Efficient Convolutional Forward Modeling and Sparse Coding in Multichannel Imaging

Han Wang[*,‡], Yhonatan Kvich[†], Eduardo Pérez[*,‡], Florian Römer[*,‡], Yonina C. Eldar[†]
[*]*Applied AI Signal Processing and Data Analysis, Fraunhofer Institute for Nondestructive Testing*, Saarbrücken, Germany
[†]*Faculty of Math and Computer Science, Weizmann Institute of Science*, Rehovot, Israel
[‡]*Dept. Electronic Measurements and Signal Processing, Technische Universität Ilmenau*, Ilmenau, Germany
{han.wang, eduardo.jose.perez.mejia, florian.roemer}@izfp.fraunhofer.de, {yonatan.kvich, yonina.eldar}@weizmann.ac.il

***Abstract*—This study considers the Block-Toeplitz structural properties inherent in traditional multichannel forward model matrices, using Full Matrix Capture (FMC) in ultrasonic testing as a case study. We propose an analytical convolutional forward model that transforms reflectivity maps into FMC data. Our findings demonstrate that the convolutional model excels over its matrix-based counterpart in terms of computational efficiency and storage requirements. This accelerated forward modeling approach holds significant potential for various inverse problems, notably enhancing Sparse Signal Recovery (SSR) within the context LASSO regression, which facilitates efficient Convolutional Sparse Coding (CSC) algorithms. Additionally, we explore the integration of Convolutional Neural Networks (CNNs) for the forward model, employing deep unfolding to implement the Learned Block Convolutional ISTA (BC-LISTA).**

***Index Terms*—Forward Modeling, Convolutional Sparse Coding, Deep Unfolding, Toeplitz Matrix, Multichannel Imaging**

I. Introduction

When tackling inverse problems to obtain descriptive parameters based on observations, the solution approach is directly related to the forward model describing the observed data [1]. In applications such as ultrasound nondestructive testing and medical imaging, as well as radar localization, the prior knowledge of structural sparsity has led to the usage of algorithms such as Orthogonal Matching Pursuit (OMP) [2], [3], Iterative Shrinkage-Thresholding Algorithm (ISTA) [4], and its accelerated variant, Fast ISTA (FISTA) [5], which require iterative optimization of forward model parameters. Recent strides in model-based deep learning [6] have given rise to algorithms like Learned ISTA (LISTA) [7], [8], reducing the number of iterations in inference, and to task-based compressed sensing applications, ranging from optimal sensor placement to sophisticated subsampling techniques [9]–[13]. Despite their innovations, traditional matrix-based forward models, alongside Multilayer Perceptron (MLP)-based LISTA implementations, grapple with the computational demands imposed by burgeoning data volumes.

In response, deep learning-based solutions [14], [15] and matrix-free forward models [16], [17] have emerged as innovative alternatives. Among these, convolutional forward models, which utilize specific matrix structures to facilitate Convolutional Sparse Coding (CSC) algorithms, represent a significant advancement. Notably, previous studies have proposed convolutional dictionaries as effective substitutes for matrix-based models with a Toeplitz structure [18]. Furthermore, the introduction of a LISTA-Toeplitz model has demonstrated the potential of replacing Gram matrices appearing in the gradient of linear least squares problems, which have a Toeplitz structure in some applications, with convolutional filters, though the approach still harbors a significant count of trainable parameters due to its reliance on linear layers for representing the remaining components of the gradient [19]. Nonetheless, such assumptions do not always align with the realities of multichannel time-delay-based measurements, where instead a more general Block-Toeplitz structure is present [20], [21].

Recognizing this gap, we demonstrate that the inherent structure of Uniform Linear Array (ULA) measurements allows the forward model to be precisely rewritten using ensembles of strided convolutions. Our main contributions are as follows: first, we develop an accurate convolutional forward model by leveraging the block Toeplitz structure of the physical model matrix. Second, we introduce the Block Convolutional FISTA (BC-FISTA) and Block Convolutional LISTA (BC-LISTA) algorithms for signal recovery. BC-FISTA is shown to be mathematically equivalent to the naive FISTA implementation but requires significantly less memory. BC-LISTA enables the learning of convolutional kernels and hyperparameters, providing enhanced flexibility and improved performance. Third, our comparative analysis reveals that the convolutional model significantly outperforms traditional matrix-based models regarding memory efficiency. Furthermore, both proposed CSC algorithms are capable of generating high-quality images.

II. Matrix-Based Forward Model

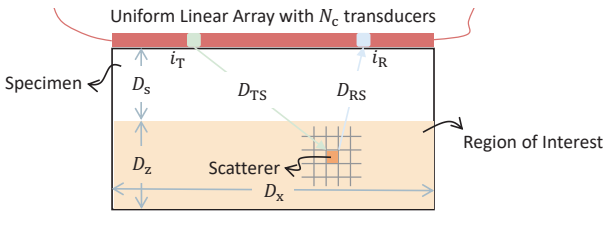


Fig. 1: Two-dimensional illustration of FMC measurement.

Consider a ULA comprising N_c transducers with a pitch size of d_c, strategically positioned on the specimen targeted for

This work was partially supported by the Fraunhofer Internal Programs under Grant No. Attract 025-601128, and the Thuringian Ministry of Economic Affairs, Science and Digital Society (TMWWDG), as well as by Amazon Fund.

detection. Each transducer in this array has dual functionality, capable of both transmitting and receiving signals. The total numbers of transmitters and receivers are respectively denoted by N_{T} and N_{R}. There is a distance D_{s} between the array and the Region of Interest (ROI) to avoid artifacts. The ROI is of shape $(D_{\mathrm{z}} \times D_{\mathrm{x}})$ and discretized into $(N_{\mathrm{z}} \times N_{\mathrm{x}})$ pixels, each measuring $(d_{\mathrm{z}} \times d_{\mathrm{x}})$ in size. Each pixel is a potential scatterer that can reflect the pulse to receivers.

In the experiment, the Full Matrix Capture (FMC) refers to sequentially excite element by element, and all receivers capture the response signals for each transmission. We assume that the received signal $p(t)$ at a given receiver constitutes a superposition of time-delayed, amplitude-scaled replicas of the emitted pulse, each subject to phase inversion and attributable to the interaction with discrete scatterers. Mathematically, the received signal is given by the real-valued Gabor function:

$$p(t) = \sum_{s=1}^{N_{\mathrm{s}}} a_s e^{-\alpha(t-\tau_s)^2} \cos(2\pi f_c(t-\tau_s)) + n(t), \quad (1)$$

where s and N_{s} are the index and total number of scatterers within the ROI, respectively. The variable τ_s signifies the Time of Flight (ToF) and is a function of the spatial coordinates of the transmitter, the receiver, and the individual scatterer. The parameter a_s represents the reflectivity coefficient associated with each scatterer, encapsulating the relative strength of the echo. The term α is indicative of the bandwidth factor, which, along with the center frequency f_c, characterizes the pulse shape. The additional term $n(t)$ accounts for Additive White Gaussian Noise (AWGN) introduced during signal reception. Given the sampling frequency f_{s} and duration, N_{t} samples are obtained per A-scan. Finally, we assemble a three-dimensional measurement data denoted as $\mathbf{Y} \in \mathbb{R}^{N_{\mathrm{t}} \times N_{\mathrm{R}} \times N_{\mathrm{T}}}$

We characterize the ROI by a two-dimensional reflectivity map $\mathbf{X} \in \mathbb{R}^{N_{\mathrm{z}} \times N_{\mathrm{x}}}$. The conventional approach for constructing the forward model commences with the vectorization of the measurement data and the reflectivity map, adhering to the Fortran-style (column-major) format. Subsequently, each column of the model matrix $\mathbf{A}$ is designated to represent the anticipated FMC dataset that would result from a scatterer located at the respective coordinate. In this context, the forward model $\mathbf{A} \in \mathbb{R}^{(N_{\mathrm{t}} N_{\mathrm{R}} N_{\mathrm{T}}) \times (N_{\mathrm{z}} N_{\mathrm{x}})}$ is formulated. This model maps the vectorized reflectivity map $\mathbf{x} \in \mathbb{R}^{N_{\mathrm{z}} N_{\mathrm{x}}}$ to the vectorized FMC measurement data $\mathbf{y} \in \mathbb{R}^{N_{\mathrm{t}} N_{\mathrm{R}} N_{\mathrm{T}}}$. Then this measurement procedure is effectively expressed as a linear transformation:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \quad (2)$$

The matrix-based forward model (2) encounters severe limitations regarding storage capacity and computational efficiency, especially with larger ULAs and wider ROIs. The convolutional forward model presents a viable solution to these challenges, which we introduce next.

III. Convolutional Forward Model

A. Block-Toeplitz Structure Forward Model

As depicted in Fig. 2, the multi-dimensional FMC measurement data $\mathbf{Y}$ can be conceptualized as comprising $(2N_{\mathrm{c}} - 1)$ diagonal slices. Owing to the reciprocity principle between transmitters and receivers, two slices positioned symmetrically about the center slice exhibit identical characteristics. Consequently, the data $\mathbf{Y}$ can be effectively represented by only N_{c} unique slices. We denote a specific 2D data slice by $\mathbf{Y}_{i_{\mathrm{s}}}$, where i_{s} ranges from 0 to $N_{\mathrm{c}} - 1$, and each slice contains $(N_{\mathrm{s}} - i_{\mathrm{s}})$ A-scans. Each A-scan is associated with a unique transmitter-receiver pair, with the relationship between their indices formulated as $i_{\mathrm{R}} = i_{\mathrm{T}} + i_{\mathrm{s}}$. To elucidate further, within the i_{s} slice, the index i_{T} spans the set $[0, ..., N_{\mathrm{c}} - 1 - i_{\mathrm{s}}]$, while i_{R} correspondingly traverses the range $[i_{\mathrm{s}}, ..., N_{\mathrm{c}} - 1]$.

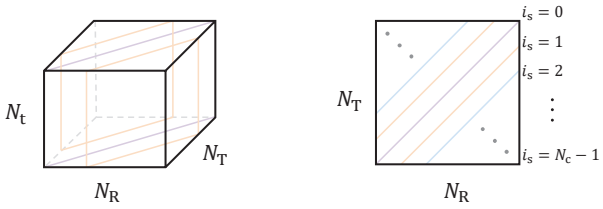


Fig. 2: Slices of 3D FMC data array.

In the context of the slice-wise matrix-based forward model illustrated in Fig. 3, the matrix $\mathbf{B}_{i_{\mathrm{s}}} \in \mathbb{R}^{(N_{\mathrm{c}} - i_{\mathrm{s}}) N_{\mathrm{t}} \times (N_{\mathrm{z}} N_{\mathrm{x}})}$ can derive the vectorized slice $\mathbf{y}_{i_{\mathrm{s}}}$ from the reflectivity map $\mathbf{x}$. The matrix $\mathbf{B}_{i_{\mathrm{s}}}$ is intricately structured into $(N_{\mathrm{c}} - i_{\mathrm{s}}) \times N_{\mathrm{x}}$ blocks. Each block encapsulates the signal determined by the specific coordinates of transmitters, receivers, and scatterers:

- Transmitters: $[i_{\mathrm{T}} \cdot d_{\mathrm{c}},\ 0]$
- Receivers: $[(i_{\mathrm{T}} + i_{\mathrm{s}}) \cdot d_{\mathrm{c}},\ 0]$, where $i_{\mathrm{T}} \in [0, ..., N_{\mathrm{c}} - 1 - i_{\mathrm{s}}]$
- Scatterers: $[i_{\mathrm{x}} \cdot d_{\mathrm{x}},\ i_{\mathrm{z}} \cdot d_{\mathrm{z}} + D_{\mathrm{s}}]$, where $i_{\mathrm{x}} \in [0, ..., N_{\mathrm{x}} - 1]$ and $i_{\mathrm{z}} \in [0, ..., N_{\mathrm{z}} - 1]$.

Using these coordinates and assuming that the pixel width is equal to the pitch size, such that $d_x = d_c$, the ToF is computed via the equation:

$$\tau = \frac{1}{c_0}\Bigg(\sqrt{[(i_{\mathrm{T}} - i_{\mathrm{x}})d_{\mathrm{x}}]^2 + (i_{\mathrm{z}} d_{\mathrm{z}} + D_{\mathrm{s}})^2} + \sqrt{[(i_{\mathrm{T}} + i_{\mathrm{s}} - i_{\mathrm{x}})d_{\mathrm{x}}]^2 + (i_{\mathrm{z}} d_{\mathrm{z}} + D_{\mathrm{s}})^2}\Bigg). \quad (3)$$

It is important to observe that the ToF parameter, τ, remains invariant for constant values of the difference $(i_{\mathrm{T}} - i_{\mathrm{x}})$. The constancy leads to a situation where the blocks along any given diagonal of the matrix $\mathbf{B}_{i_{\mathrm{s}}}$ are identical. Consequently, Fig. 3 demonstrates the Block-Toeplitz structure inherent in the slice-wise matrix-based forward model. The matrix $\mathbf{B}_{i_{\mathrm{s}}}$ is amenable to a more compact representation, utilizing only $(N_{\mathrm{c}} - i_{\mathrm{s}} + N_{\mathrm{x}} - 1)$ unique blocks to encapsulate the entire structure. This tailored configuration implies that the forward model can be efficiently distilled into a series of one-dimensional convolutions, significantly simplifying computational efforts.

B. Convolutional Model for Block-Toeplitz Matrix Structure

To reformulate the matrix-vector multiplication in the Block-Toeplitz structured matrix as convolution operations, the unique blocks are first reoriented and restructured into an elongated matrix $\tilde{\mathbf{B}}_{i_{\mathrm{s}}} \in \mathbb{R}^{N_{\mathrm{t}} \times (N_{\mathrm{c}} - i_{\mathrm{s}} + N_{\mathrm{x}} - 1) N_{\mathrm{z}}}$. Subsequently, executing a row-wise inversion of the matrix $\tilde{\mathbf{B}}_{i_{\mathrm{s}}}$ is essential to aligning

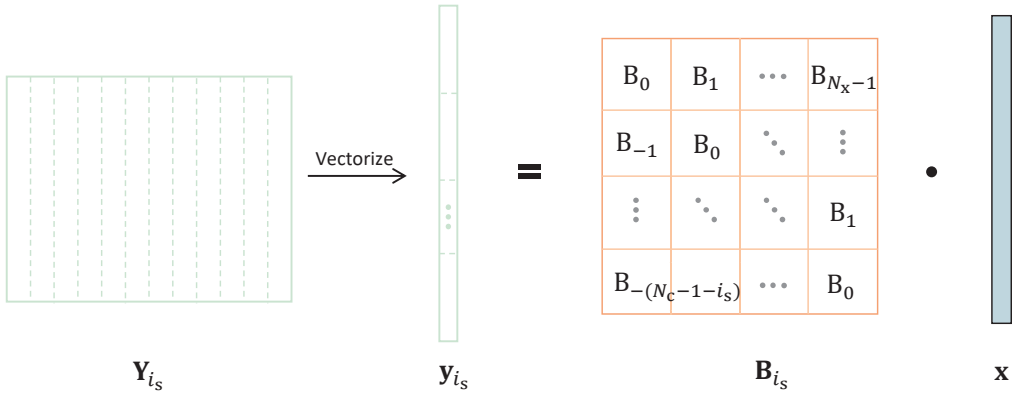


Fig. 3: Slice-wise matrix-based forward model.

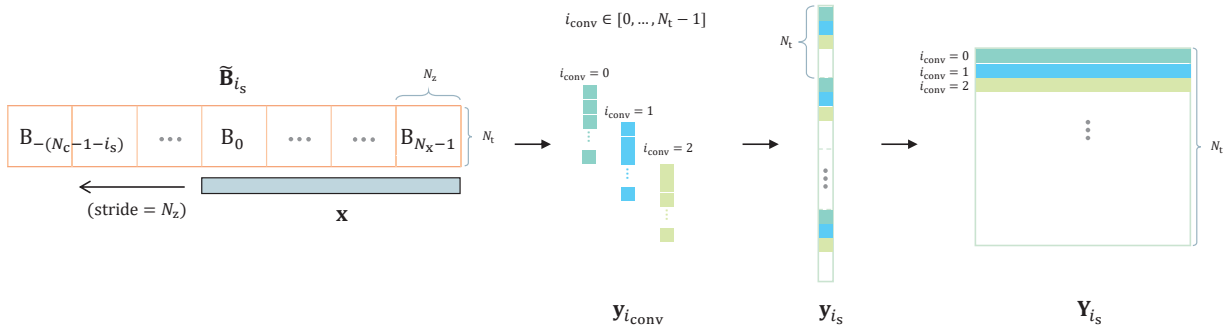


Fig. 4: Matrix-vector product from Fig. 3 rewritten exactly as 1D strided convolutions.

the convolution kernels with the standard convolution operation paradigm. This adjustment allows for the vector $\mathbf{x}$ to be subject to convolution with each flipped row, utilizing a stride of N_z. Each individual one-dimensional convolution operation yields a vector $\mathbf{y}_{i_{conv}} \in \mathbb{R}^{N_c - i_s}$. Upon completion of N_t such convolutions, the corresponding row vectors are concatenated in sequence to construct the data slice $\mathbf{Y}_{i_s}$, a process depicted in Fig. 4.

Transitioning to a CNN framework for the forward model offers compelling advantages such as expedited inference, and efficient backpropagation. This CNN-based methodology can be smoothly integrated with other network architectures or algorithms, thereby enriching its utility and operational effectiveness. To accommodate the CNN structure, it is necessary to append zeros to either end of the input vector $\mathbf{x}$, with the padding determined by $(N_c - i_s - 1)N_z$ and maintaining a stride of N_z. This zero-padding preprocessing step does not impinge upon computational efficiency as the convolution count and kernel dimensions remain unaltered. By setting the number of output channels to N_t, the output is congruent with $\mathbf{Y}_{i_s}$, thereby obviating the need for iterative convolutions and manual assembly of the row vectors.

In our proposed methodology, we instantiate N_c parallel Conv1D layers within our model architecture, with each layer responsible for the computation of a distinct data slice. These computed slices are then meticulously integrated into their corresponding positions to construct the FMC data array, which

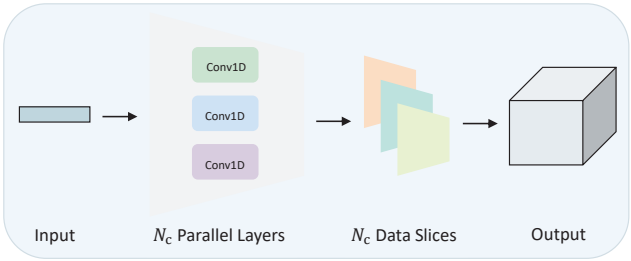


Fig. 5: CNN-based forward model.

is visually delineated in Fig. 5. To summarize, the analytic convolutional forward model comprises N_c different matrices of varying dimensions, denoted $[\tilde{\mathbf{B}}_0, \tilde{\mathbf{B}}_1, ..., \tilde{\mathbf{B}}_{N_c-1}]$. Each matrix serves to initialize the weights of a Conv1D layer configured with N_t output channels.

The analytical convolutional forward model exhibits several advantages over the matrix-based model, with the number of parameters serving as a pivotal aspect. To illustrate a quantitative evaluation of their respective performances, we consider a ULA with channel number, N_c, ranging from 1 to 128. For simplicity, we set $N_x = N_z = N_c$, $d_x = d_z = d_c$, and $D_s = D_z$. The total number of samples, N_t, is dictated by the longest ToF within the ROI. By assuming a single-precision float data type for the parameters, we calculate and compare the storage demands of both models in Megabytes, as shown in Fig. 6. The convolutional model is obviously storage-friendly especially when the array size is larger. Consider a commonly-used 128-channel ULA

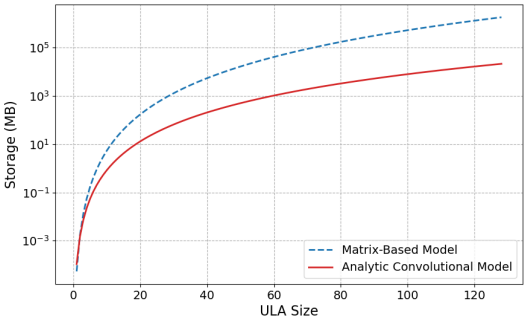


Fig. 6: Model storage requirement comparison.

as an example, under this scenario, the matrix-based forward model requires approximately 1789 GB. In stark contrast, the convolutional forward model necessitates merely 21 GB.

IV. Efficient Convolutional Sparse Coding

The efficiency of our convolutional forward modeling facilitates effective gradient computation for backward passes, making it particularly suited for SSR. This capability underpins the development of efficient CSC algorithms for solving the LASSO function:

$$\min_{\mathbf{x}} \frac{1}{2}\|f(\mathbf{x}) - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{4}$$

where $f(\mathbf{x})$ is the forward model, $\mathbf{y}$ is the observed data, $\mathbf{x}$ is the signal to be recovered, and λ is a regularization parameter that controls the sparsity of the solution.

A. Block Convolutional FISTA

Algorithm 1 Block Convolutional FISTA

Input: Initial guess $\mathbf{x}_0$, auxiliary sequence $\mathbf{z}_0 = \mathbf{x}_0$, Nesterov momentum $t_0 = 1$, number of iterations N_{iter}

1: **for** i to N_{iter} **do**
2: Compute gradient of $\frac{1}{2}\|f(\mathbf{z}_k) - \mathbf{y}\|_2^2$, denoted $\nabla f(\mathbf{z}_k)$
3: Perform a gradient descent step: $\mathbf{x}_k = \mathbf{z}_k - \frac{1}{L}\nabla f(\mathbf{z}_k)$
4: Apply the proximal operator corresponding to the $L1$ regularization term to $\mathbf{x}_k$: $\mathbf{x}_{k+1} = S_{\lambda/L}(\mathbf{x}_k)$
5: $S_\theta(v)$ is the soft-thresholding operator defined as: $S_\theta(v) = \text{sign}(v)\max(|v| - \theta, 0)$
6: Update the momentum term t: $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
7: Update the auxiliary variable $\mathbf{z}$: $\mathbf{z}_{k+1} = \mathbf{x}_{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_{k+1} - \mathbf{x}_k)$
8: **end for**

Output: Reconstructed signal $\hat{\mathbf{x}}$.

The mathematical process of Block Convolutional FISTA (BC-FISTA) is summarized in Algorithm 1. In practical implementations, the calculation of the gradient $\nabla f(\cdot)$ and the Lipschitz constant L is contingent upon the explicit formulation of $f(\cdot)$. For instance, in scenarios where $f(\mathbf{x})$ represents a linear operation $f(\mathbf{x})$, the gradient is determined as $\mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{x} - \mathbf{y})$, and L can be identified as the largest eigenvalue of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$. However, in the context of extensive ULAs and ROIs, the dimensions of the matrix $\mathbf{A}$ may surpass the available Random Access Memory (RAM) of CPUs or GPUs. This scenario can lead to suboptimal values of L and λ, potentially resulting in slow convergence rates for the BC-FISTA. Therefore, in such cases, adopting a Learned Block Convolutional ISTA (BC-LISTA) emerges as the superior strategy.

B. Learned Block Convolutional ISTA

Within the framework of deep algorithm unrolling [8], the BC-LISTA requires a transpose model, denoted as $g(\cdot)$, for the computation of the back-projected residual. The mathematical formulation of the single-iteration update is presented as follows:

$$\mathbf{x}_{k+1} = S_{\lambda/L}\Big(\mathbf{x}_k + \frac{1}{L}(g(\mathbf{y} - f(\mathbf{x}_k)))\Big). \tag{5}$$

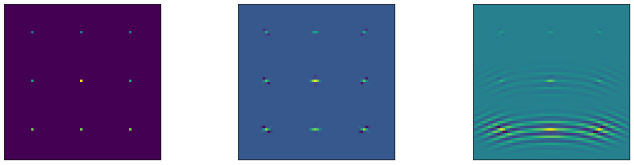
In the network architecture, $g(\cdot)$ is implemented with ConvTranspose1D layers. Each ConvTranspose1D layer is designed to deduce a reconstructed signal from an individual data slice. These reconstructed signals are subsequently aggregated through a weighted sum mechanism, endowed with trainable weights, to produce the refined signal estimate. This estimate is then processed through a soft-thresholding function, facilitating further updates.

The BC-LISTA architecture provides significant versatility in defining trainable parameters, including the kernels for the forward and transpose models, soft-thresholding parameters, and weights in the weighted sum mechanism. It is recommended to keep the forward model fixed to ensure consistency, while parameters such as λ, L, and aggregation weights are optimized as trainable to enhance network adaptability and performance. Typically, the kernels of the transpose model are adjustable to facilitate robust signal reconstruction, except in cases with optimal initial conditions or simpler inference tasks where fixed kernels may be adequate.

C. Evaluation

In the evaluation, we employ a 64-channel ULA, adhering to the parameters establishment scheme for Fig. 6, specifically, $N_{\text{c}} = 64$ and $d_{\text{c}} = 0.5\,\text{mm}$. Our training process for BC-LISTA involved synthetic data generation; for each epoch, we produced a new set of random reflectivity maps $\{\mathbf{x}_n\}$ as in [13], and converted these into FMC data $\{\mathbf{Y}_n\}$, where $n \in [1, \ldots, 20]$. The Adam optimizer with learning rate 1×10^{-4} is used to minimize the mean squared error $\frac{1}{N}\|\text{BC-LISTA}(\mathbf{Y}_n) - \mathbf{x}_n\|$ across 50 epochs. The learned parameters are the final superposition weights, as well as the kernels of the transpose model $g(\cdot)$, the soft-thresholding parameter λ, and step size L for each layer.

We deploy a 5-layer BC-LISTA alongside a 5-iteration BC-FISTA, applied to identical FMC data sets. The comparative analysis of the resultant images is visually depicted in Fig. 7, where it is evident that BC-LISTA significantly outperforms BC-FISTA in terms of image quality.



(a) Ground Truth (b) BC-LISTA (c) BC-FISTA

Fig. 7: Reconstructed images comparison. The reconstruction achieved through BC-FISTA is mathematically identical to that of the naive FISTA algorithm but at a lower cost.

Subsequently, both algorithms were executed 100 times on the same dataset, with processing times recorded in seconds on both CPU and GPU platforms. The CPU tests were conducted on a laptop with an Intel Core i7-8550U processor, while the GPU tests were performed on a cluster equipped with AMD EPYC 7742 processors and NVIDIA A100 GPUs. This analysis yielded maximum, minimum, and average execution times, detailed in TABLE I. The findings highlight the computational efficiency of the CSC algorithms on both platforms, with BC-LISTA demonstrating superior performance on both platforms. Additionally, the rapid processing times achieved on the GPU indicate an efficient training process for BC-LISTA.

TABLE I: Computation Efficiency Comparison

	Intel Core i7-8550U			NVIDIA A100		
	Max	Ave	Min	Max	Ave	Min
BC-FISTA	22.317	19.136	16.628	1.657	1.183	1.175
BC-LISTA	8.769	6.518	5.316	0.885	0.884	0.883

*Unit: seconds (s)

Performance differences are attributed to BC-FISTA's reliance on automatic differentiation for inference, which is computationally more intensive than BC-LISTA's convolutions and transpose operations. BC-LISTA benefits from using original convolution kernels as initial guesses, requiring fewer training epochs. Consistently, BC-LISTA outperforms BC-FISTA with an equal number of iterations, due to the advantages of employing learned linear transformations.

V. Conclusion

In this investigation, we suggest an innovative convolutional forward modeling technique that capitalizes on the Block-Toeplitz structure inherent in multichannel, time-delay-based matrices. This methodology is particularly advantageous for applications exhibiting shift-invariance along at least one dimension, where the corresponding axes are integer multiples of one another. We have demonstrated that our convolutional model markedly improves memory efficiency, an essential feature for managing large ULAs and expansive ROIs. Moreover, we managed to implement and assess two CSC algorithms, BC-FISTA and BC-LISTA, which not only perform effectively on high-end GPUs but also on older-generation CPUs. Notably, BC-LISTA delivers superior image quality and computational efficiency over (BC-)FISTA, underscoring the potential to accelerate data processing tasks significantly. The adaptability of our proposed model extends to other unrolled inversion algorithms, offering remarkable flexibility in the customization of trainable parameters. The detection of sparsity in the kernels provides a foundation for future research aimed at devising a more streamlined learned convolutional forward model, potentially simplifying the architecture further and reducing the requisite parameters.

References

[1] D. P. Palomar and Y. C. Eldar. *Convex optimization in signal processing and communications*. Cambridge University Press, 2010.

[2] J. Wang, S. Kwon, and B. Shim. Generalized orthogonal matching pursuit. *IEEE Transactions on signal processing*, 2012.

[3] F. Boßmann, G. Plonka, T. Peter, O. Nemitz, and T. Schmitte. Sparse deconvolution methods for ultrasonic ndt: Application on tofd and wall thickness measurements. *Journal of Nondestructive Evaluation*, 2012.

[4] 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: A Journal Issued by the Courant Institute of Mathematical Sciences*, 2004.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2009.

[6] N. Shlezinger, Y. C. Eldar, and S. P. Boyd. Model-based deep learning: On the intersection of deep learning and optimization. *IEEE Access*, 2022.

[7] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *27th International Conference on Machine Learning (ICML-2010)*, 2010.

[8] V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 2021.

[9] H. Wang, E. Pérez, and F. Römer. Deep learning-based optimal spatial subsampling in ultrasound nondestructive testing. In *31st European Signal Processing Conference (EUSIPCO-2023)*. IEEE, 2023.

[10] S. Mulleti, H. Zhang, and Y. C. Eldar. Learning to sample: Data-driven sampling and reconstruction of FRI signals. *IEEE Access*, 2023.

[11] H. Wang, E. Pérez, and F. Römer. Data-driven subsampling matrices design for phased array ultrasound nondestructive testing. In *2023 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2023.

[12] I. A. M. Huijben, B. S. Veeling, K. Janse, M. Mischi, and R. J. G. van Sloun. Learning sub-sampling and signal recovery with applications in ultrasound imaging. *IEEE Transactions on Medical Imaging*, 2020.

[13] H. Wang, Y. Zhou, E. Pérez, and F. Römer. Jointly learning selection matrices for transmitters, receivers and fourier coefficients in multichannel imaging. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2024)*. IEEE, 2024.

[14] Z. Wei and X. Chen. Deep-learning schemes for full-wave nonlinear inverse scattering problems. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[15] X. Chen, Z. Wei, L. Maokun, P. Rocca, et al. A review of deep learning approaches for inverse scattering problems (invited review). *ELECTROMAGNETIC WAVES*, 2020.

[16] A. Zacharopoulos, A. Garofalakis, J. Ripoll, and S. Arridge. Development of in-vivo fluorescence imaging with the matrix-free method. In *Journal of Physics: Conference Series*. IOP Publishing, 2010.

[17] A. Besson, D. Perdios, F. Martinez, Z. Chen, R. Carrillo, M. Arditi, Y. Wiaux, and J. Thiran. Ultrafast ultrasound imaging as an inverse problem: Matrix-free sparse image reconstruction. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2017.

[18] H. Sreter and R. Giryes. Learned convolutional sparse coding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[19] R. Fu, Y. Liu, T. Huang, and Y. C. Eldar. Structured LISTA for multidimensional harmonic retrieval. *IEEE Transactions on Signal Processing*, 2021.

[20] T. Kailath and J. Chun. Generalized displacement structure for block-toeplitz, toeplitz-block, and toeplitz-derived matrices. *SIAM Journal on Matrix Analysis and Applications*, 1994.

[21] S. Semper, J. Kirchhof, C. Wagner, F. Krieg, F. Römer, A. Osman, and G. Del Galdo. Defect Detection from 3D Ultrasonic Measurements Using Matrix-free Sparse Recovery Algorithms. In *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018.