scale dataset* which contains 10 independent input signals of size $(25 \times 25 \times 25)$. Each signal is generated as follows. We draw $K = 3$ atoms of size $(5 \times 5 \times 5)$ according to an Uniform distribution with values in $[-1, 1]$ and normalize them. We set the maximal CP-rank of the sparse activations to $R^* = 2$. They are drawn from a Bernoulli-Uniform distribution with Bernoulli parameter equals to 0.2, and range of values in $[-1, 1]$. Finally, we generate the input tensor according to the convolutional model induced by the K-CDL (4).
- A *large-scale dataset* which is generated as the small-scale dataset but with input signals of size $(128 \times 128 \times 128)$ and Bernoulli parameter equals to 0.02.

These two datasets are extended with their noisy counterpart called *noisy small-scale dataset* and *noisy large-scale dataset*. Following [27], for each input, we construct noisy input signals by adding Multivariate Gaussian noise of progressively high variance to obtain a Signal to Noise Ratio (SNR) with respect to the original input of $25.5, 9.5$, 4.5, and 0.09 dB.

Metrics. We evaluate the methods with the Root Mean Square Error (RMSE) between the true input signal (resp. the true activation maps) and the reconstruction. These metrics are respectively denoted RMSE($\boldsymbol{\mathcal{Y}}$) and RMSE($\boldsymbol{\mathcal{Z}}$) and should be as small as possible.

Evaluation on the CSC task. On this experiment, the dictionary is assumed to be known and we only evaluate the CSC task. For both the noiseless and noisy cases and for each one of the 20 input signals, we run our methods with $R = 1, 2, 3, 4$ and for 5 different initializations giving a total of 400 runs. The activations $\{\boldsymbol{\mathcal{Z}}_{k,q}\}$ are initialized with random Uniform matrices. Each time, the initialization returning the lowest RMSE($\boldsymbol{\mathcal{Y}}$) of the 5 tries is kept.

a) Noisless case: Quantitative results for the noiseless case are collected in Table II. Both proposed methods give competitive results with RMSE under $1.e{-}7$ as soon as $R \geq 2$. Furthermore, as expected, the best results are obtained when the estimated rank R is equals to the true one, i.e. when $R = R^* = 2$. Notice that, although surprising, an overestimation of the rank does not degrade the performance and still leads to very low RMSE – under $1.e-7$. We also collected results of the standard methods in Table II (bottom). With RMSE only around $1.e{-}5$, we clearly outperformed FCSC-ShM and ConvFISTA.

b) Noisy case: Results on the noisy case with $R = 2$ are displayed on Figure 2. Interestingly, even with a lot of noise (e.g. SNR< 6), we observe that T-ConvADMM and T-ConvFISTA reconstruct the input signal with high accuracy while FCSC-ShM and ConvFISTA is completely defective and mostly overfits the noise. This was expected as the noise does not share the low-rank structure the signals. Table IV in the Appendix also shows that the RMSEs of the large-scale data are lower than those of the small-scale data. This was also expected because regardless of the size of the data, $R^* = 2$. Hence, we inherently impose much more structure on large signals. In conclusion, while our two methods give better results than the standard ones on these experiments, we cannot yet conclude whether T-ConvADMM or T-ConvFISTA is the best, as small differences between the two methods may be due to multiple factors.

Evaluation on the whole CDL task. We now evaluate our method on the full K-CDL problem i.e. both the dictionary and the activations are unknown. We use the datasets of the previous section, set $R = 2$, and use T-ConvFISTA combined with the FCSC solver with Sherman–Morrison iterates for the

CSC CP-rank	Metrics	*Small-scale dataset* T-ConvADMM	T-ConvFISTA	*Large-scale dataset* T-ConvADMM	T-ConvFISTA
$R=1$	RMSE($\boldsymbol{\mathcal{Y}}$) $\downarrow$	$\mathbf{0.016}\,(\pm\mathbf{0.005})$	$\mathbf{0.016}\,(\pm\mathbf{0.005})$	$\mathbf{0.025}\,(\pm\mathbf{0.002})$	$\mathbf{0.025}\,(\pm\mathbf{0.002})$
	RMSE($\boldsymbol{\mathcal{Z}}$) $\downarrow$	$\mathbf{0.013}\,(\pm\mathbf{0.003})$	$\mathbf{0.013}\,(\pm\mathbf{0.003})$	$\mathbf{0.016}\,(\pm\mathbf{0.003})$	$\mathbf{0.016}\,(\pm\mathbf{0.003})$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ $\uparrow$	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ $\uparrow$	0%	0%	0%	0%
$R=2$	RMSE($\boldsymbol{\mathcal{Y}}$) $\downarrow$	$1.346 \cdot e{-}7\,(\pm 8.996 \cdot e{-}8)$	$\mathbf{1.966 \cdot e{-}8}\,(\pm\mathbf{6.716 \cdot e{-}9})$	$\mathbf{7.575 \cdot e{-}11}\,(\pm\mathbf{5.528 \cdot e{-}12})$	$2.804 \cdot e{-}10\,(\pm 2.672 \cdot e{-}10)$
	RMSE($\boldsymbol{\mathcal{Z}}$) $\downarrow$	$8.041 \cdot e{-}8\,(\pm 5.312 \cdot e{-}8)$	$\mathbf{1.261 \cdot e{-}8}\,(\pm\mathbf{4.025 \cdot e{-}9})$	$\mathbf{4.476 \cdot e{-}11}\,(\pm\mathbf{2.556 \cdot e{-}12})$	$1.736 \cdot e{-}10\,(\pm 1.723 \cdot e{-}10)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ $\uparrow$	94%	**96%**	80%	**85%**
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ $\uparrow$	**96%**	**98%**	**90%**	**90%**
$R=3$	RMSE($\boldsymbol{\mathcal{Y}}$) $\downarrow$	$\mathbf{3.195 \cdot e{-}7}\,(\pm\mathbf{4.351 \cdot e{-}7})$	$7.126 \cdot e{-}7\,(\pm 2.348 \cdot e{-}7)$	$\mathbf{6.439 \cdot e{-}10}\,(\pm\mathbf{3.972 \cdot e{-}10})$	$1.771 \cdot e{-}8\,(\pm 7.898 \cdot e{-}8)$
	RMSE($\boldsymbol{\mathcal{Z}}$) $\downarrow$	$\mathbf{1.954 \cdot e{-}7}\,(\pm\mathbf{2.533 \cdot e{-}7})$	$4.266 \cdot e{-}7\,(\pm 1.355 \cdot e{-}7)$	$\mathbf{4.253 \cdot e{-}10}\,(\pm\mathbf{2.833 \cdot e{-}10})$	$1.200 \cdot e{-}8\,(\pm 4.459 \cdot e{-}9)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ $\uparrow$	**72%**	44%	**90%**	60%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ $\uparrow$	**96%**	**96%**	**90%**	56%
$R=4$	RMSE($\boldsymbol{\mathcal{Y}}$) $\downarrow$	$\mathbf{4.154 \cdot e{-}7}\,(\pm\mathbf{1.494 \cdot e{-}7})$	$9.290 \cdot e{-}7\,(\pm 2.851 \cdot e{-}7)$	$\mathbf{8.893 \cdot e{-}10}\,(\pm\mathbf{4.922 \cdot e{-}10})$	$4.365 \cdot e{-}8\,(\pm 1.030 \cdot e{-}8)$
	RMSE($\boldsymbol{\mathcal{Z}}$) $\downarrow$	$\mathbf{2.646 \cdot e{-}7}\,(\pm\mathbf{9.219 \cdot e{-}8})$	$5.512 \cdot e{-}7\,(\pm 1.796 \cdot e{-}7)$	$\mathbf{5.248 \cdot e{-}10}\,(\pm\mathbf{2.771 \cdot e{-}10})$	$2.689 \cdot e{-}8\,(\pm 5.989 \cdot e{-}9)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ $\uparrow$	**72%**	20%	**100%**	**100%**
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ $\uparrow$	**98%**	**100%**	**100%**	**100%**
CSC CP-rank	**Metrics**	***Small-scale dataset* FCSC-ShM** [25]	**ConvFISTA** [31]	***Large-scale dataset* FCSC-ShM** [25]	**ConvFISTA** [31]
–	RMSE($\boldsymbol{\mathcal{Y}}$) $\downarrow$	$3.072 \cdot e{-}5\,(\pm 7.682 \cdot e{-}6)$	$3.211 \cdot e{-}5\,(\pm 5.364 \cdot e{-}6)$	$2.840 \cdot e{-}5\,(\pm 3.403 \cdot e{-}6)$	$1.630 \cdot e{-}5\,(\pm 1.000 \cdot e{-}6)$
	RMSE($\boldsymbol{\mathcal{Z}}$) $\downarrow$	$2.031 \cdot e{-}5\,(\pm 4.601 \cdot e{-}6)$	$8.746 \cdot e{-}5\,(\pm 5.234 \cdot e{-}6)$	$1.873 \cdot e{-}5\,(\pm 2.128 \cdot e{-}6)$	$1.435 \cdot e{-}5\,(\pm 1.376 \cdot e{-}6)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ $\uparrow$	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ $\uparrow$	0%	0%	0%	0%

TABLE II: Results return on the CSC task on dataset without noise. For T-ConvADMM and T-ConvFISTA, $R = 1, 2, 3$, or 4. Mean and standard deviation are reported. For the RMSE the lowest the better. For the other ones, the higher the better.

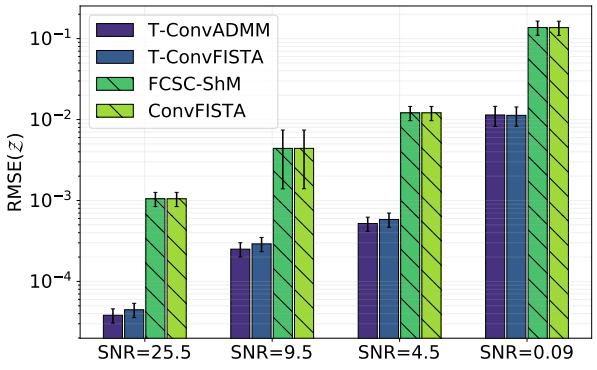


Fig. 2: RMSE($\boldsymbol{\mathcal{Z}}$) when only the CSC is evaluated on the large dataset with noise. For T-ConvADMM and T-ConvFISTA, R is set to its true value, $R^* = 2$. Mean and standard deviation are reported. The standard deviation are indicated using black lines.

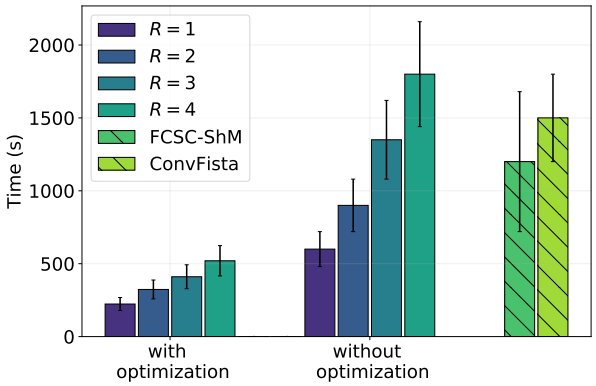


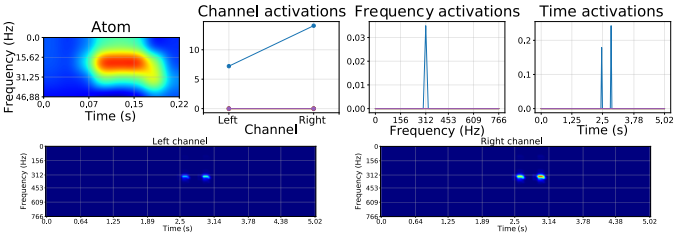
Fig. 3: Time until convergence of T-ConvFISTA on the dictionary learning process ($\boldsymbol{\mathcal{Z}}$ + $\boldsymbol{\mathcal{D}}$ steps), with and without the optimizations discussed in Section IV-C. The standard deviation are indicated using black lines.

$\boldsymbol{\mathcal{D}}$-step [27]. This solver is preferred to T-ConvADMM as it provides similar results on the K-CSC without the necessity of tuning the ρ parameter (we calculate the Lipschitz constant instead). The activations $\{\boldsymbol{Z}_{k,q}\}$ and the atoms $\{\boldsymbol{\mathcal{D}}_k\}$ are initialized with random Uniform matrices or tensors. Then, we normalize the atoms to satisfy the ℓ_2 constraint. On noiseless signals, we obtain a range of RMSEs comparable to those obtained with standard methods when $R \geq R^*$. However, on noisy signals, we observe that T-ConvFISTA returns better results than FCSC-ShM and ConvFista even if the number of active coefficients is lower.

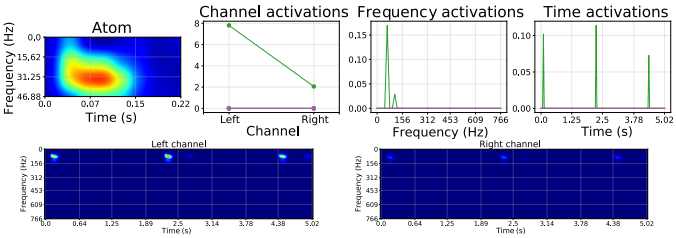
Regarding the time performances of T-ConvFISTA, we compare them to the ones of FCSC-ShM and ConvFista. For sake of fairness, these two methods have been re-implemented in `Python` using the same core functions. Figure 3 shows the average time until convergence (i.e. until the relative convergence tolerance becomes lower than $1e-4$ [29]). While it is important to remind that the relative speeds of each methos are dependent of their choice of hyperparameters as well as on the sparsity of the signals, we observe that (i) T-ConvFISTA with the optimizations discussed in Section IV-C is significantly faster than its regular counterpart and (ii) T-ConvFISTA is faster than FCSC-ShM and ConvFista, even if the advantage decreases as R increases. This is in line with the complexity of Table I.

B. Evaluation on audio data

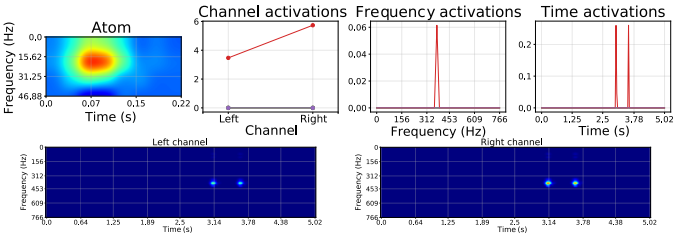
Data. Identifying recurring patterns in audio signal is an important problem in many scientific domains. A popular model to achieve this is nonnegative matrix factorization (NMF) [57]. A more recent model is the convolutive nonnegative matrix factorization (CNMF) [58]. It extends the classic NMF



(a) Left stereo audio: atom of drums **(b)** Right stereo audio: atom of drums



(c) Left stereo audio: atom of drums **(d)** Right stereo audio: atom of drums



(e) Left stereo audio: atom of guitar **(f)** Right stereo audio: atom of guitar

Fig. 4: On each of the four group of images: From left to right, the learned atom, the activations relative to the first dimension (channel), the activations relative to the second dimension (frequency), and the activations relative to the third dimension (time). Then, the two spectrograms corresponding to the reconstruction.

by introducing a convolutional structure into the low-rank model reconstruction and thus, captures short-term temporal dependencies in the data. However, these two methods never deal with stereo or multidimensional signals. In this example, we propose to use T-ConvFISTA to learn a dictionary (i.e. short-lived temporal patterns) on a stereo audio signal.

This stereo signal is a 5 seconds audio file resampled at 8000Hz and issued from the Blind Source Separation Database (BASS-DB) [59] – file Latino 1. Thus, we have a total of $2 \times 5 \times 8000 = 80000$ data points. For the two sub-signals (one per channel), we compute a Short-Time Fourier Transform (STFT) to obtain a spectrogram. Window size is set to 512 with 50% overlap: only the first 50 bins have been conserved $(0 - 781.25\text{Hz})$. To obtain sparse spectrograms, before any learning, we threshold them in order to only keep the important part of them. The final data consists in a third order tensor of size $(2 \times 50 \times 158)$ where the two spectrograms are stacked. The goal is now to extract relevant time-frequency atoms from this tensor data. To do so, using T-ConvFISTA, we reconstruct the input using $K = 25$ atoms of size $(1 \times 4 \times 8)$ (i.e. atoms of 0.224 seconds covering a band of frequencies of 46.875Hz). The maximal CP-rank of each associated activation is set to $R = 5$.

Metrics	T-ConvFISTA	FCSC-Sh	Zeros*
RMSE	$\mathbf{3.415 \cdot e{-}3}$	$4.048 \cdot e{-}3$	$1.060 \cdot e{-}2$
Sparsity	**0.17 %**	0.34 %	0.0 %

* reconstruction full of zeros.

TABLE III: Quantitative results on the audio signal.

Results. Results are displayed on Figure 4. We obtain a RMSE($\boldsymbol{\mathcal{Y}}$) of $3.415e{-}3$ with 0.17% active coefficients while with FCSC-ShM we obtain a RMSE of $4.048e{-}3$ with more than 0.34% active coefficients. For reference, the RMSE is equal to $1.060e{-}2$ when the reconstruction is full of 0. These results are collected in Table III. Atoms and activations returned by our method are displayed on Figure 4. Since in this audio signal the different instruments play in distinct frequency bands we can isolate them. For instance, the first two atoms of Figure 4 correspond to the drums and the last one to the guitar. This shows that the algorithm allows to extract the different instruments by identifying the time-frequency patterns related to each of them. Regarding the activations, we see that the algorithm captures the temporal structure of the musical piece. For instance, the second drum pattern is activated periodically, which is probably linked to the tempo used in the audio file.

C. Evaluation on EEG data

Data.

We now consider multichannel EEG signals that record brain activity with sensors covering a large part of the head. Note that this is a difficult dataset due to the small signal-to-noise ratio and the presence of impulsive noise. This two-dimensional measurement is stored in a matrix $\boldsymbol{X}$ in $\mathbb{R}^{N_s \times N_t}$ where N_s is the number of sensors and N_t is the number of samples.

The data consists in 32 EEG signals during a General Anesthesia (GA). We crop the full signal to keep only an important phase of the GA known as the "Recovery of Consciousness" (RoC) [60]. Recorded at 250Hz, each signal has a duration of 1000 seconds. With all channels included, it corresponds to $8,000,000$ points. Signals are pre-processed with a bandpass filter between 1 and 20Hz, to remove the potential drift below 1Hz, and to keep the frequencies below 20Hz that characterize GA [61]. Then, on each channel the STFT is computed with a window size of 1024 and 50% overlap: only the first 82 bins have been conserved $(0 - 20$ Hz). To obtain sparse spectrograms, we hard-threshold them before any learning in order to only keep the important part of them. We stack the 32 spectrograms in a final tensor $\boldsymbol{\mathcal{Y}}$ of size $(32 \times 82 \times 490)$. As for the audio signal, the goal is now to extract relevant time-frequency atoms from this tensor data with T-ConvFISTA. During a GA, patients are static and EEG signals do not present many patterns. As a consequence, we set $R = 2$, and only learn $K = 5$ atoms of size $(1 \times 15 \times 5)$ corresponding to time-frequency atoms covering 8.19 seconds and a band of frequencies of 3.42Hz. To reconstruct the 1-D initial signal from the spectrograms we apply the inverse STFT.

Learned dictionary and activations. Three learned atoms with their activations are displayed in Figure 5. One important

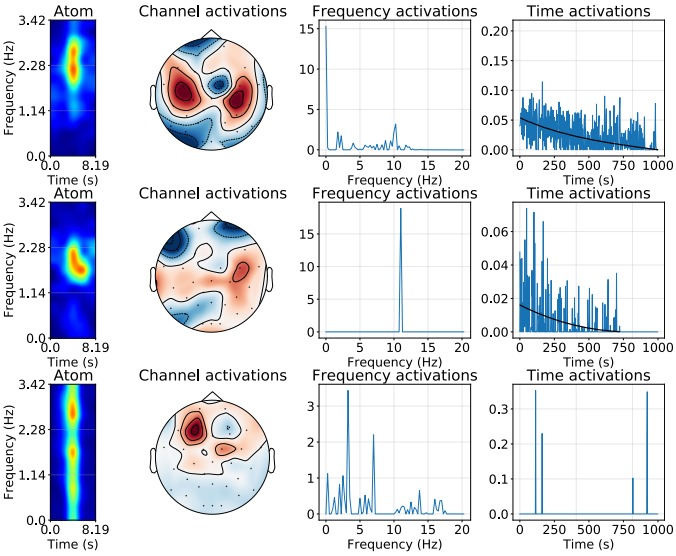


Fig. 5: Three atoms of interest with their activations. From left to right: the time-frequency atom, the channel activations (mode 1), the frequency activations (mode 2), and the time activations (mode 3).

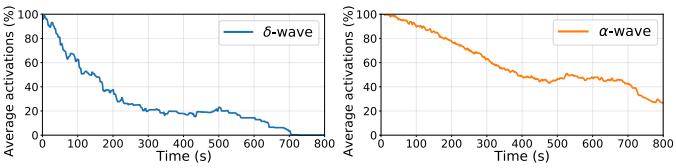


Fig. 6: Evolution of the time activations for the first and second atoms of Figure 5 which are relative to the δ and α waves.

property is the high interpretability of our results. Indeed, as we decompose the activations into the modes (channels $\times$ frequencies $\times$ times), we can study each one of them independently. For example, from the frequency activations (mode 2), we see that the first two atoms are related to important frequencies in anesthesia referred as α and θ-waves. Regarding their time activations (mode 3), they decrease with time (see Figure 6). This is a common behavior that occurs during a GA induced by propofol [60]. Indeed, it is known that when sedation begins, α and θ-waves appear. Then, during the ROC stage, they gradually disappear and fade away. The third atom corresponds to important spikes which may be explain by impulsive noise. From the channel activations (mode 1), we see that most of its contribution is on one channel. However, due to the propagation of the electricity on all the scalp, the other sensors also record these spikes at the same time. The activation tensor relative to this particular atom is therefore rank-1 (as found by the algorithm). Notice that, thanks to its identification, we can remove its contribution from the final reconstruction in order to not observe the spikes.

Robustness to noise and reconstruction. Via the channel activations (mode 1) of one learned atom we identify three deficient channels: 10 (CP1), 21 (CP2), and 28 (F4). In a clinical context, these channels are at spatial positions where the cap can come off. The sensors then only pick up noise at these positions. Fortunately, as shown in the synthetic experiments, due to the low-rank constraint, the model assumes

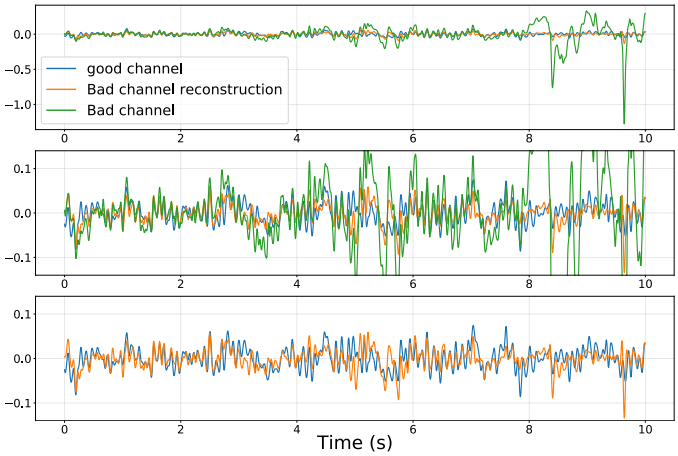


Fig. 7: On top, raw signal of a good channel (blue), a bad channel (green), and a reconstruction of the bad channel with T-ConvFISTA (orange). The other two figures are more focused on signals.

links between the channels and is robust to strong noise. In our case, this leads to an automatic reconstruction of the bad channels using the good ones. In Figure 7 for instance, we see a bad channel (in green) presenting a lot of noise, especially after 8 seconds. Using the other channels (e.g. the blue one), our algorithm reconstructs a plausible underlying true signal (in orange). Note that in this final experiments, we used the Inverse STFT in order to re-obtain temporal signals (and not only spectrograms).

VI. Conclusion

In this article, we extend the CDL problem to multivariate signals using tensor algebra. More particularly, we supposed that the activation maps are sparse and CP low-rank. We proposed two algorithms based either on ADMM and FISTA to efficiently solve the associated minimization problem. These two algorithms are evaluated and compared on both synthetic and real data. We showed that they provide better results than conventional algorithms in term of reconstruction, sparsity, and interpretability. On real datasets, we showed that the ability of our methods to split the activation maps in each mode allows a better comprehension of the input signal.

Several directions of research are available to improve multivariate CDL methods. Fist, some optimization steps of our two algorithms could be improved e.g. Equation (10) could maybe be solved more efficiently using conjugate gradient. Second, it is difficult, for the two proposed methods (but for CDL methods in general) to tune the hyperparameters. For instance, some supervision could be introduced to learn the adequate hyperparameters from an annotated database. Finally, a thorough understanding of the theoretical aspect of such model need to be investigated. Indeed, at the moment, important questions remain unanswered (e.g. conditions of ineffability of the atoms/activations).

References

[1] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2009, pp. 1033–1040.

[2] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2007, pp. 609–616.

[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing (TSP)*, vol. 54, no. 11, pp. 4311–4322, 2006.

[4] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research (JMLR)*, vol. 11, pp. 19–60, 2010.

[5] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: a comparative review and new algorithms," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 366–381, 2018.

[6] J. Wang, C. Xu, X. Xie, and J. Kuang, "Multichannel audio signal compression based on tensor decomposition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 286–290.

[7] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[8] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 540–552, 2013.

[9] G. Rabusseau and H. Kadri, "Low-rank regression with tensor responses," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 1867–1875.

[10] X. Li, J. Haupt, and D. Woodruff, "Near optimal sketching of low-rank tensor regression," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3466–3476.

[11] L. He, K. Chen, W. Xu, J. Zhou, and F. Wang, "Boosted sparse and low-rank tensor regression," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 1009–1018.

[12] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 1, pp. 208–220, 2012.

[13] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of EEG signals: a brief review," *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015.

[14] R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua, "Learning separable filters," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2754–2761.

[15] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, "Learning separable filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 37, no. 1, pp. 94–106, 2014.

[16] J. Quesada, P. Rodriguez, and B. Wohlberg, "Separable dictionary learning for convolutional sparse coding via split updates," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4094–4098.

[17] G. Silva, J. Quesada, and P. Rodríguez, "Efficient separable filter estimation using rank-1 convolutional dictionary learning," in *IEEE 28th International Workshop on Machine Learning for Signal Processing*. IEEE, 2018, pp. 1–6.

[18] J. Quesada, G. Silva, P. Rodriguez, and B. Wohlberg, "Combinatorial separable convolutional dictionaries," in *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*. IEEE, 2019, pp. 1–5.

[19] C. Hildreth, "A quadratic programming procedure," *Naval Research Logistics Quarterly*, vol. 4, no. 1, pp. 79–85, 1957.

[20] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[21] M. Nikolova and P. Tan, "Alternating proximal gradient descent for nonconvex regularised problems with multiconvex coupling terms," 2017.

[22] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2528–2535.

[23] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, vol. 9, no. R2, pp. 41–76, 1975.

[24] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[25] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 391–398.

[26] B. Wohlberg, "Efficient convolutional sparse coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7173–7177.

[27] ——, "Efficient algorithms for convolutional sparse representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, 2015.

[28] D. Gabay, "Chapter ix applications of the method of multipliers to variational inequalities," in *Studies in Mathematics and its Applications*. Elsevier, 1983, vol. 15, pp. 299–331.

[29] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[30] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[31] R. Chalasani, J. C. Principe, and N. Ramakrishnan, "A fast proximal method for convolutional sparse coding," in *International Joint Conference on Neural Networks (IJCNN)*, 2013, pp. 1–5.

[32] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[33] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons, 2005.

[34] A.-H. Phan, P. Tichavskỳ, and A. Cichocki, "Low rank tensor deconvolution," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 2169–2173.

[35] P. Humbert, J. Audiffren, L. Oudre, and N. Vayatis, "Low rank activations for tensor-based convolutional sparse coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3252–3256.

[36] T. Ahmed, H. Raja, and W. U. Bajwa, "Tensor regression using low-rank and sparse tucker decompositions," *SIAM Journal on Mathematics of Data Science*, vol. 2, no. 4, pp. 944–966, 2020.

[37] J. B. Kruskal, "Rank, decomposition, and uniqueness for 3-way and n-way arrays," *Multiway Data Analysis*, pp. 7–18, 1989.

[38] A. Uschmajew, "Local convergence of the alternating least squares algorithm for canonical tensor approximation," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 2, pp. 639–652, 2012.

[39] E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *Journal of Chemometrics*, vol. 25, no. 2, pp. 67–86, 2011.

[40] P. Paatero, "A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, no. 2, pp. 223–242, 1997.

[41] W. W. Sun and L. Li, "Store: sparse tensor response regression and neuroimaging analysis," *Journal of Machine Learning Research (JMLR)*, vol. 18, no. 1, pp. 4908–4944, 2017.

[42] B. Hao, A. R. Zhang, and G. Cheng, "Sparse and low-rank tensor estimation via cubic sketchings," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2020, pp. 1319–1330.

[43] Y. Liu, J. Liu, Z. Long, and C. Zhu, "Tensor computation for data analysis," 2021.

[44] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of N-way arrays," *Journal of chemometrics*, vol. 14, no. 3, pp. 229–239, 2000.

[45] E. F. Lock, "Tensor-on-tensor regression," *Journal of Computational and Graphical Statistics*, vol. 27, no. 3, pp. 638–647, 2018.

[46] R. Xu, Y. Xu, and Y. Quan, "Factorized tensor dictionary learning for visual tensor data completion," *IEEE Transactions on Multimedia*, vol. 23, pp. 1225–1238, 2020.

[47] F. Huang and A. Anandkumar, "Convolutional dictionary learning through tensor factorization," in *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges*, 2015, pp. 116–129.

[48] M. Boussé, N. Vervliet, I. Domanov, O. Debals, and L. De Lathauwer, "Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications," *Numerical Linear Algebra with Applications*, vol. 25, no. 6, p. e2190, 2018.

[49] G. Silva, J. Quesada, P. Rodríguez, and B. Wohlberg, "Fast convolutional sparse coding with separable filters," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 6035–6039.

[50] A. Bibi and B. Ghanem, "High order tensor formulation for convolutional sparse coding," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1772–1780.

[51] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.

[52] F. Jiang, X.-Y. Liu, H. Lu, and R. Shen, "Efficient multi-dimensional tensor sparse coding using t-linear combination," in *AAAI Conference on Artificial Intelligence*, 2018.

[53] X. Gong, W. Chen, and J. Chen, "A low-rank tensor dictionary learning method for hyperspectral image denoising," *IEEE Transactions on Signal Processing (TSP)*, vol. 68, pp. 1168–1180, 2020.

[54] B. Wohlberg, "Convolutional sparse representation of color images," in *IEEE Southwest Symposium on Image Analysis and Interpretation*. IEEE, 2016, pp. 57–60.

[55] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "Tensorly: tensor learning in Python," *Journal of Machine Learning Research (JMLR)*, vol. 20, no. 1, pp. 925–930, 2019.

[56] B. Wohlberg, "SPORCO: a Python package for standard and convolutional sparse representations," in *Proceedings of the 15th Python in Science Conference*, 2017, pp. 1–8.

[57] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[58] P. D. O'grady and B. A. Pearlmutter, "Convolutive non-negative matrix factorisation with a sparseness constraint," in *IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, 2006, pp. 427–432.

[59] E. Vincent, R. Gribonval, C. Févotte, A. Nesbit, M. D. Plumbley, M. E. Davies, and L. Daudet, "Bass-db multitrack recordings," Jun. 2005. [Online]. Available: https://doi.org/10.5281/zenodo.4064926

[60] P. L. Purdon, E. T. Pierce, E. A. Mukamel, M. J. Prerau, J. L. Walsh, K. F. K. Wong, A. F. Salazar-Gomez, P. G. Harrell, A. L. Sampson, A. Cimenser *et al.*, "Electroencephalogram signatures of loss and recovery of consciousness from propofol," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 110, no. 12, pp. E1142–E1151, 2013.

[61] E. N. Brown, R. Lydic, and N. D. Schiff, "General anesthesia, sleep, and coma," *New England Journal of Medicine (NEJM)*, vol. 363, no. 27, pp. 2638–2650, 2010.

[62] J. Liu, C. Garcia-Cardona, B. Wohlberg, and W. Yin, "First-and second-order methods for online convolutional dictionary learning," *SIAM Journal on Imaging Sciences*, vol. 11, no. 2, pp. 1589–1628, 2018.

APPENDIX

SNR	CSC Metrics	$\mathbf{R=2}$ *noisy small-scale dataset* T-ConvADMM	T-ConvFISTA	*noisy large-scale dataset* T-ConvADMM	T-ConvFISTA
25.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{3.988 \cdot e{-}4\,(\pm 4.121 \cdot e{-}5)}$	$\mathbf{3.999 \cdot e{-}4\,(\pm 4.427 \cdot e{-}5)}$	$\mathbf{6.523 \cdot e{-}5\,(\pm 5.970 \cdot e{-}7)}$	$\mathbf{7.606 \cdot e{-}5\,(\pm 1.253 \cdot e{-}6)}$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{2.397 \cdot e{-}4\,(\pm 2.331 \cdot e{-}5)}$	$\mathbf{2.403 \cdot e{-}4\,(\pm 2.528 \cdot e{-}5)}$	$\mathbf{3.820 \cdot e{-}5\,(\pm 4.431 \cdot e{-}7)}$	$\mathbf{4.469 \cdot e{-}5\,(\pm 1.1666 \cdot e{-}6)}$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}3\}$ ↑	98%	94%	86%	90%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}3\}$ ↑	98%	96%	86%	90%
9.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{2.513 \cdot e{-}3\,(\pm 1.046 \cdot e{-}4)}$	$\mathbf{2.492 \cdot e{-}3\,(\pm 9.024 \cdot e{-}5)}$	$\mathbf{4.254 \cdot e{-}4\,(\pm 9.016 \cdot e{-}6)}$	$\mathbf{4.958 \cdot e{-}4\,(\pm 7.733 \cdot e{-}6)}$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{1.509 \cdot e{-}3\,(\pm 6.113 \cdot e{-}5)}$	$\mathbf{1.495 \cdot e{-}3\,(\pm 5.066 \cdot e{-}5)}$	$\mathbf{2.504 \cdot e{-}4\,(\pm 8.241 \cdot e{-}6)}$	$\mathbf{2.913 \cdot e{-}4\,(\pm 7.194 \cdot e{-}6)}$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 2.5e{-}3\}$ ↑	84%	84%	84%	88%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 2.5e{-}3\}$ ↑	96%	98%	84%	90%
4.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{5.224 \cdot e{-}3\,(\pm 3.302 \cdot e{-}4)}$	$\mathbf{4.847 \cdot e{-}3\,(\pm 3.166 \cdot e{-}4)}$	$\mathbf{8.835 \cdot e{-}4\,(\pm 2.140 \cdot e{-}5)}$	$\mathbf{9.918 \cdot e{-}4\,(\pm 1.529 \cdot e{-}5)}$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{3.147 \cdot e{-}3\,(\pm 2.039 \cdot e{-}4)}$	$\mathbf{2.894 \cdot e{-}3\,(\pm 1.805 \cdot e{-}4)}$	$\mathbf{5.187 \cdot e{-}4\,(\pm 1.708 \cdot e{-}5)}$	$\mathbf{5.828 \cdot e{-}4\,(\pm 1.434 \cdot e{-}5)}$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 5.e{-}3\}$ ↑	41%	52%	84%	88%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 4.e{-}3\}$ ↑	84%	86%	84%	88%
0.09dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{7.796 \cdot e{-}2\,(\pm 1.156 \cdot e{-}3)}$	$\mathbf{7.805 \cdot e{-}2\,(\pm 2.893 \cdot e{-}3)}$	$\mathbf{1.938 \cdot e{-}2\,(\pm 5.476 \cdot e{-}3)}$	$\mathbf{1.923 \cdot e{-}2\,(\pm 5.212 \cdot e{-}3)}$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{4.857 \cdot e{-}2\,(\pm 1.948 \cdot e{-}3)}$	$\mathbf{4.848 \cdot e{-}2\,(\pm 2.566 \cdot e{-}3)}$	$\mathbf{1.139 \cdot e{-}2\,(\pm 3.176 \cdot e{-}3)}$	$\mathbf{1.127 \cdot e{-}2\,(\pm 3.033 \cdot e{-}3)}$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 2.e{-}2\}$ ↑	0%	0%	71%	71%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 3.e{-}2\}$ ↑	0%	0%	91%	91%

SNR	CSC Metrics	*noisy small-scale dataset* FCSC-ShM [25]	ConvFISTA [31]	*noisy large-scale dataset* FCSC-ShM [25]	ConvFISTA [31]
25.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$2.292 \cdot e{-}3\,(\pm 1.220 \cdot e{-}5)$	$2.109 \cdot e{-}3\,(\pm 2.547 \cdot e{-}4)$	$1.732 \cdot e{-}3\,(\pm 8.707 \cdot e{-}6)$	$1.732 \cdot e{-}3\,(\pm 8.703 \cdot e{-}6)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$1.454 \cdot e{-}3\,(\pm 7.326 \cdot e{-}5)$	$1.311 \cdot e{-}3\,(\pm 2.027 \cdot e{-}4)$	$1.050 \cdot e{-}3\,(\pm 2.794 \cdot e{-}6)$	$1.050 \cdot e{-}3\,(\pm 2.791 \cdot e{-}5)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}3\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}3\}$ ↑	0%	0%	10%	10%
9.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$6.734 \cdot e{-}3\,(\pm 7.117 \cdot e{-}4)$	$6.673 \cdot e{-}3\,(\pm 6.847 \cdot e{-}4)$	$6.689 \cdot e{-}3\,(\pm 3.344 \cdot e{-}5)$	$6.689 \cdot e{-}3\,(\pm 3.344 \cdot e{-}4)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$4.393 \cdot e{-}3\,(\pm 6.060 \cdot e{-}4)$	$4.367 \cdot e{-}3\,(\pm 5.919 \cdot e{-}4)$	$4.405 \cdot e{-}3\,(\pm 3.011 \cdot e{-}4)$	$4.406 \cdot e{-}3\,(\pm 3.010 \cdot e{-}4)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 2.5e{-}3\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 2.5e{-}3\}$ ↑	0%	0%	0%	0%
4.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$1.215 \cdot e{-}2\,(\pm 1.252 \cdot e{-}3)$	$1.209 \cdot e{-}2\,(\pm 1.202 \cdot e{-}3)$	$1.211 \cdot e{-}2\,(\pm 6.243 \cdot e{-}4)$	$1.211 \cdot e{-}2\,(\pm 6.242 \cdot e{-}4)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$7.800 \cdot e{-}3\,(\pm 1.033 \cdot e{-}3)$	$7.774 \cdot e{-}3\,(\pm 1.009 \cdot e{-}3)$	$7.812 \cdot e{-}3\,(\pm 5.252 \cdot e{-}4)$	$7.813 \cdot e{-}3\,(\pm 5.251 \cdot e{-}4)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 5.e{-}3\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 4.e{-}3\}$ ↑	0%	0%	0%	0%
0.09dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$2.780 \cdot e{-}1\,(\pm 1.622 \cdot e{-}3)$	$2.591 \cdot e{-}1\,(\pm 1.650 \cdot e{-}3)$	$2.593 \cdot e{-}1\,(\pm 6.895 \cdot e{-}4)$	$2.595 \cdot e{-}1\,(\pm 4.980 \cdot e{-}4)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$1.506 \cdot e{-}1\,(\pm 1.506 \cdot e{-}3)$	$1.372 \cdot e{-}1\,(\pm 2.169 \cdot e{-}3)$	$1.372 \cdot e{-}1\,(\pm 1.461 \cdot e{-}3)$	$1.367 \cdot e{-}1\,(\pm 1.081 \cdot e{-}4)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 2.e{-}3\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 3.e{-}3\}$ ↑	0%	0%	0%	0%

TABLE IV: Results when only the CSC is evaluated on dataset with noise. For T-ConvADMM and T-ConvFISTA, R is set its true value, $R^* = 2$. Mean and standard deviation are reported. For the RMSE the lowest the better. For the other ones, the higher the better.

A. Technical proofs

This section provides the technical proofs of the different propositions exposed in the main paper. To be as complete as possible, we recall well-known but important definitions and theoretical properties from multidimensional Fourier analysis. We begin with two important definitions.

Definition 1. (Discrete Fourier Transform (DFT)) – *Let consider a function $\boldsymbol{\mathcal{F}}$ defined on $\{0\dots,N_1-1\}\times\cdots\times\{0\dots,N_p-1\}$ with period $(N_1,\cdots,N_p)$. The Discrete Fourier Transform (DFT) of $\boldsymbol{\mathcal{F}}$ is given by*

$$\widehat{\boldsymbol{\mathcal{F}}}[k_1,\cdots,k_p] = \sum_{n_1=0}^{N_1-1}\cdots\sum_{n_p=0}^{N_p-1}\boldsymbol{\mathcal{F}}[n_1,\dots,n_p]\exp\left(-i2\pi\left(\frac{k_1n_1}{N_1},\dots,\frac{k_pn_p}{N_p}\right)\right)\,,$$

and the Inverse DFT (IDFT) of $\widehat{\boldsymbol{\mathcal{F}}}$ is given by

$$\boldsymbol{\mathcal{F}}[n_1,\cdots,n_p] = \left(\frac{1}{\prod_{i=1}^p N_i}\right)\cdot\sum_{k_1=0}^{N_1-1}\cdots\sum_{k_p=0}^{N_p-1}\widehat{\boldsymbol{\mathcal{F}}}[k_1,\dots,k_p]\exp\left(i2\pi\left(\frac{k_1n_1}{N_1},\dots\frac{k_pn_p}{N_p}\right)\right)\,.$$

Let now consider the periodization of two discrete function $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$,

$$\tilde{\boldsymbol{\mathcal{F}}}[n_1,\cdots,n_p] = \boldsymbol{\mathcal{F}}[n_1 \bmod N_1,\cdots,n_p \bmod N_p]$$
$$\tilde{\boldsymbol{\mathcal{G}}}[n_1,\cdots,n_p] = \boldsymbol{\mathcal{G}}[n_1 \bmod N_1,\cdots,n_p \bmod N_p]\,.$$

The two functions $\tilde{\boldsymbol{\mathcal{F}}}$ and $\tilde{\boldsymbol{\mathcal{G}}}$ are now two discrete functions with period $(N_1,\cdots,N_p)$ (each of the modes are periodic one-dimensional signals). The *circular convolution* is defined as follow.

Definition 2. (Circular discrete convolution) – *Let consider two functions $\boldsymbol{\mathcal{F}},\boldsymbol{\mathcal{G}}$ defined on $\{0\dots,N_1-1\}\times\cdots\times\{0\dots,N_p-1\}$ with both a period of $(N_1,\cdots,N_p)$. The circular convolution between $\tilde{\boldsymbol{\mathcal{F}}}$ and $\tilde{\boldsymbol{\mathcal{G}}}$ is given by*

$$(\tilde{\boldsymbol{\mathcal{F}}}\circledast\tilde{\boldsymbol{\mathcal{G}}})[n_1,\cdots,n_p] = \sum_{k_1=0}^{N_1-1}\cdots\sum_{k_p=0}^{N_p-1}\tilde{\boldsymbol{\mathcal{F}}}[k_1,\cdots,k_p]\tilde{\boldsymbol{\mathcal{G}}}[n_1-k_1,\cdots,n_p-k_p]\,.$$

$\tilde{\boldsymbol{\mathcal{F}}} \circledast \tilde{\boldsymbol{\mathcal{G}}}$ is a signal of period $(N_1, \cdots, N_p)$ and can be decomposed in a Fourier basis like classical periodic signals which give rises to the following important theorem.

Theorem 1. (Discrete convolution theorem) – *If $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ have period $(N_1, \cdots, N_p)$, then the DFT of $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{F}} \circledast \boldsymbol{\mathcal{G}}$ is*

$$\widehat{\boldsymbol{\mathcal{H}}}[n_1, \cdots, n_p] = \widehat{\boldsymbol{\mathcal{F}}}[n_1, \cdots, n_p] * \widehat{\boldsymbol{\mathcal{G}}}[n_1, \cdots, n_p] \ , \quad \textit{or in tensor notation} \quad \widehat{\boldsymbol{\mathcal{H}}} = \widehat{\boldsymbol{\mathcal{F}}} * \widehat{\boldsymbol{\mathcal{G}}} \ ,$$

where $*$ is the component-wise product or Hadamard product.

We now present a simple lemma showing the important advantage of separable signals (i.e. low-rank signal) over non-separable ones in term of complexity.

Lemma 4. (Mode-wise DFT) – *Given the CP-decomposition of a tensor $\boldsymbol{\mathcal{X}} = [\![\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p]\!]$, the Discrete Fourier Transform (DFT) can be performed mode-wise, i.e.*

$$\widehat{\boldsymbol{\mathcal{X}}} = \sum_{r=1}^{R} \widehat{\boldsymbol{x}}_1^r \circ \cdots \circ \widehat{\boldsymbol{x}}_p^r \triangleq [\![\widehat{\boldsymbol{X}}_1, \cdots, \widehat{\boldsymbol{X}}_p]\!] \ , \tag{18}$$

where $\widehat{\cdot}$ denotes the frequency representation of a signal. The complexity of the computation of $\widehat{\boldsymbol{\mathcal{X}}}$ using the FFT goes from $\boldsymbol{\mathcal{O}}(\prod_{i=1}^p n_i \log(\prod_{i=1}^p n_i))$ to $\boldsymbol{\mathcal{O}}(R \sum_{i=1}^p n_i \log(n_i)))$. Notice that the DFT is only performed on the second dimension of each factor matrix, i.e. $\widehat{\boldsymbol{X}}_q = [\widehat{\boldsymbol{x}_q(:,1)} \mid \ldots \mid \widehat{\boldsymbol{x}_q(:,R)}]$.

Proof. Using the definition of both the DFT and the CP-decomposition, the proof is straightforward. Furthermore, as we only perform 1-D FFT, we obtain the given complexity. □

To prove the next results, we will extensively used matricization and vectorization techniques. One important formula in tensor algebra is given by the above proposition.

Proposition 1. (Matricization of the Kruskal operator [7]) – *Let $\boldsymbol{\mathcal{X}}$ be a tensor in $\mathbb{X}$ with CP-decomposition $[\![\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(p)}]\!]$. Then,*

$$\boldsymbol{X}_{(q)} = \boldsymbol{X}_q \left(\boldsymbol{X}_p \odot \cdots \odot \boldsymbol{X}_{q+1} \odot \boldsymbol{X}_{q-1} \odot \cdots \odot \boldsymbol{X}_1\right)^\intercal = \boldsymbol{X}_q \left(\overset{\hookleftarrow p}{\odot}_{i=1, i\neq q} \boldsymbol{X}_i\right)^\intercal \ ,$$

where $\odot$ is the Khatri–Rao product and $\overset{\hookleftarrow p}{\odot}_{i=1}$ denotes the product of p Khatri–Rao products in reverse order. We can also also vectorized this formula which gives

$$vec(\boldsymbol{X}_{(q)}) = \left(\boldsymbol{X}_p \odot \cdots \odot \boldsymbol{X}_{q+1} \odot \boldsymbol{X}_{q-1} \odot \cdots \odot \boldsymbol{X}_1 \otimes \boldsymbol{I}_{n_q}\right) vec(\boldsymbol{X}_q) = \left(\overset{\hookleftarrow p}{\odot}_{i=1, i\neq q} \boldsymbol{X}_i \otimes \boldsymbol{I}_{n_q}\right) vec(\boldsymbol{X}_q) \ ,$$

where $\boldsymbol{I}_{n_q}$ is an identity matrix of size $(n_q \times n_q)$.

Lemma 5. (Equality in the Fourier domain) – *The orthogonality of the Fourier basis implies a Plancherel formula. Therefore the fidelity term $f(\cdot)$ is equal in the Fourier domain to*

$$f\left(\boldsymbol{Z}_{k,q}\right) = \frac{1}{2\prod_{i=1}^p n_i} \left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^K \widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_{k,1}, \cdots, \widehat{\boldsymbol{Z}}_{k,p}]\!] \right\|_F^2 \triangleq \frac{1}{\prod_{i=1}^p n_i} \widehat{f}\left(\widehat{\boldsymbol{Z}}_{k,q}\right) \ ,$$

where $*$ is the component-wise product and $\widehat{f}$ denotes the fidelity term in the Fourier domain up to the factor $1/\prod_{i=1}^p n_i$.

Proof. The proof rests on several equalities and properties.

$$\begin{aligned}
\frac{1}{2}\left\| \boldsymbol{\mathcal{Y}} - \sum_{k=1}^K \boldsymbol{\mathcal{D}}_k \circledast \sum_{r=1}^R \boldsymbol{z}_{k,r}^{(1)} \circ \cdots \circ \boldsymbol{z}_{k,r}^{(p)} \right\|_F^2 &= \frac{1}{2\prod_{i=1}^p N_i} \left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^K DFT(\boldsymbol{\mathcal{D}}_k \circledast \sum_{r=1}^R \boldsymbol{z}_{k,r}^{(1)} \circ \cdots \circ \boldsymbol{z}_{k,r}^{(p)}) \right\|_F^2 \quad \text{(Parseval's theorem – Plancherel)} \\
&= \frac{1}{2\prod_{i=1}^p N_i} \left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^K \widehat{\boldsymbol{\mathcal{D}}}_k * \sum_{r=1}^R DFT(\boldsymbol{z}_{k,r}^{(1)} \circ \cdots \circ \boldsymbol{z}_{k,r}^{(p)}) \right\|_F^2 \quad \text{(Convolution theorem 1)} \\
&= \frac{1}{2\prod_{i=1}^p N_i} \left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^K \widehat{\boldsymbol{\mathcal{D}}}_k * \sum_{r=1}^R \widehat{\boldsymbol{z}}_{k,r}^{(1)} \circ \cdots \circ \widehat{\boldsymbol{z}}_{k,r}^{(p)} \right\|_F^2 \quad \text{(Lemma 4)} \\
&= \frac{1}{2\prod_{i=1}^p N_i} \left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^K \widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_k^{(1)}, \cdots, \widehat{\boldsymbol{Z}}_k^{(p)}]\!] \right\|_F^2 \quad \text{(Kruskal operator)} \ .
\end{aligned}$$

□

Lemma 6. (A compact vectorized formulation) – *The following equality holds*

$$\widehat{f}\left(\widehat{\boldsymbol{Z}}_{k,q}\right) = \frac{1}{2}\left\| \widehat{\boldsymbol{y}}^{(q)} - \widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{z}}^{(q)} \right\|_F^2 \ , \tag{19}$$

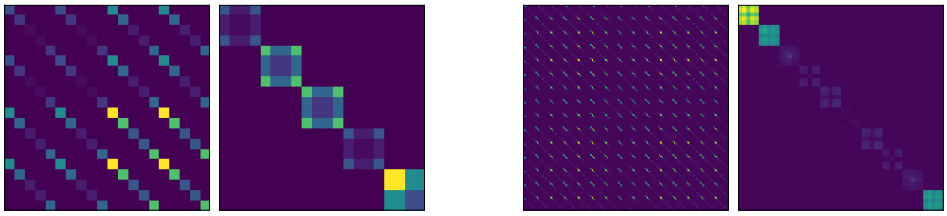


Fig. 8: Visualization of $\boldsymbol{G} = (\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ before and after a reordering. The two left matrices correspond to the Gram matrix without and with reordering. The two right matrices also correspond to the Gram matrix without and with reordering but for a higher dimension.

where $\widehat{\boldsymbol{y}}^{(q)}$ *and is the vectorization of the folding of* $\widehat{\boldsymbol{\mathcal{Y}}}$ *along the dimension* q, $\widehat{\boldsymbol{z}}^{(q)} = [\widehat{\boldsymbol{z}}_1^{(q)^\intercal}, \dots, \widehat{\boldsymbol{z}}_K^{(q)^\intercal}]^\intercal$ *where* $\forall k, \widehat{\boldsymbol{z}}_k^{(q)}$ *is the vectorization of the matrix* $\widehat{\boldsymbol{Z}}_{k,q}$, $\widehat{\boldsymbol{\Gamma}} = [diag(\widehat{\boldsymbol{d}_1}^{(q)}), \dots, diag(\widehat{\boldsymbol{d}_K}^{(q)})]$ *with* $\boldsymbol{d}_k^{(q)}$ *the vectorization of the folding of* $\widehat{\boldsymbol{\mathcal{D}}_k}$ *along the dimension* q, *and*

$$\widehat{\boldsymbol{A}} = \begin{pmatrix} \widehat{\boldsymbol{B}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K \end{pmatrix} \quad \text{where} \quad \widehat{\boldsymbol{B}}_k = (\overset{\hookleftarrow p}{\odot}_{i=1, i\neq q} \widehat{\boldsymbol{Z}}_{k,i}) \ . \tag{20}$$

Here, $\widehat{\boldsymbol{\Gamma}} \in \mathbb{C}^{n_1 \cdots n_p \times K n_1 \cdots n_p}$, $\widehat{\boldsymbol{A}} \in \mathbb{C}^{K \prod_{1, i\neq q}^p n_i \times KR}$, $\boldsymbol{I} \in \mathbb{R}^{n_q \times n_q}$, *and* $\widehat{\boldsymbol{z}}^{(q)} \in \mathbb{C}^{KRn_q}$. *Thus, the design matrix* $\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ *is in* $\mathbb{C}^{n_1 \cdots n_p \times KRn_q}$.

Proof. The proof mainly rests on the Proposition 1 and on the formulation of the previous lemma.

$$\begin{aligned}
\left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^K \widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_{k,1}, \cdots, \widehat{\boldsymbol{Z}}_{k,p}]\!] \right\|_F^2 &= \left\| \widehat{\boldsymbol{Y}}^{(q)} - \sum_{k=1}^K \widehat{\boldsymbol{D}}_k^{(q)} * \widehat{\boldsymbol{Z}}_{k,q} \left(\overset{\hookleftarrow p}{\odot}_{i=1} \widehat{\boldsymbol{Z}}_k^{(i)} \right)^\intercal \right\|_F^2 \quad \text{(matricization)} \\
&= \left\| \widehat{\boldsymbol{y}}^{(q)} - \sum_{k=1}^K \widehat{\boldsymbol{d}}_k^{(q)} * \left(\overset{\hookleftarrow p}{\odot}_{i=1} \widehat{\boldsymbol{Z}}_k^{(i)} \otimes I \right) \text{vec}(\widehat{\boldsymbol{Z}}_{k,q}) \right\|_F^2 \quad \text{(vectorization)} \\
&= \left\| \widehat{\boldsymbol{y}}^{(q)} - \sum_{k=1}^K \text{diag}(\widehat{\boldsymbol{d}}_k^{(q)}) \left(\overset{\hookleftarrow p}{\odot}_{i=1} \widehat{\boldsymbol{Z}}_k^{(i)} \otimes I \right) \text{vec}(\widehat{\boldsymbol{Z}}_{k,q}) \right\|_F^2 \quad (\boldsymbol{x} * \boldsymbol{y} = \text{diag}(\boldsymbol{x})\boldsymbol{y}) \\
&= \left\| \widehat{\boldsymbol{y}}^{(q)} - \sum_{k=1}^K \text{diag}(\widehat{\boldsymbol{d}}_k^{(q)}) \widehat{\boldsymbol{C}}_k \widehat{\boldsymbol{z}}_k \right\|_F^2 ,
\end{aligned}$$

where the last line is just notations. To obtain the final equality, we stack the matrices $\{\text{diag}(\widehat{\boldsymbol{d}}_k^{(q)})\}$ and construct a block-diagonal matrix such that the block are the $\{\boldsymbol{C}_k\}$. Finally we obtain the following equality.

$$\begin{pmatrix} \widehat{\boldsymbol{C}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{C}}_K \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{B}}_1 \otimes \boldsymbol{I} & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K \otimes \boldsymbol{I} \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{B}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K \end{pmatrix} \otimes \boldsymbol{I} \, ,$$

with $\widehat{\boldsymbol{B}}_k = (\overset{\hookleftarrow p}{\odot}_{i=1, i\neq q} \widehat{\boldsymbol{Z}}_{k,i})$. This end the proof. □

Proposition 2. (Gradient of f) – *With the notation of Theorem 2, the gradient of* f *with respect to* $\boldsymbol{z}^{(q)} = [\texttt{vec}(\boldsymbol{Z}_1^{(q)})^\intercal, \dots, \texttt{vec}(\boldsymbol{Z}_K^{(q)})^\intercal]^\intercal$ *is given by*

$$\nabla_{\boldsymbol{z}^{(q)}} f \left(\{\boldsymbol{Z}_k^{(r)}\}_{k=1,r=1}^{K,p} \right) = IDFT \left[\left((\widehat{\boldsymbol{A}}^{(q)} \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^{(q)} \right)^H \left(\widehat{\boldsymbol{\Gamma}}^{(q)} (\widehat{\boldsymbol{A}}^{(q)} \otimes \boldsymbol{I}) \widehat{\boldsymbol{z}}^{(q)} - \widehat{\boldsymbol{y}}^{(q)} \right) \right] , \tag{21}$$

where $IDFT[\cdot]$ *stands for the Inverse Discrete Fourier Transform.*

Proof. See [62] *Section 2.3.2.* for a complete and detailed proof. □

Proposition 3. *The matrix* $\boldsymbol{G} \triangleq (\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ *can be obtained by computing* K^2 *blocks* $\boldsymbol{G}_{k,\ell}, 1 \leq k, \ell \leq K$, *where*

$$\boldsymbol{G}_{k,\ell} = \left((\overset{\hookleftarrow p}{\odot}_{i=1, i\neq q} \widehat{\boldsymbol{Z}}_{k,i})^H \otimes I \right) \overline{diag(\widehat{\boldsymbol{d}_k}^{(q)})} diag(\widehat{\boldsymbol{d}_\ell}^{(q)}) \left((\overset{\hookleftarrow p}{\odot}_{i=1, i\neq q} \widehat{\boldsymbol{Z}}_{\ell,i}) \otimes I \right) \ , \tag{22}$$

and each of these blocks can be computed in $\mathcal{O}(R^2 \prod_{i=1, i\neq q}^p n_i)$ *operations.*

Proof. The first step of the proof requires to write $\widehat{\boldsymbol{\Gamma}}$ as the Kronecker product of two specific matrices in order to use the equality $(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D})=(\boldsymbol{AC}\otimes\boldsymbol{BD})$. Recall that $\widehat{\boldsymbol{\Gamma}}$ is a block-diagonal matrix, i.e. $\widehat{\boldsymbol{\Gamma}}=[\text{diag}(\widehat{\boldsymbol{d}_1}^{(q)}),\cdots,\text{diag}(\widehat{\boldsymbol{d}_K}^{(q)})]$. We can decompose each diagonal-block, $\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})$, into smaller diagonal matrices using the kronecker product as follow

$$\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})=\sum_{i=1}^{N_{\setminus q}}\text{diag}(e_i)\otimes\Delta_{k,i}\quad\text{with}\quad N_{\setminus q}=\prod_{i=1,i\neq q}^{p}n_i\ ,$$

where $\text{diag}(e_i)\in\mathbb{R}^{N_{\setminus q}\times N_{\setminus q}}$ and $\Delta_{k,i}\in\mathbb{C}^{n_q\times n_q}$ being the i-th diagonal block of $\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})$ (i.e. $\Delta_{k,i}=\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})_{(i\cdot n_q:(i+1)\cdot n_q),(i\cdot n_q:(i+1)\cdot n_q)}$). As $(\text{diag}(e_i)\otimes\Delta_{k,i})$ is decomposed into two matrices of the proper dimension, we can used the equality $(\boldsymbol{A}\otimes\boldsymbol{B})(\boldsymbol{C}\otimes\boldsymbol{D})=(\boldsymbol{AC}\otimes\boldsymbol{BD})$ leading to

$$\begin{aligned}
&\left((\odot_{i=1,i\neq q}^{p}\widehat{\boldsymbol{Z}}_{k,i})^H\otimes I\right)\overline{\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})}\text{diag}(\widehat{\boldsymbol{d}_\ell}^{(q)})\left((\odot_{i=1,i\neq q}^{p}\widehat{\boldsymbol{Z}}_{\ell,i})\otimes I\right)\\
&=\left(\widehat{\boldsymbol{B}}_k^H\otimes I\right)\sum_{i=1}^{N_{\setminus q}}\left(\text{diag}(e_i)\otimes\overline{\Delta_{k,i}}\right)\sum_{j=1}^{N_{\setminus q}}\left(\text{diag}(e_j)\otimes\Delta_{\ell,j}\right)\left(\widehat{\boldsymbol{B}}_\ell\otimes I\right)\\
&=\sum_{i=1}^{N_{\setminus q}}\sum_{j=1}^{N_{\setminus q}}\left(\widehat{\boldsymbol{B}}_k^H\otimes I\right)\left(\text{diag}(e_i)\otimes\overline{\Delta_{k,i}}\right)\left(\text{diag}(e_j)\otimes\Delta_{\ell,j}\right)\left(\widehat{\boldsymbol{B}}_\ell\otimes I\right)\\
&=\sum_{i=1}^{N_{\setminus q}}\sum_{j=1}^{N_{\setminus q}}\left(\widehat{\boldsymbol{B}}_k^H\text{diag}(e_i)\text{diag}(e_j)\widehat{\boldsymbol{B}}_\ell\otimes\overline{\Delta_{k,i}}\Delta_{\ell,j}\right)\\
&=\sum_{i=1}^{N_{\setminus q}}\left(\widehat{\boldsymbol{B}}_k^H\text{diag}(e_i)\text{diag}(e_i)\widehat{\boldsymbol{B}}_\ell\otimes\overline{\Delta_{k,i}}\Delta_{\ell,i}\right)\\
&=\sum_{i=1}^{N_{\setminus q}}\left((\text{diag}(e_i)\widehat{\boldsymbol{B}}_k)^H\text{diag}(e_i)\widehat{\boldsymbol{B}}_\ell\otimes\overline{\Delta_{k,i}}\Delta_{\ell,i}\right)=\sum_{i=1}^{N_{\setminus q}}\left(\overline{\widehat{\boldsymbol{B}}}_k(i,:)\circ\widehat{\boldsymbol{B}}_\ell(i,:)\otimes\overline{\Delta_{k,i}}\Delta_{\ell,i}\right)\ .
\end{aligned}$$

The outer product of two vectors in $\mathbb{C}^{1\times R}$ is of complexity $\boldsymbol{\mathcal{O}}(R^2)$. This product is made for each $1\leq i\leq N_{\setminus q}$ and for each K^2 blocks. Hence, the overall complexity is $\boldsymbol{\mathcal{O}}((KR)^2\prod_{i=1,i\neq q}^{p}n_i)$. In addition, as this matrix has a particular block structure, i.e. this is a band-matrix, its product with a vector of size KRn_q is of small complexity equals to $\boldsymbol{\mathcal{O}}((KR)^2n_q)$. □