mathematics

Article

Improved Analytic Learned Iterative Shrinkage Thresholding Algorithm and Its Application to Tomographic Synthetic Aperture Radar Building Object Height Inversion

Weiqiu Liang [1], Jiying Liu [1,*] and Jubo Zhu [2,*]

[1] College of Science, National University of Defense Technology, Changsha 410073, China; lwq22@nudt.edu.cn
[2] School of Artificial Intelligence, Sun Yat-sen University (Zhuhai Campus), Zhuhai 519082, China
* Correspondence: jiying.liu@nudt.edu.cn (J.L.); zhujubo@mail.sysu.edu.cn (J.Z.)

Abstract: Tomographic Synthetic Aperture Radar (TomoSAR) building object height inversion is a sparse reconstruction problem that utilizes the data obtained from several spacecraft passes to invert the scatterer position in the height direction. In practical applications, the number of passes is often small, and the observation data are also small due to the objective conditions, so this study focuses on the inversion under the restricted observation data conditions. The Analytic Learned Iterative Shrinkage Thresholding Algorithm (ALISTA) is a kind of deep unfolding network algorithm, which is a combination of the Iterative Shrinkage Thresholding Algorithm (ISTA) and deep learning technology, and it has the advantages of both. The ALISTA is one of the representative algorithms for TomoSAR building object height inversion. However, the structure of the ALISTA algorithm is simple, which has neither the excellent connection structure of a deep learning network nor the acceleration format combined with the ISTA algorithm. Therefore, this study proposes two directions of improvement for the ALISTA algorithm: firstly, an improvement in the inter-layer connection of the network by introducing a connection similar to residual networks obtains the Extragradient Analytic Learned Iterative Shrinkage Thresholding Algorithm (EALISTA) and further proves that the EALISTA achieves linear convergence; secondly, there is an improvement in the iterative format of the intra-layer iteration of the network by introducing the Nesterov momentum acceleration, which obtains the Fast Analytic Learned Iterative Shrinkage Thresholding Algorithm (FALISTA). We first performed inversion experiments on simulated data, which verified the effectiveness of the two proposed algorithms. Then, we conducted TomoSAR building object height inversion experiments on limited measured data and used the deviation metric P to measure the robustness of the algorithms to invert under restricted observation data. The results show that both proposed algorithms have better robustness, which verifies the superior performance of the two algorithms. In addition, we further analyze how to choose the most suitable algorithms for inversion in engineering practice applications based on the results of the experiments on measured data.

Keywords: ALISTA; residual structure; Nesterov acceleration; height inversion

MSC: 68T99

check for updates

Citation: Liang, W.; Liu, J.; Zhu, J. Improved Analytic Learned Iterative Shrinkage Thresholding Algorithm and Its Application to Tomographic Synthetic Aperture Radar Building Object Height Inversion. *Mathematics* **2024**, *12*, 1464. https://doi.org/10.3390/math12101464

Academic Editor: Ioannis G. Tsoulos

Received: 11 April 2024
Revised: 6 May 2024
Accepted: 7 May 2024
Published: 9 May 2024

1. Introduction

TomoSAR is a three-dimensional imaging technique. When imaging the object of a building in an urban area, it inverts the height on the azimuth-distance-resolved unit of the observed object on a two-dimensional image, thus obtaining the azimuth distance height information of the object, which is also known as the height inversion. It processes SAR complex images obtained from multiple passes through observation and then obtains the scatterer distribution in the height direction, which has the ability of three-dimensional resolution of the radar object and can reconstruct the observation object in the height direction [1]. TomoSAR object height inversion technology can really solve the problem of overlay masking and has a wide range of applications in field topographic mapping, field biomass estimation, resource survey, forest cover, three-dimensional imaging of man-made buildings, detection of underground buried objects, and other civil and military applications. The traditional TomoSAR building object height inversion algorithm is a fast Fourier transform, where the image is de-skewed, and then a fast Fourier transform is performed in the height direction to obtain the scatterer distribution in the height direction [2]. This algorithm requires a large number of passes, which means that the observation data need to be sufficient, and the baseline distribution needs to be uniform. However, in practical applications, due to objective limitations, the number of flight passes is generally small, so the observation data are also limited. This leads to high sidelobes in the inversion results based on the fast Fourier transform algorithm, which has a significant impact on the quality of the inversion. In order to improve the resolution in the height direction and reduce the sidelobes in the case of a limited number of passes, the theory of compressed sensing has been introduced into the field of TomoSAR building object height inversion [3]. Compared with the traditional object height inversion algorithm, the TomoSAR building object height inversion algorithm based on compressed sensing has a greater improvement in the accuracy of the inversion results under the condition of limited observation data, so the TomoSAR object height inversion based on compressed sensing is a popular research direction. A mainstream traditional algorithm for TomoSAR building object height inversion based on compressed sensing is the Iterative Shrinkage Thresholding Algorithm (ISTA), which is a convex optimization algorithm that requires multiple iterations, and the parameters of the algorithm need to be set manually, resulting in low inversion efficiency.

Due to an improvement in arithmetic power, deep learning technology has been greatly developed. Deep learning has been introduced into the field of compressed perception, and under this trend, a series of research results have been achieved by combining the TomoSAR building object height inversion technique based on compressed sensing with deep learning [4–7]. The current TomoSAR object height inversion algorithms formed based on the combination of compressed sensing and deep learning are mainly deep unfolding network algorithms. The ISTA-based deep unfolding network algorithm is one of the typical representatives of this class of algorithms. The idea of the deep unfolding network algorithm is to consider each iteration step in the traditional iterative algorithm as a layer of an artificial neural network and define some iteration parameters as learnable parameters in the network, such as the step size of each iteration and the threshold value. Finally, the network is trained to determine the optimal parameter values for the algorithm. In fact, in traditional optimization algorithms, the step size is an important parameter that has a great impact on the convergence speed, and different values of the step size can significantly affect the performance of the algorithm. In practical applications, the algorithm's step size is usually set to a fixed value, or the optimal step size is solved by a line search, which is very slow to converge and time-consuming, so it is extremely meaningful to solve the optimal step size parameter using deep learning methods. This method is a combination of traditional optimization algorithms and deep learning techniques, so it combines the advantages of both and generally has better performance and faster convergence than traditional algorithms. Moreover, since the deep unfolding network algorithm is obtained on the basis of the traditional algorithm, it retains the interpretability of the original algorithm.

Gregor and LeCun first proposed this class of algorithms in 2010, which unfolds the classical ISTA into the form of a network and sets the known matrix parameters in the algorithm as well as the thresholds in the threshold function as learnable parameters, resulting in the Learning Iterative Shrinkage Thresholding Algorithm (LISTA) [8]. After LISTA was proposed, many depth unfolding algorithms related to it were proposed, and various variant algorithms based on LISTA-related algorithms appeared, such as LISTA-CPSS [9], TiLISTA [10], ALISTA [10], SLISTA [11], GLISTA [12], Ada-LISTA [13], LISTA-AT [14], HyperLISTA [15], ALISTA-Polyak [15], and so on. These algorithms mainly improve the initial LISTA and involve two main directions; one is an improvement in the network structure and the other is an improvement in the threshold function. In the improvement work of the network structure, the first is to prove the existence of a coupling relationship between the matrix parameters in LISTA, to propose a new deep unfolding network algorithm, LISTA-CPSS, to reduce the number of parameters to be learned, and to prove the linear convergence of LISTA-CPSS. Next, the matrix parameters of each layer of the network are shared on the basis of the LISTA network structure, which greatly reduces the number of parameters that the network needs to train, thus obtaining the deep unfolding network algorithm TiLISTA. Immediately after, it is further demonstrated that the matrix parameters in the network can be obtained by solving an optimization problem independently, without being set as learnable parameters; therefore, based on the TiLISTA, the parameters that need to be trained are further reduced, which yields the ALISTA. ALISTA only needs to learn a small number of scalar parameters compared to other previous networks, which in turn greatly further reduces the number of parameters to be learned and greatly improves the training efficiency. The excellence of the ALISTA makes it one of the current advanced algorithms for TomoSAR building object height inversion. Studies have shown that when the ALISTA is used for TomoSAR building object height inversion, its performance is superior to the commonly used traditional algorithms such as OMP, IHT, and ISTA [16,17].

However, the improvement from LISTA to ALISTA is only the selection of training parameters on the original LISTA iteration format, which improves the training efficiency by reducing the number of parameters to be trained, and the network structure obtained is just a simple structure of layer-by-layer connection. In the network inter-layer connection structure, the excellent network structure in deep learning is not introduced into the deep unfolding network algorithm; in the network intra-layer iteration format, the accelerated format of the ISTA is not introduced. Li et al. proposed the Extragradient Learned Iterative Shrinkage Thresholding Algorithm (ELISTA) based on the extragradient method, combining the idea of the extragradient method with LISTA, thus obtaining the ELISTA, which has a network structure similar to that of a deep residual network, and their work links deep unfolding networks to the classical deep residual networks [18,19]. However, the training parameters of ELISTA are matrix parameters, and the training is time-consuming. Therefore, we further combine the idea of ELISTA with ALISTA to obtain EALISTA, which only needs to train scalar parameters and prove its convergence. The EALISTA requires training only scalar parameters like the ALISTA and has the same excellent residual structure as the ELISTA network. In addition, Chen et al. found faster convergence after adding a Polyak ball momentum acceleration term to the iterative format of the ALISTA network [15]. However, in first-order optimization algorithms, Polyak ball momentum acceleration does not necessarily guarantee convergence. Inspired by this, in order to accelerate the ALISTA, we introduce the Nesterov momentum acceleration term and obtain an accelerated deep unfolding network algorithm, which we call the FALISTA. It is generally believed that the FALISTA converges faster than the ALISTA.

After improving the algorithm, we first conduct experiments on simulation data to verify the effectiveness of the proposed algorithm, followed by experiments on real measurement data to verify the superiority of the proposed algorithm. Due to the lack of a large number of measured TomoSAR datasets as training data, in this study, we use the simulation dataset with fixed sparsity to perform simulation and real experiments.

There are two main theoretical innovations in this study: combining the two algorithms ALISTA and ELISTA, an improved algorithm EALISTA with the advantages of both algorithms is given, and a proof of convergence of EALISTA is given; combining the advantages of the ALISTA with the Nesterov acceleration algorithm yields an acceleration algorithm, FALISTA, with relatively few network parameters. We improved the ALISTA in two dimensions: the connection method between network layers and the iteration format within network layers, respectively.

This paper is structured as follows: Section 2 elucidates the model and solution of the height inversion; Section 3 proposes the EALISTA and proves its convergence; Section 4 proposes the FALISTA; Section 5 shows the results of the simulated data experiments and the measured data experiments; Section 6 discusses the results of the experiments; and Section 7 is the conclusion, which sums up this study and gives an outlook.

2. Mathematical Modeling and Solution of Height Inversion

2.1. Mathematical Modeling of Height Inversion

TomoSAR utilizes multiple aligned 2D SAR image data with different viewing angles obtained from multiple observations of the same scene to perform aperture synthesis in the altitude direction to obtain altitude-direction information, so as to obtain 3D scattering information of the observed object. By observing the same area several times and obtaining multiple SAR images with different incidence angles, TomoSAR can synthesize the aperture in the height direction perpendicular to the azimuth distance plane to obtain the height information of the object. Figure 1 illustrates the principle of the TomoSAR building object height inversion.

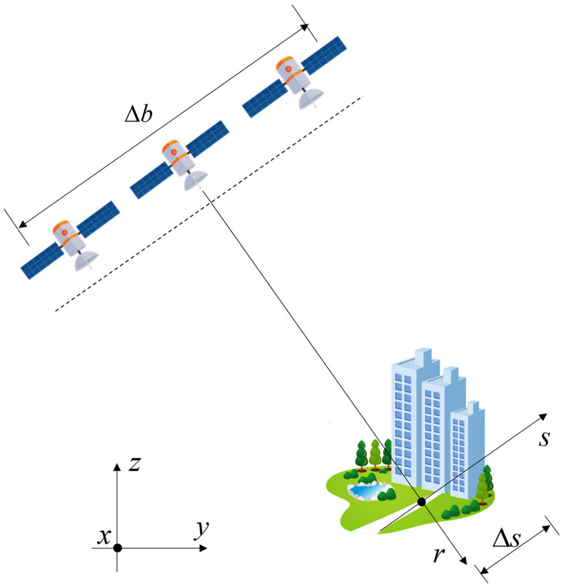


Figure 1. Schematic diagram of TomoSAR building object height inversion model.

Let $B = [b_1, b_2, \cdots, b_M]$, $m = 1, 2, \cdots, M$ denote the height-directed aperture distribution, Δs is the height-directed span, indicating the extent of the height inversion, for a selected azimuth-distance-resolving cell (x_0, R_0), the measurement $y_m(x_0, R_0)$, corresponding to the mth SAR image with aperture size b_m can be expressed as:

$$y_m(x_0, R_0) = \int_{\Delta s} \gamma(s) \exp(-j2\pi\xi_m s) ds, \tag{1}$$

where $\gamma(s)$ is the backward scattering coefficient along the height toward s. The backward scattering coefficient reflects the position of the scatterer, and we only need to ask for the backward scattering coefficient to obtain the position distribution of the scatterer in the height direction; $\xi_m = -2b_m/\lambda R$ is the spatial frequency, where b_m is the size of the height-directed aperture; λ is the emitted pulse wavelength; and R is the instantaneous slant distance. Discretizing the height-directed backscattering coefficient $\gamma(s)$, inverse model (1) can be approximated as follows:

$$y_m \approx \delta_s \sum_{l=1}^{L} \gamma(s_l) \exp(-j2\pi\xi_m s_l), \tag{2}$$

where L is the number of points in the height-wise discretization; the constant $\delta_s = \Delta s/(L-1)$ is the height-wise discretization interval.

Let $y = [y_1, y_2, \cdots, y_M]^T$ and $\gamma = [\gamma(s_1), \gamma(s_2), \cdots, \gamma(s_L)]^T$ denote all baseline data and discrete backward scattering coefficients at the studied azimuth-distance-resolving cell (x_0, R_0), respectively; then, (2) can be written as:

$$y_{M\times 1} = \Phi_{M\times L}\gamma_{L\times 1} \tag{3}$$

where $\Phi \in C^{M\times L}$ is the observation matrix constructed from the TomoSAR imaging geometry relations, which can be expressed as $\Phi(m,l) = \exp(j(4\pi/\lambda R)b_m s_l)$.

2.2. Solving the Model

The object of interest in this study is man-made buildings, and the distribution of backscattering coefficients of these buildings in the height direction can be considered to be sparse. Therefore, for the height-direction sparse distribution, the sparse reconstruction of the height-direction scattering coefficient can be realized by solving the l_q regularization model:

$$\begin{aligned} &\min_{\gamma}\|\gamma\|_q^q,\ 0 < q \leq 1 \\ &s.t. \quad \text{y} = \Phi\gamma \end{aligned} \tag{4}$$

When the height-wise sparsity K is unknown and there is measurement noise, the above model can be solved approximately by solving the following optimization problem:

$$\hat{\gamma} = \underset{\gamma}{\text{argmin}}\{\frac{1}{2}\|y - \Phi\gamma\|_2^2 + \lambda\|\gamma\|_q^q\}, \tag{5}$$

where λ is the regularization parameter. The precondition for TomoSAR height inversion is that the height to backward scattering coefficient or its transformation is sparse, which is the requirement of the theory of compressed perception, and this condition is obviously valid for the man-made building object; this optimization problem is a composite optimization problem. Generally, $q = 1$, at this point, is a typical LASSO problem:

$$\hat{\gamma} = \underset{\gamma}{\text{argmin}}\{\frac{1}{2}\|y - \Phi\gamma\|_2^2 + \lambda\|\gamma\|_1\}. \tag{6}$$

Since neural networks are generally based on the real number domain for signal processing in image processing, but the two-dimensional images obtained from radar imaging are in a complex form, they need to be transformed into the real number domain before processing:

$$\begin{pmatrix} \text{Re}(y) \\ \text{Im}(y) \end{pmatrix} = \begin{pmatrix} \text{Re}(\Phi) & -\text{Im}(\Phi) \\ \text{Im}(\Phi) & \text{Re}(\Phi) \end{pmatrix} \begin{pmatrix} \text{Re}(\gamma) \\ \text{Im}(\gamma) \end{pmatrix}, \tag{7}$$

At this point:

$$y = \begin{pmatrix} \text{Re}(y) \\ \text{Im}(y) \end{pmatrix}, \Phi = \begin{pmatrix} \text{Re}(\Phi) & -\text{Im}(\Phi) \\ \text{Im}(\Phi) & \text{Re}(\Phi) \end{pmatrix}, \gamma = \begin{pmatrix} \text{Re}(\gamma) \\ \text{Im}(\gamma) \end{pmatrix} \tag{8}$$

3. EALISTA and Its Convergence Analysis

3.1. ALISTA and ELISTA

Convex optimization algorithms are generally used to solve the optimal solution of problem (6), and the ISTA is one of the typical mainstream solution algorithms. The kth iteration of the ISTA is formulated as follows:

$$\gamma^k = \eta_{\lambda/L}\Big(\gamma^{k-1} + t_{k-1}\Phi^T(y - \Phi\gamma^{k-1})\Big), \tag{9}$$

where $\eta_{\lambda/L}(\cdot)$ is the shrinkage threshold function, which is defined as $\eta_\theta(x) = sign(x)$ $\max\{|x| - \theta, 0\}$. A fixed step size $t_k = 1/L$ is generally taken, L is the maximum singular value of $\Phi^T\Phi$, and λ is a parameter. The general ISTA parameters are set manually, the convergence speed is slow, and different parameter settings have a great impact on the convergence speed. With the development of deep learning, using the self-learning characteristics of deep learning to train the parameter values of the ISTA can greatly improve the performance of the algorithm. LISTA is the first deep unfolding network algorithm, which is based on the unfolding of the ISTA. The ISTA is similar to the structure of the recurrent neural network, so the k layer LISTA network can be regarded as a k Layer Forward Feedback Network, and each layer of its network structure is expressed as follows:

$$\gamma^k = \eta_{\theta^k}\left(W_1^k y + W_2^k \gamma^{k-1}\right), \tag{10}$$

where the initial values $\theta^k = \lambda/L$, $W_1^k = (1/L)\Phi^T$, $W_2^k = I - (1/L)\Phi^T\Phi$, W_1^k and W_2^k are the weight matrices of the network, and the parameters to be learned in the LISTA network are $\left\{\theta^k, W_1^k, W_2^k\right\}$. The number of parameters of LISTA is large, the network is complex, and the training is time-consuming. After further research, the LISTA-CPSS network is obtained, which reduces the complexity of the network, while utilizing the information of the measurement matrix and its network structure is:

$$\gamma^k = \eta_{\theta^k}\left(\gamma^{k-1} - W^k(\Phi\gamma^{k-1} - y)\right), \tag{11}$$

where W^k is the weight matrix of the network and the parameters to be learned by the network are $\left\{\theta^k, W^k\right\}$. The parameters of LISTA-CPSS are significantly reduced from LISTA, and the training speed is accelerated. Further, sharing the parameters of each layer of the network can continue to reduce the number of parameters, which is the TiLISTA network, structured as:

$$\gamma^k = \eta_{\theta^k}\left(\gamma^{k-1} - \alpha^{k-1}W(\Phi\gamma^{k-1} - y)\right), \tag{12}$$

where the network parameters to be trained are $\left\{\theta^k, \alpha^k, W\right\}$, and the weight matrices of each layer are shared. It can then be shown that the weight matrix parameters in the network can be obtained by solving an optimization problem independent of the training data without the need for network training, which yields the ALISTA network with the network structure:

$$\gamma^k = \eta_{\theta^k}\left(\gamma^{k-1} - \alpha^{k-1}\overline{W}(\Phi\gamma^{k-1} - y)\right), \tag{13}$$

where $\overline{W}$ can be pre-calculated by solving the optimization, and the network parameters to be trained are $\left\{\theta^k, \alpha^k\right\}$. From LISTA to ALISTA, the network structure parameters become reduced, and the training complexity becomes lower and lower.

Li et al. combined the extragradient method with LISTA and proposed the ELISTA, which has the following iterative format for the $k+1$th layer of the network:

$$\begin{array}{l} \gamma^{k+1/2} = \eta_{\theta_1^k}(\gamma^k - \alpha_1^k W(\Phi\gamma^k - y)) \\ \gamma^{k+1} = \eta_{\theta_2^k}(\gamma^k - \alpha_2^k W(\Phi\gamma^{k+1/2} - y)) \end{array}, \tag{14}$$

The training parameters of the ELISTA are $\left\{\theta_1^k, \theta_2^k, \alpha_1^k, \alpha_2^k, W\right\}$, where W is a shared parameter. It is clear from Figure 2 that the network structure of the ELISTA is similar to the residual network structure.

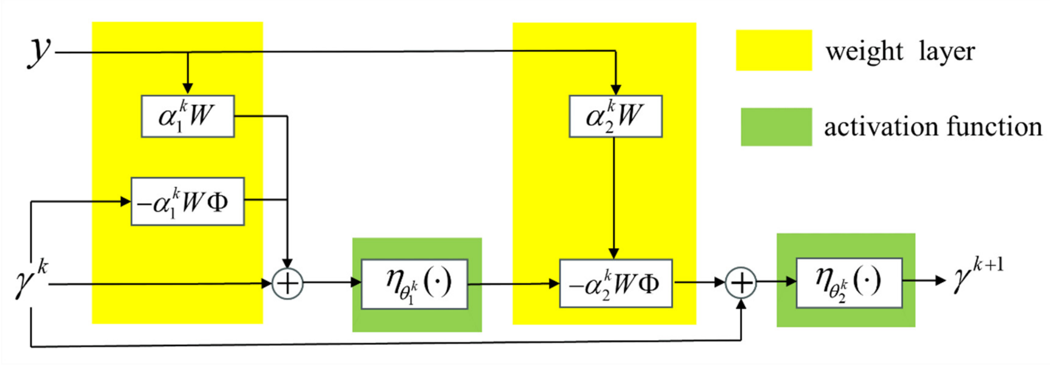


Figure 2. ELISTA network structure [19].

The network structure of ELISTA introduces the residual structure of deep learning and still maintains the interpretability of the deep unfolding network algorithm since it is introduced based on the extragradient algorithm. However, its training parameters come with matrix parameters, and the training efficiency is low. Therefore, considering the lightness of the ALISTA network, we combine ELISTA with ALISTA to obtain the EALISTA with the advantages of both.

3.2. EALISTA Format

According to [10], the matrix parameter $\overline{W}$ in the ALISTA can be computed in advance without network training to obtain it; inspired by this, the matrix parameter in the ELISTA network is also computed in advance without changing the rest of the network, so that the EALISTA can be obtained, and the iterative format of the network at the $k+1$th layer is:

$$\begin{aligned} \gamma^{k+1/2} &= \eta_{\theta_1^k}(\gamma^k - \alpha_1^k \overline{W}(\Phi\gamma^k - y)) \\ \gamma^{k+1} &= \eta_{\theta_2^k}(\gamma^k - \alpha_2^k \overline{W}(\Phi\gamma^{k+1/2} - y))' \end{aligned} \tag{15}$$

where the $\overline{W}$ matrix is computed as in [8], and the training parameter of the EALISTA is $\left\{\theta_1^k, \theta_2^k, \alpha_1^k, \alpha_2^k\right\}$. Figure 3 is a one-layer network structure of EALISTA, which shows that the network structure of EALISTA is similar to the network structure of ELISTA.

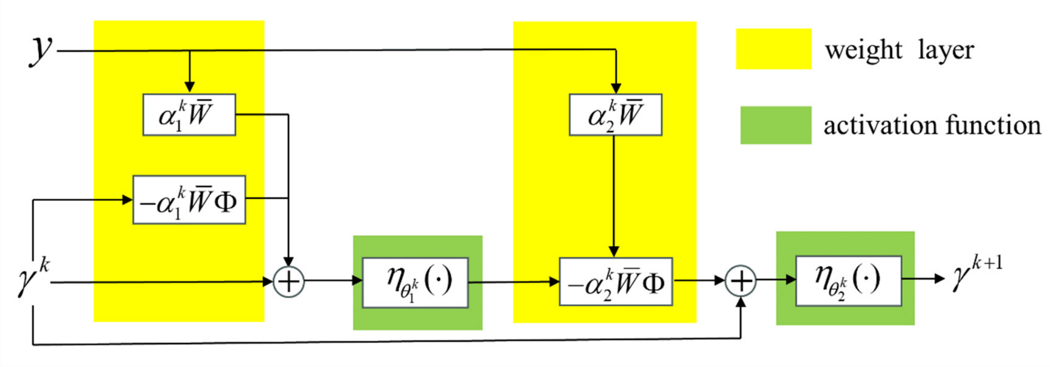

y
$\alpha_1^k \overline{W}$
$-\alpha_1^k \overline{W}\Phi$
γ^k
$\eta_{\theta_1^k}(\cdot)$
$\alpha_2^k \overline{W}$
$-\alpha_2^k \overline{W}\Phi$
$\eta_{\theta_2^k}(\cdot)$
γ^{k+1}
weight layer
activation function

Figure 3. EALISTA network structure.

The EALISTA has a superior residual-like network structure compared to the ALISTA, which in turn has a higher training efficiency compared to the ELISTA.

3.3. Convergence Analysis of the EALISTA

Assumption 1 (Base Assumption) [9]. *Assume that the vector γ^* is sampled from the following set with noise ε:*

$$\gamma^* \in X(B,s) \triangleq \{\gamma^* | \, |\gamma_i^*| \leq B, \forall i, \ \| \gamma_i^* \|_0 \leq s\}. \tag{16}$$

That is, γ^* is bounded and s-sparse ($s \geq 2$). Further, we assume that $\varepsilon = 0$.

Definition 1 ([10]). *Given a matrix $\Phi \in \mathbb{R}^{m\times n}$, its generalized mutual consistency is defined as follows:*

$$\mu(\Phi) = \inf_{W\in\mathbb{R}^{n\times m},\ (W_{:\ ,i}\ )^T\Phi_{:\ ,\ i}=1,\ \forall i} \left\{ \max_{i\neq j,\ 1\leq i,j\leq n} (W_{:\ ,i}\ )^T\Phi_{:,\ j} \right\}. \tag{17}$$

Let $\mathcal{W}(\Phi)$ be the set of matrices satisfying the generalized mutual consistency of $\mu(\Phi)$, that is,

$$\mathcal{W}(\Phi) = \left\{ W| \max_{i\neq j,\ 1\leq i,j\leq n} (W_{:\ ,i}\ )^T\Phi_{:,\ j} = \mu(\Phi),\ (W_{:\ ,i}\ )^T\Phi_{:\ ,\ i} = 1,\ \forall i \right\}. \tag{18}$$

Theorem 1 ([19]). *If Assumption 1 holds and a suitable parameter matrix W is chosen to satisfy $W \in \mathcal{W}(\Phi)$, the following equation holds:*

$$\begin{array}{l} \theta_1^k = \alpha_1^k\omega_{k+1/2}(t_{k+1/2}\Big|\Theta)\mu(\Phi)\mathrm{sup}_{\gamma^*} \parallel \gamma^k - \gamma^* \parallel_1 \\ \theta_2^k = \alpha_2^k\omega_{k+1}(t_{k+1}\Big|\Theta)\mu(\Phi)\mathrm{sup}_{\gamma^*} \parallel \gamma^{k+1/2} - \gamma^* \parallel_1 \end{array}, \tag{19}$$

If $\alpha_1^k, \alpha_1^k \in (0, 2/(1 + (2s-1)\mu(\Phi)))$, s is small enough, then for the sequence generated by the ELISTA, there are "false positive" $t_{k+1/2}$, t_{k+1}, which satisfy $0 < t_{k+1}, t_{k+1/2} < s$, and

$$\parallel \gamma^k - \gamma^* \parallel_2 \leq sB\exp(\sum_{i=1}^{k} c_i^*) < sB\exp(ct), \tag{20}$$

where $c_i^* < 0, c = \max_{i=1,2,\ldots,k}\{c_i^*\} < 0, \Theta$ is a model parameter. $\omega_{k+1/2}(t_{k+1/2}|\Theta)$ and $\omega_{k+1}(t_{k+1}|\Theta)$ describe the relationship between Θ and the rate of false positives.

Refs. [12,18] describe the definitions of "false positive", Θ, c_i^*, $\omega_{k+1/2}(t_{k+1/2}|\Theta)$ and $\omega_{k+1}(t_{k+1}|\Theta)$ in detail. Theorem 1 shows that the ELISTA can achieve a linear convergence rate.

Theorem 2. *EALISTA can reach linear convergence.*

Proof of Theorem 2. The only difference between the EALISTA and the ELISTA format is that the parameter matrices in the EALISTA are computed by solving the optimization problem and are not the network parameters, whereas in the ELISTA, they are the learnable matrix parameters. According to Theorem 1, the ELISTA reaches linear convergence whenever $W \in \mathcal{W}(\Phi)$ holds. Similarly, for the EALISTA, as long as $\overline{W} \in \mathcal{W}(\Phi)$ holds, then the EALISTA reaches linear convergence. According to the definition of $\overline{W}$ in [10], $\overline{W} \in \mathcal{W}(\Phi)$ holds, so the EALISTA can reach linear convergence. □

4. FALISTA Format

ALISTA is obtained by unfolding the ISTA, which only achieves sub-linear convergence with a convergence speed of $\mathcal{O}(1/k)$, which affects the convergence speed of ALISTA. In order to improve the convergence speed of the first-order optimization algorithm, Nesterov proposed three acceleration algorithms in 1983 [20], 1988 [21], and 2005 [22], and the convergence speed can reach $\mathcal{O}(1/k^2)$. These three acceleration algorithms can be used on the ISTA. Beck and Teboulle in 2009 used the acceleration algorithm proposed by Nesterov in 1983 on ISTA to obtain the FISTA [23]. An iterative format of FISTA is as follows:

$$\begin{array}{l} d^k = \gamma^{k-1} + \frac{k-2}{k+1}(\gamma^{k-1} - \gamma^{k-2}) \\ \gamma^k = \eta_{\theta^k}(d^k - \beta^k\Phi^T(\Phi d^k - y)) \end{array}. \tag{21}$$

If the chosen step size satisfies $\beta^k \leq 1/L$, then the convergence rate can reach $\mathcal{O}(1/k^2)$. The gradient descent part can be obtained by substituting d^k:

$$\begin{array}{l} \gamma^{k-1} + \frac{k-2}{k+1}(\gamma^{k-1} - \gamma^{k-2}) - \beta^k \Phi^T(\Phi(\gamma^{k-1} + \frac{k-2}{k+1}(\gamma^{k-1} - \gamma^{k-2})) - y) \\ = \gamma^{k-1} + \frac{k-2}{k+1}(\gamma^{k-1} - \gamma^{k-2}) - \beta^k \Phi^T(\Phi\gamma^{k-1} - y) - \beta^k \Phi^T \Phi \frac{k-2}{k+1}(\gamma^{k-1} - \gamma^{k-2}) \end{array}. \quad (22)$$

Observing the above equation, we find that the equation can be divided into two parts; the first part keeps the original format of ISTA in order to be combined with the ALISTA network. Reference [15] added the Polyak ball momentum acceleration format directly after the ALISTA format, and inspired by this, the second part can be used as the acceleration format similar to [15]; adjusting the order of the equation can obtain:

$$\begin{array}{l} \gamma^{k-1} + \frac{k-2}{k+1}(\gamma^{k-1} - \gamma^{k-2}) - \beta^k \Phi^T(\Phi\gamma^{k-1} - y) - \beta^k \Phi^T \Phi \frac{k-2}{k+1}(\gamma^{k-1} - \gamma^{k-2}) \\ = \gamma^{k-1} - \beta^k \Phi^T(\Phi\gamma^{k-1} - y) + \frac{k-2}{k+1}(I - \beta^k \Phi^T \Phi)(\gamma^{k-1} - \gamma^{k-2}) \end{array}, \quad (23)$$

In order to reduce the network parameters that need to be trained, the network matrix parameters of the first part are calculated in advance like ALISTA; at the same time, in order to ensure that the convergence speed reaches more than $\mathcal{O}(1/k^2)$ as much as possible, $\frac{k-2}{k+1}$ is fixed as a constant not to be involved in the training, and so the FALISTA is obtained, and the iterative format of the kth layer of the network is as follows:

$$\gamma^k = \eta_{\theta^k}(\gamma^{k-1} - \alpha^{k-1}\overline{W}(\Phi\gamma^{k-1} - y) + \frac{k-2}{k+1}G^k(\gamma^{k-1} - \gamma^{k-2})). \quad (24)$$

The network training parameters are $\left\{\theta^k, \alpha^k, G^k\right\}$, where the initial values $\theta^k = \lambda/L$, $\alpha^k = 1/L$, $G^k = I - \beta^k \Phi^T \Phi$, α^k of FALISTA has the same definition as α^k of ALISTA. The FALISTA both introduces the Nesterov acceleration format, which accelerates the speed of convergence, and combines the ALISTA in advance by solving the optimization problem to calculate the matrix parameters, which reduces the training parameters to a certain extent.

5. Results

5.1. Simulation Experiment Results

We first verify the effectiveness of the two algorithms proposed in this study on a simulation dataset by applying the EO-College tutorial [24] to generate the measurement matrix as well as the observation data, with the number of scattering points and their positions in the height direction set in advance as the real scattering coefficient distribution. The wave number maps used in the simulation experiments in this study are taken from the test dataset provided in the EQ-College tutorial. This dataset is a simple tutorial, where the wave number information used is given directly in the equivalent wave number matrix without the radar parameters involved in the process for ease of learning. An altitude interval between −70 and 70 was chosen for the calculation of the measurement matrix under such wave number conditions. Nine single-view complex images were selected as observation data. Due to the lack of a real training dataset, the sparse training dataset used is given by a Bernoulli distribution with the sparsity fixed in advance [25]. The training dataset is 50,000, the test dataset is 10,000, and the batch size for training is 100. The network model is recorded when the loss in the test set reaches its minimum, and the results are predicted using this model. Considering the characteristics of the TomoSAR building object height inversion model, the MSE loss function and the loss function SL1 based on the L1 norm are chosen as the loss functions for the network training, in order to achieve a better inversion effect. The MSE loss function is as follows:

$$MSELoss = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \|X_{predict}^{(i)} - X_{Label}^{(i)}\|_2^2, \quad (25)$$

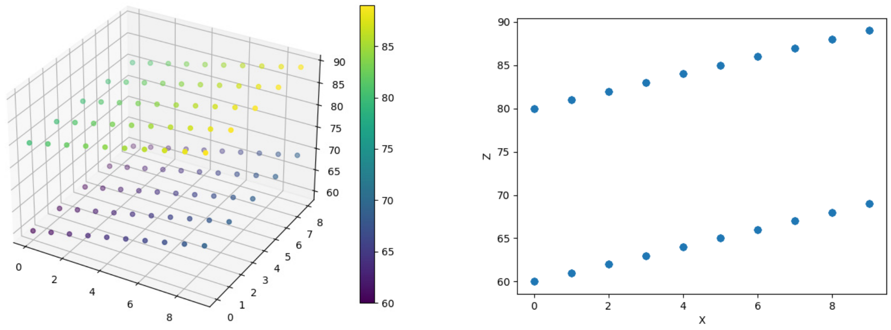
The SL1 loss function is as follows:

$$SL1Loss = \frac{1}{N_{batch}} \sum_{i=1}^{N_{batch}} \|X_{predict}^{(i)}\|_1, \tag{26}$$

where N_{batch} denotes the number of samples in a training batch, $X_{predict}^{(i)}$ denotes the network output, and $X_{Label}^{(i)}$ denotes the training labels.

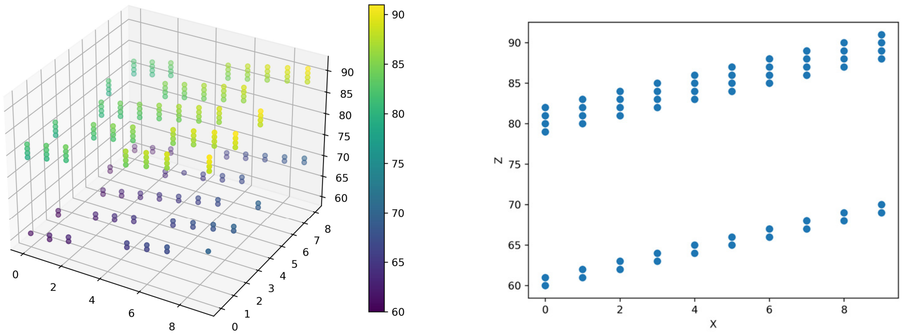
Using the simulation data, the model is trained, and the inversion results are obtained to compare with the predefined true-value results to verify the effectiveness of the algorithm. According to the general experience, the number of network layers in the deep unfolding network algorithm is 10–20 layers when the algorithm can converge [26]. Therefore, we set the number of network layers to 10 for both algorithms. Considering the characteristics of building scattering coefficients, we set the sparsity of the training dataset to 4 and 5 to train the network and obtain the results as follows.

Figure 4 represents the three-dimensional positions of scattering points that we set in advance using the EQ-College tutorial, with two scattering points set in the height direction, and the intervals of the scattering points are 60 to 70 and 80 to 90, respectively. Figures 5–8 show the three-dimensional positions of scattering points for the two algorithms that we proposed for the inversion of our algorithms when the training sparsity of the training dataset is set to 4 and 5, respectively. For the EALISTA, Figure 5 shows that when the sparsity of the training dataset is 4, the inversion is not good enough, and some scattering points are missing; Figure 6 indicates that the algorithm is able to accurately invert the true positions of the scattering points when the training sparsity is 5. For the FALISTA, Figures 7 and 8 show that the training dataset sparsity is either 4 or 5, and the algorithm is able to accurately invert the true positions of the scattering points.



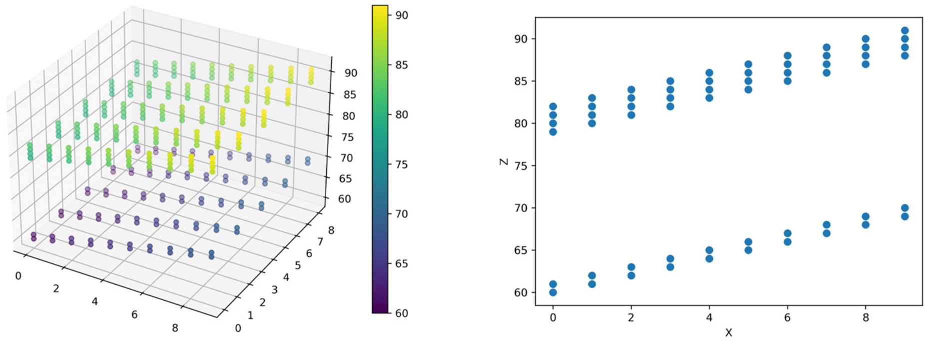
(**a**) (**b**)

Figure 4. (**a**) Three-dimensional map of the true location of the scattering point; (**b**) side view of the true position of the scattering point.



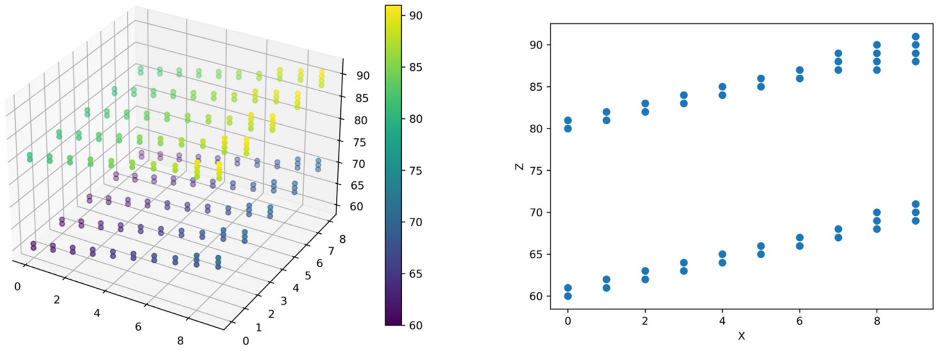
(**a**) (**b**)

Figure 5. (**a**) Three-dimensional map of the scattering point inversion position of the EALISTA for a sparsity of 4; (**b**) side view of the scattering point inversion position of the EALISTA for a sparsity of 4.



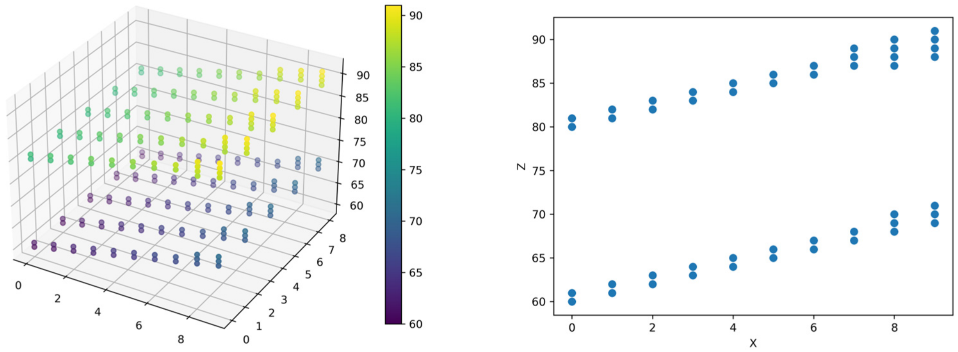
(**a**) (**b**)

Figure 6. (**a**) Three-dimensional map of the scattering point inversion position of the EALISTA for a sparsity of 5; (**b**) side view of the scattering point inversion position of the EALISTA for a sparsity of 5.



(**a**) (**b**)

Figure 7. (**a**) Three-dimensional map of the scattering point inversion position of the FALISTA for a sparsity of 4; (**b**) side view of the scattering point inversion position of the FALISTA for a sparsity of 4.



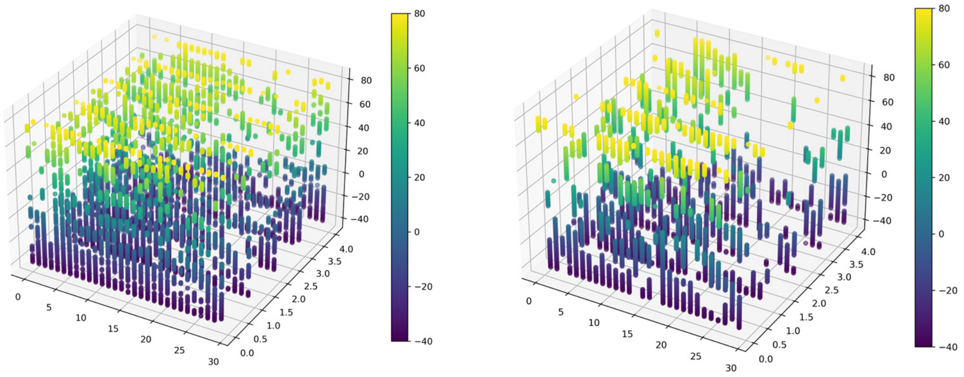
(**a**) (**b**)

Figure 8. (**a**) Three-dimensional map of the scattering point inversion position of the FALISTA for a sparsity of 5; (**b**) side view of the scattering point inversion position of the FALISTA for a sparsity of 5.

5.2. Experimental Results of Measured Data

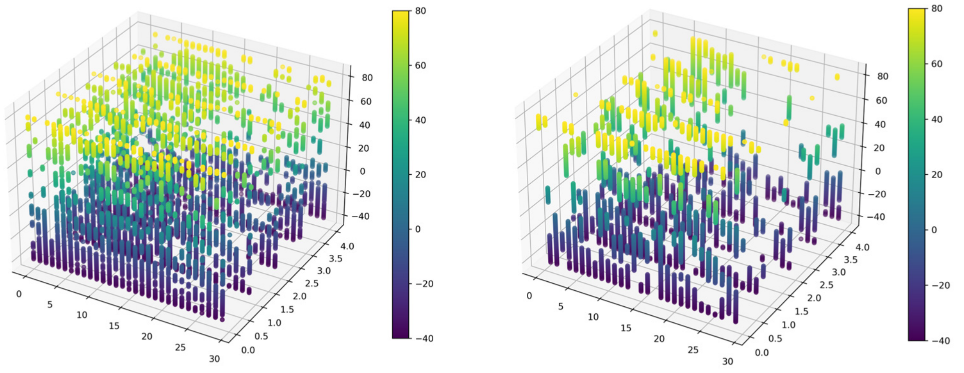
We use the urban building data collected by the Chinese Academy of Sciences (CAS) in the Yuncheng area of China in 2015 to perform the real TomoSAR building object height inversion [27]. The number of channels for this data is eight, the bandwidth is 500 MHZ, the image size is 3100 × 1200, and the polarization information is HH. We intercepted a range size of 30 × 5 on the image for inversion and set the inversion height interval from −40 to 80. The loss function used for network training is the same as the simulation experiment described above. In order to study the inversion effect under restricted observation data, eight, six, five, four, and three single-view complex images are randomly selected as observation data for the object height inversion of buildings, respectively. The EALISTA and FALISTA proposed in this study are used to conduct the experiments and compared with the results of the inversion of the ALISTA and the ALISTA-Polyak [16]. The dataset used for training is the same as that for simulation experiments, which is also a sparse dataset, obeying the Bernoulli distribution. In the experiments, we set the sparsity of the training dataset to 4 and 2, respectively, to train the network and obtain the inversion results, and some of the results are shown below:

Figures 9–12 show the scattering point inversion results of the four algorithms under real data when the training dataset is 4. We intercepted the plane size of 30 × 5 and set the inversion height range from −40 to 80. The (a) of each image represents the scattering point location inverted when the observed data are eight images, and the (b) of each image represents the scattering point location inverted when the observed data are four images.



(**a**) (**b**)

Figure 9. (**a**) ALISTA: inversion results under training dataset sparsity of 4 and 8 single-view complex images; (**b**) ALISTA: inversion results under training dataset sparsity of 4 and 4 single-view complex images.



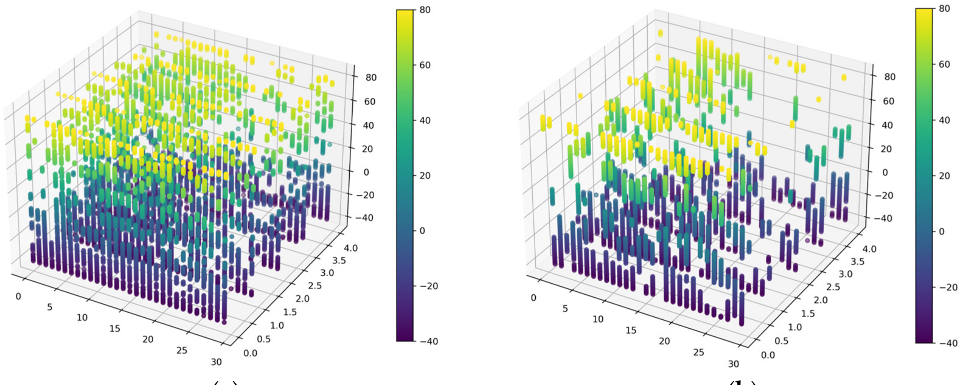
(**a**) (**b**)

Figure 10. (**a**) ALISTA-Polyak: inversion results under training dataset sparsity of 4 and 8 single-view complex images; (**b**) ALISTA-Polyak: inversion results under training dataset sparsity of 4 and 4 single-view complex images.

We use the deviation P to measure the algorithm's inversion performance under restricted observation data. The index P reflects the robustness of the algorithm's inversion ability as the observation data decrease, and the smaller P represents the better performance of the algorithm. The inversion results of eight images in each case are used as the true value of the distribution of scattering points, and the deviation is calculated for the inversion results of six, five, four, and three images. The expression for the deviation P is:

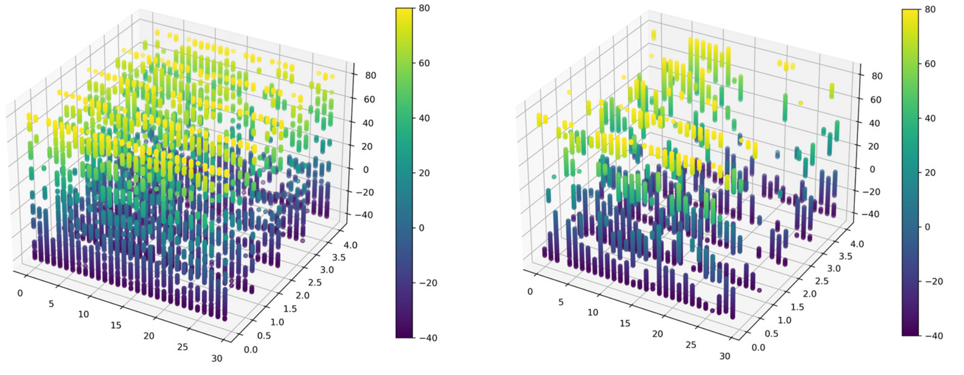
$$P = (\sum_{l=1}^{l} \frac{|h_p - h_r|}{h_r})/n, \tag{27}$$

where h_p and h_r are the average heights at point l when both contain scattering points at a pixel location l. The summation process traverses all n common pixel locations. P value reflects the magnitude of the deviation in the imaging results from different observations. P value metrics are shown in the table below.



(**a**) (**b**)

Figure 11. (**a**) EALISTA: inversion results under training dataset sparsity of 4 and 8 single-view complex images; (**b**) EALISTA: inversion results under training dataset sparsity of 4 and 4 single-view complex images.



(**a**) (**b**)

Figure 12. (**a**) FALISTA: inversion results under training dataset sparsity of 4 and 8 single-view complex images; (**b**) FALISTA: inversion results under training dataset sparsity of 4 and 4 single-view complex images.

The columns of Tables 1 and 2 indicate the number of observed data images used for inversion, which are six images, five images, four images, and three images, respectively. The P values are calculated by substituting (27) with the results obtained from eight images for inversion as the true value of the scattering point location, and the results obtained from six, five, four, and three images for inversion as the contrast value, respectively. Tables 1 and 2 represent the P values when the training dataset sparsity is 4 and 2, respectively, and the observations used for inversion are changed to six images, five images, four images, and three images, respectively.

In addition, the value of sparsity in the training dataset also affects the algorithm inversion performance, and we measure the degree of influence of the sparsity value on the algorithm's inversion performance by the mean value $\overline{H}$. The expression for $\overline{H}$ is:

$$\overline{H} = \sum_{i=3}^{6} \frac{|P_2^i - P_4^i|}{4}, \tag{28}$$

where P_2^i denotes the value of P when the observed data are i images and the sparsity of the training dataset is 2. P_4^i is defined similarly. The values of each algorithm are listed in the table below.

Table 1. P values of different algorithms for training dataset sparsity of 4.

	6	**5**	**4**	**3**
ALISTA	0.0577	0.2772	0.3934	0.4471
EALISTA	0.0509	0.2426	0.3488	0.4302
ALISTA-Polyak	0.0511	0.2470	0.3594	0.4241
FALISTA	0.0536	0.2482	0.3287	0.4159

Table 2. P values of different algorithms for training dataset sparsity of 2.

	6	**5**	**4**	**3**
ALISTA	0.0594	0.3010	0.4540	0.5259
EALISTA	0.0520	0.2438	0.3496	0.4316
ALISTA-Polyak	0.0513	0.2460	0.3576	0.4126
FALISTA	0.0550	0.2558	0.3237	0.4088

Table 3 shows the $\overline{H}$ values of different algorithms, the $\overline{H}$ values reflect the extent to which the inversion performance of the algorithms under the restriction of observation data is affected by the change in the sparsity value of the training dataset, and smaller $\overline{H}$ values indicate that the algorithms are less affected by the change in the sparsity value of the training dataset.

Table 3. $\overline{H}$ values for different algorithms.

	ALISTA	**EALISTA**	**ALISTA-Polyak**	**FALISTA**
$\overline{H}$	0.0412	0.0011	0.0036	0.0052

6. Discussion

Through the simulation numerical experiments, we can see that both of our proposed algorithms are effective and are able to invert the positions of the scattering points, and although a small number of virtual points appear, the virtual points are surrounding the real points. It can also be found that the effect of inversion is closely related to the sparsity setting of the training dataset, which reveals that we have to choose the appropriate sparsity according to the characteristics of the real scattering scene.

In the real data inversion experiments, we can see that both the EALISTA and the FALISTA perform significantly better than the ALISTA when the sparsity of the training set is 2 and 4. The EALISTA outperforms the ALISTA-Polyak when the observations are five and four images, but the performance is not as good as the ALISTA-Polyak at three images. Whereas the FALISTA does not perform as well as the EALISTA and the ALISTA-Polyak algorithm when the observation data are five images, it outperforms both algorithms at three and four images. In addition, we use the strategy of fixing the sparsity of the training dataset to train the network, and the value of sparsity is also an important factor that affects the performance of the algorithm. We can see that the EALISTA is least affected by the change in sparsity value, followed by ALISTA-Polyak and FALISTA, and the most affected is the ALISTA, which also shows that our proposed new algorithms are superior from this perspective. Overall, both of our proposed algorithms perform better than ALISTA and ALISTA-Polyak, and the superiority of EALISTA is mainly reflected in the time when the data restriction is not so serious, while FALISTA is superior in the case of a more serious data restriction.

In engineering applications, in general, the observation data used for inversion are constrained, but the degree of this constraint will vary according to the real situation.

According to our experimental results on real data, it may be more robust to choose the EALISTA for inversion when the data limitation is not so severe, while it is more robust to choose the FALISTA for inversion when the observation data are extremely limited. The choice of training dataset sparsity is related to prior knowledge of the scattering scenario. In practical applications, if we have sufficient a priori knowledge of the scattering scene, we can choose the most suitable sparsity and then choose the most suitable inversion algorithm according to the degree of limitation of the observation data; if we lack the physical a priori knowledge of the scattering scene, it may be more suitable to directly use the EALISTA as the inversion algorithm.

7. Conclusions

On the theoretical side, in this study, the ALISTA is improved in two dimensions: the method of connecting between network layers and the iteration format within network layers. The EALISTA and the FALISTA are obtained, and proof of convergence for the EALISTA is given. In the engineering application, according to the lack of a training dataset, the simulation dataset obeying Bernoulli distribution is applied to train the network, and the training strategy of fixed sparsity is proposed. According to the experimental results of real data, we analyze how to choose the most suitable algorithm according to the actual situation in practical applications. The research in this study has both theoretical and engineering significance. However, in this study, we did not conduct a convergence analysis of FALISTA, which combines the advantages of ALISTA and FISTA, and it may converge faster than both ALISTA and FISTA, so proving the convergence of FALISTA is the next step in research. In addition, in this study, the ALISTA is improved in two dimensions, which are the inter-layer connection method of the network and the intra-layer iteration format of the network, respectively, and the EALISTA and the FALISTA are obtained. Inspired by this, we can consider unifying these two dimensions and improving both the inter-layer connection method of the network and the intra-layer iteration format of the network, so that we can possibly obtain an algorithm with better performance, which is also a meaningful research direction.

Author Contributions: Conceptualization, W.L. and J.L.; methodology, W.L. and J.L.; validation, W.L.; formal analysis, W.L. and J.L.; writing—original draft preparation, W.L.; writing—review and editing, J.L. and J.Z.; supervision, J.L. and J.Z.; project administration, J.L. and J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China, No. 2020YFA0713504.

Data Availability Statement: The data in this article are all publicly available, and the sources of the data are stated in the article.

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

References

1. Zhu, X.X.; Wang, Y.; Montazeri, S. A review of ten-year advances of multi-baseline SAR interferometry using TerraSAR-X data. *Remote Sens.* **2018**, *10*, 1374. [CrossRef]
2. Zhu, X.X.; Bamler, R. Very high resolution spaceborne SAR tomography in urban environment. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4296–4308. [CrossRef]
3. Aguilera, E.; Nannini, M.; Reigber, A. Multisignal compressed sensing for polarimetric SAR tomography. *IEEE Geosci. Remote Sens. Lett.* **2012**, *9*, 871–875. [CrossRef]
4. Budillon, A.; Johnsy, A.C.; Schirinzi, G. Sar tomography based on deep learning. In Proceedings of the International Geoscience and Remote Sensing Symposium 2019, Yokohama, Japan, 28 July–2 August 2019; pp. 3625–3628.
5. Rong, S.; Shunjun, W.; Yanbo, W.; Jun, S.; Xiaoling, Z. A 3-D Imaging Method of Building with Tomosar Based on DUADMM-Net. In Proceedings of the 2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 16–21 July 2023; pp. 7933–7936.
6. Ren, Y.; Zhang, X.; Zhan, X.; Shi, J.; Wei, S.; Zeng, T. A Model-data-driven Network Embedding Multidimensional Features for Tomographic SAR Imaging. In Proceedings of the 2023 IEEE Radar Conference, San Antonio, TX, USA, 1–5 May 2023; pp. 1–6.
7. Li, J.; Xu, Z.; Li, Z.; Zhang, Z.; Zhang, B.; Wu, Y. An Unsupervised CNN-Based Multichannel Interferometric Phase Denoising Method Applied to TomoSAR Imaging. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2023**, *16*, 3784–3796. [CrossRef]

8. Gregor, K.; LeCun, Y. Learning fast approximations of sparse coding. In Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 399–406.
9. Chen, X.; Liu, J.; Wang, Z. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 3–8 December 2018; pp. 9079–9089.
10. Liu, J.; Chen, X.; Wang, Z. ALISTA: Analytic weights are as good as learned weights in LISTA. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019; pp. 1–13.
11. Ablin, P.; Moreau, T.; Massias, M. Learning step sizes for unfolded sparse coding. In Proceedings of the Conference and Workshop on Neural Information Processing Systems 2019, Vancouver, BC, Canada, 8–14 December 2019; pp. 13100–13110.
12. Wu, K.; Guo, Y.; Li, Z. Sparse Coding with Gated Learned ISTA. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–12.
13. Aberdam, A.; Golts, A.; Elad, M. Ada-LISTA: Learned solvers adaptive to varying models. *arXiv* **2020**, arXiv:2001.08456. [CrossRef] [PubMed]
14. Kim, D.; Park, D. Element-wise adaptive thresholds for learned iterative shrinkage thresholding algorithms. *IEEE Access* **2020**, *8*, 45874–45886. [CrossRef]
15. Chen, X.; Liu, J.; Wang, Z. Hyperparameter tuning is all you need for LISTA. *arXiv* **2021**, arXiv:2110.15900.
16. Wang, M.; Zhang, Z.; Wang, Y.; Gao, S. TomoSAR-ALISTA: Efficient TomoSAR imaging via deep unfolded network. In Proceedings of the 2022 International Conference on Radar Systems, Edinburgh, UK, 25–27 October 2022; pp. 528–533.
17. Wang, M.; Zhang, Z.; Qiu, X. ATASI-Net: An Efficient Sparse Reconstruction Network for Tomographic SAR Imaging with Adaptive Threshold. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4701918. [CrossRef]
18. Li, Y.; Kong, L.; Shang, F. Learned extragradient ISTA with interpretable residual structures for sparse coding. In Proceedings of the 35th AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; pp. 8501–8509.
19. Kong, L.; Sun, W.; Shang, F. Learned Interpretable Residual Extragradient ISTA for Sparse Coding. *arXiv* **2021**, arXiv:2106.11970.
20. Nesterov, Y. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Sov. Math. Dokl.* **1983**, *27*, 372–376.
21. Nesterov, Y. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekon. I Mateaticheskie Metod.* **1988**, *24*, 509–517.
22. Nesterov, Y. Smooth minimization of non-smooth functions. *Math. Program.* **2005**, *103*, 127–152. [CrossRef]
23. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
24. EO-College Tomography Tutorial. Available online: https://github.com/johntruckenbrodt/tomography.git (accessed on 1 April 2022).
25. Freya, B.; Jonathan, S.; Peter, J. Neurally Augmented ALISTA. *arXiv* **2020**, arXiv:2010.01930.
26. Zheng, Z.; Dai, W. Hybrid ISTA: Unfolding ISTA with Convergence Guarantees Using Free-Form Deep Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3226–3244. [CrossRef] [PubMed]
27. Qiu, X.; Jiao, Z.; Peng, L. SARMV3D-1.0: Synthetic aperture radar microwave vision 3D imaging dataset. *J. Radars* **2021**, *10*, 485–498.

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.