ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096525

INTERPRETABLE, UNROLLED DEEP RADAR BEAMPATTERN DESIGN

Kareem Metwaly, Junho Kweon*, Khaled Alhujaili^{+}, Maria Greco*, Fulvio Gini* and Vishal Monga**

* Pennsylvania State University, USA $+$ Taibah University, Saudi Arabia $\star$ University of Pisa, Italy

ABSTRACT

Optimizing a transmit MIMO radar waveform subject to the non-convex constant modulus constraint remains a problem of enduring interest. The past decade has seen a variety of tailored iterative approaches with various performance-complexity trade-offs. Despite promising work, iterative algorithms have a speed handicap and require meticulous parameter tuning. Once trained, a deep network can quickly regress the desired waveform coefficients, but it is a black box and may excel only when generous training is available. We present a fast, learned, and - for the first time - interpretable (FLI) deep learning approach by unrolling a state-of-the-art iterative optimization approach. We particularly leverage the recently proposed projection, descent, and retraction (PDR) algorithm and design a deep network where each PDR step is mapped to a layer in the neural network while preserving the non-convex constant modulus constraint. FLI breaks the trade-off between complexity and performance. It is near real-time with boosted performance – fidelity to the desired beampattern – compared to the state-of-the-art alternatives.

1. INTRODUCTION

Thanks to the potential advantages of independent transmit waveforms, multiple-input multiple-output (MIMO) radar systems have received significant attention in recent years [1]. MIMO radar transmits waveforms from its transmitting elements and extracts information about targets and the surrounding environment from received signals containing reflected echoes. MIMO radar performance is often evaluated by signal to interference plus noise ratio (SINR) [2–4]. Especially, SINR can be enhanced by optimizing the transmit waveform to minimize the deviation against an idealized beampattern, called the beampattern design problem [5–9].

An idealized beampattern of a wideband radar system is defined in two domains: *spectral* and *spatial* [10]. In the spectral domain, a radar system can have a limited range of frequency use due to the coexistence of other radio systems. Regarding spatial information, once the expected location of targets is known, the transmit beampattern should be designed to focus the transmitted power in certain directions of interest.

The practical limitation of constant modulus constraint (CMC) accompanies the radar waveform design problem due to the existence of non-linear amplifiers [11]. The MIMO waveform design under CMC is a well-known hard, non-convex problem [5, 12], whose global optimum remains elusive. Numerous proposed approaches produce suitable waveforms dealing with the non-convexity of this problem. Many iterative algorithms solve the beampattern design problem under CMC with varying analytical techniques. For instance, the methods proposed in [7, 9, 10, 13] employ a mix of convex relaxations, sequence of convex problems, and majorization-minimization methods to address the vexing CMC. In most methods, CMC is usually not maintained in each algorithm step but achieved at algorithm convergence/termination.

In contrast with the aforementioned methods, *Project-descent-retract (PDR)* [8] is a recent state-of-the-art (SOTA) algorithm that develops a gradient-based method directly on the non-convex CMC set (with guaranteed convergence) by employing principles of optimization over manifolds. PDR has been validated in various scenarios with different desired beampatterns and demonstrated SOTA performance in [8, 10].

Despite substantial progress in beampattern design [5, 6, 8, 10, 12, 14, 15], iterative algorithms dealing with CMC generally have the inherent trade-off between computational complexity and achieved performance. Specifically, the complexity increases with the problem dimension, *i.e.*, more transmit antennas and time samples. Eventually, many iterative algorithms may incur a computational burden that prohibits them from being applied to practical radar systems.

As an alternative to iterative algorithms, deep learning (DL) methods have recently been proposed [16, 17] for the waveform design problem under CMC. These DL-based methods use pre-trained neural networks from training data, and they use the modeling capacity of well-known deep architectures to approximate hard non-linear mappings. However, DL requires extensive and diverse training data to mitigate overfitting to the training set. For radar waveform design, the known DL methods [16, 17] are invariably black-box; that is, they lack the interpretability of the process of producing a solution. Algorithm unrolling [18] has recently emerged as a promising solution to mitigate the problems mentioned above with DL. Rather than building an arbitrary network structure for DL, the network designed through algorithm unrolling mimics an existing iterative algorithm such that each iteration maps to a custom-designed layer. The structure inspired by a well-founded iterative algorithm may enhance training performance and provides interpretability. Algorithm

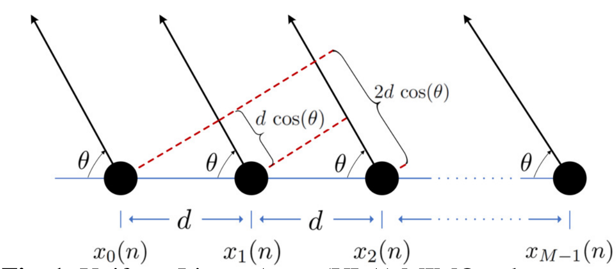


Fig. 1. Uniform Linear Array (ULA) MIMO radar system

unrolling has recently found compelling applications across sparse coding, imaging, and communications [19–23].

In this paper, we develop a first-of-its-kind unrolled deep network algorithm for the beampattern design problem under the CMC constraint, named fast, learned, and interpretable (FLI) deep beampattern design. Inspired by the PDR algorithm [8], FLI produces an always-feasible deep learning solution. It can estimate a waveform satisfying the CMC constraint in a relatively short time and with a relatively small error compared to other SOTA algorithms. Our work makes the following contributions: **1)** to the best of our knowledge, we developed the first unrolled algorithm for radar beampattern design problem under CMC, **2)** a customized deep architecture is developed that implements the PDR steps and introduces pruning and expansion layers that can help discover better minima candidates, **3)** FLI uses significantly fewer iterative steps (network layers) than PDR, as it learns the parameter values of the PDR algorithm from the dataset introduced in training, **4)** FLI does not require an input-output pair of the desired beampatterns and their corresponding optimal waveforms, instead we use the deviation from the idealized beampattern cost function as a self-supervised training loss.

2. BACKGROUND AND PROBLEM-SET UP

The objective of the beampattern design problem is to minimize the deviation between the desired beampattern and the actual beampattern generated by the transmitted waveform $\mathbf{x}$. Waveforms are transmitted from M antennas with inter-element spacing d, as shown in Fig.1. If we use N samples of the signal in the time and frequency domain, the beampattern design problem under the CMC can be formulated as [5, 8, 10]

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) = \sum_{s=1}^{S} \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} [d_{sp} - |\mathbf{a}_{sp}^H \mathbf{F}_p \mathbf{x}|]^2 \tag{1a}$$

$$\text{s.t.:} \quad |\mathbf{x}| = \mathbf{1} \tag{1b}$$

where $\mathbf{x} \in \mathbb{C}^L$, $L = M \times N$, $d_{sp} \in \mathbb{R}^+$ is the desired beampattern at discrete angle s and frequency p, $\mathbf{a}_{sp}$ is the steering vector and $\mathbf{F}_p$ is a Fourier matrix as in [8]. CMC ($|\mathbf{x}| = \mathbf{1}$) implies that $|x_l| = 1$ for $l = 0, 1, \cdots, L$. To enable a quadratically based cost function, [5] rewrote $f(\mathbf{x})$ as

$$f(\mathbf{x}) = \sum_{s=1}^{S} \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} |d_{sp} e^{j\phi_{sp}} - \mathbf{a}_{sp}^H \mathbf{F}_p \mathbf{x}|^2 \tag{2}$$

where $\phi_{sp} = \arg\{\mathbf{a}_{sp}^H \mathbf{F}_p \mathbf{x}\}$. Since both $\mathbf{x}$ and ϕ_{sp} are unknowns, this problem can be solved iteratively by minimizing (2) with respect to $\mathbf{x}$ and then fixing $\mathbf{x}$ and minimizing with respect to $\{\phi_{sp}\}$ [5, 8]. For fixed $\{\phi_{sp}\}$, simplify (2) as

$$f(\mathbf{x}) = \mathbf{x}^H \mathbf{P} \mathbf{x} - \mathbf{q}^H \mathbf{x} - \mathbf{x}^H \mathbf{q} + r \tag{3}$$

where $\mathbf{P} = \sum_p \mathbf{F}_p^H \mathbf{A}_p^H \mathbf{A}_p \mathbf{F}_p$, $\mathbf{q} = \sum_p \mathbf{F}_p^H \mathbf{A}_p^H \mathbf{d}_p$, $r = \sum_p \mathbf{d}_p^H \mathbf{d}_p$ and

$$\mathbf{A}_p = \begin{bmatrix} \mathbf{a}_{1p}^H \\ \vdots \\ \mathbf{a}_{Sp}^H \end{bmatrix}, \quad \mathbf{d}_p = \begin{bmatrix} d_{1p} e^{j\phi_{1p}} \\ \vdots \\ d_{Sp} e^{j\phi_{Sp}} \end{bmatrix}.$$

Let $\mathcal{S}^L$ be the CMC waveform set, (1) with fixed $\{\phi_{sp}\}$ is:

$$\min_{\mathbf{x} \in \mathcal{S}^L} \bar{f}(\mathbf{x}) \ = \mathbf{x}^H (\mathbf{P} + \gamma \mathbf{I}) \mathbf{x} - \mathbf{q}^H \mathbf{x} - \mathbf{x}^H \mathbf{q} \tag{4}$$

where $\gamma \geq 0$ is for numerical convergence control but does not affect the optimal solution. PDR is an iterative algorithm to solve (4) using three steps: projection, descent, and retraction. This unique algorithm utilizes gradient descent while maintaining CMC by updating iteration index k as [8]:

1. **P**rojection of search direction onto the tangent space.

   $$\mathbf{P}_{\mathcal{T}_{\mathbf{x}_{(k)}} \mathcal{S}^L} \left(\boldsymbol{\eta}_{(k)} \right) = \boldsymbol{\eta}_{(k)} - \text{Re}\{\boldsymbol{\eta}_{(k)}^* \odot \mathbf{x}_{(k)}\} \odot \mathbf{x}_{(k)} \tag{5}$$

   where $\boldsymbol{\eta}_{(k)} = -\nabla_{\mathbf{x}} \bar{f}(\mathbf{x}_{(k)})$ is the search direction, and $\mathcal{T}_{\mathbf{x}_{(k)}} \mathcal{S}^L$ is the tangent space

2. **D**escent update of $\mathbf{x}_{(k)}$ on this tangent space.

   $$\bar{\mathbf{x}}_{(k)} = \mathbf{x}_{(k)} + \beta \mathbf{P}_{\mathcal{T}_{\mathbf{x}_{(k)}} \mathcal{S}^L} \left(\boldsymbol{\eta}_{(k)} \right) \tag{6}$$

3. **R**etraction of $\bar{\mathbf{x}}_{(k)}$ to the complex circle manifold.

   $$\mathbf{x}_{(k+1)} = \bar{\mathbf{x}}_{(k)} \odot \frac{1}{|\bar{\mathbf{x}}_{(k)}|} \tag{7}$$

   where $\odot$ denotes the Hadamard product.

3. FLI FOR RADAR BEAMPATTERN DESIGN

FLI uses interpretable as opposed to the typical black-box DL blocks. Fig. 2 depicts the complete network structure. It consists of cascaded blocks mimicking the steps of an iterative algorithm. In each block, FLI progresses to a new set of solutions closer to the minimum value of the cost function. At each step (layer in the unrolled network), FLI operates similarly to the Project Descent Retract (PDR) algorithm [8]. However, unlike PDR, FLI utilizes training data and adapts to the loss function characteristics to speed up the convergence, as shown in Section 4. Thus, it requires a significantly lower number of steps to converge. In other words, we reduce the memory footprint and increase the speed of our DL model, which are common problems with many DL models that hinder their application to the beampattern design problem. At initialization, $\mathcal{I}(\cdot)$ utilizes the training dataset to find good initial values $\{\mathbf{x}_0^{(j)}\}_{j=1}^{N_e}$ for the given desired beampattern $\mathbf{d}$.

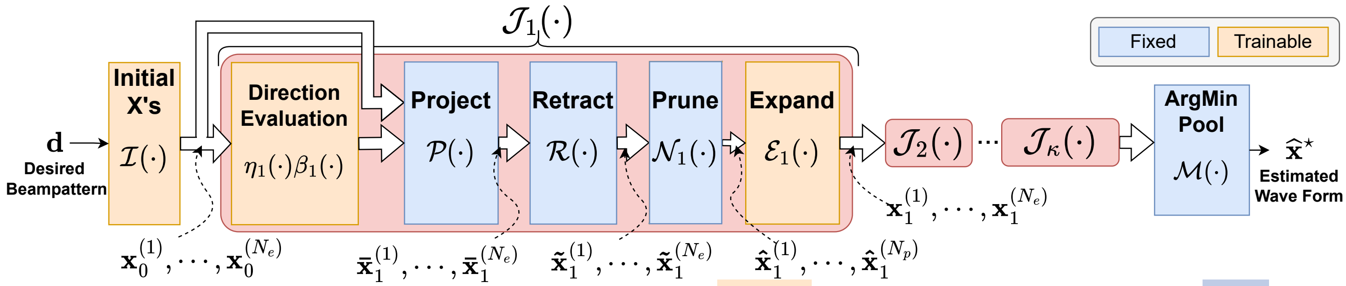


Fig. 2. Proposed FLI Architecture. Trainable blocks are colored in Orange and blocks with fixed parameters in Blue.

$\mathcal{I}(\cdot)$ consists of multiple fully connected layers with the input of size $\mathbb{R}^{NS\times 1}$ and the output of size $\mathbb{C}^{L\times 1}$. At each layer $\mathcal{J}_i(\cdot)$, we estimate (learning) a descent direction $\boldsymbol{\eta}_i$ and a step size β_i. Then, we perform projection $\mathcal{P}(\cdot)$ and retraction $\mathcal{R}(\cdot)$ similar to PDR:

$$\overline{\boldsymbol{\eta}}_i^{(j)} = \mathcal{P}(\boldsymbol{\eta}_i^{(j)}) = \boldsymbol{\eta}_i^{(j)} - \mathrm{Re}\{\boldsymbol{\eta}_i^{(j)} \odot \mathbf{x}_i^{(j)}\} \odot \mathbf{x}_i^{(j)} \tag{8}$$

$$\overline{\mathbf{x}}_i^{(j)} = \mathbf{x}_i^{(j)} + \beta_i \overline{\boldsymbol{\eta}}_i^{(j)} \tag{9}$$

$$\tilde{\mathbf{x}}_i^{(j)} = \mathcal{R}(\overline{\mathbf{x}}_i^{(j)}) = \overline{\mathbf{x}}_i^{(j)} \odot \frac{1}{|\overline{\mathbf{x}}_i^{(j)}|} \tag{10}$$

Note also that we introduce two new layers; *Prune* $\mathcal{N}_i(\cdot)$ and *Expand* $\mathcal{E}_i(\cdot)$. *Prune* reduces the number of generated vectors $\{\mathbf{x}_i^{(j)}\}_{j=1}^{N_e}$ from N_e to N_p by discarding vectors with the highest values for (1a). On the other hand, *Expand* increases the number of vectors $\{\mathbf{x}_i^{(j)}\}_{j=1}^{N_p}$ back to N_e. Thus, it generates new possible candidates that may lead to a better solution. To summarize, the operators in orange in Fig. 2 are trainable. Hence, key algorithm parameters such as descent direction and step size, *which are otherwise fixed in PDR, are now learned from training data,* thereby boosting performance and enhancing convergence speed. Finally, while many existing iterative algorithms could be unrolled, our choice of PDR is in part motivated by the fact that in addition to projection and descent, the retraction step guarantees feasibility, *i.e.*, that we necessarily estimate a CMC waveform.

We use fully connected layers throughout the implementation[1] in contrast to convolution layers, as there is no spatial/temporal locality that can be utilized in the vectors $\{\mathbf{x}_i^{(j)}\}$. $\mathcal{I}(\cdot)$ consists of 7 fully connected layers with intermediate number of neurons $= 1000$. After each layer, we use Complex Batch Normalization and Complex ReLU, as defined in [24, 25].Our realization of FLI was based on $\kappa = 5$ layers in contrast to 100 or so iterations typical for PDR convergence.

Dataset and Training: We generated a set of $20k$ beampattern specifications to train our network in a self-supervised manner; we do not have ground-truth solutions for them. Instead, we devised the training loss below using $f(\cdot)$ as in (1a).

$$\mathcal{L} = f(\widehat{\mathbf{x}}^\star) + \lambda_{f_0} \sum_{j=1}^{N_e} f(\mathbf{x}_0^{(j)}) \tag{11}$$

[1]FLI code and architecture specifics can be found here: `https://github.com/kareem-metwaly/BeampatternDesign`; we have verified that FLI results are robust to hyperparameter choices (λ_{f0}, etc.)

Table 1. Deviation from the desired beampattern.

Method	Type	Error (dB)	Time (sec)	# Iter.
Unconstrained		19.52	-	-
WBFIT [5]		33.11	0.39	199
SDR [26] [14]		28.41	1190	33
IA-CPC [15]	Iterative	30.86	14.47	180
ADMM [12]	Methods	28.91	20.36	208
PDR [8]		26.69	7.38	115
VSGP [27]		31.63	4.01	42
CON [17]	Deep	27.15	**0.13**	–
FLI (Ours)	Learning	**24.82**	**0.15**	–

4. EXPERIMENTS

Parameter Settings and Simulation Set-up: We set λ_{f_0} to be 0.5 and the learning rate to be 10^{-3} via cross-validation. Our simulation sets the number of time-samples N to be 32, the number of transmit antennas M to be 40, and the number of discretized angular sub-intervals S to be 90, respectively.

Results: We employ the desired beampattern specification (non-overlapping with training) used widely in previous work

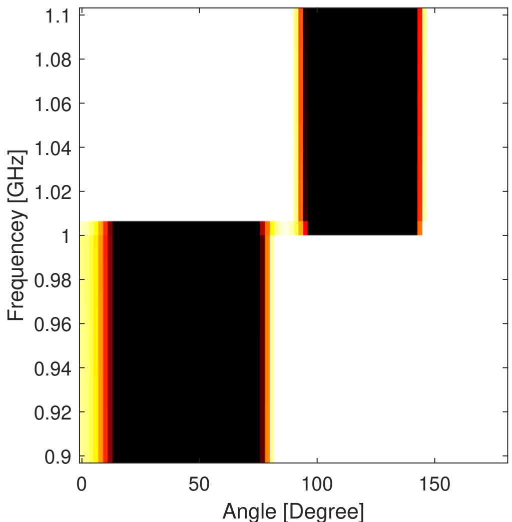
$$d(\theta, f) = \begin{cases} 0 & \theta = [10°, 80°],\ -\frac{B}{2} + f_c \leq f \leq f_c \\ 0 & \theta = [95°, 145°],\ f_c \leq f \leq \frac{B}{2} + f_c \\ 1 & \text{Otherwise.} \end{cases} \tag{12}$$

As shown in Table 1, FLI achieves better results compared to other iterative and deep learning-based methods[2] with a computational footprint (inference) that is near real-time. The measured error is defined as $10\log_{10}(f(\widehat{\mathbf{x}}^\star))$. The lower bound of the error is the one obtained from unconstrained optimization (*i.e.*, without the constant modulus constraint). Fig. 3 depicts the visual results of FLI and selected competing state-of-the-art methods. As is evident from Fig. 3, FLI generates the closest beampattern to the unconstrained case.

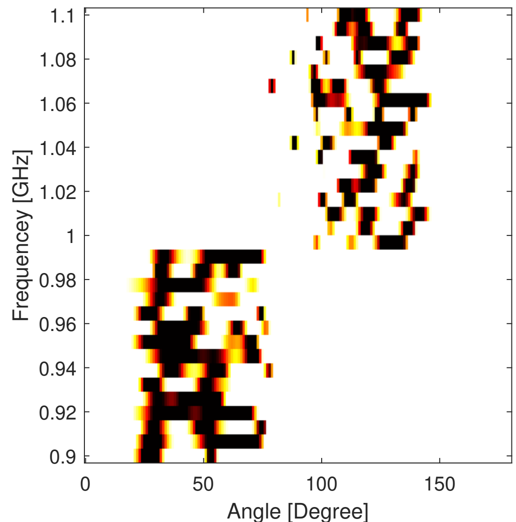
5. CONCLUSION

We present FLI, an unrolled, interpretable deep learning algorithm for the beampattern design problem under the constant modulus constraint. FLI offers both performance (fidelity to the desired beampattern) and computational benefits over the state-of-the-art. Future work may investigate the generalizability of FLI vs. known black-box deep learning alternatives.

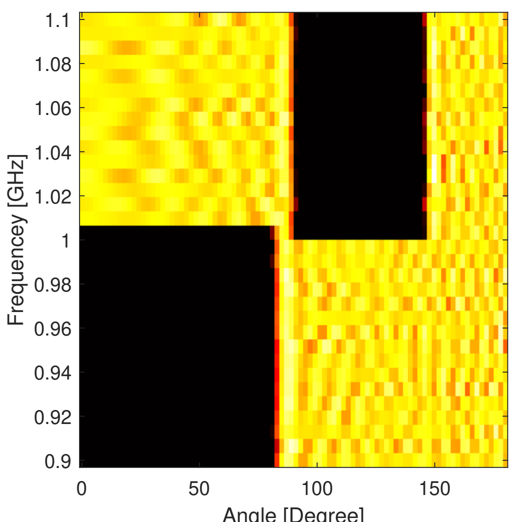
[2]For fairness of comparison, we implement the phase-prediction architecture exactly as in CON [17] but train it with the same data as FLI.



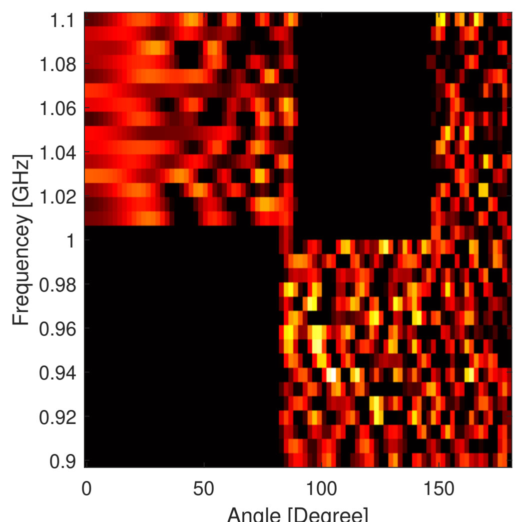
(a) Unconstrained



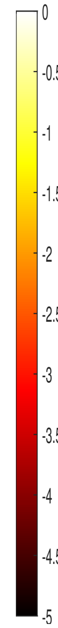
(b) SDR

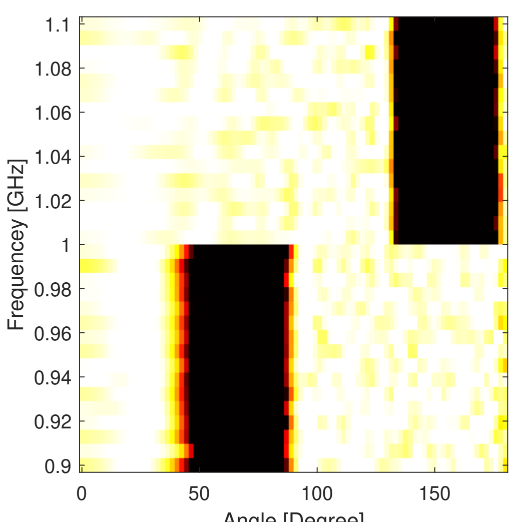


(c) PDR

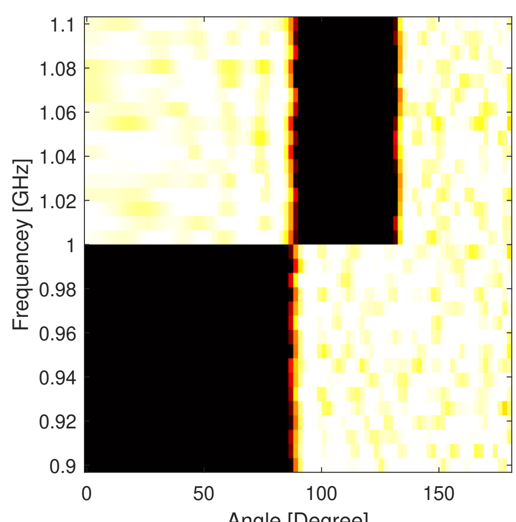


(d) VSGP





(e) CON



(f) FLI

Fig. 3. Visual results of FLI compared to other state-of-the-art (iterative and deep learning based) methods.

6. REFERENCES

[1] J. Li, *et al. MIMO Radar Signal Processing*. John Wiley & Sons, 2008. ISBN 978-0-470-39143-3.

[2] D. Fuhrmann, *et al.* "Transmit beamforming for MIMO radar systems using signal cross-correlation." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, 2008, pp. 171–186.

[3] D. Fuhrmann, *et al.* "Transmit beamforming for MIMO radar systems using partial signal correlation." *Asilomar Conference on Signals, Systems and Computers*, 2004, vol. 1, pp. 295–299.

[4] H. Li, *et al.* "Transmit Subaperturing for MIMO Radars With Co-Located Antennas." *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, 2010, pp. 55–65.

[5] H. He, *et al.* "Wideband MIMO Systems: Signal Design for Transmit Beampattern Synthesis." *IEEE Transactions on Signal Processing*, vol. 59, 2011, pp. 618–628.

[6] A. Aubry, *et al.* "MIMO Radar Beampattern Design Via PSL/ISL Optimization." *IEEE Transactions on Signal Processing*, vol. 64, 2016, pp. 3955–3967.

[7] O. Aldayel, *et al.* "Tractable Transmit MIMO Beampattern Design Under a Constant Modulus Constraint." *IEEE Transactions on Signal Processing*, vol. 65, 2017, pp. 2588–2599.

[8] K. Alhujaili, *et al.* "Transmit MIMO Radar Beampattern Design via Optimization on the Complex Circle Manifold." *IEEE Transactions on Signal Processing*, vol. 67, 2019, pp. 3561–3575.

[9] W. Fan, *et al.* "MIMO Radar Waveform Design for Quasi-Equiripple Transmit Beampattern Synthesis via Weighted lp-Minimization." *IEEE Transactions on Signal Processing*, vol. 67, 2019, pp. 3397–3411.

[10] K. Alhujaili, *et al.* "Spectrally Compatible MIMO Radar Beampattern Design Under Constant Modulus Constraints." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, 2020, pp. 4749–4766.

[11] L. Patton, *et al.* "Modulus constraints in adaptive radar waveform design." *IEEE Radar Conf.*, 2008, pp. 1–6.

[12] J. Liang, *et al.* "Unimodular Sequence Design Based on Alternating Direction Method of Multipliers." *IEEE Transactions on Signal Processing*, vol. 64, 2016, pp. 5367–5381.

[13] B. Kang, *et al.* "Spatio-Spectral Radar Beampattern Design for Coexistence With Wireless Communication Systems." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, 2019, pp. 644–657.

[14] G. Cui, *et al.* "MIMO Radar Waveform Design With Constant Modulus and Similarity Constraints." *IEEE Transactions on Signal Processing*, vol. 62, 2014, pp. 343–353.

[15] G. Cui, *et al.* "Quadratic Optimization With Similarity Constraint for Unimodular Sequence Synthesis." *IEEE Transactions on Signal Processing*, vol. 65, 2017, pp. 4756–4769.

[16] K. Zhong, *et al.* "MIMO Radar Waveform Design via Deep Learning." *IEEE Radar Conf.*, 2021, pp. 1–5.

[17] J. Hu, *et al.* "Designing Unimodular Waveform(s) for MIMO Radar by Deep Learning Method." *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, 2021, pp. 1184–1196.

[18] V. Monga, *et al.* "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing." *IEEE Signal Processing Magazine*, vol. 38, 2021, pp. 18–44.

[19] K. Gregor, *et al.* "Learning fast approximations of sparse coding." *International Conf. on Machine Learning*, 2010, pp. 399–406.

[20] Y. Li, *et al.* "Efficient and interpretable deep blind image deblurring via algorithm unrolling." *IEEE Transactions on Computational Imaging*, 2020, pp. 666–681.

[21] W. Wu, *et al.* "URetinex-Net: Retinex-Based Deep Unfolding Network for Low-Light Image Enhancement." *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022, pp. 5901–5910.

[22] O. Solomon, *et al.* "Deep Unfolded Robust PCA with Application to Clutter Suppression in Ultrasound." *IEEE Transactions on Medical Imaging*, 2019, pp. 1–1.

[23] N. Samuel, *et al.* "Learning to detect." *IEEE Transactions on Signal Processing*, 2019, pp. 2554–2564.

[24] C. Trabelsi, *et al.* "Deep Complex Networks." *International Conference on Learning Representations*, 2018, pp. 1–19.

[25] C. Lee, *et al.* "Complex-Valued Neural Networks: A Comprehensive Survey." *IEEE/CAA Journal of Automatica Sinica*, vol. 9, 2022, pp. 1406–1426.

[26] Z.-q. Luo, *et al.* "Semidefinite Relaxation of Quadratic Optimization Problems." *IEEE Signal Processing Magazine*, vol. 27, 2010, pp. 20–34.

[27] M. Zhang, *et al.* "Fast and Simple Gradient Projection Algorithms for Phase-Only Beamforming." *IEEE Transactions on Vehicular Technology*, vol. 70, 2021, pp. 10620–10632.