electronics

MDPI

Communication

A Bayesian Deep Unfolded Network for the Off-Grid Direction-of-Arrival Estimation via a Minimum Hole Array

Ninghui Li [1], Xiaokuan Zhang [2], Fan Lv [1], Binfeng Zong [2,*] and Weike Feng [2]

[1] Graduate School, Air Force Engineering University, Xi'an 710051, China; li6667778882022@163.com (N.L.); lf995407681@163.com (F.L.)
[2] Air Defence and Antimissile School, Air Force Engineering University, Xi'an 710051, China; ezxk@sina.com (X.Z.); fengweike007@163.com (W.F.)
* Correspondence: zongbinfeng@163.com

Abstract: As an important research focus in radar detection and localization, direction-of-arrival (DOA) estimation has advanced significantly owing to deep learning techniques with powerful fitting and classifying abilities in recent years. However, deep learning inevitably requires substantial data to ensure learning and generalization abilities and lacks reasonable interpretability. Recently, a deep unfolding technique has attracted widespread concern due to the more explainable perspective and weaker data dependency. More importantly, it has been proven that deep unfolding enables convergence acceleration when applied to iterative algorithms. On this basis, we rigorously deduce an iterative sparse Bayesian learning (SBL) algorithm and construct a Bayesian deep unfolded network in a one-to-one correspondence. Moreover, the common but intractable off-grid errors, caused by grid mismatch, are directly considered in the signal model and computed in the iterative process. In addition, minimum hole array, little considered in deep unfolding, is adopted to further improve estimation performance owing to the maximized array degrees of freedom (DOFs). Extensive simulation results are presented to illustrate the superiority of the proposed method beyond other state-of-the-art methods.

Keywords: direction-of-arrival estimation; deep unfolding; sparse Bayesian learning; minimum hole array

check for updates

Citation: Li, N.; Zhang, X.; Lv, F.; Zong, B.; Feng, W. A Bayesian Deep Unfolded Network for the Off-Grid Direction-of-Arrival Estimation via a Minimum Hole Array. *Electronics* **2024**, *13*, 2139. https://doi.org/10.3390/electronics13112139

Academic Editor: Dimitra I. Kaklamani

Received: 19 April 2024
Revised: 11 May 2024
Accepted: 16 May 2024
Published: 30 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

As an active research focus, direction-of-arrival (DOA) estimation has advanced significantly over the past four decades owing to the adopted subspace, compressed sensing, and deep learning that led to technology leaps [1–3]. In particular, deep learning is highly favored by researchers due to the powerful fitting and classifying abilities that play important roles in image and signal processing [4,5]. However, as a data-driven technique, deep learning requires abundant sampling data for training and empirically complicated network architectures for nonlinear fitting [6], while numerous data are not always easy to collect, and too deep and elaborate networks may lead to gradient explosion or vanishing. Additionally, the lack of explainable theory simultaneously leads to the limitation of application and development [7]. Hence, the pursuit of more explainable learning networks is becoming a research focus. Recently, deep unfolding has been proven to achieve them [8].

In contrast to empirical deep learning, deep unfolding, first developed for sparse coding [9], is rigorously based on iterative algorithms [10]. Specifically, deep unfolding enhances interpretability by constructing a network corresponding to the specific iterative steps of theoretically believable algorithms. Therefore, deep unfolding essentially belongs to a compromise between data-driven and model-based methods. On the one hand, deep unfolding is more explainable since the effect of every layer and each neuron explicitly

originated from iterative algorithms. On the other hand, deep unfolding inherits the advantages of deep learning, i.e., excellent robustness to modeling errors and strong adaptability to complicated cases that model-based algorithms tend to be at a loss. More importantly, deep unfolding is able to improve convergence rates [11,12], though only LSTA-net [13] and ADMM-net [14] have been proven theoretically. In fact, deep unfolding still empirically shows desirable performance. In the DOA estimation field, a deep unfolded FOCUSS network is developed in [15] for on-grid and off-grid DOA estimation, resulting in low complexity and high estimation accuracy. Additionally, a deep Alternating Projection Network for gridless DOA estimation is proposed in [16], achieving super-resolution estimation. Moreover, ISTA is also adopted for deep unfolding [17,18] so that the proposed method can handle anti-jamming, array imperfections, and noise. In the channel estimation field, there are many works using deep unfolding to improve estimation performance. ISTA-based deep unfolded networks are developed for high running efficiency and excellent estimation performance [19,20], while the same operation is adopted for ADMM and AMP. Moreover, RIS and SSCA algorithms are introduced for constructing deep unfolded networks [21] so that the computation complexity and signaling overhead are decreased when compared with the original algorithms.

Overall, many works have adopted iterative algorithms converging slowly to improve convergence efficiency for DOA or channel estimation. Nonetheless, the developed deep unfolded networks are limited to their original algorithms. In other words, the original iterative algorithms dominate the main performance of the deep unfolded networks. Thus, it seems to be the best selection to apply excellent iterative algorithms to be unfolded. As we know, sparse Bayesian learning (SBL) [22,23] is an excellent probabilistic algorithm for linear regression. Unlike the above deterministic algorithms with explicit regularization constraints, SBL leverages prior distributions assigned to latent variables for regularization constraint, so the sparsity performance is guaranteed. Further, the estimation performance is prominent. Moreover, the convergence performance of SBL is robust and fast since each iteration enables an increase in the posterior probability. Based on this, many works have used SBL for deep unfolding. Among them, SBL-based deep unfolded networks are mainly designed by replacing the entire iterative process with independent network layers. To be specific, SBL is directly integrated into deep networks for DOA estimation in [24,25] and can be unfolded for channel estimation when combined with DDGP or AMP [26,27].

However, the present SBL-based deep unfolded networks are coarse and vague in theory due to the direct deep unfolding. Therefore, in this paper, we deform the iterative formulas of SBL reasonably so that the proposed Bayesian deep unfolded network is explicit. We fit the linear/nonlinear operations in the iterative formulas with linear/nonlinear layers so that the theoretical interpretability is enhanced.

Moreover, to compensate for intractable off-grid errors appearing in DOA estimation problems, we take the errors into the signal model and re-derive corresponding real-valued SBL iterative formulas. Further, the off-grid error compensations are directly yielded in the proposed Bayesian deep unfolded networks, while [24,25] just design an additional network for off-grid DOA estimation. Thus, the proposed unfolded network is simpler and more useful.

In addition, to improve estimation accuracy, the minimum hole array (MHA) [28], little considered in deep unfolding, is adopted to maximize array degrees of freedom (DOFs). Here, the principle and rationale of different co-arrays for large DOFs are as follows. For a steer vector $\mathbf{a}(\theta) = [\exp(-j2\pi\xi_1 \sin(\theta)/\lambda), \ldots, \exp(-j2\pi\xi_N \sin(\theta)/\lambda)]^T$ composed of N sensors, if vectorization is adopted for the covariance of received signals, the Kronecker product of steer vectors is yielded, i.e., $\mathbf{a}(\theta)^* \otimes \mathbf{a}(\theta) = \left\{\exp(-j2\pi(\xi_i - \xi_j)\sin(\theta)/\lambda)\right\}_{i,j=1,\ldots,N}$. Obviously, the difference position set $\left\{\xi_i - \xi_j\right\}_{i,j=1,\ldots,N}$ is produced. For a uniformed linear array (ULA) composed of N sensors, its position set is $\{0, 1, \ldots, N-1\}$, and its difference position set is $\{-(N-1), -(N-2), \ldots, 0, 1, \ldots, N-1\}$ and is composed of $2N-1$ different elements. For MHA composed of N sensors, its position set satisfies some constraint [28] so that its difference position set has $N(N-1)$ different elements, resulting in

an $N(N-1)$ DOF, which is the maximal DOF in a sparse array [29] Figure 1 shows the case at $N = 5$ sensors.

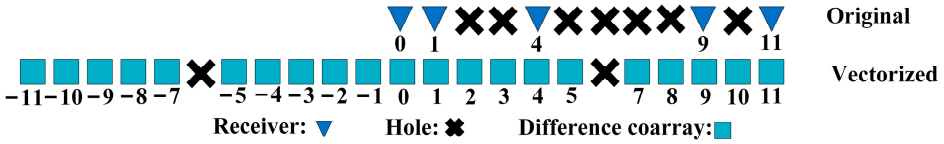


Figure 1. Minimum hole array of five sensor receivers.

Overall, the main contributions and novelties are as follows:

- Based on the general signal model, real-valued SBL iterative formulas are rigorously derived and adjusted, and the corresponding Bayesian deep unfolded network is constructed according to each operation in the iterative formulas rather than an entire formula;
- Off-grid errors are directly modeled and estimated by the iterative formulas, and the corresponding network architecture is considered;
- MHA is adopted to maximize the DOF so that the estimation performance is improved.

The rest of this paper is organized as follows. In Section 2, the signal model is presented, and the DOA problem is clarified and transformed. In Section 3, SBL iterative formulas are derived, and the corresponding Bayesian deep unfolded network is constructed. Simulation results are shown in Section 4, and conclusions are drawn in Section 5.

To simplify notations, we, respectively, show functions to obtain the real/imaginary part, trace, p norm, expectation, and transpose by $\Re(\cdot)/\Im(\cdot)$, $\mathrm{tr}(\cdot)$, $\|\cdot\|_p$, and $E(\cdot)$, $(\cdot)^T$. We, respectively, denote real/complex number sets, conditional/marginal probability density distribution, the identity matrix, the zero vector or matrix, constant, Moore–Penrose inverse, "independent and identically distributed", Kronecker product, and Khatri–Rao product by $\mathbb{R}/\mathbb{C}$, $p(a|b)/p(a;b)$, $\mathbf{I}$, $\mathbf{0}$, $const$, $(\cdot)^+$, "i.i.d.", $\otimes$, and $\circ$.

2. Problem Formulation

In radar detection and localization and navigation fields, the target direction is vital information in practice. Thus, how to locate signal sources well has been a research focus for more than 40 years. To be specific, array antenna receivers can receive different signals from various directions. Taking far-field radar detection as an example, adjacent antenna sensors have the same phase difference due to the plane electromagnetic wave. When the entire sensor array satisfies the configuration of MHA, a steer vector can be inferred as $\mathbf{a}(\theta) = [\exp(-j2\pi\xi_1 \sin(\theta)/\lambda), \exp(-j2\pi\xi_2 \sin(\theta)/\lambda), \ldots, \exp(-j2\pi\xi_N \sin(\theta)/\lambda)]^T \in \mathbb{C}^{N\times 1}$, where θ is the direction of the source signal, ξ_n is the $n-th$ element of the MHA position set $\mathbb{D}_{MHA} = \{\xi_n | \xi_n = \eta_n d, n = 1, \ldots, N\}$, λ is the wavelength, d is the unit distance, and N is the number of sensors. For all the K sources from different directions $\boldsymbol{\theta} = \{\theta_{k=1}^K\}$, the corresponding steer matrix is yielded as $\mathbf{A} = [\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_K)] \in \mathbb{C}^{N\times K}$. $\mathbf{A}$ is vital in DOA estimation because it implies the directions of all the sources based on the entire sensor array composed of N sensors. Radio frequency signals are received by an antenna array, whose sensors transfer individually received signals to independent channels, and down conversion is conducted to produce intermediate-frequency signals, i.e., baseband signals. Later, using prior signals to execute matched filtering at time t, the receiver can gain the complex reflection factor of all the K sources, i.e., $\mathbf{s}(t) = [s_1(t), \ldots, s_k(t), \ldots, s_K(t)]^T \in \mathbb{C}^{K\times 1}$ and $\forall k = 1, \ldots, K$, where $s_k(t)$ is the product of the complex reflection coefficient and Doppler shift. Usually, the radar adopts pulse accumulation to enhance signal processing ability when the echo pulses are in the same coherent processing interval (CPI), in which little fluctuation happens between different originally received signals and processed signals. In other words, sources are motionless, and the processing signals are nearly consistent

during a CPI. Overall, the entire process is briefly shown in Figure 2. The received signal at time t is modeled as

$$\mathbf{y}(t) = \sum_{k=1}^{K} \mathbf{a}(\theta_k) s_k(t) + \mathbf{n}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t) \tag{1}$$

where $\mathbf{y}(t) \in \mathbb{C}^{N\times 1}$ is the received signal vector, $\mathbf{a}(\theta_k) = [1, \ldots, \exp(-j2\pi\xi_N \sin\theta_k/\lambda)]^T \in \mathbb{C}^{N\times 1}$ is the steer vector of the $k-th$ source signal, $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_K)] \in \mathbb{C}^{N\times K}$ is the array manifold matrix, and $\mathbf{n}(t) \in \mathbb{C}^{N\times 1}$ is the vector representing the complex zero-mean Gaussian white noise with variance σ^2. All the complex reflection factors of source signals $s_k(t)$ are assumed to be additive complex i.i.d. Gaussian distributed with zero mean and variance σ_k^2, so the covariance of the received data is expressed as

$$\mathbf{R}_y = E[\mathbf{y}(t)\mathbf{y}(t)^H] = \mathbf{A}(\boldsymbol{\theta})\mathbf{R}_s\mathbf{A}(\boldsymbol{\theta})^H + \sigma^2\mathbf{I}_N \tag{2}$$

where $\mathbf{R}_s = E[\mathbf{s}(t)\mathbf{s}(t)^H] = \mathrm{diag}[\sigma_1^2, \ldots, \sigma_K^2]$. Vectorizing both sides of (2) yields

$$\mathbf{r} = \overline{\mathbf{A}}(\boldsymbol{\theta})\boldsymbol{\sigma} + \mathrm{vec}(\sigma^2\mathbf{I}_N) \tag{3}$$

where $\mathbf{r} = \mathrm{vec}(\mathbf{R}_y) \in \mathbb{C}^{N^2\times 1}$, $\overline{\mathbf{A}}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1)^* \otimes \mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_K)^* \otimes \mathbf{a}(\theta_K)] \in \mathbb{C}^{N^2\times K}$, and $\boldsymbol{\sigma} = [\sigma_1^2, \ldots, \sigma_K^2]^T \in \mathbb{R}^{K\times 1}$. In practice, the covariance is approximated by

$$\hat{\mathbf{R}}_y = \frac{1}{T}\sum_{t=1}^{T} \mathbf{y}(t)\mathbf{y}(t)^H \tag{4}$$

where T is the number of snapshots. Compared with (3), vectorizing (4) results in an additive approximation error term $\Delta\mathbf{r}$ [30], i.e.,

$$\hat{\mathbf{r}} = \overline{\mathbf{A}}(\boldsymbol{\theta})\boldsymbol{\sigma} + \mathrm{vec}(\sigma^2\mathbf{I}_N) + \Delta\mathbf{r} \tag{5}$$

where $\hat{\mathbf{r}} = \mathrm{vec}(\hat{\mathbf{R}}_y) \in \mathbb{C}^{N^2\times 1}$ and $\Delta\mathbf{r} \in \mathbb{C}^{N^2\times 1}$ follow an asymptotic complex Gaussian distribution, i.e.,

$$p(\Delta\mathbf{r}) = \mathcal{CN}(\Delta\mathbf{r}|\mathbf{0}_{N^2\times 1}, \mathbf{R}_y^H \otimes \mathbf{R}_y/T) \tag{6}$$

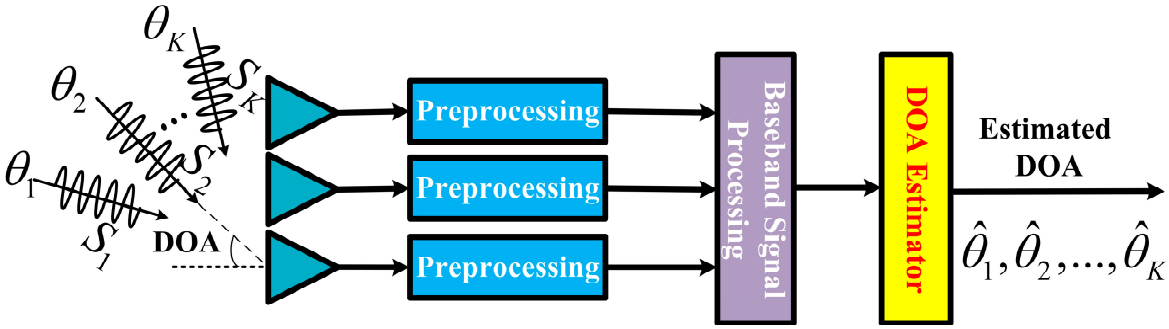


Figure 2. Flowchart of radar signal processing for DOA estimation.

In practice, $\mathbf{R}_y^H \otimes \mathbf{R}_y/T$ is replaced by $\mathbf{R}_y^H \otimes \mathbf{R}_y/T$ due to the unavailable $\mathbf{R}_y$. To avoid the later influence of σ^2, we directly eliminate the noise term $\mathrm{vec}(\sigma^2\mathbf{I}_N)$ by a selection matrix $\mathbf{J} \in \mathbb{R}^{N(N-1)\times N^2}$ [31]. Multiplying both sides of (5) yields

$$\bar{\mathbf{r}} = \widetilde{\mathbf{A}}(\boldsymbol{\theta})\mathbf{x} + \Delta\widetilde{\mathbf{r}} \tag{7}$$

where $\bar{\mathbf{r}} = \mathbf{J}\hat{\mathbf{r}} \in \mathbb{C}^{N(N-1)\times 1}$, $\widetilde{\mathbf{A}}(\theta) = \mathbf{J}\overline{\mathbf{A}}(\theta) \in \mathbb{C}^{N(N-1)\times K}$, $\Delta\widetilde{\mathbf{r}} = \mathbf{J}\Delta\mathbf{r} \in \mathbb{C}^{N(N-1)\times 1}$, and $\mathbf{J} = [\mathbf{J}_1^T, \mathbf{J}_2^T, \ldots, \mathbf{J}_i^T, \ldots \mathbf{J}_{N-1}^T]^T$ with the $i-th$ entry $\mathbf{J}_i = [\mathbf{0}_{N\times[(i-1)N+i]}, \mathbf{I}_N, \mathbf{0}_{N\times(N^2-iN-i)}] \in \mathbb{R}^{N\times N^2}$. After noise removal, $\Delta\widetilde{\mathbf{r}}$ follows $\mathcal{CN}(\mathbf{0}_{N(N-1)\times 1}, \mathbf{J}(\mathbf{R}_y^H \otimes \mathbf{R}_y)\mathbf{J}^T/T)$. Normalizing (7) so that the new approximation error follows a complex standard norm distribution and multiplying both sides of (7) by a normalization matrix $\mathbf{W}^{-1/2}$ yields

$$\widehat{\mathbf{r}} = \widehat{\mathbf{A}}(\theta)\boldsymbol{\sigma} + \Delta\widehat{\mathbf{r}} \tag{8}$$

where $\widehat{\mathbf{r}} = \mathbf{W}^{-1/2}\bar{\mathbf{r}} \in \mathbb{C}^{N(N-1)\times 1}$, $\widehat{\mathbf{A}}(\theta) = \mathbf{W}^{-1/2}\widetilde{\mathbf{A}}(\theta) \in \mathbb{C}^{N(N-1)\times K}$, $\Delta\widehat{\mathbf{r}} = \mathbf{W}^{-1/2}\Delta\widetilde{\mathbf{r}} \in \mathbb{C}^{N(N-1)\times 1}$, and $\mathbf{W} = \frac{1}{T}\mathbf{J}(\hat{\mathbf{R}}_y^H \otimes \hat{\mathbf{R}}_y)\mathbf{J} \in \mathbb{C}^{N(N-1)\times N(N-1)}$. After normalization, $\Delta\widehat{\mathbf{r}}$ follows $\mathcal{CN}(\mathbf{0}_{N(N-1)\times 1}, \mathbf{I}_{N(N-1)})$.

Let $\vartheta = \{\vartheta_l\}_{l=1}^L$ be an L-grid angular set that uniformly covers spatial range $[-\pi/2, \pi/2]$. Specially, if true sources lie on the grid, (8) can be rewritten as

$$\widehat{\mathbf{r}} = \breve{\mathbf{A}}(\vartheta)\mathbf{x} + \Delta\widehat{\mathbf{r}} \tag{9}$$

where $\breve{\mathbf{A}}(\vartheta) = \mathbf{W}^{-1/2}\mathbf{J}\overline{\mathbf{A}}(\vartheta) \mathbb{C}^{N(N-1)\times L}$ and $\overline{\mathbf{A}}(\vartheta) = [\mathbf{a}(\vartheta_1)^* \otimes \mathbf{a}(\vartheta_1), \ldots, \mathbf{a}(\vartheta_L)^* \otimes \mathbf{a}(\vartheta_L)] \in \mathbb{C}^{N^2\times L}$. $\mathbf{x} \in \mathbb{R}^{L\times 1}$ is the sparse extension of $\boldsymbol{\sigma}$ with the $l-th$ element, satisfying

$$x_l = \begin{cases} \sigma_k^2, \ if \ \vartheta_l = \theta_k \\ 0, \ elsewhere \end{cases} \tag{10}$$

Obviously, $\mathbf{x}$ is sparse with many zero elements since L is usually large enough for high estimation accuracy. Despite large L being adopted, there still exists errors if true sources are off-grid, i.e., $\vartheta_l \neq \theta_k$ and $\forall l = 1, \ldots, L$. Considering that, we approximate the $k-th$ true source DOA θ_k by the Taylor expansion at the corresponding nearest gird ϑ_{l_k}, i.e.,

$$\mathbf{a}(\theta_k) \approx \mathbf{a}(\vartheta_{l_k}) + \mathbf{b}(\vartheta_{l_k})(\theta_k - \vartheta_{l_k}) \tag{11}$$

where $\mathbf{b}(\vartheta_{l_k}) = [\mathbf{a}(\vartheta_{l_k})]'$ is the derivative of $\mathbf{a}(\vartheta_{l_k})$ with respect to ϑ_{l_k}. For simplicity, the $i-th$ element of $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_L]^T$ is introduced as

$$\delta_l = \begin{cases} \theta_k - \vartheta_{l_k}, \ if \ l = l_k \\ 0, \quad elsewhere \end{cases} \tag{12}$$

where δ_l is limited in $[-r/2, r/2]$ and r is the gird interval. If δ_l exceeds the range, $\pm r/2$ is assigned to δ_l. Using (11) and (12) and ignoring the quadratic term with respect to δ_l, the following equation approximately holds:

$$\begin{aligned} &\mathbf{a}(\theta_k)^* \otimes \mathbf{a}(\theta_k) = [\mathbf{a}(\vartheta_{l_k}) + \mathbf{b}(\vartheta_{l_k})\delta_l]^*[\mathbf{a}(\vartheta_{l_k}) + \mathbf{b}(\vartheta_{l_k})\delta_l] \\ &\approx \mathbf{a}(\vartheta_{l_k})^* \otimes \mathbf{a}(\vartheta_{l_k}) + [\mathbf{b}(\vartheta_{l_k})^* \otimes \mathbf{a}(\vartheta_{l_k}) + \mathbf{a}(\vartheta_{l_k})^* \otimes \mathbf{b}(\vartheta_{l_k})]\delta_l \end{aligned} \tag{13}$$

Let $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_L]^T \in \mathbb{R}^{L\times 1}$ and $\boldsymbol{\Delta} = \mathrm{diag}(\boldsymbol{\delta}) \in \mathbb{R}^{L\times L}$. For all the K sources, (13) is extended to

$$\overline{\mathbf{A}}(\theta) \approx \boldsymbol{\Phi}(\vartheta) = \overline{\mathbf{A}}(\vartheta) + [\mathbf{A}(\vartheta)^* \circ \mathbf{B}(\vartheta) + \mathbf{B}(\vartheta)^* \circ \mathbf{A}(\vartheta)]\boldsymbol{\Delta} \tag{14}$$

where $\boldsymbol{\Phi}(\vartheta) \in \mathbb{C}^{N^2\times L}$, $\mathbf{A}(\vartheta) = [\mathbf{a}(\vartheta_1), \ldots, \mathbf{a}(\vartheta_L)] \in \mathbb{C}^{N\times L}$, $\mathbf{B}(\vartheta) = [\mathbf{b}(\vartheta_1), \ldots, \mathbf{b}(\vartheta_L)] \in \mathbb{C}^{N\times L}$. Therefore, (9) is rewritten as

$$\widehat{\mathbf{r}} = \boldsymbol{\psi}(\vartheta)\mathbf{x} + \Delta\widehat{\mathbf{r}} \tag{15}$$

where $\boldsymbol{\psi}(\vartheta) = \mathbf{W}^{-1/2}\mathbf{J}\boldsymbol{\Phi}(\vartheta) \in \mathbb{C}^{N(N-1)\times L}$. As neural networks of deep unfolding are originally designed for real-valued data, we transform (15) into a real-valued equation as

$$\breve{\mathbf{r}} = \breve{\boldsymbol{\psi}}(\vartheta)\mathbf{x} + \Delta\breve{\mathbf{r}} \tag{16}$$

where $\breve{\mathbf{r}} = \begin{bmatrix} \Re(\widehat{\mathbf{r}}) \\ \Im(\widehat{\mathbf{r}}) \end{bmatrix} \in \mathbb{R}^{2N(N-1)\times 1}$, $\mathbf{D} = \begin{bmatrix} \Re(\mathbf{W}^{-1/2}\mathbf{J}[\mathbf{A}(\vartheta)^* \circ \mathbf{B}(\vartheta) + \mathbf{B}(\vartheta)^* \circ \mathbf{A}(\vartheta)]) \\ \Im(\mathbf{W}^{-1/2}\mathbf{J}[\mathbf{A}(\vartheta)^* \circ \mathbf{B}(\vartheta) + \mathbf{B}(\vartheta)^* \circ \mathbf{A}(\vartheta)]) \end{bmatrix} \in \mathbb{R}^{2N(N-1)\times L}$, $\mathbf{C} = \begin{bmatrix} \Re(\mathbf{W}^{-1/2}\mathbf{J}\overline{\mathbf{A}}(\vartheta)) \\ \Im(\mathbf{W}^{-1/2}\mathbf{J}\overline{\mathbf{A}}(\vartheta)) \end{bmatrix} \in \mathbb{R}^{2N(N-1)\times L}$, $\breve{\boldsymbol{\psi}}(\vartheta) = \mathbf{C} + \mathbf{D}\boldsymbol{\Delta}$, and $\Delta\breve{\mathbf{r}} = \begin{bmatrix} \Re(\Delta\widehat{\mathbf{r}}) \\ \Im(\Delta\widehat{\mathbf{r}}) \end{bmatrix} \in \mathbb{R}^{2N(N-1)\times 1}$.

Overall, the DOA estimation problem is equivalently cast as a sparse recovery problem for solving real-valued $\mathbf{x}$ and $\boldsymbol{\delta}$ with real and imaginary parts of known complex quantities $\widehat{\mathbf{r}}$, $\mathbf{A}(\vartheta)$, $\mathbf{B}(\vartheta)$, and $\overline{\mathbf{A}}(\vartheta)$.

3. Proposed Method

In this section, the DOA estimation is realized in accordance with the real-valued framework in (16) by Bayesian inference based on the posterior. The proposed method is developed via an iterative procedure to jointly recover the sparse support $\mathbf{x}$ and the grid mismatch error $\boldsymbol{\delta}$.

3.1. Bayesian Inference

As widely known, Bayesian inference is based on the posterior decided by likelihood and priors. According to (16), the likelihood is

$$p(\breve{\mathbf{r}} \,\big|\, \mathbf{x}; \boldsymbol{\Delta}) = \mathcal{N}(\breve{\mathbf{r}} \,\big|\, \breve{\boldsymbol{\psi}}(\vartheta)\mathbf{x}, \mathbf{I}_{2N(N-1)}) \tag{17}$$

Then, i.i.d. Gaussian priors are assigned to all the elements of $\mathbf{x}$ as follows:

$$p(\mathbf{x}; \boldsymbol{\Gamma}) = \mathcal{N}(\mathbf{x}|\mathbf{0}_L, \boldsymbol{\Gamma}) \tag{18}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{L\times L}$ is a diagonal matrix, i.e., $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma})$ and $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_L]^T \in \mathbb{R}^{L\times 1}$. **Remark**: $\boldsymbol{\gamma}$ is finally regarded as the solved $\mathbf{x}$ since $\boldsymbol{\gamma}$ represents the potential source power so as to reflect the potential source location.

According to the Bayesian formula, the posterior is

$$p(\mathbf{x}|\breve{\mathbf{r}}; \boldsymbol{\Delta}, \boldsymbol{\Gamma}) = \frac{p(\breve{\mathbf{r}} \,\big|\, \mathbf{x}; \boldsymbol{\Delta}) p(\mathbf{x}; \boldsymbol{\Gamma})}{\int p(\breve{\mathbf{r}} \,\big|\, \mathbf{x}; \boldsymbol{\Delta}) p(\mathbf{x}; \boldsymbol{\Gamma}) d\mathbf{x}} \tag{19}$$

(19) is analytically solved to be a Gaussian distribution with mean and variance as follows:

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{\Sigma}\breve{\boldsymbol{\psi}}(\vartheta)^T\breve{\mathbf{r}} \\ \boldsymbol{\Sigma} &= (\boldsymbol{\Gamma}^{-1} + \breve{\boldsymbol{\psi}}(\vartheta)^T\breve{\boldsymbol{\psi}}(\vartheta))^{-1} \end{aligned} \tag{20}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{L\times 1}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{L\times L}$. Please refer to Appendix A for the derivation. Finally, Bayesian inference is realized based on the maximum a posteriori (MAP) criterion. Specifically, we update hyperparameters $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$ to maximize the logarithmic expectation of (19), i.e.,

$$\begin{aligned} \boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma}) &= \underset{\boldsymbol{\Gamma}}{\mathrm{argmax}}\{\mathrm{E}[\ln p(\mathbf{x}|\breve{\mathbf{r}}; \boldsymbol{\Delta}, \boldsymbol{\Gamma})]\} \\ \boldsymbol{\Delta} = \mathrm{diag}(\boldsymbol{\delta}) &= \underset{\boldsymbol{\Delta}}{\mathrm{argmax}}\{\mathrm{E}[\ln p(\mathbf{x}|\breve{\mathbf{r}}; \boldsymbol{\Delta}, \boldsymbol{\Gamma})]\} \end{aligned} \tag{21}$$

Using (21), $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ are analytically solved as follows:

$$\begin{aligned} \boldsymbol{\gamma} &= \operatorname{diag}(\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}) \\ \boldsymbol{\delta} &= \operatorname{diag}(\mathbf{D}^+ \breve{\mathbf{r}}\, \boldsymbol{\mu}^T (\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma})^{-1} - \mathbf{D}^+\mathbf{C}) \end{aligned} \tag{22}$$

Please refer to Appendix B for the derivation.
Overall, the SBL algorithm is summarized in Table 1.

Table 1. The proposed method is named **Algorithm 1**.

Initialize $\boldsymbol{\delta}$, $\boldsymbol{\gamma}$, $\varepsilon^{(0)}$, error tolerance ε, set elements in $\boldsymbol{\delta}$ or $\boldsymbol{\gamma}$ as small and the same values.
While $\varepsilon^{(k)} > \varepsilon$ **do:**
compute $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ according to (20).
compute $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ according to (22).
$\varepsilon^{(k)} = \left\|\boldsymbol{\gamma}^{(k+1)} - \boldsymbol{\gamma}^{(k)}\right\| / \left\|\boldsymbol{\gamma}^{(k)}\right\|$.
$k = k + 1$.
End While
Return $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$.
Output refined DOAs.

The proposed method has guaranteed convergence since the posterior is bound to increase at each iteration. However, the SBL converges quite slowly, so we construct the corresponding deep unfolded network based on **Algorithm 1** to accelerate the convergence rate.

3.2. Deep Unfolding

In order to unfold **Algorithm 1** into deep neural network architecture, iterative equations (20) and (22) are rewritten as

$$\begin{aligned} \boldsymbol{\Sigma}^{(m)} &= \{\mathbf{C}^T\mathbf{C} + (\boldsymbol{\Gamma}^{(m-1)})^{-1} + \underbrace{\mathbf{C}^T\mathbf{D}\boldsymbol{\Delta}^{(m-1)} + (\boldsymbol{\Delta}^{(m-1)})^T\mathbf{D}^T\mathbf{C}}_{\boldsymbol{\Delta}_1} + \underbrace{(\boldsymbol{\Delta}^{(m-1)})^T\mathbf{D}^T\mathbf{D}\boldsymbol{\Delta}^{(m-1)}}_{\boldsymbol{\Delta}_2}\}^{-1} \\ \boldsymbol{\mu}^{(m)} &= \boldsymbol{\Sigma}^{(m)}(\mathbf{C}\breve{\mathbf{r}} + \underbrace{\mathbf{D}\boldsymbol{\Delta}^{(m-1)}\breve{\mathbf{r}}}_{\boldsymbol{\Delta}_3}) \\ \boldsymbol{\Gamma}^{(m)} &= \underbrace{\boldsymbol{\mu}^{(m)}(\boldsymbol{\mu}^{(m)})^T}_{\boldsymbol{\mu}_1} + \boldsymbol{\Sigma}^{(m)} \\ \boldsymbol{\Delta}^{(m)} &= \underbrace{2\mathbf{D}^+\breve{\mathbf{r}}(\boldsymbol{\mu}^{(m)})^T}_{\boldsymbol{\mu}_2}(\boldsymbol{\Gamma}^{(m)})^{-1} - \mathbf{D}^+\mathbf{C} \end{aligned} \tag{23}$$

via linear transformation, where $m = 1, \ldots, M$ denotes the $m - th$ iteration. **Remark**: In our deep unfolded neural networks, $\boldsymbol{\Delta}_1$, $\boldsymbol{\Delta}_3$, $\boldsymbol{\mu}_2$, matrix inversion, and matrix transposition with respect to iterative hyperparameters are fitted by linear layers since they are linear transformations, while the nonlinear quadratic terms $\boldsymbol{\Delta}_2$ and $\boldsymbol{\mu}_1$ are fitted by nonlinear layers. Constant terms are input, and the rest in (23) are realized by adders and multipliers. Specifically, the constructed network architecture is shown in Figures 3 and 4.

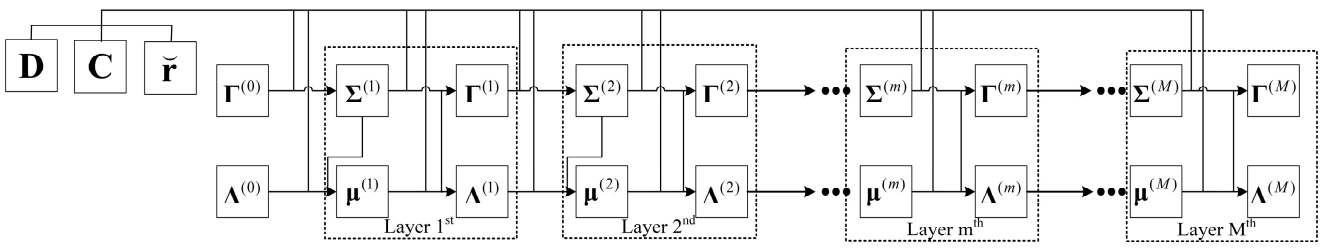


Figure 3. Scheme of the proposed Bayesian deep unfolded network.

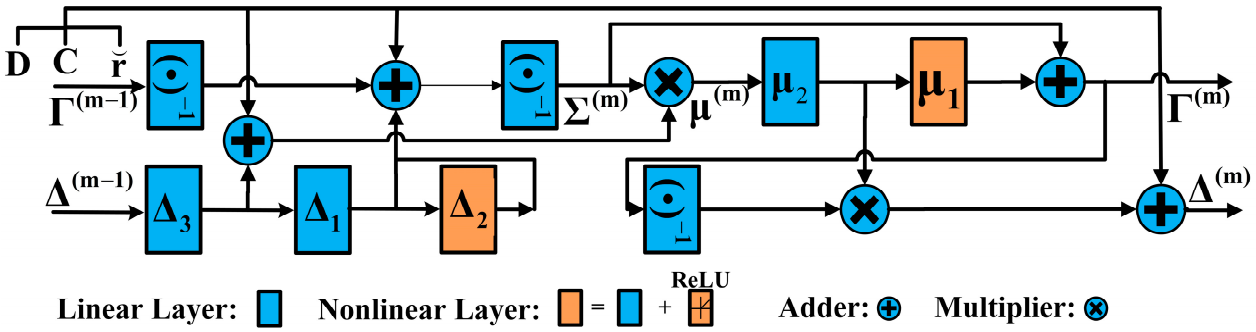


Figure 4. Scheme of the m-th layer of the Bayesian deep unfolded network.

4. Numerical Simulation

In this section, the superiority of the proposed method is fully evaluated by three simulations.

Simulation 1 tests the convergence rates of the 10-layer proposed method and **Algorithm 1**, ISTA [32], and 20-layer LISTA [9]. Simulation conditions are as follows. (i) Baseline conditions: Inter(R) Xeon(R) Platinum 8270 CPU @ 2.70 GHz, three uncorrelated source signals from random on-grid angles $\{-20°, 5°, 30°\}$ with random off-grid gaps $\{0.13°, 0.45°, 0.32°\}$, grid interval $1°$, $SNR = 20$dB, $T = 60$ snapshots, $N = 6$ sensors with position set $\mathbb{D}_{MHA} = \{0, 1, 8, 11, 13, 17\}$, and grid range $[-90°, 90°]$. (ii) Neural network training conditions: NVIDIA Quadro RTX 8000 GPU, on-gird space $[-90°, 90°]$ with $1°$ interval, and off-grid space $[-0.5°, 0.5°)$ with $0.01°$ interval; thus, the total is 18,100 labels, and each label corresponds to 10 samples so as to yield 181,000 samples $\breve{\mathbf{r}}$, epoch 500, batch size 32, a training–validating ratio of 8:2, an Adam optimizer with a learning rate of 0.001, and a designed normalized mean square error (NMSE) function as a loss function. The NMSE is defined as

$$NMSE = \frac{1}{2P}\sum_{p=1}^{P}\left(\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2/\|\mathbf{x}\|_2^2 + \|\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}\|_2^2/\|\boldsymbol{\delta}\|_2^2\right) \tag{24}$$

where P is the batch size of the validation set. $\hat{\mathbf{x}}$ and $\hat{\boldsymbol{\delta}}$ are the estimated values of the validated labels $\mathbf{x}$ and $\boldsymbol{\delta}$ yielded by (i) baseline conditions.

The simulation results are depicted in Figure 5. Intuitively, the algorithms based on deep unfolding, i.e., LISTA and the proposed method, converge faster than their original algorithms, i.e., ISTA and **Algorithm 1**. The proposed method requires fewer layers and achieves a lower bound than LISTA at convergence, which implies the proposed method has a faster convergence rate and better sparse recovery ability. The simulation results can be explained by the fact that the deep unfolded techniques indeed enable accelerating convergence, and SBL converges faster than ISTA originally. Owing to the reasonable and effective network architecture, the proposed method achieves remarkable convergence performance.

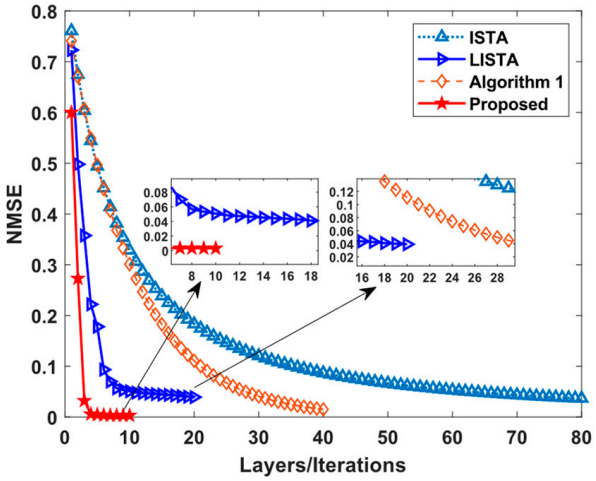


Figure 5. NMSE versus layers/iterations.

The following simulations focus on the root-mean-square error (RMSE) performance comparison of LISTA, FAAN-MUSIC [33], RVSBL [34], StructCovMLE [35], CRLB [36], and the proposed method. The RMSE is defined as

$$RMSE = \sqrt{\frac{1}{M_c K} \sum_{m_c=1}^{M_c} \sum_{k=1}^{K} \left(\hat{\theta}_{m_c,k} - \theta_k\right)^2} \tag{25}$$

where $M_c = 200$ is the Monte Carlo number, $\hat{\theta}_{m_c,k}$ is the estimated angle for the $k-th$ source in the $m_c - th$ trial, and θ_k is the true angle for the $k-th$ source.

Simulation 2 investigates the RMSE performance with respect to SNRs. As shown in Figure 6, the proposed method realizes a lower RMSE than others and approaches the CRLB. In particular, the proposed method outperforms RVSBL, which indicates that the used MHA indeed improves DOA estimation accuracy. On the one hand, the proposed method inherits the advantage of SBL, i.e., excellent sparsity recovery ability. On the other hand, the real-valued transformation further enhances its ability.

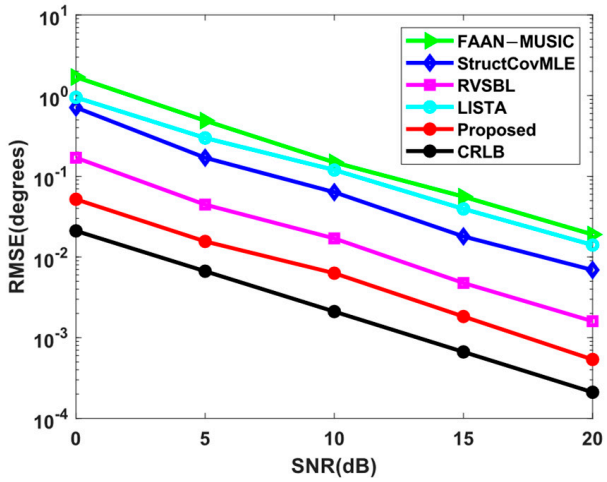


Figure 6. RMSE versus SNR.

Simulation 3 tests the RMSE performance with respect to the number of snapshots. The simulation results in Figure 7 show that the proposed method is robust to snapshot varying, i.e., it requires slight sampling data but achieves high estimation accuracy, which can be explained by the fact that data-driven methods have powerful mapping and fitting abilities. Undoubtedly, the constructed deep unfolded network realizes excellent mapping effects, and, simultaneously, prominent generalization capability.

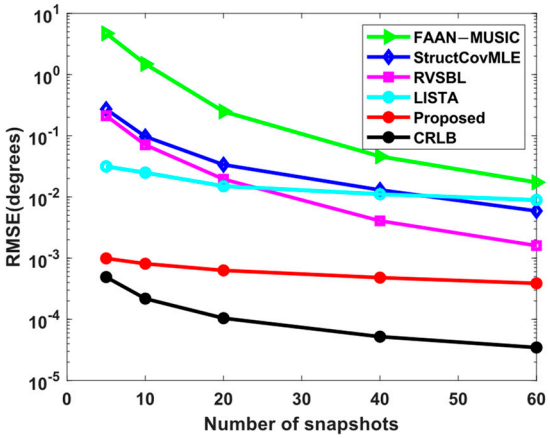


Figure 7. RMSE versus the number of snapshots.

Simulation 4 examines the RMSE performance with respect to the number of sources. Here, all the source angles are uniformly selected from the interval of $[-90°, 90°]$. Intuitively, in Figure 8, all

the algorithms inevitably suffer performance reduction, but the proposed method still outperforms others over the whole range of various numbers of sources. The results indicate that the proposed method has the potential to locate more sources when the angle accuracy is not strictly constrained.

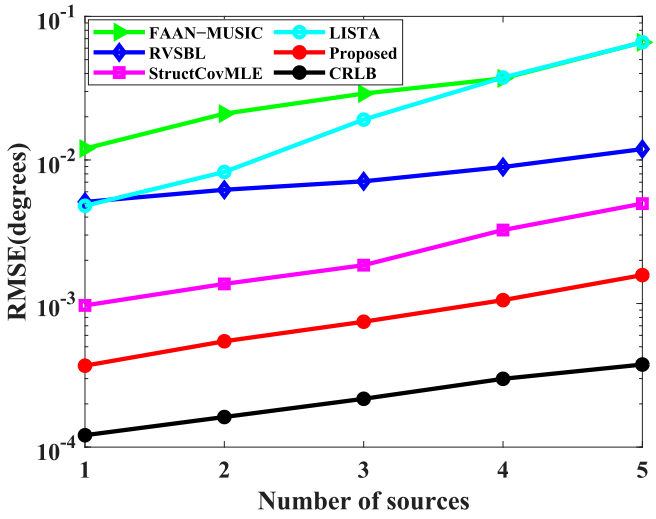


Figure 8. RMSE versus the number of sources.

Simulation 5 focuses on the RMSE performance with respect to the number of sensors. Notice that the number of sensors ranges from 4 to 8. The reason for this is that the MUSCI-based method, i.e., FAAN–MUSIC, can only locate maximal $N-1$ sources with N sensors. Thus, $N-1 \geq 3$ must be satisfied so that FAAN–MUSIC works normally. In the proposed method, the adopted MHA position set is $\mathbb{D}_{MHA} = \{0, 1, 4, 9, 15, 22, 32, 34\}$ and contains the maximized eight sensors. In Figure 9, the proposed method achieves the fewest RMSEs. This result can be explained by the fact that the proposed method can inherit the ability of SBL to handle highly underdetermined problems, and the adopted MHA indeed plays an important role in improving estimation accuracy.

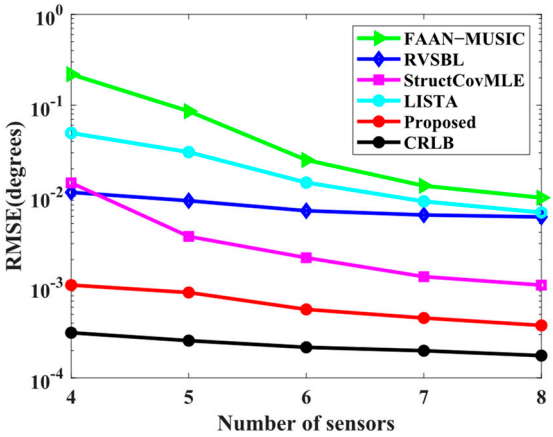


Figure 9. RMSE versus the number of sensors.

Based on Simulation 2, Simulation 6 examines the average computational time for each Monte Carlo trial. As shown in Table 2, the proposed method runs faster than others, except for LISTA, which proves that the used deep unfolded network indeed improves SBL efficiency significantly. **Remark**: The result, i.e., the proposed method, is slower than LISTA and has no conflict against the conclusions in Simulation 1 because the computational complexity of each layer of the proposed method is larger than that in LISTA.

Table 2. Computational time of the algorithms.

Algorithms	Computational Time
FAAN-MUSIC	1.223 s
RVSBL	0.294 s
StructCovMLE	0.537 s
LISTA	0.00137 s
Proposed	0.00472 s

5. Conclusions

In this paper, a Bayesian deep unfolded network is constructed according to our derived SBL iterative formulas and is based on the adopted minimum hole array for a high estimation accuracy. Simulation results sufficiently illustrate the superiority of the proposed method, such as its high convergence rate and excellent estimation performance. On the one hand, the proposed method inherits the advantages of SBL, e.g., robustness to various snapshots, excellent sparsity performance, and the ability to handle underdetermined cases. On the other hand, the proposed method reserves and extends the superiority of deep learning, such as steady estimation ability owing to real-valued data and fast convergence ability. More importantly, this paper empirically proves the advantages of deep unfolding again, though rigorously theoretical identification is not provided.

Author Contributions: Conceptualization, N.L. and W.F.; methodology, N.L.; software, F.L.; validation, B.Z., X.Z. and W.F.; writing—original draft preparation, N.L.; writing—review and editing, N.L. and W.F.; visualization, F.L.; supervision, B.Z.; project administration, X.Z.; funding acquisition, X.Z., and W.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by [the Natural Science Basic Research Program of Shaanxi] grant number [2023-JC-YB-488]. This work is supported in part by the Youth Talent Lifting Project of the China Association for Science and Technology No. 2021-JCJQ-QT-018 and The Youth Innovation Team of Shaanxi Universities.

Data Availability Statement: No new data were created.

Acknowledgments: The authors would like to thank Menghui Fan for his investigation and participation.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

For (19), the logarithmic form with respect to variable $\mathbf{x}$ is

$$\begin{aligned}\ln(19) &= \ln[p(\breve{\mathbf{r}}|\mathbf{x};\boldsymbol{\Delta})p(\mathbf{x};\boldsymbol{\Gamma})] + const = -\tfrac{1}{2}[(\breve{\mathbf{r}} - \breve{\boldsymbol{\psi}}(\vartheta)\mathbf{x})^T(\breve{\mathbf{r}} - \breve{\boldsymbol{\psi}}(\vartheta)\mathbf{x}) + \mathbf{x}^T\boldsymbol{\Gamma}^{-1}\mathbf{x}] + const \\ &= -\tfrac{1}{2}\mathbf{x}^T(\boldsymbol{\Gamma}^{-1} + \breve{\boldsymbol{\psi}}(\vartheta)^T\breve{\boldsymbol{\psi}}(\vartheta))\mathbf{x} + \mathbf{x}^T\breve{\boldsymbol{\psi}}(\vartheta)^T\breve{\mathbf{r}} + const = -\tfrac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + const\end{aligned} \tag{A1}$$

Patently, $\mathbf{x}$ follows a Gaussian distribution, whose mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ can be solved via simple linear transformation.

Appendix B

To solve (21), i.e., the extreme points of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$, we take the derivative of $\mathrm{E}[\ln p(\mathbf{x}|\breve{\mathbf{r}};\boldsymbol{\Delta},\boldsymbol{\Gamma})]$ with respect to $\boldsymbol{\Gamma}$, $\boldsymbol{\Delta}$.

$$\begin{aligned}&\frac{\partial\left\{\mathrm{E}[\ln p(\mathbf{x}|\breve{\mathbf{r}};\boldsymbol{\Delta},\boldsymbol{\Gamma})]\right\}}{\partial\boldsymbol{\Gamma}} = \frac{\partial\{\mathrm{E}[\ln p(\mathbf{x};\boldsymbol{\Gamma})]+const\}}{\partial\boldsymbol{\Gamma}} = \frac{\partial\left\{\mathrm{E}[\ln|\boldsymbol{\Gamma}|^{-1/2} - \mathbf{x}^T\boldsymbol{\Gamma}^{-1}\mathbf{x}/2]+const\right\}}{\partial\boldsymbol{\Gamma}} \\ &= \frac{\partial\left\{\mathrm{E}[\ln\left|\boldsymbol{\Gamma}^{-1}\right|^{1/2} - \mathbf{x}^T\boldsymbol{\Gamma}^{-1}\mathbf{x}/2]+const\right\}}{\partial\boldsymbol{\Gamma}^{-1}}\frac{\partial\mathrm{tr}[(\boldsymbol{\Gamma}^{-1})^T]}{\partial\boldsymbol{\Gamma}} = \frac{\partial\left\{\frac{1}{2}\ln\left|\boldsymbol{\Gamma}^{-1}\right| - \frac{1}{2}\mathrm{E}[\mathrm{tr}(\boldsymbol{\Gamma}^{-1}\mathbf{x}\mathbf{x}^T)]+const\right\}}{\partial\boldsymbol{\Gamma}^{-1}}(-\boldsymbol{\Gamma}^{-2}) \\ &= \left\{\left[(\boldsymbol{\Gamma}^{-1})^{-1}\right]^T/2 - \mathrm{E}[(\mathbf{x}\mathbf{x}^T)^T]/2\right\}(-\boldsymbol{\Gamma}^{-2}) = [\boldsymbol{\Gamma}^T - (\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma})](-\boldsymbol{\Gamma}^{-2})/2\end{aligned} \tag{A2}$$

where $\boldsymbol{\Sigma}$ is a real symmetric matrix. Obviously, $\boldsymbol{\Gamma} = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma}$ is the solution of (27).

Likewise, the derivative with respect to $\boldsymbol{\Delta}$ is

$$\begin{aligned}&\frac{\partial\left\{\mathrm{E}[\ln p(\mathbf{x}|\breve{\mathbf{r}};\boldsymbol{\Delta},\boldsymbol{\Gamma})]\right\}}{\partial\boldsymbol{\Delta}}=\frac{\partial\left\{\mathrm{E}[\ln p(\breve{\mathbf{r}}|\mathbf{x};\boldsymbol{\Delta})]+const\right\}}{\partial\boldsymbol{\Delta}}=\frac{\partial\left\{\mathrm{E}[(\breve{\mathbf{r}}-(\mathbf{C}+\mathbf{D}\boldsymbol{\Delta})\mathbf{x})^{T}(\breve{\mathbf{r}}-(\mathbf{C}+\mathbf{D}\boldsymbol{\Delta})\mathbf{x})+const]\right\}}{\partial\boldsymbol{\Delta}}\\&=\frac{\partial\left\{\mathrm{E}[-2\mathbf{x}^{T}\boldsymbol{\Delta}^{T}\mathbf{D}^{T}\breve{\mathbf{r}}-2\mathbf{x}^{T}\boldsymbol{\Delta}^{T}\mathbf{D}^{T}\mathbf{C}\mathbf{x}+\mathbf{x}^{T}\boldsymbol{\Delta}^{T}\mathbf{D}^{T}\mathbf{D}\boldsymbol{\Delta}\mathbf{x}]+const\right\}}{\partial\boldsymbol{\Delta}}=\mathrm{E}[-2\mathbf{D}^{T}\breve{\mathbf{r}}\mathbf{x}^{T}+2\mathbf{D}^{T}\mathbf{C}\mathbf{x}\mathbf{x}^{T}+2\mathbf{D}^{T}\mathbf{D}\boldsymbol{\Delta}\mathbf{x}\mathbf{x}^{T}]\\&=-2\mathbf{D}^{T}\breve{\mathbf{r}}\boldsymbol{\mu}+2\mathbf{D}^{T}\mathbf{C}(\boldsymbol{\mu}\boldsymbol{\mu}^{T}+\boldsymbol{\Sigma})+2\mathbf{D}^{T}\mathbf{D}\boldsymbol{\Delta}(\boldsymbol{\mu}\boldsymbol{\mu}^{T}+\boldsymbol{\Sigma})\end{aligned}\tag{A3}$$

Using linear transformation, the solution of (28) is gained so that (22) is yielded. The above derivation process is based on matrix properties and derivative rules, i.e., $\mathbf{x}^{T}\mathbf{A}\mathbf{x}=\mathrm{tr}(\mathbf{A}\mathbf{x}\mathbf{x}^{T})$, $\mathrm{tr}(\mathbf{A}\mathbf{B})=\mathrm{tr}(\mathbf{B}\mathbf{A})$, $\mathrm{tr}(\mathbf{A}^{T})=\mathrm{tr}(\mathbf{A})$, $\partial\mathrm{tr}(\mathbf{A}\mathbf{B})/\partial\mathbf{B}=\mathbf{A}^{T}$, $\partial\ln|\mathbf{A}|/\partial\mathbf{A}=(\mathbf{A}^{-1})^{T}$, $\partial\mathrm{tr}(\mathbf{A}\mathbf{B}\mathbf{A}^{T})/\partial\mathbf{A}=\mathbf{A}(\mathbf{B}+\mathbf{B}^{T})$, $\partial\mathrm{tr}(\mathbf{A})/\partial\mathbf{C}=[\partial\mathrm{tr}(\mathbf{A})/\partial\mathbf{B}][\partial\mathrm{tr}(\mathbf{B})/\partial\mathbf{C}]$, $\partial\mathrm{tr}(\mathbf{A}^{-1})/\partial\mathbf{A}=-(\mathbf{A}^{-2})^{T}$, $\forall$ any real square matrix $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and vector $\mathbf{x}$.

References

1. Krim, H.; Viberg, M. Two decades of array signal processing research: The parametric approach. *IEEE Signal Proc. Mag.* **1996**, *13*, 67–94. [CrossRef]
2. Pesavento, M.; Trinh-Hoang, M.; Viberg, M. Three More Decades in Array Signal Processing Research: An optimization and structure exploitation perspective. *IEEE Signal Proc. Mag.* **2023**, *40*, 92–106. [CrossRef]
3. Liu, W.; Haardt, M.; Greco, M.S.; Mecklenbräuker, C.F.; Willett, P. Twenty-Five Years of Sensor Array and Multichannel Signal Processing: A review of progress to date and potential research directions. *IEEE Signal Proc. Mag.* **2023**, *40*, 80–91. [CrossRef]
4. Purwins, H.; Li, B.; Virtanen, T.; Schlüter, J.; Chang, S.Y.; Sainath, T. Deep learning for audio signal processing. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 206–219. [CrossRef]
5. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
6. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
7. Nielsen, I.E.; Dera, D.; Rasool, G.; Ramachandran, R.P.; Bouaynaya, N.C. Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. *IEEE Signal Proc. Mag.* **2022**, *39*, 73–84. [CrossRef]
8. Monga, V.; Li, Y.; Eldar, Y.C. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Proc. Mag.* **2021**, *38*, 18–44. [CrossRef]
9. Gregor, K.; LeCun, Y. Learning fast approximations of sparse coding. In Proceedings of the 27th International Conference on International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 399–406.
10. Hershey, J.R.; Roux, J.L.; Weninger, F. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv* **2014**, arXiv:1409.2574.
11. Liu, J.; Chen, X. ALISTA: Analytic weights are as good as learned weights in LISTA. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
12. Chen, X.; Liu, J.; Wang, Z.; Yin, W. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
13. Zhang, J.; Ghanem, B. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1828–1837.
14. Sun, J.; Li, H.; Xu, Z. Deep ADMM-Net for compressive sensing MRI. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 10–18.
15. Su, X.; Liu, Z.; Shi, J.; Hu, P.; Liu, T.; Li, X. Real-valued deep unfolded networks for off-grid DOA estimation via nested array. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 4049–4062. [CrossRef]
16. Su, X.; Hu, P.; Liu, Z.; Shi, J.; Li, X. Deep alternating projection networks for gridless DOA estimation with nested array. *IEEE Signal Proc. Let.* **2022**, *29*, 1589–1593. [CrossRef]
17. Wu, L.; Liu, Z.M.; Liao, J. DOA estimation using an unfolded deep network in the presence of array imperfections. In Proceedings of the 2022 7th International Conference on Signal and Image Processing (ICSIP), Suzhou, China, 20–22 July 2022; pp. 182–187.
18. Tang, H.; Zhang, Y.; Luo, J.; Zhang, Y.; Huang, Y.; Yang, J. Sparse DOA Estimation Based on a Deep Unfolded Network for MIMO Radar. In Proceedings of the IGARSS 2023—2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 16–21 July 2023; pp. 5547–5550.
19. Li, Y.; Wang, X.; Olesen, R.L. Unfolded deep neural network (UDNN) for high mobility channel estimation. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March–1 April 2021; pp. 1–6.
20. Johnston, J.; Wang, X. Model-based deep learning for joint activity detection and channel estimation in massive and sporadic connectivity. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 9806–9817. [CrossRef]
21. Kang, K.; Hu, Q.; Cai, Y.; Yu, G.; Hoydis, J.; Eldar, Y.C. Mixed-timescale deep-unfolding for joint channel estimation and hybrid beamforming. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 2510–2528. [CrossRef]
22. Tipping, M.E. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.* **2001**, *1*, 211–244.
23. Li, N.; Zhang, X.K.; Zong, B.; Lv, F.; Xu, J.; Wang, Z. An off-grid direction-of-arrival estimator based on sparse Bayesian learning with three-stage hierarchical Laplace priors. *Signal Process.* **2024**, *218*, 109371. [CrossRef]

24. Gong, Z.; Su, X.; Hu, P.; Liu, S.; Liu, Z. Deep Unfolding Sparse Bayesian Learning Network for Off-Grid DOA Estimation with Nested Array. *Remote Sens.* **2023**, *15*, 5320. [CrossRef]
25. Wan, Q.; Fang, J.; Huang, Y.; Duan, H.; Li, H. A variational Bayesian inference-inspired unrolled deep network for MIMO detection. *IEEE Trans. Signal Process.* **2022**, *70*, 423–437. [CrossRef]
26. Hu, Q.; Shi, S.; Cai, Y.; Yu, G. DDPG-driven deep-unfolding with adaptive depth for channel estimation with sparse Bayesian learning. *IEEE Trans. Signal Process.* **2022**, *70*, 4665–4680. [CrossRef]
27. Gao, J.; Cheng, X.; Li, G.Y. Deep Unfolding-Based Channel Estimation for Wideband TeraHertz Near-Field Massive MIMO Systems. *arXiv* **2023**, arXiv:2308.13381.
28. Linebarger, D.A.; Sudborough, I.H.; Tollis, I.G. Difference bases and sparse sensor arrays. *IEEE Trans. Inf. Theory* **1993**, *39*, 716–721. [CrossRef]
29. Pal, P.; Vaidyanathan, P.P. Nested arrays: A novel approach to array processing with enhanced degrees of freedom. *IEEE Trans. Signal Process.* **2010**, *58*, 4167–4181. [CrossRef]
30. Ottersten, B.; Stoica, P.; Roy, R. Covariance matching estimation techniques for array signal processing applications. *Digit. Signal Process.* **1998**, *8*, 185–210. [CrossRef]
31. Yang, J.; Liao, G.; Li, J. An efficient off-grid DOA estimation approach for nested array signal processing by using sparse Bayesian learning strategies. *Signal Process.* **2016**, *128*, 110–122. [CrossRef]
32. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
33. Stoica, P.; Babu, P. Low-rank covariance matrix estimation for factor analysis in anisotropic noise: Application to array processing and portfolio selection. *IEEE Trans. Signal Process.* **2023**, *71*, 1699–1713. [CrossRef]
34. Das, A. Real-valued sparse Bayesian learning for off-grid direction-of-arrival (DOA) estimation in ocean acoustics. *IEEE J. Ocean. Eng.* **2020**, *46*, 172–182. [CrossRef]
35. Pote, R.R.; Rao, B.D. Maximum likelihood-based gridless doa estimation using structured covariance matrix recovery and sbl with grid refinement. *IEEE Trans. Signal Process.* **2023**, *71*, 802–813. [CrossRef]
36. Stoica, P.; Larsson, E.G.; Gershman, A.B. The stochastic CRB for array processing: A textbook derivation. *IEEE Signal Process. Lett.* **2001**, *8*, 1448–1501. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.