Design of Smoothed L0 UnfoldingNet Architecture for OFDM Channel Estimation

Zhu Lin Tao
School of Electronics & Information
Soochow University
Suzhou, China
693489137@qq.com

Jia Hou[1,2]
[1] Yangtze Delta Region Institute (Quzhou), University of Electronic Science and Technology of China
[2] School of Electronics & Information,Soochow University
Suzhou, China
houjia@suda.edu.cn

***Abstract*—Recently, the sparse recovery (SR) techniques could be used to achieve better channel estimates with fewer pilot resources in the OFDM system. To improve the priori information of SR techniques, in this paper, the smoothed L0 UnfoldingNet (SL0-UNet) architecture is designed. In the proposed architecture, two different learning methods are presented to enhance the recovery performance for channel estimation. The results show that both two proposed learning methods could achieve more robust performance with limited complexity increased compared to the traditional SR technique.**

Keywords—Compressed Sensing, Channel Estimation, Deep learning, Sparse Recovery, OFDM

I. Introduction

Sparse recovery (SR) techniques, such as Matching Pursuit (MP) [1] and Orthogonal Matching Pursuit (OMP) [2], has been studied in recent years, however, those algorithms are all rely on the priori knowledge of the channel model. It has been shown that a small uncertainty on system parameters for priori information could lead to higher estimation performance losses [3]. Otherwise, to reduce the computation complexity, the smoothed L0 norm algorithm (SL0) was introduced in [4]. But the key for SL0 algorithm to obtain better estimation performance is also based on the initialization and priori information. Further, combined the deep neural network (DNN) [5] with the SR techniques for OFDM channel estimation were proposed. In this approach the iterations can be unfolded as DNN, where the optimized initial parameters and priori information of the SR techniques are obtained from DNN training. To reduce the computations complexity, in this paper, a SL0-UnfoldingNet (SL0-UNet) is considered, where two different learning methods are designed. One is from the traditional SL0 initialization with lower complexity, the other is from the training results of each layer within DNN with better performance in the case of lower signal to noise ratio (SNR).

II. System Model

The channel is expressed as a superposition of L Plane waves arriving at the antenna array. After sampling with period T, the discrete time equivalent channel is obtained as:

$$\mathbf{h}_0[n]=\sum_{i=0}^{L-1} a_i e^{j\phi_i}\delta(nT-\tau_i) \tag{1}$$

Each plane is characterized by a per path attenuation a_i, delay τ_i, and phase shift ϕ_i. If $\tau_0=0$ and $\tau_i>\tau_{i-1}$, then the delay spread is τ_L. The length of the equivalent channel $k=[\tau_L/T]$, where $\mathbf{h}_0[n]$ is a $k\times 1$ column vector. Then the received signal after cyclic prefix removal is:

$$\mathbf{y}=\mathbf{x}*\mathbf{h}+\mathbf{w} \tag{2}$$

where x is a length $N\gg k$ OFDM symbol, y, x and w are column vectors, $\mathbf{h}$ is padded with $N-k$ zeros and the noise is $w_m\sim CN(0,\sigma)\,\forall m=1,...,N$. Also, (2) can be expressed in matrix-vector form $\mathbf{y}=\mathbf{X}_c*\mathbf{h}+\mathbf{w}$ where $\mathbf{X}_c=\mathbf{F}_N\mathrm{D}(\mathbf{x})\mathbf{F}_N^H$. $\mathbf{F}_N$ is the size N discrete Fourier transform (DFT) matrix, and $\mathrm{D}(\bullet)$ is a diagonalization matrix which place it's argument on the diagonal of a square matrix. For an OFDM symbol, the set $\mathbb{C}=\{1,2,...,N\}$ of sub-carriers with a subset $\mathbb{Q}_P\subset\mathbb{C}$ reserved for $|\mathbb{Q}_P|=P$ pilot symbols. The received pilot subcarriers are then expressed as:

$$\mathbf{y}_P=\mathbf{F}_{N,\mathbb{Q}_P}\mathbf{y}=\mathrm{I}_{\mathbb{Q}_P}\mathrm{D}(\mathbf{x})\mathbf{F}_N\mathbf{h}+\mathbf{F}_{N,\mathbb{Q}_P}\mathbf{w} \tag{3}$$

where $\mathbf{F}_{N,\mathbb{Q}_P}$ are the P rows of the size N DFT matrix indexed by the $\mathbb{Q}_P$ pilot subcarriers and where the matrix $\mathrm{I}_{\mathbb{Q}_P}$ consists of P rows of $N\times N$ identity matrix indexed by $\mathbb{Q}_P$. According to the compressed sensing theory, equation (3) is converted into the following form:

$$\mathbf{y}_P=\mathbf{T}\mathbf{h}+\mathbf{w}_P \tag{4}$$

where $\mathbf{T}\in\mathbb{C}^{P\times N}$ is the product of the basis matrix $\mathrm{I}_{\mathbb{Q}_P}\mathrm{D}(\mathbf{x})$ and the observed matrix $\mathbf{F}_N$. Since vector $\mathbf{h}\in\mathbb{C}^{N\times 1}$ is a k sparse vector and contains only a few non-zero elements in $\mathbf{h}$, the channel estimation problem can be transformed into a sparse recovery problem as follows:

$$\arg\min_{\mathbf{h}}\|\mathbf{h}\|_0,\ s.t.\|\mathbf{y}_P-\mathbf{T}\mathbf{h}\|_2\le\varepsilon \tag{5}$$

where ε represents the noise level, $\|\bullet\|_0,\|\bullet\|_2$ represent the L0 norm and L2 norm of the vector respectively.

SL0 uses a continuous function to approximate the L0 norm and minimizes the cost function formed by the smoothed function through the steepest descent method and gradient projection method. For (5), the following Gaussian function is considered as:

$$f_\sigma(x)=\exp\left(-|x|^2/2\sigma^2\right) \tag{6}$$

which satisfies:

$$\lim f_\sigma(x)=\begin{cases}1, & x=0\\ 0, & x\neq 0\end{cases} \tag{7}$$

Therefore, assuming $\mathbf{h}_l$ is the lth element of $\mathbf{h}$, define the following cost function:

$$F_\sigma(\mathbf{h}) = N - \sum_{l=0}^{L-1} f_\sigma(\mathbf{h}_l) \tag{8}$$

Then (5) can be converted to the following formula:

$$\arg\min_h \lim_{\sigma\to\infty} F_\sigma(\mathbf{h}), s.t. \|\mathbf{y}_P - \mathbf{Th}\|_2 \le \varepsilon \tag{9}$$

The value of σ determines how smooth the function $F_\sigma(\mathbf{h})$ is: the larger value of σ, the smoother $F_\sigma(\mathbf{h})$ (but worse approximation to L0 norm); and the smaller value of σ, the closer behavior of $F_\sigma(\mathbf{h})$ to L0 norm. σ is the approximation parameter of $F_\sigma(\mathbf{h})$. In order to obtain the optimal solution of (7), a decreasing sequence of approximation parameter is usually selected. The steps of SL0 is as follows:

- **Initialization**:
  1) Let $\mathbf{h}^{(0)}$ be equal to the minimum L2 norm solution of $\mathbf{y}_P = \mathbf{Th}$, obtained by pseudo-inverse of $\mathbf{T}$.
  2) Choose a suitable decreasing sequence for σ, μ.
- For $i = 1, \ldots, I$:
  1) Let $\sigma = \sigma_i$, $\mu = \mu_i$
  2) Maximize(approximately) the function $F_\sigma(\mathbf{h})$ on the feasible set $\mathbf{h} = \{\mathbf{h} : \|\mathbf{y}_P - \mathbf{Th}\|_2 \le \varepsilon\}$ using J iterations of the steepest ascent algorithm:
     - Initialization: $\mathbf{h}_0^{(i)} = \mathbf{h}^{(i-1)}$.
     - For $j = 1, \ldots, J$ (loop J times):
       Let $\Delta F_{\sigma_i}(\mathbf{h}_{j-1}^{(i)}) = -\mathbf{h}_{j-1}^{(i)} \odot \exp\left(-\left|\mathbf{h}_{j-1}^{(i)}\right| / 2\sigma_i^2\right) / \sigma_i^2$
       Let $\mathbf{h}_j^{(i)} = \mathbf{h}_{j-1}^{(i)} + \mu_i \sigma_i^2 \Delta F_{\sigma_i}(\mathbf{h}_{j-1}^{(i)})$.
       Project $\mathbf{h}_j^{(i)}$ back onto the feasible set: $\mathbf{h}_j^{(i)} \leftarrow \mathbf{h}_j^{(i)} + \mathbf{T}^+\left(\mathbf{y}_P - \mathbf{Th}_j^{(i)}\right)$.
  3) Set $\mathbf{h}^{(i)} = \mathbf{h}_J^{(i)}, i \leftarrow i+1$.
- Final answer: $\mathbf{h} = \mathbf{h}^{(I)}$.

In general, the initial parameter sequence $\{\sigma_1, \ldots, \sigma_I\}$ and $\{\mu_1, \ldots, \mu_I\}$ of SL0 algorithm are calculated in advance. $\sigma_1 = 2\max\left|\mathbf{h}^{(0)}\right|$, $\sigma_i = c\sigma_{i-1}$, where c is usually between 0.5 and 1, $\mu_1 = \cdots = \mu_I = \mu > 0$. However, such fixed initialization method cannot guarantee the most convergent result of SL0 solution, and improper parameter settings may cause the accuracy of the algorithm to decrease and the convergence speed to slow down. Therefore, it is limited in practical application.

III. Desigh of SL0-UNet

As shown in (4), the vector $\mathbf{h}$ follows a certain sparse distribution and matrix $\mathbf{T}$ is fixed, vector $\mathbf{y}_P$ will also follow a certain sparse distribution. The proposed network is the unfolded version of the SL0 iterative algorithm. It is used to learn the optimal σ sequence and μ sequence through gradient descent in the unfolded network:

- The forward pass of SL0-UNet performs the estimation as the architecture of SL0-UNet is the unfolded SL0 algorithm. Each layer will approach the real channel and finally estimate the channel.
- The backward pass of SL0-UNet performs the parameter learning. As the training weight of SL0-UNet is initialized with general SL0 parameter initialization method, the goal of the backward pass is to update the parameter so that it can perform efficient sparse recovery for all data following a certain distribution.

A. Network Structure

According to the steps of SL0 algorithm, it can be regarded as the network's I th layer in Figure 1. The input of SL0-UNet is $\mathbf{T}$, $\mathbf{y}_P$ and $\mathbf{h}^{(0)}$, and the training parameters are $\{\sigma_1, \ldots, \sigma_I\}$ and $\{\mu_1, \ldots, \mu_I\}$. $P_J(\bullet)$ represents the inner loop iteration, which can be regarded as a nonlinear activation function, and the output is $\mathbf{h}^{(I)}$. The ith layer's output of SL0-UNet is expressed as:

$$\mathbf{h}^{(i)} = P_J(\mathbf{y}_P, \mathbf{T}, \mathbf{h}^{(i-1)}, \sigma_i, \mu_i) \tag{10}$$

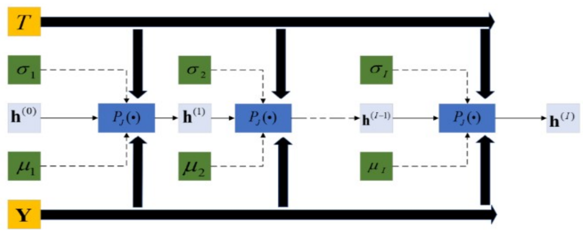

Figure 1 The network structure of SL0-UNet

$P_J(\bullet)$ corresponds to step 2 in Table 1 and it is an internal loop in the SL0 algorithm. In SL0-UNet, it is regarded as a J-layer subnetwork with input $\mathbf{y}_P, \mathbf{T}, \mathbf{h}_0^{(i)}, \sigma_i, \mu_i$ and output $\mathbf{h}_J^{(i)}$. The structure of the $j(j = 1, 2, \ldots, J)$ th layer of $P_J(\bullet)$ is shown in Figure 2, which mainly consists of two parts: update module and projection module.

Update module ($\mathrm{M}_{j=1:J}^{(i)}$): After the J th data of layer $i-1$ in SL0-UNet is passed through the projection module, the output $\mathbf{h}_J^{(i-1)}$ is taken as the initial value $\mathbf{h}_0^{(i)}$ of layer i, then $\mathrm{M}_{j=1:J}^{(i=1:I)}$ can be expressed by the following equation:

$$\mathbf{h}_j^{(i)} = \mathbf{h}_{j-1}^{(i)} + \mu_i \sigma_i^2 \Delta F_{\sigma_i}(\mathbf{h}_{j-1}^{(i)}) \tag{11}$$

where $\Delta F_{\sigma_{i-1}}(\mathbf{h}_{j-1}^{(i)}) = -\mathbf{h}_{j-1}^{(i)} \odot \exp(-\left|\mathbf{h}_{j-1}^{(i)}\right|^2 / 2\sigma_i^2) / \sigma_i^2$, σ_{i-1}, μ_{i-1} are the approximation parameter and iterative step size parameter of layer $i-1$ respectively, and the output $\mathbf{h}_J^{(i)}$ is the input of the projection module.

Projection module($Z_{j=1:J}^{(i)}$): The output $\mathbf{h}_J^{(i)}$ of update module $\mathrm{M}_j^{(i)}$ is taken as input, $Z_{j=1:J}^{(i=1:I)}$ can be expressed by

$$\mathbf{h}_j^{(i)} = \mathbf{h}_j^{(i)} + \mathbf{T}^+(\mathbf{y}_P - \mathbf{T}\mathbf{h}_j^{(i)}) \tag{12}$$

Where $\mathbf{T}^+ = \mathbf{T}^H(\mathbf{TT}^H)^{-1}$, when $j \le J-1$ and $i \le I$, the output $\mathbf{h}_j^{(i)}$ serves as the input of update module $\mathrm{M}_{j+1}^{(i)}$; when

$j = J$ and $i \le I - 1$, the output $\mathbf{h}_J^{(i)}$ serves as the input of the first update module $\mathrm{M}_1^{(i+1)}$ of layer $i+1$ in SL0-UNet; when $j = J$ and $i = I$, the output $\mathbf{h}_J^{(I)}$ serves as the final output $\mathbf{h}^{(I)}$ of the network.

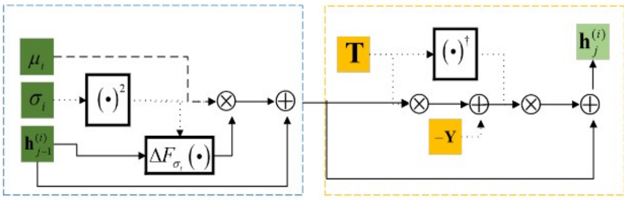

Figure 2 The structure of J th layer in subnetwork $P_J(\bullet)$

The insertion method of the pilot is shown in Table I, and the frequency domain data of the pilot is 1. According to (1), $\{\mathbf{h}_q^{train}\}_{q=1}^{Q}$ is generated as the training label set, and $\{\mathbf{h}_o^{test}\}_{o=1}^{O}$ is generated as the test label set. The sparsity k is set to 5. According to $\mathbf{T}$, $\{\mathbf{h}_q^{train}\}_{q=1}^{Q}$, $\{\mathbf{h}_o^{test}\}_{o=1}^{O}$, the corresponding training datasets $\{\mathbf{y}_q^{train}\}_{q=1}^{Q}$ and test datasets $\{\mathbf{y}_o^{test}\}_{o=1}^{O}$ are generated, where $\mathbf{y}_q^{train} = \mathbf{T}\mathbf{h}_q^{train}$, $\mathbf{y}_o^{test} = \mathbf{T}\mathbf{h}_o^{test} + \mathbf{w}$, and the SNR given is 1dB to 30dB with an interval of 1dB.

TABLE I. PILOT INSERTION MODE

Continuous - distributed	32 pilots are continuously inserted on 128 ~ 159 subcarriers
Well-distributed	Pilots are spaced at 8 intervals and evenly distributed over 256 subcarriers
Random-distributed	Pilots are randomly inserted on each of the 32 subcarriers

B. *Learning Methods of Parameters in SL0-UNet*

To optimize the parameters,, two learning methods are considered corresponding to SL0-UNet1 and SL0-UNet2.

- SL0-UNet1

SL0-UNet1 sets $\sigma_i = c\sigma_{i-1}$, $\mu_1 = \cdots = \mu_I$, c is the attenuation factor, so the network only trains σ_1, μ_1, and c to get better $\{\sigma_1,\ldots,\sigma_I\}$ and $\{\mu_1,\ldots,\mu_I\}$. The advantage of this learning method is that only three parameters need to be trained and the probability of obtaining the local optimal solution gets smaller. However, the disadvantage is that the nonlinear fitting ability of the network is insufficient.

- SL0-UNet2

SL0-UNet2 is trained for σ_i and μ_i of each layer. The number of network learning parameters correspondingly changes to $2I$. This method has high flexibility and strong nonlinear fitting ability. But the disadvantage is that the training time is longer and it is easy to fall into local optimality. It should be emphasized that under the same number of layers, the trained SL0-UNet1 and SL0-UNet2 will have the same time complexity in application

C. *Initialization and training methods*

1) Initialization

The input is set to $\mathbf{h}^{(0)} = 0$, $\mathbf{h}_q^{(0)} = (\mathbf{T})^{+}\mathbf{y}_q^{train}$ and two different initialization is expressed in Table II.

TABLE II. TWO DIFFERENT INITIALIZATION METHOD

	σ	c	μ
SL0-UNet1	$\sigma_1 = 2\max\left\{\lvert\mathbf{h}_q^{(0)}\rvert\right\}_q^Q$	$c = 0.5$	$\mu_1 = 1$
SL0-UNet2	$\sigma_i = 0.5^{i-2}\max\left\{\lvert\mathbf{h}_q^{(0)}\rvert\right\}_{q=1}^{Q}$		$\mu_1 = \cdots = \mu_I = 1$

With the minimum L2 norm solution the next value for σ may be chosen about two to four times of the maximum absolute value of the obtained sources (max $\lvert\mathbf{h}\rvert^{(i)}$). Different from SL0 algorithm, the SL0-UNet avoids calculating the least square norm solution of $\mathbf{y}_p = \mathbf{T}\mathbf{h}$ during initialization, so the operation complexity is lower.

2) Training methods

Given the number of network layers I and the training data sets $\Upsilon\{\{\mathbf{h}_q^{train}\}_{q=1}^{Q}, \{\mathbf{y}_q^{train}\}_{q=1}^{Q}\}_{q=1}^{Q}$, the normalized Mean Square Error (NMSE) is defined as the network loss function,

$$\mathrm{E}(\Phi) = \frac{1}{Q}\sum_{(\mathbf{y}_q^{train},\mathbf{h}_q^{train})\in\Gamma} \frac{\left\|\hat{\mathbf{h}}_q^{(I)}(\Phi,\mathbf{T},\mathbf{h}^{(0)},\mathbf{y}_q^{train}) - \mathbf{h}_q^{train}\right\|_2^2}{\left\|\mathbf{h}_q^{train}\right\|_2^2} \quad (13)$$

where, $\hat{\mathbf{h}}_q^{(I)}(\Phi,\mathbf{T},\mathbf{h}^{(0)},\mathbf{y}_q^{train})$ the output of I th layer , Φ is the parameter and $\mathbf{T},\mathbf{h}^{(0)},\mathbf{y}_q^{train}$ are the inputs.

- SL0-UNet1

The parameters $\Phi^* = \{\sigma_1^*, \mu_1^*, c^*\}$ can be updated directly by minimizing the $\mathrm{E}(\Phi)$ through the back propagation algorithm.

- SL0-UNet2

To be more specific, on the premise of $\Phi_{1:i}^* = \{\sigma_i^*, \mu_i^*\}_{1:i}$ obtained after training the i layers of the network, the layer $i+1$ is trained in two steps. First, keep $\Phi_{1:i}^*$ unchanged and learn the parameter Φ_{i+1} of layer $i+1$ in the network by using (13)

$$\Phi_{i+1}^* = \arg\min_{\Phi_{i+1}} \frac{1}{Q}\sum_{(\mathbf{y}_q^{train},\mathbf{h}_q^{train})\in\Gamma} \frac{\left\|\hat{\mathbf{h}}_q^{(i+1)}(\{\Phi_{1:i}^*,\Phi_{i+1}\},\mathbf{T},\mathbf{h}^{(0)},\mathbf{y}_q^{train}) - \mathbf{h}_q^{train}\right\|_2^2}{\left\|\mathbf{h}_q^{train}\right\|_2^2} \quad (14)$$

were, $\hat{\mathbf{h}}_q^{(i+1)}(\{\Phi_{1:i}^*,\Phi_{i+1}\},\mathbf{T},\mathbf{h}^{(0)},\mathbf{y}_q^{train})$ represents the output of layer $i+1$ in the network, and then takes $\{\Phi_{1:i}^*,\Phi_{i+1}\}$ as the initial value, and solves (15) to update $\Phi_{1:i+1}$:

$$\Phi_{1:i+1}^* = \arg\min_{\Phi_{1:i+1}} \frac{1}{Q}\sum_{(\mathbf{y}_q^{train},\mathbf{h}_q^{train})\in\Gamma} \frac{\left\|\hat{\mathbf{h}}_q^{(i+1)}(\{\Phi_{1:i+1}\},\mathbf{T},\mathbf{h}^{(0)},\mathbf{y}_q^{train}) - \mathbf{h}_q^{train}\right\|_2^2}{\left\|\mathbf{h}_q^{train}\right\|_2^2} \quad (15)$$

the optimal parameters $\Phi^* = \{\sigma_i^*, \mu_i^*\}_{i=1}^{I}$ of layer 1 to layer i is obtained. After optimizing the network parameters, the estimation result can be obtained by new test datasets $\mathbf{y}_o^{test}$

$$\hat{\mathbf{h}}_o^{test} = \hat{\mathbf{h}}_o^{(I)}(\Phi^*,\mathbf{T},\mathbf{h}^{(0)},\mathbf{y}_o^{test}) \quad (16)$$

IV. EXPERIMENTAL RESULTS

In Fig.3, three pilot insertion methods under comb pilot sequence structures are tested respectively by using SL0 algorithm. It is shown that the SL0 algorithm has good performance when using fewer pilot resources. The number of network layer in SL0-UNet is set 10, the training data sets $Q = 3000$ and the number of iterations is 1600. The initial values are obtained and shown in Table III and Table IV. It is shown that the typical initial value settings obtained by the two kinds of network training are different from those of traditional SL0 algorithm, and the operation efficiency is higher than that of SL0 algorithm.

In Fig4, it is shown that the channel sparse recovery performance of SL0-UNet is better than SL0 algorithm, especially, SL0-Unet2 performs better with higher SNR.

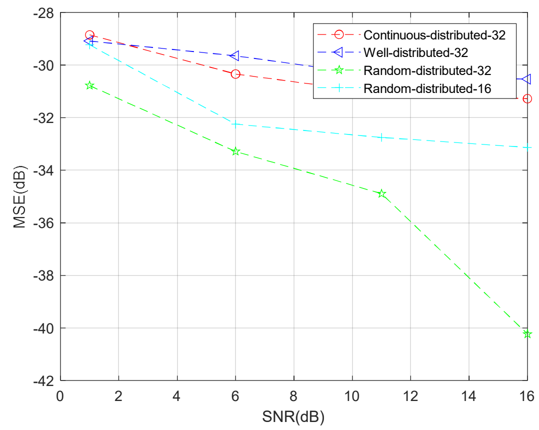

Figure 3 The comparison of three pilot insertion methods

TABLE III. THE OPTIMIZATION OF PARAMETERS IN SL0-UNET1 (WITHOUT NOISE)

	σ_1	μ_1	c
Before optimization	0.0626	1	0.5
After optimization	0.0528	1.1793	0.6368

TABLE IV. THE OPTIMIZATION OF PARAMETERS IN SL0-UNET 2 (WITHOUT NOISE)

Layer	Initial σ	Final σ	Initial μ	Final μ
1	0.1083	0.1083	1	1.0000
2	0.2165	0.0511	1	1.8608
3	0.1083	0.0562	1	1.6221
4	0.0541	0.0090	1	1.2975
5	0.0271	0.0102	1	1.3709
6	0.0135	0.0145	1	1.3229
7	0.0068	0.0149	1	1.4483
8	0.0034	0.0148	1	1.6202
9	0.0017	0.0130	1	1.6732
10	0.0008	0.0120	1	1.6840

TABLE V. THE OPERATION COMPLEXITY OF SL0-UNET,SL0 AND OMP

SR techniques	$O(\bullet)$
SL0-UNet	$(4n+8mn+1)IJ$
SL0	$(8mn+4n+1)IJ+8mn+4m^3/3$
OMP	$4mnk+2nk+4mk^3+4k^4/3+8mk^2$

In Fig.5, it is show that with the increasing of network layer I, the complexity of SL0 algorithm is higher than SL0-UNet1 and SL0-UNet2. For SL0-UNet1 the number of parameters that needed to be trained is 2 in each network layer, and for SL0-UNet2 the number is $2I$.

In Fig.6, it is shown that when the SNR is lower than 15dB, the NMSE of SL0-UNet2 is almost consistent with SL0 and OMP algorithms, but better than LS algorithm. However, when the SNR is higher than 15dB, the NMSE of SL0-UNet2 decreases rapidly, while the performance of OMP algorithm deteriorates because the priori information of sparsity is unknown.

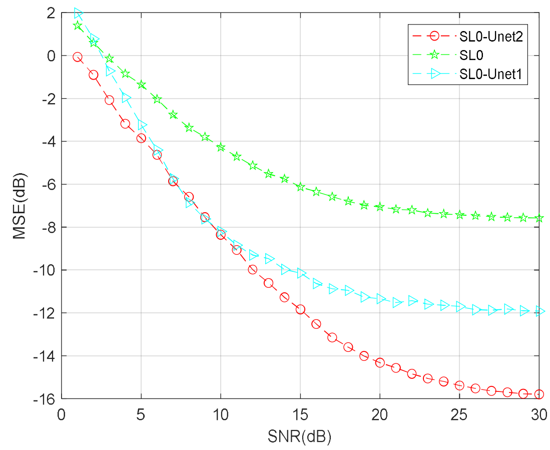

Figure 4 The comparison between SL0, SL0-UNet1, SL0-UNet2

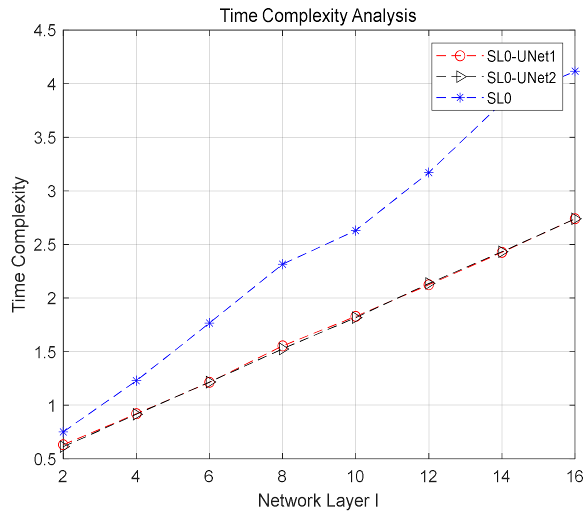

Figure 5 The time complexity between SL0-UNet1, SL0-UNet2

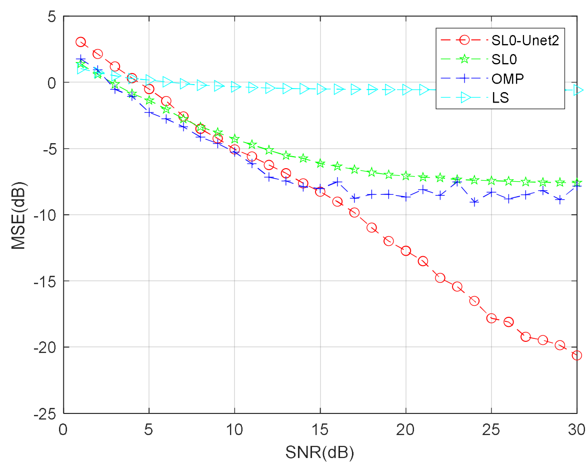

Figure 6 The comparison between SL0, SL0-UNet2, OMP, LS

In Fig.7 it is shown that SL0-UNet2 can recover the channel well in any situation while the performance of SL0 and OMP algorithm is not satisfactory.

Then, in order to investigate whether the proposed network can perform channel estimation with fewer pilot resources, the number of pilots is set to be 16, 32, 64 when constructing the datasets, and the number of pilots that SL0, OMP and LS algorithm used is still set to be 32. The comparison is shown in Fig8. It is shown that the sparse recovery performance of SL0-Unet increases steadily with the increase of pilot resources, and is not sensitive to noise. And compared with the traditional sparse recovery algorithm, it can use less pilot resources to get the same or even better results.

V. Conclusions

A deep neural network model with iterative optimization algorithm based on SL0 algorithm is introduced, and the network structure, data sets construction, initial parameter learning methods and training methods are given in details. The simulation results show that under different SNR, SL0-UNet2 has better recovery effect than SL0-UNet1.

Acknowledgment

This work was supported by NSFC, China (62071319).

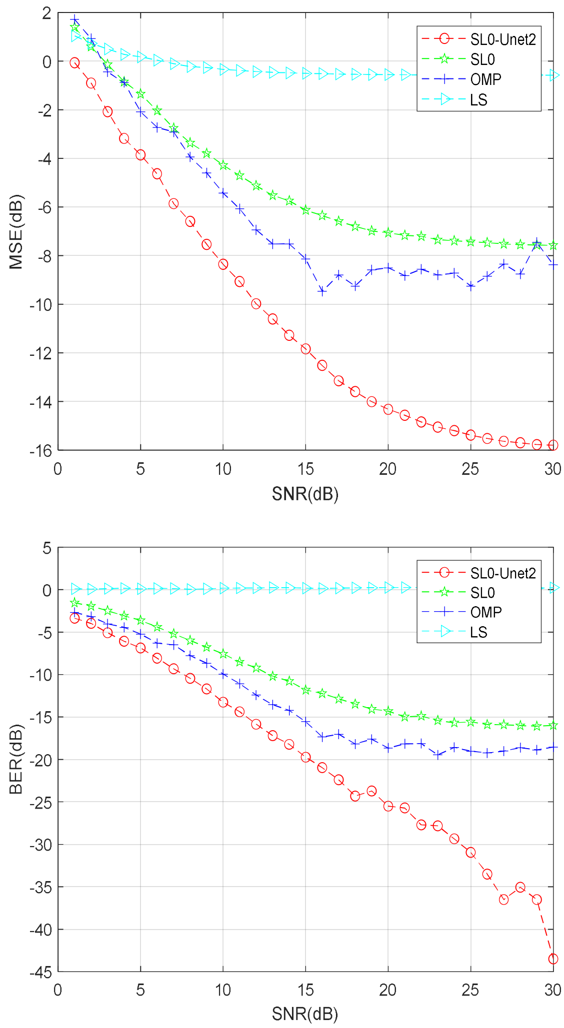

Figure 7 The comparison between SL0, SL0-UNet2, OMP, LS

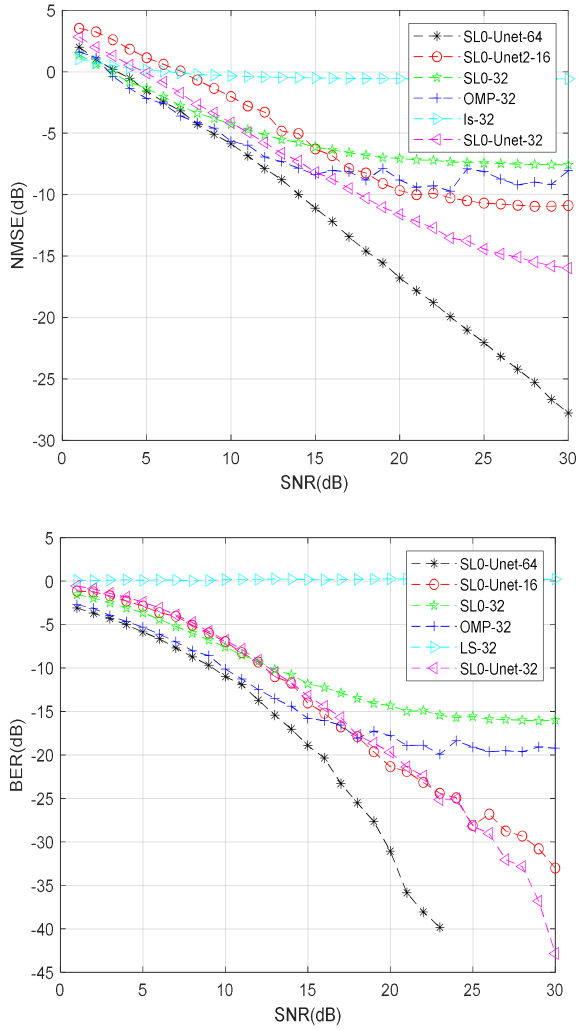

Figure 8 The comparison between SL0-UNet-16, SL0-UNet-32,SL0-UNet-64,SL0-32 OMP-32, LS-32

References

[1] Franklin J, Cooper III A B. Stopping Criteria for Compressive Sensing OFDM Channel Estimation using OMP[C]//2023 57th Annual Conference on Information Sciences and Systems (CISS). IEEE, 2023:

[2] TROPP J A, GILBERT A C. Signal recovery from random measurements via orthogonal matching pursuit[J]. IEEE Transactions on Information Theory, 2007, 53(12): 4655-4666.

[3] T. Yassine and L. Le Magoarou, "mpNet: variable depth unfolded neural network for massive MIMO channel estimation," IEEE Trans. Wireless Commun., vol. 21, no. 7, pp. 5703–5714, 2022.

[4] MOHIMANI H, BABAIE-ZADEH M, JUTTEN C. A fast approach for overcomplete sparse decomposition based on smoothed l0 norm[J]. IEEE Transactions on Signal Processing, 2009, 57(1): 289-301.

[5] Ye H, Li G Y, Juang B H. Power of deep learning for channel estimation and signal detection in OFDM systems[J]. IEEE Wireless Communications Letters, 2017, 7(1): 114-117.

[6] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," IEEE Signal Process. Mag., vol. 38, no. 2, pp. 18–44, 2021.

[7] Youssef M M, Ibrahim M, Abdelhamid B. Deep Learning-aided Channel Estimation For Universal Filtered Multi-carrier Systems[C]//2023 40th National Radio Science Conference (NRSC). IEEE, 2023, 1: 159-166.

[8] GREGOR K, LECUN Y. Learning fast approximations of sparse coding[C]. Proceedings of the 27th International Conference on International Conference on Machine Learning, 2010: 399-406.