On learning with shift-invariant structures

Cristian Rusu

Abstract—We describe new results and algorithms for two different, but related, problems which deal with circulant matrices: learning shift-invariant components from training data and calculating the shift (or alignment) between two given signals. In the first instance, we deal with the shift-invariant dictionary learning problem while the latter bears the name of (compressive) shift retrieval. We formulate these problems using circulant and convolutional matrices (including unions of such matrices), define optimization problems that describe our goals and propose efficient ways to solve them. Based on these findings, we also show how to learn a wavelet-like dictionary from training data. We connect our work with various previous results from the literature and we show the effectiveness of our proposed algorithms using synthetic, ECG signals and images.

I. INTRODUCTION

Circulant matrices [1] are highly structured matrices where each column is a circular shift of the previous one. Because of their structure and their connection to the fast Fourier transform [2] (circulant matrices are diagonalized by the Fourier matrix [1] Section 3), these matrices have seen many applications in the past: computing the shift between two signals (the shift retrieval problem exemplified in the circular convolution and cross-correlation theorems) for the GPS locking problem [3], time delay estimation [4], compressed shift retrieval from Fourier components [5], matching or alignment problems for image processing [6], designing numerically efficient linear transformations [7] and overcomplete dictionaries [8], matrix decompositions [9], convolutional dictionary learning [10], [11], [12] and sparse coding [13]. Learning shift invariant structured from data [14],[15] for medical imaging [16], EEG [17] and audio [18] signal analysis. A recent review of the methods, solutions and applications related to circulant and convolutional representations is given in [19].

In this paper we propose several, numerically efficient, algorithms to extract shift-invariant components or alignments from data using circulant matrices.

Since this work is based on circulant matrices, we start by outlining in Section II some of their properties that we will use continuously, particularly their factorization with the Fourier matrix. Then, we deal with learning shift-invariant (circulant and unions of circulant) and wavelet-like components from training data (Section III) and the shift retrieval problem (Section IV). Finally, in Section V, we show experimental results with various data sources (synthetic, ECG, images) that highlight the learning capabilities of the proposed methods.

Notation: bold lowercase $x \in \mathbb{R}^n$ is used to denote column a vector, bold uppercase $X \in \mathbb{R}^{n \times m}$ is used to denote a matrix, non-bold lowercase Greek letters like $\alpha \in \mathbb{R}$ are used to denote scalar values, calligraphic letters $\mathcal{K}$ are used to denote sets and $|\mathcal{K}|$ is the cardinality of $\mathcal{K}$ (abusing this notation, $|\alpha|$ is the magnitude of a scalar). Then $\|x\|_2$ is the $\ell_2$ norm, $\|X\|_F$ is the Frobenius norm, $\text{tr}(X)$ denotes the trace, $\text{vec}(X) \in \mathbb{R}^{nm}$ vectorizes the matrix $X \in \mathbb{R}^{n \times m}$ columnwise, $\text{diag}(x)$ denotes the diagonal matrix with the vector $x$ on its diagonal, $X^H$ is the complex conjugate transpose, $X^T$ is the matrix transpose, $X^\ast$ is the complex conjugate, $X^{-1}$ denotes the inverse of a square matrix, $x_{kj}$ is the $(k,j)^{th}$ entry of $X$. Tilde variables like $\tilde{X}$ represents the columnwise Fourier transform of $X$, $X \otimes Y$ denotes the Kronecker product, $X \otimes Y$ and $X \otimes Y$ denote elementwise multiplication and division, respectively, between two matrices of same size.

II. BRIEF OVERVIEW OF CIRCULANT MATRICES

We consider in this paper circulant matrices $C$. These square matrices are completely defined by their first column $c \in \mathbb{R}^n$: every column is a circular shift of the first one. With a down shift direction the right circulant matrices are:

$$C = \text{circ}(c) = \begin{bmatrix} c & P^2c & \ldots & P^{n-1}c \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (1)$$

The matrix $P \in \mathbb{R}^{n \times n}$ denotes the orthonormal circulant matrix that circularly down-shifts a target vector $c$ by one position, i.e., $P = \text{circ}(e_2)$ where $e_2$ is the second vector of the standard basis of $\mathbb{R}^n$. Notice that $P^{n-1} = \text{circ}(e_n)$ is also orthonormal circulant and denotes a cyclic shift by $q-1$. The main property of circulant matrices [1] is their eigenvalue factorization which reads:

$$C = F^H \Sigma F, \quad \Sigma = \text{diag}(\sigma) \in \mathbb{C}^{n \times n}, \quad (2)$$

where $F \in \mathbb{C}^{n \times n}$ is the unitary Fourier matrix ($F^H F = FF^H = I$) and the diagonal $\sigma = \sqrt{n} e_1$, $\sigma \in \mathbb{C}^n$.

Note that the multiplication with $F$ on the right is equivalent to the application of the Fast Fourier Transform, i.e., $Fc = \text{FFT}(c)$, while the multiplication with $F^H$ is equivalent to the inverse Fourier transform, i.e., $F^H c = \text{IFFT}(c)$. Both transforms are applied in $O(n \log n)$ time and memory.

III. LEARNING CIRCULANT AND CONVOLUTIONAL STRUCTURES FROM DATA

Previously, several dictionary learning techniques that accommodate for shift invariance have been proposed: extending the well-known K-SVD algorithm to deal with shift-invariant structures [17], [20], [21], proposing a shift-invariant iterative least squares dictionary learning algorithm [22], extending the dictionary while solving an eigenvalue problem [23], fast online learning approach [24], research that combines shift and 2D rotation invariance [25] and proposing new algorithms that optimize directly the dictionary learning objective functions.
with circulant matrices \([15], [14]\). The convolutional sparse representation model \([26], [27], [28]\) where the dictionary is a concatenation of circulant matrices has been extensively studied in the past. Furthermore, recent work \([13]\) uses tools developed in the sparse representations literature to provide theoretical insights into convolutional sparse coding where the dictionary is a concatenation of banded circulant matrices and its connection to convolutional neural networks \([29]\). Detailed literature reviews of these learning and convolutional sparse representations problems and proposed solutions have been recently described in \([19]\) Section II.

Our goal is to propose numerically efficient dictionary update rules while the sparse recovery is done using standard tools like the orthogonal matching pursuit (OMP) \([30]\).

### A. Shift-invariant dictionary learning

Given a dataset \(Y \in \mathbb{R}^{n \times N}\) and a maximum sparsity level \(s \geq 1\) for the representations \(X \in \mathbb{R}^{n \times N}\), the work in \([15]\) introduces an efficient way of learning a circulant dictionary \(C \in \mathbb{R}^{n \times n}\) by solving the optimization problem:

\[
\begin{align*}
\min_{c, X} & \quad \|Y - CX\|_F^2 \\
\text{subject to} & \quad \|\text{vec}(X)\|_0 \leq sN, \ C = \text{circ}(c), \ |c|_2 = 1.
\end{align*}
\]

(3)

For fixed \(X\), to update \(c\) we develop the objective function to

\[\|Y - CX\|_F^2 = \|FY - \Sigma FX\|_F^2 = \|\tilde{Y} - \Sigma \tilde{X}\|_F^2,\]

(4)

and in order to minimize it we set

\[
\sigma_1 = \frac{\tilde{X}^H \tilde{y}_1}{\|\tilde{X}\|_2^2} \quad \sigma_k = \frac{\tilde{X}^H \tilde{y}_k}{\|\tilde{X}\|_2^2} \quad \sigma_{n-k+2} = \sigma_k, \quad k = 2, \ldots, n,
\]

(5)

where \(\tilde{Y}\) and \(\tilde{X}\) are the rows of \(\tilde{Y} = FY\) and \(\tilde{X} = FX\).

**Result 1.** Given \(Y \in \mathbb{R}^{n \times N}\) and \(X \in \mathbb{R}^{n \times N}\) the best circulant dictionary \(C\) in terms of the Frobenius norm achieves

\[
\min_c \|Y - CX\|_F^2 = \sum_{k=1}^n \left( \|y_k\|_2^2 - \frac{\|\tilde{x}_k \tilde{y}_k\|_2^2}{\|\tilde{x}_k\|_2^2} \right).
\]

(6)

**Proof.** Expand the objective of (3) using the optimal (5) as

\[
\begin{align*}
\|Y - CX\|_F^2 &= \|Y\|_F^2 + \|CX\|_F^2 - 2\text{tr}(CXY^H) \\
&= \|Y\|_F^2 + \|F^H \Sigma FX\|_F^2 - 2\text{tr}(F^H \Sigma FXY^H) \\
&= \|Y\|_F^2 + \|\Sigma \tilde{X}\|_F^2 - 2\text{tr}(\Sigma \tilde{X} \tilde{Y}^H) \\
&= \|Y\|_F^2 + \sum_{k=1}^n \frac{\|\tilde{x}_k \tilde{y}_k\|_2^2}{\|\tilde{x}_k\|_2^2} - 2\sum_{k=1}^n \frac{\|\tilde{x}_k \tilde{y}_k\|_2^2}{\|\tilde{x}_k\|_2^2} \\
&= \|Y\|_F^2 - \sum_{k=1}^n \frac{\|\tilde{x}_k \tilde{y}_k\|_2^2}{\|\tilde{x}_k\|_2^2} = \sum_{k=1}^n \left( \|y_k\|_2^2 - \frac{\|\tilde{x}_k \tilde{y}_k\|_2^2}{\|\tilde{x}_k\|_2^2} \right).
\end{align*}
\]

(7)

In the context of dictionary learning, to obey the unit \(\ell_2\) norm constraint on \(c\) we should normalize the optimal solution \(\sigma \leftarrow \frac{1}{\|\sigma\|_2^2} \sigma\). This is avoided because we can always group this normalizing factor with the representations \(X\) instead of the circulant dictionary, i.e., \(\|\sigma\|_2^2 CX = C\|\sigma\|_2^2 X\). This grouping is correct because the Fourier transform preserves \(\ell_2\) norms and we have that \(\|\sigma\|_2 = |c|_2\), i.e., all the columns of \(C\) are \(\ell_2\) normalized after normalizing \(\sigma\).

The algorithm called C-DLA, first introduced in \([15]\), has low computational complexity that is dominated by the \(O(nN \log n)\) computation of \(\tilde{Y}\) (once) and that of \(\tilde{X}\) (at each iteration). The calculations in (5) take approximately \(2nN\) operations: there are \(\frac{n}{2}\) components in \(\sigma\) to be computed while \(\|\tilde{x}_k\|_2^2\) and \(\tilde{x}_k^H \tilde{y}_k\) take \(2N\) operations each.

**Remark 1** (Removing the DC component from the data). For datasets with a strong DC component the learned circulant dictionary might be \(C \approx \frac{1}{\sqrt{n}} 1_{n \times n}\). Therefore, preprocessing the dataset \(Y\) by removing the mean component is necessary and we have \(\sigma_1 = 0\) in (5) since \(\tilde{y}_1 = 0_{n \times 1}\).

**Remark 2** (On the local optimality of \(C\)). A necessary condition that the optimal circulant dictionary \(C\) obeys is \(\|\sigma\|_2^2 = \sum_{k=1}^n \frac{|\tilde{x}_k \tilde{y}_k|^2}{\|\tilde{x}_k\|_2^2} = 1\), i.e., the \(\ell_2\) norm of the optimal solution \(\sigma\) of the dictionary learning problem (3) is one.

**Remark 3** (Using only a subset of the \(n\) shifts). We can limit which and how many of all the possible \(n\) shifts of \(c\) are allowed. We achieve this by ensuring that rows of \(X\) corresponding to unwanted shifts are zero.

**Remark 4** (Approximating linear operators by circulant matrices). Given \(Y \in \mathbb{R}^{n \times N}\), let us consider the special case of (3) when \(N = n\) and we fix \(X = I\). Now we calculate the closest, in Frobenius norm, circulant matrix to a given linear transformation \(Y\). Because the Frobenius norm is elementwise the optimal solution is directly \(c_k = \frac{1}{N} \sum_{(i-j) \text{mod } n = (k-1)} y_{ij}, k = 1, \ldots, n\). Unfortunately, in general, circulant matrices do not approximate all linear transformations with high accuracy. The result is intuitive since matrices have \(n^2\) degrees of freedom while circulants have only \(n\). Furthermore, if we add other constraints, such as orthogonality for example, the situation is even worse: \([31]\) shows that the set of orthonormal circulants is finite and constructs it. Therefore, researchers proposed approximating a linear transformation by a product of \(O(n)\) circulant, or Toeplitz, and diagonal matrices \([9], [32]\).

### B. Union of circulant dictionary learning

In the previous sections, the algorithms proposed deal with a single circulant or convolutional structure. We consider now overcomplete dictionaries that are unions of such structures.

Consider a dictionary which is the union of \(L\) circulants:

\[
D = [C^{(1)} \ C^{(2)} \ \ldots \ C^{(L)}] \in \mathbb{R}^{n \times nL},
\]

(8)

where each \(C^{(l)} = \text{circ}(c^{(l)}) = F^H \Sigma^{(l)} F\), \(\Sigma^{(l)} = \text{diag}(\sigma^{(l)})\), \(\sigma^{(l)} = F c^{(l)}\), is a circulant matrix. Given training data \(Y \in \mathbb{R}^{n \times N}\), with this structure, the dictionary learning
Algorithm 1 – UCirc–DLA–SU.

Input: The dataset $Y \in \mathbb{R}^{n \times N}$, the number of circulant atoms $L$ and the sparsity $s \leq n$.

Output: The union of $L$ circulant dictionaries $D \in \mathbb{R}^{n \times nL}$ as in (8) and the sparse representations $X \in \mathbb{R}^{nL \times N}$ such that $\|Y - DX\|_F^2$ is reduced.

1. Initialization: compute the singular value decomposition of the dataset $Y = U \Sigma V^T$, set $c^{(l)} = u_k$ for $l \leq n$, set $c^{(l)}$ to random $\ell_2$ normalized vectors of size $n$ for $L \geq \ell > n$ and compute the representations $X = \text{OMP}(D, Y, s)$.

2. Compute Fourier transform $\tilde{Y}$, set its first row to zero.

3. For $1, \ldots, K$:
   - Update dictionary: – Compute all the $L$ Fourier transforms $\tilde{X}^{(l)}$, $l = 1, \ldots, L$.
     - Construct optimal $\{\sigma_k^{(l)}\}_{k=1}^L$ by (10): $\{\sigma_1^{(l)}\}_{l=1}^L = 0$ and compute $\{\sigma_k^{(l)}\}_{l=1}^L$, $\{\sigma_{n-k+2}^{(l)}\}_{l=1}^L = 1$, $k = 2, \ldots, \left\lfloor \frac{n}{2} \right\rfloor + 1$ and normalize the $L$ circulants $\sigma_k^{(l)} \leftarrow \|\sigma_k^{(l)}\|_1^{-1} \sigma_k^{(l)}$.
   - Update sparse representations $X = \text{OMP}(D, \tilde{Y}, s)$.

The problem has the objective:

$$\|Y - DX\|_F^2 = \|Y - [C^{(1)} \ C^{(2)} \ldots \ C^{(L)}] X\|^2_F$$

$$= \|Y - [F^H \Sigma^{(1)} F \ldots F^H \Sigma^{(L)} F] X\|^2_F$$

$$= \|Y - F^H [\Sigma^{(1)} F \ldots \Sigma^{(L)} F] X\|^2_F$$

$$= \|FY - L \sum_{l=1}^L \Sigma^{(l)} FX^{(l)}\|^2_F$$

$$= \|FY - L \sum_{l=1}^L \Sigma^{(l)} FX^{(l)}\|^2_F = \|\tilde{Y} - L \sum_{l=1}^L \Sigma^{(l)} \tilde{X}^{(l)}\|^2_F$$

where the tilde matrices indicate the Fourier transforms (taken columnwise) and the representations $X \in \mathbb{R}^{nL \times N}$ are separated row-wise into $L$ continuous non-overlapping blocks of size $n$ denoted $X^{(l)} \in \mathbb{R}^{n \times N}$. A way to update all circulant components using the Fourier transforms $\tilde{Y}$ and $\tilde{X}$ presents itself. Denote by $\tilde{Y}^{(k)}$ the $k$th row of $\tilde{Y}$ and by $\tilde{X}^{(k)}$ the $k$th row of $\tilde{X}^{(l)}$. The objective function of the dictionary learning problem separates into $k = 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor + 1$ (given real valued training data $Y$) distinct least squares problems like:

$$\min_{\sigma_k^{(1)}, \ldots, \sigma_k^{(L)}} \|\tilde{Y}^{(k)} - L \sum_{l=1}^L \sigma_k^{(l)} \tilde{X}^{(k)}\|^2_F.$$ (10)

Therefore, for a fixed $k$, the diagonal entries $(k, k)$ of all $\Sigma^{(l)}$ (which are denoted $\sigma_k^{(l)}$) are updated simultaneously by solving the least squares problems (10). Given real-valued data, mirror relations $\sigma_{n-k+2}^{(l)} = (\sigma_k^{(l)})^*$, $k = 2, \ldots, n$, hold analogously to (5) for all $l = 1, \ldots, L$. Notice that this formulation is just a natural extension of the one dimensional least squares problems in (5). To compute all the components of all $\Sigma^{(l)}$ we solve this least squares problem $\frac{n}{2}$ times – the computational complexity is $O(nL^2N)$.

The union of dictionaries learning method presented in [15], UC–DLA, updates each circulant block $C^{(l)}$ sequentially and separately (this can be seen as a block coordinate descent approach). The new proposed learning method, called Union of Circulant Dictionary Learning Algorithm with Simultaneous Updates (UCirc–DLA–SU), is described in Algorithm 1.

Remark 5 (Updating an unused circulant component). Assuming that $X^{(l)} = 0_{n \times N}$, i.e., the $l$th circulant matrix is never used in the representations, then we use the update $c^{(l)} = \arg \max_{c \neq \ell} \|Y - L_{\ell=1,i \neq \ell} C^{(i)} X^{(i)}\|_2^2$. This is the block update method used in UC–DLA [15]. Furthermore, similarly to [33], this update could be used also when atoms of block $\ell$ have a lower contribution to the reconstruction than atoms from other blocks, i.e., $\|X^{(l)}\|_F^2 \ll \|X^{(i)}\|_F^2$, $\forall i \neq \ell$.

C. Union of convolutional dictionary learning

Convolutional dictionary learning can be reduced to circulant dictionary learning by observing that given $c \in \mathbb{R}^n$ and $x \in \mathbb{R}^m$ with $m \geq n$ the result of their convolution is a vector $y$ of size $p = n + m - 1$

$$y = c * x = \text{circ} \left( \left[ \begin{array}{c} c \\ 0_{(m-1) \times 1} \end{array} \right], \left[ \begin{array}{c} x \\ 0_{(n-1) \times 1} \end{array} \right] \right) = C_{\text{conv}} X_{\text{conv}},$$ (11)

When the dictionary is a union of $L$ convolutional matrices as

$$D_{\text{conv}} = \left[ C^{(1)}_{\text{conv}} \ C^{(2)}_{\text{conv}} \ldots \ C^{(L)}_{\text{conv}} \right] \in \mathbb{R}^{p \times pL},$$ (12)

and the objective function to minimize with respect to this union dictionary given the fixed representations $X_{\text{conv}} \in \mathbb{R}^{p \times nL \times N}$ (separated into $L$ continuous non-overlapping blocks denoted $X_{\text{conv}}^{(l)} \in \mathbb{R}^{p \times nL}$) is developed as

$$\|Y - D_{\text{conv}} X_{\text{conv}}\|_F^2 = \|\text{vec}(\tilde{Y}) - L \sum_{l=1}^L \text{vec}(\Sigma_{\text{conv}}^{(l)} \tilde{X}^{(l)}_{\text{conv}})\|_F^2$$

$$= \|\tilde{Y} - L \sum_{l=1}^L A^{(l)} F_{1:n} c^{(l)}\|_F^2 = \|\tilde{Y} - Bc\|_F^2,$$

where $B = \left[ A^{(1)} F_{1:n} \ A^{(2)} F_{1:n} \ldots A^{(L)} F_{1:n} \right] \in \mathbb{R}^{PN \times nL}$, $A^{(l)} = \left[ \tilde{X}^{(l)}_{1:n} \otimes e_1 \ldots \tilde{X}^{(l)}_{1:n} \otimes e_p \right] \in \mathbb{R}^{PN \times p}$, with the rows of $X_{\text{conv}}$, $c \in \left[ c^{(1)} \ c^{(2)} \ldots c^{(L)} \right] \in \mathbb{R}^{nL}$ and $F_{1:n} \in \mathbb{R}^{PN \times n}$ is the $p \times p$ Fourier matrix restricted to its first $n$ columns. The solution here is given by the least squares

$$c = (B^H B) \backslash B^H \tilde{y},$$ (14)

where $B^H B \in \mathbb{R}^{nL \times nL}$ is a positive definite block symmetric matrix, where each $n \times n$ block is a Toeplitz matrix like $T_{\ell_1 \ell_2} = F_{1:n} (A^{(l_1)})^H A^{(l_2)} F_{1:n}$ for the $(\ell_1, \ell_2)$th block – the diagonal blocks are symmetric positive definite Toeplitz. Therefore, $B^H B$ is determined by $nL + (2n - 1)(\frac{L(L-1)}{2})$ parameters – the first term covers the parameters of the $L$ symmetric Toeplitz diagonal blocks and the second terms covers the parameters of all $\frac{L(L-1)}{2}$ non-diagonal Toeplitz blocks. The computational burden is highest in order to calculate the diagonals $W^{(\ell_1 \ell_2)} = (A^{(l_1)})^H A^{(l_2)}$ with entries $w_{\ell_1 \ell_2}^{(l_1 \ell_2)} = (\tilde{X}^{(l_1)}_{\ell_1})^H \tilde{X}^{(l_2)}_{\ell_2}$, i.e., the inner product of the corresponding rows from $\tilde{X}^{(l_1)}_{\text{conv}}$ and $\tilde{X}^{(l_2)}_{\text{conv}}$, respectively. These calculations take $O(pL^2N)$ operations. The inverse Fourier
Algorithm 2 – UConv-DLA-SU.

Input: The dataset \( Y \in \mathbb{R}^{p \times N} \), the number of convolutional atoms \( L \), the length of \( c \) denoted \( n \), the length of the input signals \( m \geq n \) (both \( n \) and \( m \) are chosen such that \( p = n + m - 1 \)) and the sparsity \( s \leq m \).

Output: The union of \( L \) convolutional dictionaries \( D_{\text{conv}} \in \mathbb{R}^{p \times L} \) as in (12) and the sparse representations \( X_{\text{conv}} \in \mathbb{R}^{pL \times N} \) such that \( \|Y - D_{\text{conv}}X_{\text{conv}}\|_F^2 \) is reduced.

1. Initialization: set \( c^{(l)} \) for \( l = 1, \ldots, L \), to random \( 2 \) normalized vectors of size \( p \) with non-zero elements in the first \( n \) entries and compute \( X = \text{OMP}(D_{\text{conv}}, Y, s) \).

2. Compute Fourier transform \( \tilde{Y} \), set its first row to zero.

3. For \( 1, \ldots, K \):
   - Update dictionary: compute all the \( L \) Fourier transforms \( \tilde{X}^{(l)}_{\text{conv}}, \ell = 1, \ldots, L \).
   - Construct \( v = (v^{(1)} \ldots v^{(L)}) \): for \( \ell = 1, \ldots, L \) set \( \tilde{z}^{(k)}_\ell = (\tilde{x}^{(k)}_\ell)^H \tilde{y}_\ell \) and compute \( z^{(k)}_\ell = (\tilde{z}^{(k)}_\ell)^H \tilde{x}^{(k)}_\ell + (\tilde{z}^{(k)}_\ell)^H \tilde{y}_\ell \), where we have defined \( \tilde{X}_{\text{conv}} \) as in (14).
   - Explicitly construct \( B^H B \): for \( \ell_1 = 1, \ldots, L \) and \( \ell_2 = 1, \ldots, \ell_1 \), compute first column and row of the block \( T_{\ell_1, \ell_2} \) by computing Fourier transform of \( w_{\ell_1, \ell_2} = (\tilde{X}^{(k)}_{\ell_1})^H \tilde{s}^{(k)}_{\ell_2} \), where \( \tilde{s}^{(k)}_{\ell_2} = (\tilde{s}^{(1)}_{\ell_2} \ldots \tilde{s}^{(L)}_{\ell_2})^T \).
   - Get \( c \), solve (14) by the Cholesky decomposition and normalize the \( L \) convolutions \( c^{(l)} \leftarrow \|c^{(l)}\|_2^{-1} c^{(l)} \).
   - Update sparse representations \( X = \text{OMP}(D_{\text{conv}}, Y, s) \).

Transforms of \( W^{(\ell_1, \ell_2)} \) to recover the entries of \( T_{\ell_1, \ell_2} \) take only \( O(p \log_2 p) \) operations.

The inverse problem in (14) can be solved exactly in \( O(n^3 L^3) \) via block Cholesky factorization [34] Chapter 4.2. When \( nL \) is large, an alternative approach is to use some iterative procedure like the Conjugate Gradient approach (already used in convolutional problems [35]) where the computational burden falls on computing matrix-vector products with the matrix \( B^H B \) which take only \( O(p L^2) \) operations.

The large matrix \( A_{\text{conv}}^{(1:n,1:m)} \) is never explicitly constructed in the computation of \( c = (F^H_{1:n,1:m} A^H F_{1:n,1:m})^{-1} F^H_{1:n,1:m} A^H \tilde{Y} = (F^H_{1:n,1:m} W F_{1:n,1:m})^{-1} v \), where we have defined \( W = \text{diag}(\|\tilde{x}_1\|_2^2 \ldots \|\tilde{x}_n\|_2^2) \), \( v = F^H_{1:n,1:m} A^H \tilde{Y} \).

The inverse problem in (14) can be solved exactly in \( O(n^3 L^3) \) via block Cholesky factorization [34] Chapter 4.2. When \( nL \) is large, an alternative approach is to use some iterative procedure like the Conjugate Gradient approach (already used in convolutional problems [35]) where the computational burden falls on computing matrix-vector products with the matrix \( B^H B \) which take only \( O(p L^2) \) operations.

The new proposed learning method, called Union of Convolutional Dictionary Learning Algorithm with Simultaneous Updates (UConv-DLA-SU), is described in Algorithm 2.

The importance of dictionaries that are unions of convolutional matrices where the first columns have different (and potentially overlapping) supports has been highlighted in [13].
to (13) since in the Fourier domain the overall problem is decoupled into a series of smaller size independent subproblems that can be efficiently solved in parallel.

**Remark 8 (A special case of time-frequency synthesis dictionary).** Union of circulant matrices also show up when studying discrete time-frequency analysis/synthesis matrices [37]. Consider the Gabor synthesis matrix given by

\[ G = [D^{(1)} C \ D^{(2)} C \ldots \ D^{(m)} C] \in \mathbb{C}^{m \times m^2}, \]

where \( C = \text{circ}(g) \) for \( g \in \mathbb{C}^m \) which is called the Gabor window function, i.e., the matrix \( G \) contains circular shifts and modulations of \( g \). The matrices \( D^{(t)} \), \( t = 1, \ldots, m \), are diagonal with entries \( \delta_{kk} = \omega^{(t-1)(k-1)} \) and \( \omega = e^{2\pi i/m} \).

In the context of compressed sensing with structured random matrices, Gabor measurement matrices have been used for sparse signal recovery [38]: the Gabor function of length \( m \) is chosen with random entries independently and uniformly distributed on the torus \( \{ z \in \mathbb{C} \mid |z| = 1 \} \) [38 Theorem 2.3]. Here, our goal is to learn the Gabor function \( g \) from a given dataset \( Y \) such that the dictionary \( G \) allows good sparse representations. Now the objective function develops to:

\[
\| Y - GX \|_F^2 = \| Y - D(I \otimes C)X \|_F^2 = \| y - ((I \otimes \circledast F)X^T \otimes D(I \otimes \circledast F^H)) \sigma \|_F^2
\]

\[
= \| y - \sum_{t=1}^m X_1^{(t)} \circledast d_1^{(t)} \ldots \sum_{t=1}^m X_m^{(t)} \circledast d_m^{(t)} \|_F^2 = \| y - A \sigma \|_F^2 = \| y - A \sigma_{\circ} \|_F^2,
\]

where we have \( A \in \mathbb{C}^{mN \times m} \) and we have denoted \( y = \text{vec}(Y) \), \( D = [D^{(1)} \ldots D^{(m)}] \), \( \tilde{X}_k^{(t)} \) is the \( k \)-th column of \( D^{(t)} F^H \). Similarly to the convolutional dictionary learning case, Gabor atoms typically have compact support and we can add the sparse structure to \( g \) (the support of size \( n \leq m \)) and find the minimizer by solving the least squares problem which this time is unstructured (there is no Toeplitz structure).

**D. Wavelet-like dictionary learning**

Consider now the following structured matrix

\[ C_k^{(p)} = [G_k S H_k S] \in \mathbb{R}^{p \times p}, \]

where \( G_k = \text{circ}(g_k) \) and \( H_k = \text{circ}(h_k) \) are both \( p \times p \) circulant matrices and \( S \in \mathbb{R}^{p \times \frac{p}{2}} \) is a selection matrix that keeps only every other column, i.e., \( G_k S = [g_k^0 P^2 g_k \ldots P^{(p-2)} g_k] \in \mathbb{R}^{p \times \frac{p}{2}} \) (down-sampling the columns by a factor of 2). In general we assume that the filters \( g_k \) and \( h_k \) have compact support with length denoted \( n \leq p \).

Now we define a new transformation that operates only on the first \( \frac{p}{2} \) coordinates and keeps the latter unchanged:

\[ W_k^{(p)} = \begin{bmatrix} C_k^{(p)} & 0 & \ldots & 0 \\ 0 & P^{(p-2)} & \ldots & P^{(p-\frac{p}{2})} \end{bmatrix} \in \mathbb{R}^{p \times p}. \]

Finally, we define a wavelet-like transformation that is a cascade of the fundamental stages (21) as

\[ W = W_1^{(p)} \ldots W_{m-1}^{(p)} W_m^{(p)}. \]

In the spirit of dictionary learning, our goal is to learn a transformation from given data such that it has sparse representations. The strategy we use is to update each \( W_k^{(p)} \) (actually, the \( C_k^{(p)} \) component) while keeping all the other transformations fixed. Therefore, for the \( k \)-th component we want to minimize

\[
\| Y - WX \|_F^2 = \| Y - W_k^{(p)} W_B X \|_F^2
\]

\[
= \| Y - [W_{A,1} \ W_{A,2}] \begin{bmatrix} C_{1-2}^{(p)} & 0 \\ 0 & I \end{bmatrix} X \|_F^2
\]

\[
= \| Y - W_{A,1} C_{1-2}^{(p)} X \|_F^2
\]

\[
= \| Y - W_{A,1} C_{k-1}^{(p)} X_{1-2} \|_F^2
\]

\[
= \| Y - W_{A,1} C_{k-1}^{(p)} X_{1-2} \|_F^2
\]

\[
= \| y - A \begin{bmatrix} \text{circ}(g_k) \ \text{circ}(h_k) \end{bmatrix} \|_F^2 = \| y - A \begin{bmatrix} \text{circ}(g_k) \ \text{circ}(h_k) \end{bmatrix} \|_F^2,
\]

where \( F \) is the Fourier matrix of size \( \frac{p}{2} \), we denoted \( W_A = W_{A,1} \ldots W_{A,m-1} W_{A,m} \) and \( X = W_B Y \), \( \bar{Y} = Y - W_{A,2} \hat{X}_2 \), \( \bar{y} = \text{vec}(Y) \), \( W_{A,1} \) are the first \( \frac{p}{2} \) columns of \( W_A \), \( \hat{X}_1 \) are the first \( \frac{p}{2} \) rows of \( \hat{X} \). We have also denoted \( \Sigma_{g_k} = \text{diag}(\text{circ}(g_k)) \) and similarly for \( \Sigma_{h_k} \). The matrix \( A \in \mathbb{C}^{mN \times \frac{p}{2}} \) is made up of a subset of the columns from \( (I_2 \otimes \text{FS}) \tilde{X}_1^T \otimes W_{A,1} F^H \) corresponding to the non-zero entries from \( \text{vec}((\Sigma_{g_k}, \Sigma_{h_k})) \). Minimizing the quantity in (23) leads to a least squares problem where both \( g_k \) and \( h_k \) have a fixed non-zero support of known size \( n \). It obeys \( n \leq \frac{p}{2} \) such that the circulants for \( C_{1-2}^{(p)} \) can be constructed.

Similarly to the union of circulant cases described before, some computational benefits arise when minimizing (23), i.e., computing \( (I_2 \otimes F^H) A^H (I_2 \otimes F) \), where \( A = (I_2 \otimes \text{FS}) \tilde{X}_1^T \otimes W_{A,1} F^H \) and \( R = W_{A,1} F^H \), such that \( A = Q \otimes R \) where \( \otimes \) is the Khatri-Rao product [36]. Notice that \( A^H A = \begin{bmatrix} D^{(1)} & D^{(2)} \end{bmatrix} \begin{bmatrix} D^{(1)*} & D^{(2)*} \end{bmatrix} \),

where the blocks are diagonal with entries \( d_{ii}^{(1)} = |q_i|^2 \| r_i \|_2^2 \) and \( d_{ii}^{(2)} = (q_i^T r_i + i \| r_i \|_2^2) \) where \( q_i \) and \( r_i \) are columns of \( Q \) and \( R \), respectively, \( i = 1, \ldots, \frac{p}{2} \). Therefore, \( (I_2 \otimes F^H) A^H A (I_2 \otimes F) \) is a \( 2 \times 2 \) block matrix whose blocks are real-valued circulant matrices (and the diagonal blocks are also symmetric). Also, because \( A^H A \) is symmetric positive definite it allows for a Cholesky factorization \( LL^T \) where the matrix \( L \) has only the main diagonal and the secondary lower diagonal of size \( \frac{p}{2} \) of non-zero values. A further computational benefit comes from when \( n \ll p \) and we solve a least squared problem in \( 2n \) variables, as compared to \( 2p \), i.e., \( A (I_2 \otimes F) \begin{bmatrix} G_k \\ H_k \end{bmatrix} = \bar{A} (I_2 \otimes F, 1:n) \begin{bmatrix} G_k \\ H_k \end{bmatrix}, \) with both \( G_k, H_k \in \mathbb{R}^n \). Finally, notice that \( (I_2 \otimes F^H, 1:n) A^H A (I_2 \otimes F, 1:n) \) has Toeplitz blocks, like in the case of uConv–DLA–SU.

The linear transformation (23) has two major advantages: i) the computational complexity of matrix-vector multiplications \( Wx \) with a fixed \( x \in \mathbb{R}^p \) is controlled by the number of
1. Initialization: set all stages $C_k^{(n)} = I$, $k = 1, \ldots, m$; compute the singular value decomposition of the dataset $Y = USV^T$ and compute the sparse representations $X = \mathcal{T}_f(U^T Y)$, i.e., project and keep the $s$ largest entries in magnitude for each column (column) in the dataset $Y$.

2. For $1, \ldots, K$:
   - Update dictionary: with all other components fixed, update only the $k$th non-trivial component of $W$ denoted $C_k^{(m)}$ (by computing both $g_k$ and $h_k$ on the support of size $n$) for each $k = 1, \ldots, m$, at a time by minimizing the least squares problem.
   - Update $D$ such that $WD$ has unit $\ell_2$ norm columns and update sparse representations $X = \text{OMP}(WD, Y, s)$.

IV. Shift retrieval problems

Given two signals $x, y \in \mathbb{C}^n$ assuming that $y$ is a cyclic shift of $x$ in order to recover it we maximize the inner product:

$$\text{arg max}_q \mathbb{R}\{x^H P^q y\},$$

where $P^q \in \mathbb{R}^{n \times n}$ denotes a cyclic shift by $q$. Practically, to recover the shift we use the circular cross-correlation theorem:

$$\text{max} \mathbb{R}\{\text{IFFT}(\text{FFT}(x)^* \odot \text{FFT}(y))\}.$$ (25)

If the two signals are circularly shifted version of each other (by the amount $q - 1$) then the result of this calculation is $e_q$.

Alternatively we can assume that $y = P^{q-1} x$, and then $y = P^{q-1 - n} x$ with $P = \text{circ}(e_2)$. We consider the problem

$$\text{minimize}_q \|y - P^{q-1} x\|^2_F.$$ (26)

Use $P^{q-1} = F^H \Sigma F$ and similarly to develop $\|y - P^{q-1} x\|^2_F = \|\tilde{y} - \tilde{\Sigma} \tilde{x}\|^2_F$, where $\Sigma = \text{diag}(\sigma)$, $\sigma = Fe_q$ (the $q$th column of the Fourier matrix). If we relax the constraint and allow $P^{q-1}$ to be any circulant matrix, to minimize the Frobenius norm, as the special case of (5) for $N = 1$, we have $\sigma_i = \hat{y}_i / \hat{x}_i$, $\hat{x}_i \neq 0$, and $Fe_q = \tilde{y} \odot \tilde{x}$. We have:

$$\text{IFFT}(\text{FFT}(y) \odot \text{FFT}(x)) = e_q.$$ (27)
reduces the complexity of the shift retrieval problem to \(O(n)\), as also observed for the compressive shift retrieval result \(^5\).

There is of course a connection to the circular cross-correlation theorem. We can rewrite (28) as:

\[
\text{IFFT}(\text{FFT}(y) \odot \text{FFT}(x)) = \text{IFFT}(\text{FFT}(y)^* \odot \text{FFT}(y) \odot |\text{FFT}(x)|^2),
\]

where \( |\text{FFT}(x)|^2 \) computes the square absolute values of each element of the Fourier transform of \(x\). Notice that (28) represents a variant of (29). If the two signals \(x\) and \(y\) are shifted versions of each other then (26) and (28) provide the same answer. If this is not the case, or the signals are noisy, then (28) is a weaker result in general since the minimizer \(P^{q-1}\) in (27) might no longer have \(P = \text{circ}(e_2)\), but some other circulant matrix that minimizes (27).

### A. Compressive shift retrieval

Recently, the compressive shift retrieval problem has been introduced \(^5\). Define the sensing matrix \(A \in \mathbb{C}^{m \times n}, m \leq n\), and the compressed measurement signals \(z = Ay \in \mathbb{C}^m\) and \(v = Ax \in \mathbb{C}^m\). Assuming that \(y\) is a cyclic shift of \(x\), the goal is to determine the shift from \(z\) and \(v\). Similarly to (25), consider the test (Corollary 2 in \(^5\)):

\[
\arg\max_q \Re\{z^H \hat{P}^q v\},
\]

where \(\hat{P}^q = AP^q A^H\). It has been shown that when \(A\) is taken to be a partial Fourier matrix then (Corollary 4 in \(^5\)):

\[
\max_{q \in \{0, \ldots, n-1\}} \Re \left\{ \sum_{i=1}^{m} e^{2\pi j ki/n} \right\},
\]

reverses the true shift if there exists \(p \in \{1, \ldots, m\}\) such that \(\hat{x}_{kp} \neq 0\) (the \(k_i\)th coefficient of the Fourier transform of \(x\)) and \(\{1, \ldots, n-1\} \setminus \{k_i\}\) contains no integers. The set \(K = \{k_1, \ldots, k_m\}\) contains the indices of the rows contained in the partial Fourier matrix \(A\). Following \(^5\) Theorem 1, we assume that the sensing matrix \(A\) obeys: \(A^H A P^{q-1} = P^{q-1} A^H A\), \(\exists \alpha \in \mathbb{R}\) such that \(\alpha A A^H = I\) and all columns of \(A\) are different so that there is no shift ambiguity in the measurements.

The compressive shift retrieval result is partly based on the fact that \(A^H A P^{q-1} = P^{q-1} A^H A\). Notice that \(A^H A = F^H \Sigma F\) where the diagonal \(\Sigma\) contains \(\{0, 1\}\) with ones on the positions where the rows of the Fourier matrix are selected (\(K\)).

Result 2 (Circulant compressive shift retrieval with a proof based on circulant matrices). Given \(z = Ay\) and \(v = Ax\) where \(y = P^{q-1} x\), assuming \(v_i \neq 0, i = 0, 1, \ldots, m\) then:

\[
(Fe_q)_K = z \ominus v.
\]

**Proof.** We start again from the least squares problem:

\[
\begin{align*}
\min_q & \quad ||z - AP^{q-1} A^H v||_2^2.
\end{align*}
\]

With the assumption that \(y - P^{q-1} x = 0\) the objective reaches the zero minimum when \(P^{q-1} = F^H \Sigma F, \Sigma = \text{diag}(Fe_q)\): \(Ay - AP^{q-1} A^H x = A(y - P^{q-1} x)\), where we used the commutativity of circulant matrices and that \(A A^H = I\). To develop (32), start again from (2) and the expression of the matrix multiplication as \(\text{vec}(A A^H v) = (FA^H v)^T \odot (A^H) v(\Sigma)\). We finally obtain:

\[
\begin{align*}
||z - AP^{q-1} A^H v||_2^2 & = ||z - A^H A v||_2^2 \\
& = \|\text{vec}(z) - \text{vec}(A^H A v)||_2^2 \\
& = ||z - (FA^H v)^T \odot (A^H) v(\Sigma)||_2^2 \\
& = ||z - (FA^H v)^T \odot (A^H) \text{vec}(\Sigma)||_2^2 \\
& = ||z - VFe_q||_2^2,
\end{align*}
\]

where the matrix \(V \in \mathbb{R}^{m \times n}\) contains only the columns of the Kronecker product that match the non-zero elements of the diagonal matrix \(\Sigma\). The matrix contains the elements of \(v\) in positions \((k_i, i)\). The second equality holds because the Frobenius norm is elementwise. It follows that \(VFe_q = z\) and

\[
(Fe_q)_K = V^H (V V^H)^{-1} z = V^H (z \odot |v|^2)
\]

**The compressive shift retrieval is equivalent to (28), the regular shift retrieval, on the set of Fourier components \(K\). This is a unified view of the shift retrieval solutions.**

In relation to (30), we use the circulant structures to reach:

\[
\begin{align*}
\hat{z}^H \hat{P}^q v = z^H A^H \Sigma F \hat{A} F^H v = \text{vec}(z^H A^H \Sigma F v)
\end{align*}
\]

where we expressed the matrix multiplications as a linear transformation on \(\Sigma = \text{diag}(Fe_{q+1})\), with \(q \in \{0, \ldots, n-1\}\) and \(r \in \mathbb{C}^n\) is the expression in the parenthesis with \((r)_K = z^H \ominus v\). The matrix \(FA^H \in \mathbb{R}^{m \times n}\) is a partial permutation matrix – only positions \((k_i, i)\) are non-zero. The products with \(v\) and \(z\) produce extended vectors \((v)_K, (z)_K \in \mathbb{C}^n\). Thus, maximizing \(\hat{z}^H \hat{P}^q v\) reduces to the selection of \(e_{q+1}\).

Due to the natural appearance of the Fourier matrix \(F\) in the factorization of circulant matrices its rows are also the natural choice in the rows of the measurement matrix \(A\). Cancellations that occur because of this choice lead to the analytic results found. This shows a simple alternative, but equivalent, way to develop the result (30) of \(^5\).

### V. Experimental results

We now discuss numerical results that show how well the proposed methods extract shift-invariant structures from data. In our proposed algorithms, in the sparse recovery phase, we use the orthogonal matching pursuit (OMP) algorithm, but any other sparse approximation method could be chosen.

#### A. Synthetic experiments

We create a synthetic dataset generated from a fixed number of atoms and their shifts and the proceed to measure how well we recover them from the dictionary learning perspective. The experimental setup follows: generate \(N = 2000\) signals of length \(n = 20\), that are linear combinations (with sparsity \(s = 4\)) of \(L = 45\) randomly generated kernel columns which are allowed to have only \(q = 3\) circular shifts (out of the possible \(n = 20\)), i.e., \(Y \in \mathbb{R}^{n \times N}\) where each columns is \(y_i = \sum_{l=1}^{L} \alpha_i P^{q-i} e_l + n_i\) for \(i = 1, \ldots, N\) with fixed \(||e_1||_2 = 1,\)
SU achieves lower error approximately each step of the algorithm. We observe that UCirc–DLA–SU of updating all the circulant components simultaneously with the dataset. This shows the benefit (as compared to UC–DLA–SU) methods in the task of recovering the atoms used in creating how the UCirc–DLA–SU outperforms previously proposed vector representing noise.

randomly uniformly distributed and \( n \) \( \| C \| \)

\[ \| \alpha_i \|_0 = s \text{ where } \alpha_{i \ell} \in [-10, 10] \text{ and } q_{i \ell} \in \{0, \ldots, q-1\} \] are randomly uniformly distributed and \( n_i \) is a random Gaussian vector representing noise.

First, using the synthetic dataset, we show in Figure 1 how the UCirc–DLA–SU outperforms previously proposed methods in the task of recovering the atoms used in creating the dataset. This shows the benefit (as compared to UC–DLA–SU) of updating all the circulant components simultaneously with each step of the algorithm. We observe that UCirc–DLA–SU achieves lower error approximately 75% of the time. The typical counter-example is one where UC–DLA converges slower (in more iterations) to a slightly lower representation error, i.e., sub-optimal block calculations ultimately lead to a better final result. This observation is not surprising since both heuristic methods only approximately solve the overall original dictionary learning problem (with unknowns both \( C \) and \( X \)). To show this, with the same synthetic dataset for noise level \( \text{SNR} = 30 \text{dB} \) in Figure 2 we calculate how many times on average each atom in all circulant components (from all the \( L = 45 \)) is used in the sparse representations. On average, UCirc–DLA–SU recovers the correct supports (in effect, the indices of the shifts used) more often than UC–DLA–SU.

Figure 3 shows the learning times for the union of circulants \( Y \), \( L = 45 \) atoms of the \( L = 45 \) circulants \( C_i \). With perfect recovery the \( q = 3 \) peeks should be \( \frac{N_s}{q \cdot L} \approx 44 \).

\[ \frac{\alpha_i}{\| \alpha_i \|_0} = s \text{ where } \alpha_{i \ell} \in [-10, 10] \text{ and } q_{i \ell} \in \{0, \ldots, q-1\} \] are randomly uniformly distributed and \( n_i \) is a random Gaussian vector representing noise.

B. Experiments on ECG data

Electrocardiography (ECG) signals \[ 42 \] have many repetitive sub-structures that could be recovered by shift-invariant dictionary learning. Therefore, in this section, we use the proposed UConv–DLA–SU to find in ECG signals short (compact support) features that are repeated. We use the MIT-BIH arrhythmia database\[ 1 \] from which we extract a normal sinus rhythm signal composed of equality length samples from five different patients, all sampled at 128 Hz. This signal is reshaped into a matrix of centered, non-overlapping sections of length \( p = 64 \), leading to the training dataset \( Y \in \mathbb{R}^{64 \times 10100} \).

Because we are searching for sub-signals with limited support, we use the UConv–DLA–SU with parameters \( n = 12 \), \( s = 2 \) and \( L = 2 \) to recover the shift-invariant structure. In Figure 4 we show the original ECG signal and its reconstruction in the union of convolutional dictionaries. Of course, the reconstruction is not perfect but it is able to accurately capture the spikes in the data and remove some of the high-frequency features, i.e., the signal looks filtered (denoised). The second plot, Figure 5 shows the \( L = 2 \) learned atoms from the data which capture the spiky nature of the training signal.

C. Experiments on image data

The training data \( Y \) that we consider are taken from popular test images from the image processing literature (pirate, peppers, boat etc.). The test dataset \( Y \in \mathbb{R}^{8 \times 8} \) consists of \( 8 \times 8 \) non-overlapping image patches with their means removed. We consider \( N = 12288 \) and we have \( p = 64 \). To evaluate the learning algorithms, in this section we consider the relative

1 https://www.physionet.org/physiobank/database/mitdb/
representation error of the dataset $Y$ in the dictionary $D$ given the sparse representations $X$ as

$$\epsilon = \|Y - DX\|_F^2 \|Y\|_F^{-2} \%.$$  \quad (33)

We consider image data because there are well-known wavelet transforms that efficiently encode such data. We will use the filters of the Haar and Daubechies D4 wavelet transforms, i.e., with $n = 2$, $m = \log_2 p$, $\mathbf{h}_k = \left[\frac{1}{\sqrt{2}} - \frac{1}{\sqrt{2}}\right]$, $\mathbf{g}_k = \left[\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}}\right]$ and $n = 4$, $m = -1 + \log_2 p$, $\mathbf{h}_k = \left[\frac{1-\sqrt{3}}{4\sqrt{2}} \frac{-3-\sqrt{3}}{4\sqrt{2}} \frac{3+\sqrt{3}}{4\sqrt{2}} \frac{-1+\sqrt{3}}{4\sqrt{2}}\right]$, $\mathbf{g}_k = \left[\frac{1+\sqrt{3}}{4\sqrt{2}} \frac{3+\sqrt{3}}{4\sqrt{2}} \frac{3-\sqrt{3}}{4\sqrt{2}} \frac{1-\sqrt{3}}{4\sqrt{2}}\right]$, respectively, for all $k$. The filters are chosen such that the resulting $W$ is orthonormal.

First, we show in Figure 6 the experimental convergence of the proposed W–DLA. Notice from the description of the algorithm that all dictionary update steps necessarily decrease the objective function but, unfortunately, the overall algorithm may not be monotonically convergence since OMP is not guaranteed in general to reduce the objective function. If we also impose an orthogonality constraint on $W$ then we can avoid OMP for the sparse representations and just a projection operation guarantees optimal sparse representations. The figure shows that, when available, it is convenient to initialize the W–DLA with well-known wavelet filters since the algorithm converges faster and to slightly lower representation errors. Still, the differences are not significant while wavelets do not exist for every $n, m$.

Then, in Figure 7 we show how the representation error varies with the sparsity level $s$. We run W–DLA also with an initialization of well-known wavelet filter coefficients Haar and D4, respectively. W–DLA is always able to improve the representation performance, even when starting with the wavelet coefficients. In the Haar case the improvement is small, due to the small number of free filter parameters to learn, i.e., only 24: $m = \log_2 p = 6$ stages each with 2 filters and each with 2 coefficients. In the D4 case, the representation error is improved significantly. Both figures show that, when available, wavelet coefficients provide an excellent initialization even slightly better than the proposed W–DLA (also confirmed in Figure 6). Since these wavelets are not available for all choices $n, m$ the purpose of this plot is to show that the proposed initialization provides very good results in general.

Finally, in Figure 8 we show the effect that parameters $n$ and $m$ have on the representation error. For reference, we show the representation error of C–DLA which has $p = 64$ free parameters to learn. The performance of this dictionary is approximately matched by W–DLA with $n = 4$ and $m = -1 + \log_2 p$ which has 40 free parameters to learn: $m = 5$ stages each with 2 filters of support 4 each. Notice that the representation error plateaus after $n = 8$. In general, dictionaries built with W–DLA have $2nm$ degrees of freedom. We also show a version of W–DLA where we keep $m = 1$ and vary only $m$ in which case, of course, the representation error decreases. Note that in this case each run of W–DLA is initialized with a random set of coefficients. To show monotonic convergence it would help to initialize the filters of size $n$ with those previously computed of support $n - 1$.

VI. CONCLUSIONS

In this paper, we propose several algorithms that learn, under different constraints, shift-invariant structures from data.
We analyze the behavior of the algorithms on various data sources, and show we outperform previously proposed algorithms from the literature.

Our work is based on using circulant matrices and finding numerically efficient closed-form solutions, by least-squares. We analyze the behavior of the algorithms on various data sources, and compare and show we outperform previously proposed algorithms from the literature.

REFERENCES

[1] R. M. Gray, “Toeplitz and circulant matrices: a review,” Found. Trends Commun. Inf. Theory, vol. 2, no. 3, pp. 155–239, 2006.
[2] A. Hero, H. Messer, J. Goldberg, D. Thomson, M. Amin, G. Giannakis, A. Swami, J. Tugnait, A. Nehorai, A. Swindlehurst, J.-F. Cardoso, B. Wohlberg, “Efficient algorithms for convolutional sparse representations,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process., vol. 60, no. 4, pp. 1597–1612, 2012.
[3] S. Jain and J. Haupt, “Convolutional approximations to linear dimensionality reduction operators,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), 2017, pp. 5885–5889.
[4] R. Bristow and S. Lucey, “Optimization methods for convolutional sparse coding,” Technical Report, 2014.
[5] H. Strohmer and T. Ufimtsev, “A comparative review and new algorithms,” IEEE Trans. on Comp. Imag., vol. 4, no. 3, pp. 366–381, 2018.
[6] V. Papan, J. Sultan, and M. Elad, “Working locally thinking globally: Theoretical guarantees for convolutional sparse coding,” IEEE Trans. Sig. Process., vol. 65, no. 21, pp. 5687–5701, 2017.
[7] G. Pope, C. Aubel, and C. Studer, “Learning phase-invariant dictionaries,” in Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP), 2013, pp. 5979–5983.
[8] C. Rusu, B. Dumitrescu, and S. A. Tsatsaronis, “Explicit shift-invariant dictionary learning,” IEEE Sig. Proc. Let., vol. 21, no. 1, pp. 6–9, 2014.

Fig. 8. Representation error for W–DLA as a function of the size of the filter support $n$. For reference we show C–DLA [13] while W–DLA runs twice: once with fixed $m = 1$ and with largest $m$ such that $n \leq 2^m$ is obeyed.