ADMM-Net for Communication Interference Removal in Stepped-Frequency Radar

Jeremy Johnston, Yinchuan Li, Marco Lops, *Fellow, IEEE*, and Xiaodong Wang, *Fellow, IEEE*

***Abstract*—Complex ADMM-Net, a complex-valued neural network architecture inspired by the alternating direction method of multipliers (ADMM), is designed for interference removal in stepped-frequency radar super-resolution angle-range-doppler imaging. We consider an uncooperative spectrum sharing scenario where the radar is tasked with imaging a sparse scene amidst communication interference that is frequency-sparse due to spectrum underutilization, motivating an ℓ_1-minimization problem to recover the radar image and suppress the interference. The problem's ADMM iteration undergirds the neural network design, yielding a set of generalized ADMM updates with learnable hyperparameters and operations. The network is trained with random data generated according to the radar and communication signal models. In numerical experiments ADMM-Net exhibits markedly lower error and computational cost than ADMM and CVX.**

***Index Terms*—Deep unfolding, deep learning, alternating direction method of multipliers (ADMM), MIMO radar, stepped-frequency, interference, coexistence.**

I. Introduction

THE use of radar in civilian life has expanded—e.g. automotive radar, remote sensing, and healthcare applications—meanwhile next-generation communications systems have begun to encroach upon spectrum once designated solely for radar use [1]. In response, the U.S. Department of Defense declared an initiative [2] to spur research on algorithm and system designs that allow radars to cope with the changing spectral landscape. Subsequently, several system design motifs have materialized in the area of radar-communication coexistence [3].

Coordinated coexistence methods enable coexistence through system co-design and information sharing. Co-design of the radar waveform and communication system codebook may be cast as an optimization problem, e.g. to maximize the communication rate subject to constraints on the radar SNR [4]. In a radar-centric co-design, the radar waveform might be forced to lie in the null space of the channel between the radar and communication devices, enabled by channel state information provided either externally or by the radar's own means of channel estimation [5]. In some proposals the coexisting systems communicate with a data fusion center which uses the shared information to configure each system in a way that optimizes the performance of the system ensemble [6]. *Uncoordinated coexistence* methods, on the other hand, seek to minimize interference absent cooperation or external information. Instead, spectrum occupancy measurements may inform real-time adjustments to the transmit waveform, e.g. center frequency [7], and beamforming may mitigate directional interference [8].

In uncoordinated interference removal, thresholding or filtering can be effective if the interference is much stronger than the desired signal, though runs the risk of inadvertently distorting the desired signal. Parametric methods estimate the parameters of a statistical signal model via either subspace methods or optimization. Greedy methods, e.g. CLEAN and matching pursuit, iteratively remove the most dominant interference component, found by projecting the recording onto an interference dictionary, until a stopping criterion is met. If the received interference is concentrated in narrow regions along some dimension, e.g. time, frequency, or space, and hence is sparse in a known dictionary, convex relaxation methods such as ℓ_1-minimization can be effective [9], [10]. In this vein, the present paper considers an uncoordinated scenario where the interference is sparse in a known basis. In particular, we show that the stepped-frequency radar waveform's "frequency-hopping" property imposes on the interference structure that can be exploited.

Neural networks are attractive for interference suppression as they can learn an inverse mapping to recover a signal from corrupted measurements [11], [12]. So-called "black box" neural networks generalize well but provide only empirical rather than theoretical guarantees. Moreover, they neglect the corpus of model-based signal recovery theory and algorithms which exploit prior knowledge to devise computational procedures tailored to the problem [13]. Iterative algorithms grounded in either optimization or statistics are among the most computationally efficient for signal recovery, but their performance hinges on the careful selection of hyperparameters whose favorable values are generally problem-dependent. From one point of view, deep unfolding, the methodology applied in this paper,

Manuscript received September 7, 2020; revised January 18, 2021 and March 9, 2021; accepted April 21, 2021. Date of publication April 30, 2021; date of current version June 4, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bo Chen M.D. This work was supported in part by the U.S. National Science Foundation (NSF) under Grants CCF 1814803 and SHF 7995357, and in part by the U.S. Office of Naval Research (ONR) under Grant N000141712827. *(Corresponding author: Yinchuan Li.)*

Jeremy Johnston is with the Electrical Engineering Department, Columbia University, New York, NY 10027 USA (e-mail: j.johnston@columbia.edu).

Yinchuan Li is with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China, and also with the Electrical Engineering Department, Columbia University, New York, NY 10027 USA (e-mail: yinchuan.li.cn@gmail.com).

Marco Lops is with the Department of Electrical Engineering and Information Technologies, Università di Napoli "Federico II,", Napoli, NA 21 - I-80125, Italy (e-mail: lops@unina.it).

Xiaodong Wang is with the Electrical Engineering Department, Columbia University, New York, NY 10027 USA (e-mail: wangx@ee.columbia.edu).

Digital Object Identifier 10.1109/TSP.2021.3076900

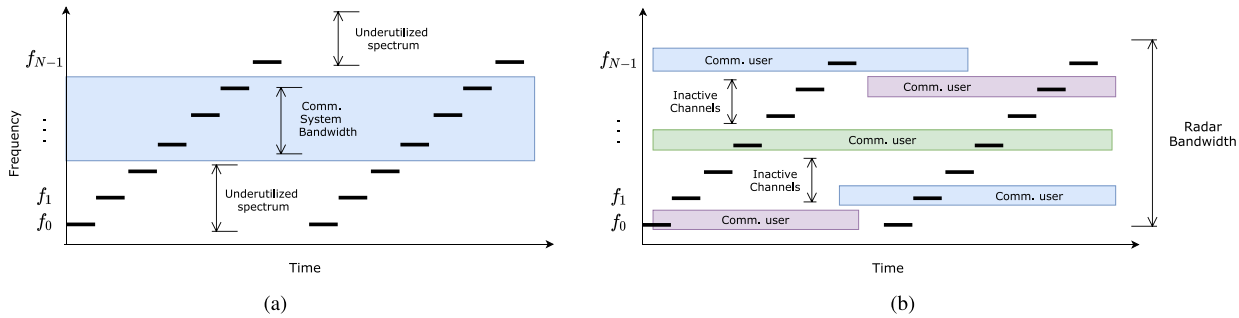


Fig. 1. Frequency occupation versus time for two representative spectrum sharing scenarios. The black strips indicate the spectrum occupied by the radar system; the colored strips indicate the spectrum occupied by the communication system. Only the overlapping regions cause interference to the radar.

casts hyperparameter selection as an instance of deep learning and thus automates cross-validation.

In the deep unfolding [14] framework a given iterative algorithm inspires a neural network design. Typically the network's forward pass has the computational complexity of a handful of algorithm iterations, a fraction of that required for the original algorithm to converge, yet the trained network may outperform the original algorithm. The algorithm's update rules are cast as a block of network layers whose forward pass emulates one complete algorithm iteration, and whose learnable parameters correspond to a chosen parameterization of the update rules, often comprising algorithm hyperparameters as well as matrices. A number of such blocks, possibly augmented in order to increase learning capacity [15], are sequenced to form the network. Network training, typically done via gradient-based optimization, employs data either gathered from the field or randomly generated according to signal models, and thus adapts the algorithm to the problem at hand. The layer parameters can be initialized either as prescribed by the algorithm, or even randomly; in one study, an unfolded vector-approximate message passing (VAMP) network randomly initialized learned a denoiser identical to the statistically matched denoiser [16]. Algorithms previously considered for deep unfolding include the iterative shrinkage thresholding algorithm (ISTA) [17], robust principal components analysis (RPCA) [18], and ADMM [19], [20]. Applications span those of iterative optimization itself, e.g. wireless communication [15], medical imaging [18], and radar [21].

In this paper, we design an ADMM-Net which simultaneously recovers a super-resolution angle-range-doppler image and removes communication interference. We target an uncooperative spectrum sharing scenario in which the radar is considered the primary function and the communications utilize portions of the shared spectrum. In the proposed multi-frame radar processing architecture, the stepped-frequency radar transmits a series of simple pulse trains to obtain a set of low-resolution radar measurements which the ADMM-Net synthesizes to form an image. Although the total radar bandwidth is large (~ 1 GHz), by virtue of the pulse-by-pulse processing only the communication signals that spectrally overlap with a given pulse interfere with the pulse's return. Moreover, communication signals tend to be sparse in the frequency domain, owing to periods of low activity or otherwise underutilized spectrum [22], [23]. Consequently, the interference manifests as sparse noise in the radar measurements.

Applicable scenarios lie between two extremes. At one (Fig. 1(a)), the total radar bandwidth overlaps with multiple communication carriers and the radar frequency step is on the order of the communication carrier bandwidth. For example, stepped frequency radars may have a step size of 20 MHz [24], while the maximum LTE bandwidth is 20 MHz [25] and in sub-6 GHz 5 G the maximum channel bandwidth is 100 MHz [26]. At the other (Fig. 1(b)), the radar overlaps with a single communication carrier. The carrier comprises sub-channels sized on the order of the radar frequency step-size that are assigned to opportunistic communication users. For example, 5 G employs channels with bandwidths in the hundreds of megahertz to a few gigahertz [27], and stepped frequency radars often have a sweep bandwidth on that order. In any case, the key property that enables the radar to coexist is that significant portions of spectrum tend to be underutilized [22] [26]. In light of this, the interference induced by the active portions can be mitigated.

Radar images may exhibit considerable sparsity, in that the number of dominant scatterers in the illuminated area may be relatively small [28, Sec. 3]. Sparsity-inducing image reconstruction algorithms exploit the relatively low information content to enable "super-resolution" finer than conventional processing methods. Compressed sensing theory has provided theoretical underpinnings for ℓ_1-minimization techniques, in particular. Such methods have been successfully applied to MIMO radar for angle-range-doppler imaging [29]–[31].

These circumstances motivate an optimization problem to jointly recover the image and remove the interference. We then use the problem's corresponding ADMM equations as a schematic for a neural network architecture. Important for radar processing, the network processes complex-valued measurements, doing so in a manner consistent with ADMM. Thus, training the network is tantamount to optimizing over the ADMM parameters and matrices a handful of ADMM iterations. In practice, training data could comprise measurements recorded by the intended radar system, i.e., the system's response to various objects with known parameters (amplitude, angle, range, speed) in various interference environments; thus the underlying model inspires the structure of the network, but the final

network is determined by the real-world measurement data. In our experiments, training data sets were randomly generated via the signal model. We find that the trained ADMM-Net recovers more accurate images than ADMM and CVX at a fraction of the computational cost.

The remainder of the paper proceeds as follows. First, we develop a model of the radar and communication signals and formulate an optimization problem to jointly recover the radar image and interference (Section II). We then derive the problem's ADMM recursion (Section III) and design an ADMM-Net by unfolding the complex-valued ADMM equations into a real-valued neural network (Section IV). Finally, numerical simulations (Section V) compare the performance of ADMM-net to that of ADMM and CVX.

II. Signal Model & Problem Formulation

A stepped-frequency MIMO radar illuminates a sparse scene in the presence of interfering communication signals, which are sparse in the frequency domain. The radar undertakes pulse-by-pulse processing over multiple measurement frames. The joint image recovery-interference removal task is cast as an optimization problem.

A. Signal Model

1) MIMO Radar Signal: Consider a frequency-stepped pulsed MIMO radar with N_T transmitters and N_R receivers. Each of the transmitted waveforms u^p, $p = 1, \ldots, N_T$, has duration T seconds and the waveforms are assumed to be approximately mutually incoherent (see (15)). The scene is illuminated by N_d trains of N pulses: within the mth train, the nth pulse emitted by the pth transmit antenna is given by

$$s^p(m, n, t) = u^p(t - nT_r - mNT_r) \exp(j2\pi f_n t), \quad (1)$$

where t is continuous time, $0 \le m \le N_d - 1$, $0 \le n \le N - 1$, $1 \le p \le N_T$; $f_n = n\Delta f + f_0$ where f_0 is the lowest carrier frequency, and $N\Delta f$ is the overall bandwidth. Each pulse echo recording length is $T_r \gg T$ seconds, the pulse repetition interval (PRI). A complete observation consists of $N_d N$ PRIs.

We consider a scene of L scatterers with scattering coefficients x_i and radial velocities v_i. The signal received by the qth receiver, $q = 1, \ldots, N_R$, is

$$r^q(n, t) = \sum_{m=0}^{N_d-1} \sum_{p=1}^{N_T} \sum_{i=1}^{L} x_i s^p(m, n, t - \tau_i^{pq}(t)), \quad (2)$$

where

$$\tau_i^{pq}(t) = \frac{2v_i}{c} t + \tau_i + \delta_i^p + \varepsilon_i^q \quad (3)$$

is the ith scatterer's delay; δ_i^p and ε_i^q are the marginal delays due to array geometry associated with antenna pair (p, q); and τ_i is the absolute round-trip delay observed by a reference antenna pair during the first PRI. We assume the velocities are constant throughout the series of sweeps.We further make the following assumptions:

- The range variation throughout the series of sweeps is negligible with respect to the range resolution of each pulse:
$$\frac{2v_i N_d N T_r}{c} \ll T.$$
- The array element spacing is much smaller than the range resolution granted by the overall transmitted bandwidth:
$$\delta_i^p + \varepsilon_i^q \ll \frac{1}{N\Delta f}. \quad (4)$$

Since the pulse is unsophisticated, $T\Delta f \simeq 1$; hence (4) implies $\delta_i^p + \varepsilon_i^q \ll T$, whereby

$$u^p(t - \tau_i^{pq}(t)) \simeq u^p(t - \tau_i). \quad (5)$$

In (2) the term $\exp(-j2\pi n\Delta f(\delta_i^p + \varepsilon_i^q))$ can be neglected since, by (4), $n\Delta f(\delta_i^p + \varepsilon_i^q) \ll 1$, $n = 0, 1, \ldots, N - 1$. Under these assumptions, (2) becomes

$$\begin{aligned} r^q(n, t) = \sum_{m=0}^{N_d-1} \sum_{p=1}^{N_T} \sum_{i=1}^{L} & x_i \exp(-j2\pi f_0(\delta_i^p + \varepsilon_i^q)) \\ & \times u^p(t - nT_r - mNT_r - \tau_i) \\ & \times \exp(j2\pi f_n(t - \tau_i - \frac{2v_i}{c} t)). \end{aligned} \quad (6)$$

2) Communication Signal: Suppose there are N_c carriers that spectrally overlap with the radar band, with center frequencies f_i^C and bandwidths B_i, $i = 0, 1, \ldots, N_c - 1$. Here the term "carrier" refers to any communication transmission within the radar band, e.g. a particular block of subcarriers within a communication band, the aggregate transmission over a communication band, etc. Thus the received communication signal has the form

$$s_c(t) = \sum_{i=0}^{N_c-1} g_i(t) \exp(j2\pi f_i^C t), \quad (7)$$

where g_i represents the information signal transmitted over carrier i and is a zero-mean random process whose power spectral density G_i satisfies

$$G_i(f) = 0 \text{ if } |f| > \frac{B_i}{2}. \quad (8)$$

B. Signal Processing At Radar Receiver Side

Receiver q's recording of the nth pulse takes the form

$$\chi^q(n, t) = r^q(n, t) + s_c(t) + e(t), \quad 0 < t < N_d N T_r, \quad (9)$$

where $e(t)$ is additive white gaussian noise (AWGN). The recording is processed as follows. Each pulse return is divided into $\lfloor T_r/T \rfloor$ range gates of size T seconds, a range interval of $\frac{cT}{2}$ meters, centered at times $t_k = kT + \frac{T}{2}$, $k = 0, \ldots, \lfloor T_r/T \rfloor - 1$. The qth receiver's recordings are projected onto the pth transmit waveform shifted to range cell k, i.e. onto the functions $\{s^p(m, n, t - t_k) : 0 \le m \le N_d - 1, 0 \le n \le N - 1, 1 \le p \le N_T, 0 \le k \le \lfloor T_r/T \rfloor - 1\}$, to obtain the output sequence $y^q(m, n, p, k)$, given by

$$\begin{aligned} y^q(m, n, p, k) &= \langle \chi^q(n, t), s^p(m, n, t - t_k) \rangle \\ &\triangleq y_R^q(m, n, p, k) + y_C^q(m, n, p, k) \end{aligned} \quad (10)$$

$$+ \overline{e}(m,n,p,k), \tag{11}$$

where $\langle y_1(t), y_2(t)\rangle \triangleq \int_{-\infty}^{\infty} y_1(t)y_2^*(t)dt$ and the terms y_R^q, y_C^q, and $\overline{e}$ are the projections of the radar echoes, the communication signal, and AWGN, respectively. This operation is equivalent to matched filtering each of the N echo recordings and sampling the output at times t_k [32]. Next we derive each of the terms in (11).

1) Radar Signal Component: We have

$$y_R^q(m,n,p,k) = \langle r^q(n,t), s^p(m,n,t-t_k)\rangle \tag{12}$$

$$\simeq \sum_{p'=1}^{N_T}\sum_{i=1}^{L} x_i \exp\left(-j2\pi f_0(\delta_i^{p'} + \varepsilon_i^q)\right) R_{u^{p'}u^p}(t_k - \tau_i)$$

$$\times \exp\left(-j2\pi f_n(\tau_i + \frac{2v_i}{c}(nT_r + mNT_r) - t_k)\right), \tag{13}$$

where $R_{uv}(\tau) \triangleq \langle u(t), v(t-\tau)\rangle$, and we have used the fact that $\{u^p(t - nT_r - mNT_r - t_k)\}_{m=0}^{N_d-1}$ is orthogonal along t. The approximation in (13) assumes the target velocities are small such that each target's position is constant within a single PRI. Since each u^p has duration T, the autocorrelation $R_{u^pu^p}$ has a duration of approximately $2T$; therefore we assume

$$R_{u^pu^p}(\tau) \simeq \begin{cases} 1 & |\tau| < T/2 \\ 0 & |\tau| > T/2 \end{cases}. \tag{14}$$

We also assume the waveforms are incoherent, i.e.,

$$R_{u^{p'}u^p}(\tau) \simeq \begin{cases} R_{u^pu^p}(\tau) & \text{if } p = p' \\ 0 & \text{if } p \neq p' \end{cases}, \ \tau \in \left[-\frac{T}{2}, \frac{T}{2}\right]. \tag{15}$$

This could be achieved, for example, through time-domain multiplexing (which would require increasing the illumination period in order to maintain a given maximum unambiguous range). Define $\mathcal{I}_k \triangleq \{i : |\tau_i - t_k| < T/2\}$, the indices of the scatterers that belong to range cell k. Applying (14) and (15), (13) becomes

$$y_R^q(m,n,p,k) = \sum_{i\in\mathcal{I}_k} x_i \exp\left(-j2\pi f_0(\delta_i^p + \varepsilon_i^q)\right)$$

$$\times \exp\left[-j2\pi(n\Delta f(\tau_i - t_k) + f_0\frac{2v_i}{c}mNT_r)\right]$$

$$\times \exp\left[-j2\pi(f_n\frac{2v_i}{c}nT_r + n\Delta f\frac{2v_i}{c}mNT_r)\right], \tag{16}$$

where we have absorbed $\exp\left(-j2\pi f_0(\tau_i - t_k)\right)$ into x_i.

In general the Tx/Rx array elements are distributed on a plane and the delays $\delta_i^p = \delta_i^p(\boldsymbol{\theta})$ and $\varepsilon_i^q = \varepsilon_i^q(\boldsymbol{\theta})$ are functions of the scatterer's angular coordinates $\boldsymbol{\theta} \in \mathbb{R}^2$, e.g. azimuth and elevation, relative to the array plane. We consider a generic array response matrix $\mathbf{H} \in \mathbb{C}^{N_T\times N_R}$ where

$$[\mathbf{H}(\boldsymbol{\theta})]_{pq} \triangleq \exp\left(-j2\pi f_0(\delta_i^p(\boldsymbol{\theta}) + \varepsilon_i^q(\boldsymbol{\theta}))\right) \tag{17}$$

and let $\mathbf{h} \triangleq \mathrm{vec}(\mathbf{H}) \in \mathbb{C}^{N_TN_R}$.

We define steering vectors for the intra- and inter-frame time scales: for intra-frame, the range steering vector $\mathbf{r}(\tau, v) \in \mathbb{C}^N$ where

$$[\mathbf{r}(\tau,v)]_n \triangleq \exp\left[-j2\pi(n\Delta f\tau + f_n\frac{2v}{c}nT_r)\right]; \tag{18}$$

for inter-frame, the velocity steering vector $\mathbf{v}(v) \in \mathbb{C}^{N_d}$ where

$$[\mathbf{v}(v)]_m \triangleq \exp\left[-j2\pi f_0\frac{2v}{c}mNT_r\right]. \tag{19}$$

Additionally, define the vector of "distortion terms" $\mathbf{c}(v) \in \mathbb{C}^{NN_d}$ where

$$[\mathbf{c}(v)]_{n+mN} \triangleq \exp\left[-j2\pi n\Delta f\frac{2v}{c}mNT_r\right]. \tag{20}$$

Now let

$$\boldsymbol{\phi}(\boldsymbol{\theta},\tau,v) \triangleq \mathbf{h}(\boldsymbol{\theta}) \otimes [(\mathbf{v}(v) \otimes \mathbf{r}(\tau,v)) \odot \mathbf{c}(v)] \in \mathbb{C}^{N_TN_RNN_d}, \tag{21}$$

where $\odot$ is the Hadamard product. Hence the radar signal component can be expressed in vector form as

$$\mathbf{y}_R(k) = \sum_{i\in\mathcal{I}_k} x_i\boldsymbol{\phi}(\boldsymbol{\theta}_i, \overline{\tau}_i(k), v_i). \tag{22}$$

where the coordinate

$$\overline{\tau}_i(k) \triangleq \tau_i - t_k \in \left[-\frac{T}{2}, \frac{T}{2}\right] \tag{23}$$

is the ith scatterer's offset from the center of the kth range cell.

2) Communication Signal Component: The interference component in the projection of receiver q's recording is

$$y_C^q(m,n,p,k) = \langle s_c(t), s^p(m,n,t-t_k)\rangle. \tag{24}$$

The power spectral density of y_C^q for any q is

$$\mathcal{S}_C(f) = \sum_{i\in\mathcal{C}_n} G_i(f - f_i^C)\left|U^p(f - f_n)\right|^2, \tag{25}$$

where

$$\mathcal{C}_n \triangleq \{i \mid |f_n - f_i^C| \leq \frac{\Delta f}{2} + \frac{B_i}{2}\} \tag{26}$$

is the set of carriers that overlap with radar pulse n. Any communication carrier spectrally overlaps with at least one radar pulse; but in general a radar pulse may or may not overlap with any carriers, in which case $\mathcal{C}_n$ would be empty. We have

$$\mathbb{E}[|y_C^q(m,n,p,k)|^2]$$

$$= \int_{-\infty}^{\infty}\sum_{i\in\mathcal{C}_n} G_i(f - f_i^C)\left|U^p(f - f_n)\right|^2 df, \tag{27}$$

implying that only the carriers $\mathcal{C}_n$ may interfere with the radar. Moreover, only a subset of the carriers $\mathcal{C}_n$ actually interfere because G_i implicitly depends on whether carrier i is in use. Therefore, $\mathbb{E}[|y_C^q(m,n,p,k)|^2] = 0$ whenever 1) $\mathcal{C}_n = \emptyset$, or 2) none of the carriers $\mathcal{C}_n$ are in use.

Define $\mathbf{B}(k) \in \mathbb{C}^{N_T\times N_R\times N_d\times N}$ such that $[\mathbf{B}(k)]_{pqmn} \triangleq y_C^q(m,n,p,k)$ and let $\mathbf{b}(k) \triangleq \mathrm{vec}(\mathbf{B}(k)) \in \mathbb{C}^{N_TN_RN_dN}$, such that element i of $\mathbf{b}(k)$ is consistent with element i of $\mathbf{y}_R(k)$. Then the number of nonzero entries in $\mathbf{b}(k)$ is equal to $N_TN_RN_d$ times the number of occurences of spectral overlap. Intuitively,

if the probability of spectrum overlap with an active carrier is small, then $\mathbf{b}(k)$ will be sparse—a plausible instance of this is explored in Section V. For now, we assume that $\mathbf{b}(k)$ contains mostly zeros.

Finally, the projection onto range cell k can be written as

$$\mathbf{y}(k) = \sum_{i \in \mathcal{I}_k} x_i \boldsymbol{\phi}(\boldsymbol{\theta}_i, \overline{\tau}_i(k), v_i) + \mathbf{b}(k) + \mathbf{e}(k), \tag{28}$$

where $\mathbf{e}(k) \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$.

C. Optimization Problem Formulation

Our task is to recover the angle-range-doppler image from the radar measurements (28). To this end, we construct an on-grid radar model and formulate an optimization problem to jointly recover the image and the interference signal. Our approach is to image the contents of a single coarse range cell k; in practice, the following would be applied separately to each desired cell.

The radar data consists of a coherent batch of echo returns from N_d sweeps, given by (9). The projection operation in (10) isolates the returns of all scatterers located in range cell k, yielding a measurement vector of length $N_T N_R N_d N$, given by (28). We assume the scatterers' coordinates in angle-range-velocity space lie on the grid $\mathcal{G} \subset \mathbb{R}^4$, where $|\mathcal{G}| \triangleq M \gg N_T N_R N_d N$. Define $\boldsymbol{\Phi} \in \mathbb{C}^{N_T N_R N_d N \times M}$ whose columns form the dictionary $\mathcal{D} \triangleq \{\boldsymbol{\phi}(\boldsymbol{\theta}, \overline{\tau}, v) \mid (\boldsymbol{\theta}, \overline{\tau}, v) \in \mathcal{G}\}$, where $\boldsymbol{\phi}$ is given by (21). By the on-grid assumption, we have $\{\boldsymbol{\phi}(\boldsymbol{\theta}_i, \overline{\tau}_i(k), v_i) \mid i \in \mathcal{I}_k\} \subseteq \mathcal{D}$. Thus, the radar signal component (22) can be expressed as

$$\mathbf{y}_R(k) = \boldsymbol{\Phi} \mathbf{w}(k), \tag{29}$$

where $\mathbf{w}(k) \in \mathbb{C}^M$ is the vectorized angle-range-doppler image. The nonzero entries of $\mathbf{w}(k)$ form $\{x_i \mid i \in \mathcal{I}_k\}$ and are positioned such that x_i weights $\boldsymbol{\phi}(\boldsymbol{\theta}_i, \overline{\tau}_i(k), v_i)$. Substituting (29) in (28), we obtain

$$\mathbf{y} = \boldsymbol{\Phi} \mathbf{w} + \mathbf{b} + \mathbf{e}, \tag{30}$$

with dependence on k hereafter implied.

Sparsity manifests in two forms: $\mathbf{b}$ is sparse because of the frequency-domain sparsity of the communication signals; $\mathbf{w}$ is sparse if the radar scene is sparse. Exploiting these properties, we formulate the following optimization problem to jointly recover $\mathbf{w}$ and $\mathbf{b}$:

$$\min_{\mathbf{w}, \mathbf{b}} \quad \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w} - \mathbf{b}\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{b}\|_1. \tag{31}$$

Given the measurement $\mathbf{y}$, (31) seeks sparse $\mathbf{w}$ and $\mathbf{b}$ that fit (30), where the hyperparameters $\lambda_1, \lambda_2 > 0$ control the sparsities. The optimal $\mathbf{w}$ is the recovered image.

III. Direct Solver Based on ADMM Algorithm

Next we derive the ADMM equations for (31). ADMM is well-suited to handle high-dimensional problems where the objective can be expressed as the sum of convex functions [33], as typically is the case in signal processing and machine learning, where dimensionality and regularization terms abound. The problem is split into smaller subproblems which often admit closed-form solutions, so an iteration typically requires only a few matrix-vector multiplies [33].

ADMM is often viewed as an approximation of the augmented Lagrange multiplier (ALM) algorithm. ALM solves via gradient ascent the dual of an ℓ_2-regularized version of the primal problem. Evaluating the dual function entails jointly minimizing the augmented Lagrangian, which may be intractable. ADMM instead "approximates" the dual function by its namesake approach, minimizing over the variables in an alternating fashion. But the resemblance to ALM is superficial, as each method can be equated to the repeated application of a unique monotone operator, revealing that each method's convergence mechanism differs from the other's [34]. Indeed, both methods belong to the broader class of proximal algorithms [35]. Nonetheless, we derive ADMM via the augmented Lagrangian.

Let $\mathbf{A} = [\boldsymbol{\Phi} \quad I_N] \in \mathbb{C}^{N \times (M+N)}$, $\mathbf{D}_1 = [I_M \quad 0] \in \mathbb{R}^{M \times (M+N)}$, $\mathbf{D}_2 = [0 \; I_N] \in \mathbb{R}^{N \times (M+N)}$, $\mathbf{x} = \begin{bmatrix} \mathbf{w}^T & \mathbf{b}^T \end{bmatrix}^T \in \mathbb{C}^{M+N}$, where I_n denotes the $n \times n$ identity matrix. Then (31) is equivalent to

$$\min_{\mathbf{x}} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda_1 \|\mathbf{D}_1 \mathbf{x}\|_1 + \lambda_2 \|\mathbf{D}_2 \mathbf{x}\|_1. \tag{32}$$

We reformulate (32) as the constrained problem

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \quad & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda_1 \|\mathbf{D}_1 \mathbf{z}\|_1 + \lambda_2 \|\mathbf{D}_2 \mathbf{z}\|_1 \\ \text{s.t.} \quad & \mathbf{x} - \mathbf{z} = 0, \end{aligned} \tag{33}$$

whose augmented Lagrangian is

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u}) = {} & \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda_1 \|\mathbf{D}_1 \mathbf{z}\|_1 + \lambda_2 \|\mathbf{D}_2 \mathbf{z}\|_1 \\ & + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2 - \frac{\rho}{2} \|\mathbf{u}\|_2^2, \end{aligned} \tag{34}$$

where $\mathbf{u}$ is the scaled dual variable [33] and $\rho \in \mathbb{R}$ is a parameter. ALM entails computing the dual function $\min_{\mathbf{x}, \mathbf{z}} \{L_\rho\}$ exactly by jointly minimizing L_ρ over $\mathbf{x}$ and $\mathbf{z}$, which may be costly because of the nonlinear term involving $\mathbf{x}$ and $\mathbf{z}$. ADMM instead minimizes along the $\mathbf{x}$ and $\mathbf{z}$ directions in an alternating fashion.

"Vanilla" ADMM comprises three steps: minimization of L_ρ over $\mathbf{x}$; minimization of L_ρ over $\mathbf{z}$; and finally a gradient ascent iteration, incrementing $\mathbf{u}$ using the gradient of L_ρ w.r.t. $\mathbf{u}$. Namely, ADMM sequentially computes

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}} \left(\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k\|_2^2 \right) \tag{35}$$

$$\begin{aligned} \mathbf{z}^{k+1} = \operatorname*{argmin}_{\mathbf{z}} \big(& \lambda_1 \|\mathbf{D}_1 \mathbf{z}\|_1 + \lambda_2 \|\mathbf{D}_2 \mathbf{z}\|_1 \\ & + \frac{\rho}{2} \|\mathbf{x}^{k+1} - \mathbf{z} + \mathbf{u}^k\|_2^2 \big) \end{aligned} \tag{36}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \nabla_{\mathbf{u}} L_\rho(\mathbf{x}^{k+1}, \mathbf{z}^{k+1}, \mathbf{u}). \tag{37}$$

Equation (35) is an ℓ_2-regularized least-squares problem. Equation (36) can be separated into

$$\mathbf{z}_1^{k+1} = \operatorname*{argmin}_{\mathbf{z}_1} \left(\lambda_1 \|\mathbf{z}_1\|_1 + \frac{\rho}{2} \|\mathbf{x}_1^{k+1} - \mathbf{z}_1 + \mathbf{u}_1^k\|_2^2 \right) \tag{38}$$

$$\mathbf{z}_2^{k+1} = \operatorname*{argmin}_{\mathbf{z}_2} \left(\lambda_2 \|\mathbf{z}_2\|_1 + \frac{\rho}{2} \|\mathbf{x}_2^{k+1} - \mathbf{z}_2 + \mathbf{u}_2^k\|_2^2 \right), \tag{39}$$

where $\mathbf{z}_i \triangleq \mathbf{D}_i \mathbf{z}$, $\mathbf{x}_i^k \triangleq \mathbf{D}_i \mathbf{x}^k$ and $\mathbf{u}_i^k \triangleq \mathbf{D}_i \mathbf{u}^k$, $i = 1, 2$. Solutions to (38) and (39) are given by the proximal operator of the

ℓ_1-norm, the soft-thresholding operator $S_\kappa : \mathbb{C}^n \to \mathbb{C}^n$. Here S_κ operates elementwise, so that element i of the output for input $\mathbf{a} = [a_1 \ \cdots \ a_n]^T$ is

$$[S_\kappa(\mathbf{a})]_i = \frac{a_i}{|a_i|} * \max(|a_i| - \kappa, 0). \tag{40}$$

Thus the vanilla ADMM equations for (31) are

$$\mathbf{x}^{k+1} = (\mathbf{A}^H\mathbf{A} + \rho I)^{-1}(\mathbf{A}^H\mathbf{y} + \rho(\mathbf{z}^k - \mathbf{u}^k)) \tag{41}$$

$$\mathbf{z}_1^{k+1} = S_{\lambda_1/\rho}(\mathbf{x}_1^{k+1} + \mathbf{u}_1^k) \tag{42}$$

$$\mathbf{z}_2^{k+1} = S_{\lambda_2/\rho}(\mathbf{x}_2^{k+1} + \mathbf{u}_2^k) \tag{43}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{x}^{k+1} - \begin{bmatrix}\mathbf{z}_1^{k+1} \\ \mathbf{z}_2^{k+1}\end{bmatrix}. \tag{44}$$

Our proposed ADMM algorithm augments vanilla ADMM in two ways. It is known that inserting a relaxation step between the $\mathbf{x}$ and $\mathbf{z}$ updates,

$$\boldsymbol{\xi}^{k+1} = \alpha\mathbf{x}^{k+1} + (1-\alpha)\mathbf{z}^k, \tag{45}$$

where $\alpha \in [0, 2]$ is a parameter, may improve convergence speed [33]. This step also arises naturally in an alternative ADMM derivation [34]. Additionally, we introduce a parameter $\eta \in \mathbb{R}$ to control the gradient step-size in the $\mathbf{u}$-update. Finally, the proposed ADMM iteration for (31) is

$$\mathbf{x}^{k+1} = (\mathbf{A}^H\mathbf{A} + \rho I)^{-1}(\mathbf{A}^H\mathbf{y} + \rho(\mathbf{z}^k - \mathbf{u}^k)) \tag{46}$$

$$\boldsymbol{\xi}^{k+1} = \alpha\mathbf{x}^{k+1} + (1-\alpha)\mathbf{z}^k \tag{47}$$

$$\mathbf{z}_1^{k+1} = S_{\lambda_1/\rho}(\boldsymbol{\xi}_1^{k+1} + \mathbf{u}_1^k) \tag{48}$$

$$\mathbf{z}_2^{k+1} = S_{\lambda_2/\rho}(\boldsymbol{\xi}_2^{k+1} + \mathbf{u}_2^k) \tag{49}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \eta\left(\boldsymbol{\xi}^{k+1} - \begin{bmatrix}\mathbf{z}_1^{k+1} \\ \mathbf{z}_2^{k+1}\end{bmatrix}\right). \tag{50}$$

where $\boldsymbol{\xi}_i^k = \mathbf{D}_i\boldsymbol{\xi}^k$, $i = 1, 2$.

This ADMM bears resemblance to that proposed in [19], [20], and may be seen as a special case thereof. The authors in [19], [20] consider the objective

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{l=1}^{L} \lambda_l g(\mathbf{D}_l\mathbf{x}),$$

for some L with arbitrary matrices $\mathbf{D}_l$ and regularizer g. For this generalized objective, multiple ADMM algorithms exist depending on how the auxiliary variable $\mathbf{z}$ is defined. Some instances of ADMM may involve challenging subproblems that may have to be approximated; in one instance of ADMM considered in [19], [20], a subproblem solution is approximated via a number of gradient descent iterations. However, in our setting ($L = 2$, $g(\cdot) \triangleq \|\cdot\|_1$, and $\mathbf{D}_l$ defined above) all instances of ADMM are equal the one we have derived.

The main pitfall of ADMM we aim to address is task of setting the parameters $\{\rho, \alpha, \eta, \lambda_1, \lambda_2\}$ which in general must be tuned for each application. Vanilla ADMM is guaranteed to converge at a linear rate for all $\rho > 0$ [36], but in practice the convergence speed as well as accuracy may significantly vary with ρ. Selection of ρ may be based on the eigenvalues of $\mathbf{A}$ [37], and ρ can be updated based on the values of the primal and dual residuals at each iteration [38]. From our experience, the ADMM parameters ρ, α, η primarily influence convergence speed, while the regularization parameters λ_1, λ_2 affect convergence accuracy. The ℓ_1 parameters can also be updated at each iteration, e.g. LARS determines a parameter schedule by calculating the solution path for every positive value of the regularization parameter [39]. Otherwise, cross-validation can be effective. The deep unfolding method we present next can be seen as a way of automating hyperparameter cross-validation, wherein algorithm hyperparameters become decision variables for optimizing a metric of algorithm performance.

IV. Complex ADMM-Net

Here we outline the unfolded network design process and then detail the proposed ADMM-Net design. One important consideration is how to handle complex valued data and parameters. Mainstream deep learning software supports only real-valued inputs and parameters, while radar measurements are typically complex-valued, so we must somehow translate ADMM's complex-valued operations into an equivalent sequence of real-valued operations. The network's forward pass, upon initialization, is identical to executing a number of ADMM iterations.

A. Towards ADMM-Net

A neural network is a composition of parameterized linear and nonlinear functions called *layers*. Deep learning is the process of adjusting the function parameters such that the network emulates some desired mapping. This amounts to optimizing over the parameters a chosen loss metric that quantifies the accuracy of the network output relative to training data, a putative sample of the desired mapping's input/output behavior. Typically gradient-based algorithms are used for the optimization, so by leveraging standard deep learning software packages, such as Tensorflow and PyTorch, which employ automatic differentiation to compute gradients, many iterative algorithms can readily be parameterized, cast as a series of network layers, and then optimized as such, a technique known as *deep unfolding*.

Unfolding an algorithm iteration into a set of feed-forward neural network layers requires specification of a) the functional dependencies between the algorithm iterates and b) the parameters to be learned. Fig. 2 depicts the *data flow graph* for the proposed ADMM iteration. Each node corresponds to an iterate, and an arrow indicates functional dependence between iterates such that the iterate associated with an arrow's head is a function of that of the arrow's tail. To generate the unfolded network, we create one layer per node. The inputs to the layer associated with a node v are the outputs of the layers associated with the tails of all arrows directed to v. A layer's input/output mapping is based on its corresponding iterate's update equation in the original algorithm or a generalized version thereof. Therefore, if the algorithm comprises n update equations, every n consecutive layers of the unfolded network together correspond to a single

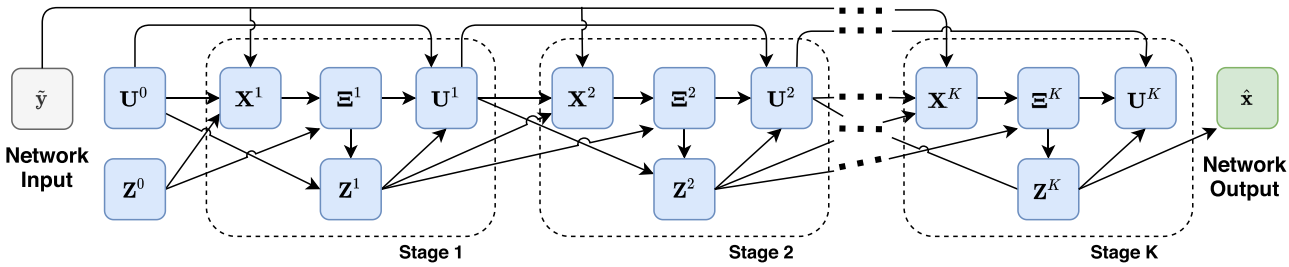
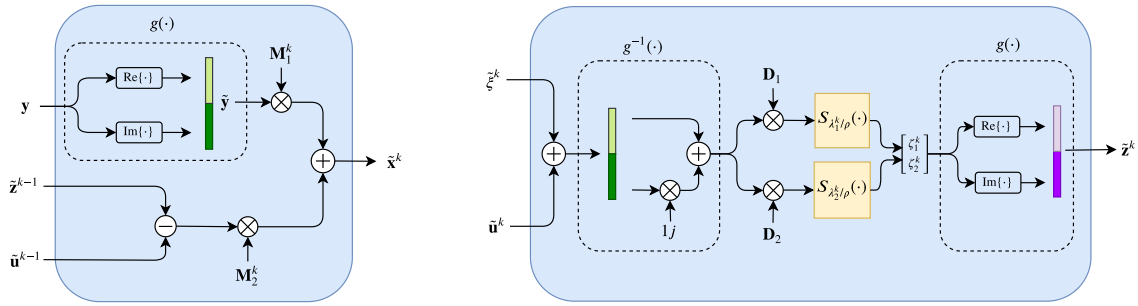


Fig. 2. Data flow graph for ADMM-net.



(a) (b)

Fig. 3. Block diagrams for reconstruction (a) and nonlinear transform (b) layers.

algorithm iteration, referred to as a network "stage" [19], [20]; see the nodes enclosed by the dashed-lines in Fig. 2.

B. ADMM-Net Structure

ADMM-Net has layer operations based on (46)-(50). Stage k of the network consists of a reconstruction layer $\mathbf{X}^k$ that corresponds to the $\mathbf{x}$-update, a relaxation layer $\mathbf{\Xi}^k$ that corresponds to the $\boldsymbol{\xi}$-update, a nonlinear transform layer $\mathbf{Z}^k$ that corresponds to the $\mathbf{z}$-update, and a dual update layer $\mathbf{U}^k$ that corresponds to the $\mathbf{u}$-update. In addition to the ADMM algorithm parameters in each layer, we also learn the linear transformations in the $\mathbf{x}$-update, initializing them as prescribed by ADMM.

Network Input: The network input $\mathbf{y} \in \mathbb{C}^N$ enters the network via the reconstruction layers.

Reconstruction Layer: This layer performs the complex $\mathbf{x}$-update prescribed by ADMM. The inputs to this layer are the network input $\mathbf{y} \in \mathbb{C}^N$, and $\tilde{\mathbf{z}}^{k-1}, \tilde{\mathbf{u}}^{k-1} \in \mathbb{R}^{2M}$ from stage $k-1$. The output of the stage k reconstruction layer is

$$\tilde{\mathbf{x}}^k = \mathbf{M}_1^k g(\mathbf{y}) + \mathbf{M}_2^k(\tilde{\mathbf{z}}^{k-1} - \tilde{\mathbf{u}}^{k-1}), \tag{51}$$

and hence $\tilde{\mathbf{x}}^k \in \mathbb{R}^{2M}$. The entries of the matrices $\mathbf{M}_1^k \in \mathbb{R}^{2M\times 2N}$ and $\mathbf{M}_2^k \in \mathbb{R}^{2M\times 2M}$ are learnable parameters. The function $g : \mathbb{C}^N \to \mathbb{R}^{2N}$ vertically concatenates the input's real and imaginary parts into a single real-valued vector: if $\mathbf{y} \in \mathbb{C}^N$, then

$$g(\mathbf{y}) := \begin{bmatrix} \mathrm{Re}\{\mathbf{y}\} \\ \mathrm{Im}\{\mathbf{y}\} \end{bmatrix} \in \mathbb{R}^{2N}. \tag{52}$$

The block diagram for g is shown in Fig. 3(a). Thus $\tilde{\mathbf{x}}^k$ is produced by stacking the real and imaginary parts of (46) into a single vector, i.e. $\tilde{\mathbf{x}}^k \equiv \begin{bmatrix} \mathrm{Re}\{\mathbf{x}^k{}^T\} & \mathrm{Im}\{\mathbf{x}^k{}^T\} \end{bmatrix}^T$. The values $\tilde{\mathbf{z}}^0 = 0$ and $\tilde{\mathbf{u}}^0 = 0$ are used for the first reconstruction layer.

Fig. 3(a) illustrates the kth reconstruction layer: the real and imaginary parts of the complex-valued observation $\mathbf{y}$ are vertically concatenated via g to form $\tilde{\mathbf{y}}$. Then $\mathbf{M}_1^k$ premultiplies $\tilde{\mathbf{y}}$ and $\mathbf{M}_2^k$ premultiplies $\tilde{\mathbf{z}}^{k-1} - \tilde{\mathbf{u}}^{k-1}$. The two resulting vectors are summed to obtain the layer output $\tilde{\mathbf{x}}^k$.

Relaxation Layer (stage k): The output of this layer is

$$\tilde{\boldsymbol{\xi}}^k = \alpha^k \tilde{\mathbf{x}}^k + (1-\alpha^k)\tilde{\mathbf{z}}^{k-1}, \tag{53}$$

where $\alpha^k > 0$ is a learnable parameter. The output $\tilde{\boldsymbol{\xi}}^k \in \mathbb{R}^{2M}$ is the concatenation of the real and imaginary parts of (47), i.e. $\tilde{\boldsymbol{\xi}}^k \equiv \begin{bmatrix} \mathrm{Re}\{\boldsymbol{\xi}^k{}^T\} & \mathrm{Im}\{\boldsymbol{\xi}^k{}^T\} \end{bmatrix}^T$.

Nonlinear Transform Layer: This layer applies the soft-thresholding operation as in the ADMM $\mathbf{z}$-update (48)-(49). The layer output is given by

$$\boldsymbol{\zeta}_1^k = S_{\lambda_1^k}(\mathbf{D}_1 g^{-1}(\tilde{\boldsymbol{\xi}}^k + \tilde{\mathbf{u}}^{k-1})) \tag{54}$$

$$\boldsymbol{\zeta}_2^k = S_{\lambda_2^k}(\mathbf{D}_2 g^{-1}(\tilde{\boldsymbol{\xi}}^k + \tilde{\mathbf{u}}^{k-1})) \tag{55}$$

$$\tilde{\mathbf{z}}^k = g\left(\begin{bmatrix} \boldsymbol{\zeta}_1^k \\ \boldsymbol{\zeta}_2^k \end{bmatrix}\right), \tag{56}$$

where $\lambda_1^k, \lambda_2^k > 0$ are the learnable ℓ_1-regularization parameters and $S_\kappa(\mathbf{a})_i = \frac{a_i}{|a_i|} * \max(|a_i| - \kappa, 0)$ is the soft-thresholding operator. The operation $g^{-1} : \mathbb{R}^{2M} \to \mathbb{C}^M$ forms a complex vector out of the top and bottom halves of the input vector: if $\tilde{\mathbf{x}} \in \mathbb{R}^{2M}$ then

$$g^{-1}(\tilde{\mathbf{x}}) := \tilde{\mathbf{x}}[0 : M-1] + j\tilde{\mathbf{x}}[M : 2M-1] \in \mathbb{C}^M, \tag{57}$$

where the notation $\mathbf{a}[k:l]$ refers to a vector containing the kth through the lth components inclusive, of the vector $\mathbf{a}$. The block diagram for g^{-1} is shown on the left-hand side of Fig. 3(b). The matrices $\mathbf{D}_1 = [I_M\ 0] \in \mathbb{R}^{M\times(M+N)}$ and $\mathbf{D}_2 = [0\ I_N] \in \mathbb{R}^{N\times(M+N)}$ partition $\mathbf{z}$ as in (38)-(39).

Fig. 3(b) illustrates this layer's operations. The layer inputs are summed and input to g^{-1}, the output of which is partitioned via premultiplication by $\mathbf{D}_1$ and $\mathbf{D}_2$. Soft-thresholding is applied with the respective thresholding parameters λ_1^k and λ_2^k. The outputs are concatenated into $\begin{bmatrix}\boldsymbol{\zeta}_1^k T & \boldsymbol{\zeta}_2^k T\end{bmatrix}^T$, whose real and imaginary parts are concatenated to form the output $\tilde{\mathbf{z}}^k$. The output of the Kth nonlinear transform layer goes to the network output layer.

Dual Update Layer: The output of this layer is

$$\tilde{\mathbf{u}}^k = \tilde{\mathbf{u}}^{k-1} + \eta^k(\tilde{\boldsymbol{\xi}}^k - \tilde{\mathbf{z}}^k), \tag{58}$$

where η^k is a learnable parameter corresponding to the gradient step size. The variable $\tilde{\mathbf{u}}^k \in \mathbb{R}^{2N}$ corresponds to the concatenation of the real and imaginary parts of (47), i.e. $\tilde{\mathbf{u}}^k \equiv \begin{bmatrix}\mathrm{Re}\{\mathbf{u}^k{}^T\} & \mathrm{Im}\{\mathbf{u}^k{}^T\}\end{bmatrix}^T$.

Network Output: The network output is derived from the output of the final nonlinear transform layer $\tilde{\mathbf{z}}^K$ via

$$\hat{\mathbf{x}} = g^{-1}(\tilde{\mathbf{z}}^K), \tag{59}$$

where g^{-1} is defined in (57).

We define $\mathbf{D}_1$ and $\mathbf{D}_2$ as constants, which differs from the architecture proposed in [19], [20]: [19], [20] declares them as unknown "sparsifying" transforms to be learned—appropriate when the desired signal is not per se sparse, as in the application considered in [19], [20]. In our setting, however, the desired signal is by assumption sparse; hence the sparsifying transforms are trivial and we elect not to learn them.

C. Training Details

1) Parameter Set: Stage k of the network has learnable parameters $\{\mathbf{M}_1^k, \mathbf{M}_2^k, \alpha^k, \lambda_1^k, \lambda_2^k, \eta^k\}$. The scalar parameters are initialized according to either theoretically or empirically justified values, as described in Section V. The matrices $\mathbf{M}_1^k \in \mathbb{R}^{2M\times 2N}$ and $\mathbf{M}_2^k \in \mathbb{R}^{2M\times 2M}$ are initialized such that the reconstruction layer's operation is initially equivalent to (46). All stages are identically initialized according to

$$\mathbf{M}_1^k \leftarrow \begin{bmatrix}\mathrm{Re}\{\mathbf{P}\mathbf{A}^T\} & -\mathrm{Im}\{\mathbf{P}\mathbf{A}^T\} \\ \mathrm{Im}\{\mathbf{P}\mathbf{A}^T\} & \mathrm{Re}\{\mathbf{P}\mathbf{A}^T\}\end{bmatrix} \in \mathbb{R}^{2M\times 2N}$$

$$\mathbf{M}_2^k \leftarrow \begin{bmatrix}\mathrm{Re}\{\rho\mathbf{P}\} & -\mathrm{Im}\{\rho\mathbf{P}\} \\ \mathrm{Im}\{\rho\mathbf{P}\} & \mathrm{Re}\{\rho\mathbf{P}\}\end{bmatrix} \in \mathbb{R}^{2M\times 2M}, \tag{60}$$

where $\mathbf{P} \triangleq (\mathbf{A}^T\mathbf{A} + \rho I)^{-1}$.

2) Training Data: The training data set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_{\text{train}}}$ of cardinality N_{train} comprises the radar system's response $\mathbf{y}_i$ to a known $\mathbf{x}_i = [\mathbf{w}_i^T \quad \mathbf{b}_i^T]^T$, a scene with parameters $\mathbf{w}_i$ illuminated amidst interference $\mathbf{b}_i$. Using real-world data for training could potentially ameliorate model mismatch due to hardware imperfections, etc. In our experiments training data were randomly generated via the signal model, as detailed in Section V-E1.

TABLE I
INDEX OF MIMO RADAR VARIABLES

Symbol	Definition
N	No. frequency steps
N_d	No. sweeps
N_T	No. transmitters
N_R	No. receivers
f_0	Start frequency
Δf	Frequency step size
f_n	$f_0 + n\Delta f,\ 0 \le n \le N-1$
u^p	Transmitter p's pulse envelope
s^p	Transmitter p's waveform
r^q	Radar return at receiver q
T	Pulse duration (all transmitters)
T_r	Pulse-repetition interval
t	Continuous fast-time, absolute
m	Sweep index
n	Pulse index within sweep
i	Scatterer index
L	No. scatterers
x_i	Scattering coeff.
τ_i^{pq}	Absolute delay, (p,q) Tx/Rx pair
δ_i^p	Marginal delay, pth Tx
ε_i^q	Marginal delay, qth Rx
τ_i	Absolute delay, reference Tx/Rx pair
$\overline{\tau}_i(k)$	Delay offset, kth range cell
v_i	Radial velocity
$\boldsymbol{\theta}_i$	Direction coordinates

3) Loss Function: The loss function of the network is the mean-squared error

$$\mathcal{L}(\hat{\mathbf{x}}_i; \mathbf{x}_i) = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2, \tag{61}$$

where $\hat{\mathbf{x}}$ is the network's output and $\mathbf{x}_i$ is the ground truth.

V. SIMULATIONS

We compare the performance of ADMM-Net, ADMM, and the CVX semi-definite program solver in a simulated interference environment in which a MIMO stepped-frequency radar shares spectrum with a SC-FDMA communication system.

A. Angle-Range-Velocity Imaging

Simulated radar measurements are generated according to the on-grid model (30). The simulated (toy-sized) stepped-frequency MIMO radar parameters are listed in Table II. The scattering coefficients x_i are independently sampled from $\mathcal{CN}(0, \sigma_x^2)$, where σ_x^2 is the variance. The columns of $\boldsymbol{\Phi}$ are scaled to have unit norm. Without loss of generality, we consider the radar processing for the range cell $k = 0$.

The Tx and Rx arrays are co-planar uniform linear arrays, with respective normalized element spacings d_T and d_R (normalized by the start carrier wavelength f_0/c), arranged in a cross-shaped geometry [40]. The array response to a scatterer at angular coordinates $(\theta_1, \theta_2) \in \mathbb{R}^2$, where θ_1 is the direction relative to the Rx array and θ_2 is the direction relative to the Tx array, is

TABLE II
RADAR SIMULATION PARAMETERS

Symbol	Value	Description
N	4	No. frequency steps
N_d	4	No. sweeps
N_T	2	No. transmitters
N_R	2	No. receivers
f_0	2 GHz	Start frequency
Δf	1 MHz	Frequency step size
T	1 μs	Pulse duration
T_r	66 μs	Pulse-repetition interval
$R_{\max}$	9900 m	Max. unambiguous range
$v_{\max}$	$\pm$142 m/s	Max. unambiguous velocity
d_T	1	Tx array normalized spacing
d_R	1	Rx array normalized spacing
σ^2	various	AWGN variance
σ_x^2	1	Scattering coefficient variance
x_i	$\sim \mathcal{CN}(0, \sigma_x^2)$	Scattering coefficient i

given by

$$\mathbf{h}(\theta_1, \theta_2) = \mathbf{h}_R(\theta_1) \otimes \mathbf{h}_T(\theta_2), \tag{62}$$

where

$$\mathbf{h}_R(\theta_1) \triangleq \begin{bmatrix} 1 & e^{-j2\pi d_R \theta_1} & \dots & e^{-j2\pi d_R \theta_1 (N_R-1)} \end{bmatrix} \tag{63}$$

$$\mathbf{h}_T(\theta_2) \triangleq \begin{bmatrix} 1 & e^{-j2\pi d_T \theta_2} & \dots & e^{-j2\pi d_T \theta_2 (N_T-1)} \end{bmatrix}. \tag{64}$$

We let $\mathcal{G} = \mathcal{T} \times \mathcal{V} \times \Theta_1 \times \Theta_2$, where

$$\mathcal{T} = \{Tm/M_\tau \mid m = -M_\tau/2, \dots, M_\tau/2 - 1\} \tag{65}$$

$$\mathcal{V} = \{v_{\max} m/M_v \mid m = -M_v/2, \dots, M_v/2 - 1\} \tag{66}$$

$$\Theta_1 = \{m/M_{\theta_1} \mid m = 0, 1, \dots, M_{\theta_1 - 1}\} \tag{67}$$

$$\Theta_2 = \{m/M_{\theta_2} \mid m = 0, 1, \dots, M_{\theta_2 - 1}\} \tag{68}$$

are the delay, velocity, and angle grids, and M_τ, M_v, and M_{θ_1} and M_{θ_2} are the respective grid sizes. Recall that $\overline{\tau} \in [-\frac{T}{2}, \frac{T}{2}]$ is the offset from the center of the coarse range cell. The absolute delay τ is recovered by $\tau = \overline{\tau} + t_k$, where t_k is the center of the coarse range cell.

We choose $M_\tau = 5$, $M_v = 5$, $M_{\theta_1} = 3$, and $M_{\theta_2} = 2$, hence $\mathbf{\Phi} \in \mathbb{C}^{64 \times 150}$. To avoid aliasing, we require $|v| \leq \frac{c}{4 f_0 N T_r} \triangleq v_{\max}$, and assuming $d_R = d_T = 1$, $0 \leq \theta_1 \leq 1$ and $0 \leq \theta_2 \leq 1$. The maximum unambiguous range is thus $R_{\max} = cT_r/2$ meters. Each coarse range cell is of size $cT/2 = 150$ meters and the conventional, DFT-based range profile resolution is $\frac{cT}{2N} = 37.5$ meters. The maximum unambiguous velocity is $\pm v_{\max} = \pm \frac{c}{4 f_0 N T_r}$.

B. Sc-Fdma

As a concrete example, consider an uplink SC-FDMA system, as specified in the 5 G New Radio standard released by 3GPP in December 2017. Suppose the system bandwidth consists of N_s subcarriers with uniform spacing Δf^C and every $K \in \mathbb{Z}^+$ consecutive subcarriers are grouped into channels with center frequencies $f_i^C = f_0^C + iK\Delta f^C,\ 0 \leq i \leq N_c - 1$, where f_0^C is the start frequency, each channel has bandwidth $K\Delta f^C$, for a total of $N_c = \lfloor N_s/K \rfloor$ channels. Users are assigned one or more channels over which to transmit. The signal transmitted over channel i has the form

$$g_i(t) = \sqrt{\gamma_i} h_i \sum_{n_c=-\infty}^{\infty} \sum_{k=0}^{K-1} a_{ik}(n_c) u_C(t - n_c T_c) \times \exp\left[j2\pi (f_i^C + k\Delta f^C) t\right], \tag{69}$$

where:

- γ_i is the power level assigned to channel i.
- $h_i \sim \mathcal{CN}(0, \beta)$ are i.i.d. channel fading coefficients. A block fading channel model is assumed and K is chosen such that $K\Delta f^C$ equals the coherence bandwidth (~ 0.5 MHz) [41]. Therefore each channel i is characterized by a single fading coefficient h_i that is statisticaly independent of all other channels. The variance β accounts for additional user-dependent effects (e.g. path loss and log-normal shadowing) [41]. For simplicity, we assume β is the same for all users.
- $\{a_{ik}(n_c) \in \mathbb{C} : 0 \leq k \leq K-1,\ 0 \leq i \leq N_c - 1,\ n_c \in \mathbb{Z}\}$ are random variables representing the transmitted symbol sequence, comprising the data and cyclic prefix, with $a_{ik}(n_c)$ transmitted on subcarrier k of channel i during the n_cth data block. In SC-FDMA the transmitted symbols $a_{ik}(n_c),\ k = 0, \dots, K-1$ are the isometric discrete Fourier transform (DFT) coefficients of the original data symbol sequence. We assume the original data symbols adhere to a memoryless modulation format.
- T_c is the block duration (cyclic prefix plus data); for example, in 5 G $\Delta f^C \sim 15$ kHz, so $T_c \sim 1/(15\text{ kHz}) = 66\ \mu s$.
- $$u_C(t) \triangleq \begin{cases} \sqrt{\frac{1}{T_c}} & 0 \leq t \leq T_c \\ 0 & \text{otherwise} \end{cases} \tag{70}$$
  is the normalized pulse envelope.

TABLE III
SC-FDMA PARAMETERS

Parameter	Value	Description
f_0^C	2 GHz	Start frequency
$K\Delta f^C$	0.5 MHz	Channel bandwidth
N_c	8	Number of channels
γ	1	Power assigned to each channel
ϵ	various	Proportion of active channels
β	various	Variance of channel fading coefficient

Table III lists the simulated SC-FDMA system parameters. Without loss of generality, in the simulations we make the following assumptions:

1) $f_0^C = f_0$. The radar and SC-FDMA system have the same start frequency, f_0.
2) $N_c K \Delta f^C = N \Delta f$. The SC-FDMA bandwidth equals the radar sweep bandwidth, and therefore the SC-FDMA system is the only source of interference. The extension to multiple interference sources is straightforward since each source would occupy a distinct frequency band. An analysis along the lines presented here would be carried out for each interference source.

3) $\frac{\Delta f}{K\Delta f^C} \triangleq L \in \mathbb{Z}^+$. The sweep bandwidth is an integer multiple of the channel bandwidth. For example, the coherence bandwidth is typically ~ 0.5 MHz and typically $\Delta f \geq 1$ MHz.
4) $\gamma_i = \gamma$ for all i, where $\gamma > 0$ is a constant.
5) We suppose the scheduling takes place on a PRI-by-PRI basis (in LTE, resource blocks are allocated in time intervals on the order of 1 ms, while the radar sweep duration may be tens of milliseconds). Let Ω denote the sample space of all possible active channel configurations—i.e., the power set of $\{n \in \mathbb{Z} \mid 0 \leq n \leq N_c - 1\}$—and let $A_i \subset \Omega$ denote the event channel i is active during any given radar pulse, where the probability of A_i is $P(A_i)$. We assume a random sample from Ω is drawn every PRI.
6) $\{a_{ik}(n_c) : \forall i, k, n_c\}$ are i.i.d., uncorrelated, and normalized, such that

$$\mathbb{E}[a_{ik}(n_c)a_{ik'}(n_c)^*|A_i] = \begin{cases} 1 & \text{if } k = k' \\ 0 & \text{if } k \neq k' \end{cases}. \tag{71}$$

In practice, the cyclic prefix violates the uncorrelatedness assumption, but the discrepancy will be small to the extent that the length of the channel impulse response is small relative to the symbol duration (e.g. in LTE the cyclic prefix duration is around 7% of the data symbol duration). Also, if the symbols are normalized, then by the norm-preserving property of the isometric DFT, the original data symbols belong to a normalized symbol set.
(7) $a_{ik}(n_c)$ and h_i are mutually independent for all i, k, and n_c.
(8) $P(A_i) \triangleq \epsilon \in [0, 1]$ for all i. This implies $\mathbf{b}$ is sparse with high probability whenever ϵ is small.

C. Signal-to-Noise Ratio

We define the signal-to-noise ratio (SNR) for a given range cell k as

$$\text{SNR} \triangleq \frac{\mathbb{E}\left[\|\mathbf{y}_R(k)\|_2^2\right]}{\mathbb{E}[\|\mathbf{e}(k)\|_2^2]} \tag{72}$$

where $\mathbf{y}_R$ is given in (22) and $\mathbf{e}(k) \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$.

D. Signal-to-Interference Ratio

The signal-to-interference ratio (SIR) for a given range cell k is defined as

$$\text{SIR} \triangleq \frac{\mathbb{E}\left[\|\mathbf{y}_R(k)\|_2^2\right]}{\mathbb{E}[\|\mathbf{b}(k)\|_2^2]} \tag{73}$$

where $\mathbf{b}$ is given by (28).

E. Algorithm Specifications

1) Admm-Net: Training data pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_{\text{train}}}$ were generated as follows. The complex-valued ground truth $\mathbf{x}_i = [\mathbf{w}_i^T \quad \mathbf{b}_i^T]^T \in \mathbb{C}^M$ was created such that $\mathbf{w}_i$ and $\mathbf{b}_i$ satisfy given sparsity levels. The nonzero entries of $\mathbf{w}_i$ were generated i.i.d. $\mathcal{CN}(0, 1)$. The nonzero entries of $\mathbf{b}_i$ were generated i.i.d. $\mathcal{CN}(0, \beta)$, where β was chosen to satisfy a given SIR. The noise $\mathbf{e}_i$ was drawn from $\mathcal{CN}(0, \sigma^2\mathbf{I})$, where σ^2 was chosen to satisfy a given SNR. The complex-valued simulated measurements are then produced by $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{e}_i$.

Training data sets contained $N_{\text{train}} = 4.5 \times 10^6$ samples and the networks were trained for 45 epochs, i.e., full passes over the training set. The Adam [42] optimizer was used with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a batch size of 500. The Adam learning rate was initialized to 10^{-3} and multiplied by 10^{-1} every 15 epochs. All networks were implemented[1] in Tensorflow 2.

The scalar network parameters were initialized identically for all layers k as

$$\alpha^k = 1.5 \qquad \lambda_1^k = 0.01 \tag{74}$$

$$\eta^k = 1 \qquad \lambda_2^k = 0.005 \tag{75}$$

$$\rho^k = 0.01. \tag{76}$$

The value for λ_2^k was determined by cross-validation; η^k was set to accord with the vanilla ADMM equations; α^k was set as recommended [33]; ρ^k was set as recommended [43]. The matrices $\mathbf{M}_1^k$ and $\mathbf{M}_2^k$ were initialized according to (60).

2) Admm: An ADMM iteration is given by Eqs. (46)-(50). We use the following parameter values:

$$\alpha = 1.5 \qquad \lambda_1 = 0.01 \tag{77}$$

$$\eta = 1 \qquad \lambda_2 = 0.005 \tag{78}$$

$$\rho = 0.01. \tag{79}$$

The justification for these values is the same as that for the ADMM-net parameter initialization, see Section V-E1.

3) Cvx: For CVX, we used the semi-definite program (SDP) solver on the problem

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & \|\mathbf{z}\|_2^2 + \lambda_1\|\mathbf{D}_1\mathbf{x}\|_1 + \lambda_2\|\mathbf{D}_2\mathbf{x}\|_1 \\ \text{s.t.} \quad & \mathbf{z} = \mathbf{y} - \mathbf{A}\mathbf{x} \end{aligned} \tag{80}$$

with parameter values

$$\lambda_1^k = 0.01 \qquad \lambda_2^k = 0.005, \tag{81}$$

where λ_2 was found through cross-validation.

4) ADMM Single-Penalty: To highlight the benefit of the proposed two-penalty objective (31), we also consider the problem

$$\min_{\mathbf{w}} \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \lambda_1\|\mathbf{w}\|_1. \tag{82}$$

We ran the associated ADMM algorithm with parameters

$$\alpha = 1.5 \qquad \lambda_1 = 0.5 \tag{83}$$

$$\eta = 1 \qquad \rho = 0.5. \tag{84}$$

F. Results

The following experiments probe network performance and robustness along four dimensions: network depth (number of stages), SNR, SIR, and sparsity level. We trained several networks, each on a data set with with certain properties. Some

[1]https://github.com/johnston-jeremy/admm_net_interference_removal

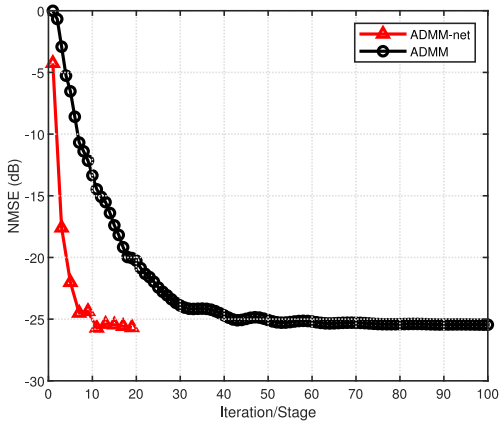


Fig. 4. NMSE versus iteration/stages of ADMM/ADMM-Net. The training and test sets have SNR $= \infty$, $\|\mathbf{w}\|_0 = 2$, $\|\mathbf{b}\|_0 = 16$ and SIR $= 0$ dB.

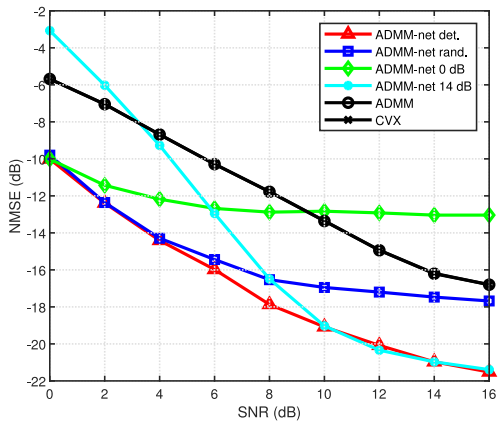


Fig. 5. NMSE versus SNR. 2 scatterers, 25% spectrum overlap, SIR $= 0$.

training sets had properties constant across samples (e.g. each sample had SNR $= 10$), others contained samples with properties drawn uniformly at random (e.g. each sample had SNR between 0 and 15). The networks were evaluated in two ways. One, on test data akin to their training sets. Two, on multiple test sets, each sampled from a particular distribution, in order to test the network's robustness to deviation from the training distribution.

We evaluate the candidate methods via the average normalized mean squared error (NMSE) of their estimates, defined as

$$\text{NMSE} = 10\log_{10}\left[\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{\text{test}}}\frac{\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2}{\|\mathbf{x}_i\|_2^2}\right] \text{dB}, \qquad (85)$$

where $\hat{\mathbf{x}}_i$ is the algorithm output and $\mathbf{x}_i$ is the ground truth. All test sets were of size $N_{\text{test}} = 10^3$.

1) Network Stages: Fig. 4 shows algorithm NMSE (dB) versus the number of stages/iterations for ADMM-Net/ADMM for the noiseless case, SNR $= \infty$ (i.e., $\sigma^2 = 0$), with $\|\mathbf{w}\|_0 = 2$ and $\|\mathbf{b}\|_0 = 16$. Evidently an 11-stage network, achieving an average NMSE of -25.72 dB, surpasses ADMM, which on average converges to an NMSE of -25.51 dB in 156 iterations (see (86) for the convergence criterion).

2) Snr: Five networks were trained: four on data sets with deterministic SNRs in $\{0, 1, \ldots, 16\}$ and one on data with random SNRs drawn from uniform$(0, 15)$. Fig. 5 plots algorithm NMSE (dB) versus SNR, where in all cases $\|\mathbf{w}\|_0 = 2$, $\|\mathbf{b}\|_0 = 16$ (25% spectral overlap), and SIR $= 0$ dB. The points on red curve are the NMSEs of the networks trained on data with a deterministic SNR equal to the point's abscissa; the points on the blue curve are the NMSEs of the single network trained on the random SNR data. The cyan and green curves each are the performance of a particular network trained on deterministic data (cyan was trained with SNR $= 14$ dB, green with SNR $= 0$ dB) evaluated on test data sets of a different SNR.

3) Sir: Four networks were trained: three were trained with deterministic SIRs in $\{-5, 0, 5\}$, and one was trained on data with random SIRs drawn from uniform$(-5, 5)$. For evaluation, we used three test sets with respective SIRs -5, 0, and 5.

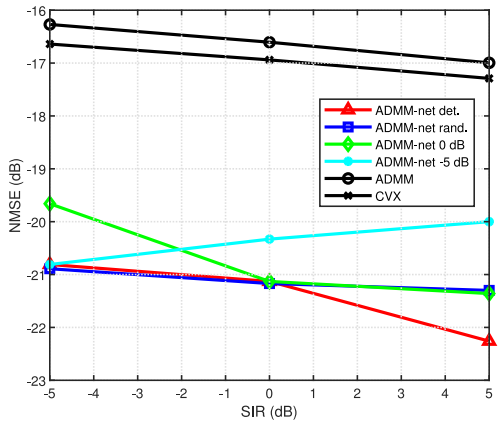


Fig. 6. ADMM-Net NMSE versus SIR. 2 scatterers, 25% spectrum overlap, SNR $= 15$.

Results are plotted in Fig. 6. The red, blue, cyan, and green curves are analagous to those in Fig. 5. Cyan was trained with SIR $= -5$ dB, green with SIR $= 0$ dB.

4) Sparsity: For radar sparsity, a total of six networks were trained. Five networks were trained on data sets with deterministic sparsity levels in $\{2, 3, 4, 5, 6\}$; within each of the five sets $\|\mathbf{w}\|_0$ was the same for all samples. One network was trained on data with random sparsity levels, where the sparsity of each sample was drawn from uniform$(2, 6)$. All six sets had $\|\mathbf{b}\|_0 = 16$, SNR $= 15$ dB, and SIR $= 0$ dB. Note that as the number of scatterers increases, the coefficients must decrease in magnitude in order to yield a given SNR. For evaluation, we fixed $\|\mathbf{b}\|_0 = 16$ and varied $\|\mathbf{w}\|_0$ from 2 to 6. Results are plotted in Fig. 7. Each point on the red curve corresponds to the test set NMSE of the particular network trained on the (deterministic) sparsity level equal to the point's abscissa. The blue curve plots the NMSE of the network trained on the data with uniformly distributed sparsity levels. The cyan and green curves each are the performance of a particular network trained on deterministic data (cyan was trained with $\|\mathbf{w}\|_0 = 6$, green with $\|\mathbf{w}\|_0 = 2$) evaluated on test data sets of a different SNR.

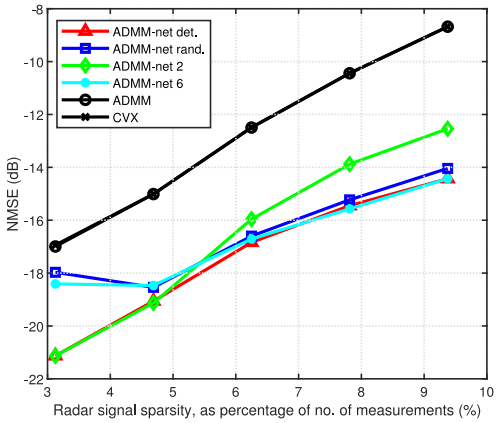


Fig. 7. NMSE versus sparsity level where $\|\mathbf{w}\|_0$, the number of scatterers, varies from 2 to 6. 25% spectrum overlap, $\mathrm{SNR} = 15$, $\mathrm{SIR} = 0$.

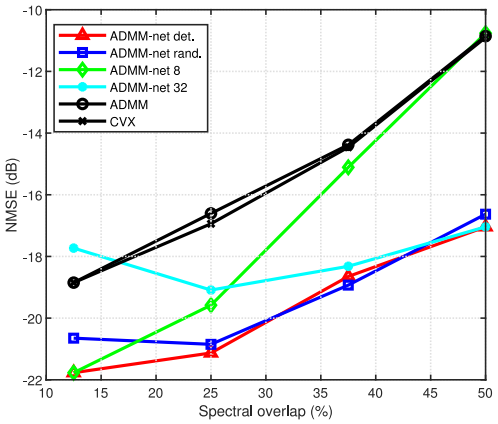


Fig. 8. NMSE versus sparsity level where $\|\mathbf{b}\|_0$ varies from 8 (12.5% overlap) to 32 (50% overlap). 2 scatterers, $\mathrm{SNR} = 15$, $\mathrm{SIR} = 0$.

Similarly, for interference sparsity, three networks were trained on data sets containing samples with a single deterministic sparsity level belonging to $\{12.5\%, 25\%, 37.5\%, 50\%\}$. The random sparsity level data was generated such that $\|\mathbf{b}\|_0 \sim \mathrm{uniform}(8, 32)$. The spectral location and number of interferers were assumed to be the same for each MIMO channel and were allowed to vary from sweep to sweep, but not within a sweep. All four sets had $\|\mathbf{w}\|_0 = 2$, $\mathrm{SNR} = 15$ dB, and $\mathrm{SIR} = 0$ dB. Note that as the number of interferers increases, their magnitudes must decrease in order to yield the same SIR. For evaluation, we fixed $\|\mathbf{w}\|_0 = 2$ and varied $\|\mathbf{b}\|_0$ from 8 to 32. Results are plotted in Fig. 8. The red, blue, cyan, and green curves are analagous to those in Fig. 7. Cyan was trained with $\|\mathbf{b}\|_0 = 32$, green with $\|\mathbf{b}\|_0 = 8$.

5) Recovered Image: To provide a qualitative account of the methods' outputs as well as demonstrate super-resolution capability, we simulate two scatterers in neighboring range grid points and the same velocity-angle grid point. Fig. 9 shows a range-velocity image slice—the slice which corresponds to the scatterers' angle location—for three methods: ADMM, ADMM single-penalty, and ADMM-Net. The respective NMSEs of the (total) recovered images are -12.0 dB, -4.7 dB, and -18.4 dB. In all scenarios, single-penalty ADMM yielded an NMSE of -5 dB or higher, except the scenario $\mathrm{SIR} = 5$ dB in which the error was -9 dB.

TABLE IV
RUN TIMES IN MILLISECONDS FOR THE SNR EXPERIMENTS, AVERAGED OVER 1000 SAMPLES

Method	5 dB	10 dB	15 dB
ADMM-Net (5 stages)	0.40	0.40	0.40
ADMM	22	26	29
CVX	510	550	600

6) Training Time: The 5-stage network training time (45 epochs, $N_{\mathrm{train}} = 4.5 \times 10^6$) was approximately 120 minutes on a 2-core server with a single Nvidia Tesla K80.

7) Run Time: Table IV lists the run times in the SNR experiment, averaged over the test set, for each algorithm, run in Matlab on a MacBook Pro with 8 GB of RAM and a 2.4 GHz Intel i5 processor. The ADMM run time is defined as the time until the convergence criterion

$$\frac{\mathrm{NMSE}(k+1) - \mathrm{NMSE}(k)}{\mathrm{NMSE}(k)} < 10^{-6} \tag{86}$$

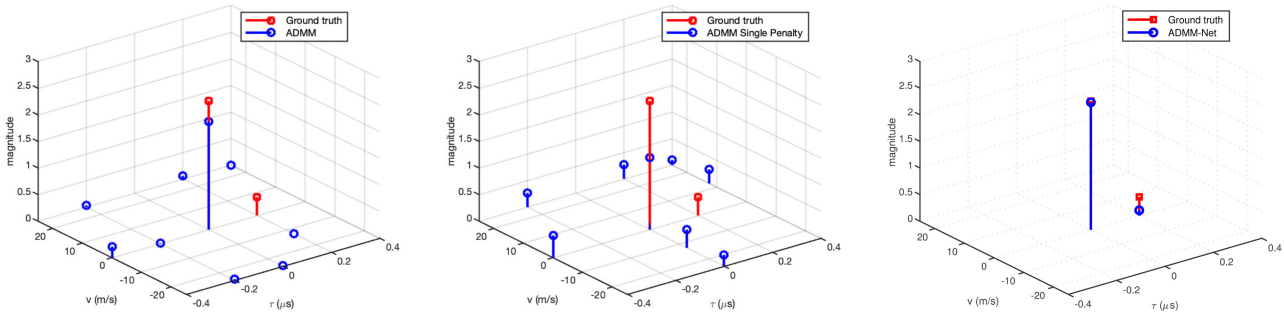
is satisfied, where $\mathrm{NMSE}(k)$ is the NMSE at iteration k. The 5-stage ADMM-Net has a constant run time, equal to the run time of 5 ADMM iterations.

G. Discussion

The deterministically trained ADMM-Nets, tested on data akin to their training sets, outperform ADMM and CVX by at least 2 dB in every scenario, and the performance gap widens to around 4 dB as SNR decreases below 15 dB, a region of significant practical interest. Moreover, the 5-stage ADMM-Net is between 50 and 80 times faster than ADMM, and between 1275 and 1500 times faster than CVX; see Table IV. Qualitatively, among the recovered images in Fig. 9 ADMM-Net's is the cleanest and most accurate. Also evident from Fig. 9 is the benefit of the two-penalty term optimization objective over the single-penalty objective.

As expected, deterministically trained networks are most accurate when used on test data with the same properties as their respective training sets, as opposed to data with properties that differ from their training sets. For example, in the SNR experiment, the network trained on high-SNR data (cyan) performs poorly on low-SNR test data, but for high-SNR test data achieves an error several dB below the low-SNR-trained network (green). Overall, the networks trained on random data (blue curves) are more robust to test set variation that the deterministically trained counterparts. Lower performance in certain regimes may trace to the fact that, since the training set size is the same as the others, fewer examples from each scenario are represented. However, the performance gap shrinks in more challenging environments, i.e. lower SNR, more spectrum overlap, etc.

While the network improves the original algorithm, it also inherits some of the algorithm's shortcomings. Chief among them



(a) (b) (c)

Fig. 9. Recovered range-velocity image slice for (a) ADMM, (b) ADMM w/ single penalty, and (c) ADMM-Net. $\mathrm{SNR} = 15$, $\mathrm{SIR} = 0$, $\|\mathbf{b}\|_0 = 16$, and $\|\mathbf{w}\|_0 = 2$. The two scatterers have magnitudes 2.4 and 0.3 and the same angular position.

perhaps is the off-grid assumption. Separate experiments, not shown, indicated that ADMM and CVX are significantly more robust in this regard. Unfolding off-grid recovery algorithms may be the subject of future work.

VI. Conclusion

We have shown that deep learning, in particular the deep unfolding framework, can be leveraged to significantly improve upon ADMM and CVX for communication interference removal in stepped-frequency radar imaging. The added cost is network training, which can be done in a matter of hours. Deep unfolding apparently makes fuller use of prior knowledge than standard iterative algorithms, adapting theoretically sound, generally applicable procedures to problem-specific data. Determining the extent to which training can compensate for model mismatch is left for future work.

How can we account for the performance ADMM-Net? Certain unfolded networks are designed to learn only algorithm hyperparameters and thus have a clear-cut "parameter-tuning" interpretation; others, such as our ADMM-Net, learn algorithm *operations*, and thus may elude such a straightforward account. In some cases the learned operations do coincide with those suggested by theory; a VAMP-inspired network, randomly initialized, learns a denoiser matched to the true signal priors [16]. ADMM-Net, on the other hand, is initialized as theoretically prescribed, whence it then deviates through training. Further insight might be found in identifying redundancies among the learnable parameters. For example, in LISTA one learnable matrix converged to a final state determined by another, thus allowing a reduction in the number of parameters without altering performance [44].

Acknowledgment

The authors would like to thank the anonymous referees and the Associate Editor for useful comments which have helped to improve the presentation of this paper.

References

[1] H. Griffiths *et al.*, "Radar spectrum engineering and management: Technical and regulatory issues," *Proc. IEEE*, vol. 103, no. 1, pp. 85–102, Jan. 2015.
[2] G. M. Jacyna, B. Fell, and D. McLemore, "A high-level overview of fundamental limits studies for the darpa ssparc program," in *Proc. IEEE Radar Conf.*, 2016, pp. 1–6.
[3] L. Zheng, M. Lops, Y. C. Eldar, and X. Wang, "Radar and communication coexistence: An overview: A review of recent methods," *IEEE Signal Process. Mag.*, vol. 36, no. 5, pp. 85–99, Sep. 2019.
[4] L. Zheng, M. Lops, X. Wang, and E. Grossi, "Joint design of overlaid communication systems and pulsed radars," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 139–154, Jan. 2018.
[5] S. Sodagari, A. Khawar, T. C. Clancy, and R. McGwier, "A projection based approach for radar and telecommunication systems coexistence," in *Proc. Global Commun. Conf. (GLOBECOM)*, 2012, pp. 5010–5014.
[6] B. Li and A. P. Petropulu, "Joint transmit designs for coexistence of MIMO wireless communications and sparse sensing radars in clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 6, pp. 2846–2864, Dec. 2017.
[7] B. H. Kirk, R. M. Narayanan, K. A. Gallagher, A. F. Martone, and K. D. Sherbondy, "Avoidance of time-varying radio frequency interference with software-defined cognitive radar," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 3, pp. 1090–1107, Jun. 2019.
[8] H. Deng and B. Himed, "Interference mitigation processing for spectrum-sharing between radar and wireless communications systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 3, pp. 1911–1919, Jul. 2013.
[9] Y. Li, X. Wang, and Z. Ding, "Multi-target position and velocity estimation using OFDM communication signals," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1160–1174, Feb. 2020.
[10] Y. Li, L. Zheng, M. Lops, and X. Wang, "Interference removal for radar/communication co-existence: The random scattering case," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4831–4845, Oct. 2019.
[11] M. Tao, J. Su, Y. Huang, and L. Wang, "Interference mitigation for synthetic aperture radar based on deep residual network," *Remote Sens.*, vol. 11, no. 14, 2019, Art. no. 1654, [Online]. Available: https://doi.org/10.3390/rs11141654
[12] A. Mousavi and R. G. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 2272–2276.
[13] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, Jun. 2010.
[14] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," MERL, Cambridge, MA, USA, Rep. TR2014-117, Aug. 2014.
[15] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[16] M. Borgerding, P. Schniter, and S. Rangan, "Amp-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.
[17] Y. Li, X. Wang, and Z. Ding, "Multi-dimensional spectral super-resolution with prior knowledge with application to high mobility channel estimation," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2836–2852, Dec. 2020.
[18] O. Solomon *et al.*, "Deep unfolded robust PCA with application to clutter suppression in ultrasound," *IEEE Trans. Med. Imag.*, vol. 39, no. 4, pp. 1051–1063, Apr. 2020.
[19] Y. Yang, J. Sun, H. Li, and Z. Xu, "Admm-csnet: A deep learning approach for image compressive sensing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 521–538, Mar. 2020.
[20] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep Admm-Net for Compressive Sensing Mri," in *Adv. Neural Inf. Process. Syst.* vol. 29, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 10–18.
[21] C. Hu, Z. Li, L. Wang, J. Guo, and O. Loffeld, "Inverse synthetic aperture radar imaging using a deep admm network," in *Proc. 20th Int. Radar Symp.*, 2019, pp. 1–9.
[22] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
[23] Y. Liang, K. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3386–3407, Sep. 2011.
[24] T. Counts, A. C. Gurbuz, W. R. Scott, J. H. McClellan, and K. Kim, "Multistatic ground-penetrating radar experiments," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 8, pp. 2544–2553, Aug. 2007.
[25] 3rd Generation Partnership Project, "Lte; Evolved Universal Terrestrial Radio Access (e-Utra); Physical Channels and Modulation," 3GPP TS 36.211 version 14.2.0 Release 14, 2017.
[26] Y. Kim *et al.*, "New radio (NR) and its evolution toward 5G-advanced," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 2–7, Jun. 2019.
[27] M. Shafi *et al.*, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, Jun. 2017.
[28] L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin, "Sparsity and compressed sensing in radar imaging," *Proc. IEEE*, vol. 98, no. 6, pp. 1006–1020, Jun. 2010.
[29] T. Strohmer and B. Friedlander, "Analysis of sparse MIMO radar," *Appl. Comput. Harmon. Anal.*, vol. 37, no. 3, pp. 361–388, 2014.
[30] Y. Yu, A. P. Petropulu, and H. V. Poor, "CSSF MIMO RADAR: Compressive-sensing and step-frequency based MIMO radar," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 2, pp. 1490–1504, Apr. 2012.
[31] M. A. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2275–2284, Jun. 2009.
[32] L. Zheng, M. Lops, X. Wang, and E. Grossi, "Joint design of overlaid communication systems and pulsed radars," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 139–154, Jan. 2018.
[33] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[34] J. Eckstein, "Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," RUTCOR, Rutgers Univ., Piscataway, NJ, Res. Rep. RRR 32-2012, Dec. 2012.
[35] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.
[36] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Rice Univ., CAAM Tech. Report: TR12-14, 2012.
[37] A. Teixeira, E. Ghadimi, I. Shames, *et al.*, "Optimal scaling of the admm algorithm for distributed quadratic programming," in *Proc. 52nd IEEE Conf. Decis. Control*, 2013, pp. 6868–6873. [Online]. Available: https://arxiv.org/abs/1303.6680v2.
[38] B. He, H. Yang, and S. Wang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *J. Optim. Theory Appl.*, vol. 106, no. 2, pp. 337–356, 2000.
[39] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, pp. 407–499, 2004.
[40] C. U. Ungan. Candan, and T.Ciloglu, "A space-time coded mills cross mimo architecture to improve doa estimation and its performance evaluation by field experiments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 1807–1818, Jun. 2020.
[41] C. DAndrea, S. Buzzi, and M. Lops, "Communications and radar coexistence in the massive mimo regime: Uplink analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 19–33, Jan. 2020.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/ abs/1412.6980.
[43] A. Ramdas and R. J. Tibshirani, "Fast and flexible admm algorithms for trend filtering," *J. Comput. Graphical Statist.*, vol. 25, no. 3, pp. 839–858, 2016.
[44] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ista and its practical weights and thresholds," in *Proc. 32nd Conf. Neural Inf. Process. Syst.*, 2018. [Online]. Available:https://proceedings.neurips.cc/paper/2018/file/cf8c9be2a4508a24ae92c9d3d379131d-Paper.pdf

Jeremy Johnston received the B.S. degree in electrical engineering from the University of Florida, Gainesville, FL, USA, in 2018. He is currently working toward the Ph.D. degree in electrical engineering with Columbia University, New York City, NY, USA.

Yinchuan Li was born in Bozhou, China, in 1994. He received the B.S. and Ph.D. degrees in electronic engineering from the Beijing Institute of Technology, Beijing, China, in 2015 and 2020, respectively. From November 2017 to November 2019, he was a Visiting Scholar with the Department of Electrical Engineering, Columbia University, New York City, NY, USA. His research interests include machine learning, deep learning, sparse signal processing, radar systems, and wireless communications. He won the Excellent Paper Award at the 2019 IEEE International Conference on Signal, Information and Data Processing.

Marco Lops (Fellow, IEEE) received the Laurea and Ph.D. degrees from Federico II University, Naples Italy. He is currently a Professor with the Department of Electrical and Information Technology, University of Naples Federico II. He was an Assistant and Associate Professor with University of Naples Federico II. In March 2000, he moved to University of Cassino and Southern Latium, Cassino, Italy, as a Full Professor, and in 2018, he returned to Federico II. From 2009 to 2012, he was also with ENSEEIHT, Toulouse, France, first as Full Professor (on leave of absence from Italy) and then as visiting Professor. In fall 2008, he was a visiting Professor with the University of Minnesota, Minneapolis, MN, USA, and in spring 2009, with Columbia University, New York City, NY, USA. Previously, he had also held visiting positions with the University of Connecticut, Mansfield, CT, USA, with Rice University, Houston, TX, USA, and with Princeton University, Princeton, NJ, USA. He has authored or coauthored more than 90 scientific papers published on refereed journals. His research interests include detection and estimation, with emphasis on communications and radar signal processing. From 2009 to 2015, he served two terms in the Sensor Array and Multichannel Signal Processing Technical Committee. He was an Associate Editor for the JOURNAL OF COMMUNICATIONS AND NETWORKS, IEEE TRANSACTIONS ON INFORMATION THEORY (Area: Detection and Estimation), IEEE SIGNAL PROCESSING LETTERS, IEEE TRANSACTIONS ON SIGNAL PROCESSING (two terms), and is currently the Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY. He was the co-recipient (with Ezio Biglieri) of the 2014 Best Paper Award from the JOURNAL OF COMMUNICATIONS AND NETWORKS. During 2018–2020, he was selected as a Distinguished Lecturer for the Signal Processing Society.

Xiaodong Wang (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA. He is currently a Professor of electrical engineering with Columbia University, New York, NY, USA. He has authored or coauthored book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception* (Prentice Hall, 2003). His current research interests include wireless communications, statistical signal processing, genomic signal processing, general areas of computing, and signal processing and communications, and has published extensively in these areas.

Dr. Wang was the recipient of the 1999 NSF CAREER Award, the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award, and the 2011 IEEE Communication Society Award for Outstanding Paper on New Communication Topics. He was an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the IEEE TRANSACTIONS ON INFORMATION THEORY. He is listed as an ISI Highly cited Author.