



Li, S., Liu, Y., Sun, Y. and Cai, Y. (2023) Deep learning-based channel estimation using Gaussian mixture distribution and expectation maximum algorithm. *Physical Communication*, 58, 102018.

There may be differences between this version and the published version. You are advised to consult the publisher's version if you wish to cite from it.

<https://eprints.gla.ac.uk/292427/>

Deposited on: 28 February 2023

Enlighten – Research publications by members of the University of Glasgow
<https://eprints.gla.ac.uk>

Deep Learning-Based Channel Estimation Using Gaussian Mixture Distribution and Expectation Maximum Algorithm

Shufeng Li^{1,*}, Yiming Liu¹, Yao Sun², Yujun Cai¹

¹ *School of Information and Communication Engineering, Communication University of China, Beijing 100024, China*

² *James Watt School of Engineering, University of Glasgow, G12 8QQ, Scotland, U.K.*

Abstract

In massive multiple-input multiple-output (MIMO), it is much challenging to obtain accurate channel state information (CSI) after radio frequency (RF) chain reduction due to the high dimensions. With the fast development of machine learning (ML), it is widely acknowledged that ML is an effective method to deal with channel models which are typically unknown and hard to approximate. In this paper, we use the low complexity vector approximate messaging passing (VAMP) algorithm for channel estimation, combined with a deep learning framework for soft threshold shrinkage function training. Furthermore, in order to improve the estimation accuracy of the algorithm for massive MIMO channels, an optimized threshold function is proposed. This function is based on Gaussian mixture (GM) distribution modeling, and the expectation maximum Algorithm (EM Algorithm) is used to recover the channel information in beamspace. This contraction function and deep neural network are improved on the vector approximate messaging algorithm to form a high-precision channel estimation algorithm. Simulation results validate the effectiveness of the proposed network.

Keywords: Massive MIMO Channel Estimation, Vector Approximate Message Passing (VAMP), Deep Learning Framework, Gaussian Mixture Distribution, Expectation Maximum Algorithm

1. Introduction

Massive multiple input multiple output (MIMO) has been regarded as a key technology for 5G. With the rise of massive antennas, many methods to reduce the number of RF chains have been applied. However, due to the high dimensions, it is much challenging to obtain accurate channel state information (CSI) after RF chain reduction. Therefore, in recent years, researchers have put forward a lot of channel estimation schemes under large-scale MIMO. Considering the high path loss of MIMO channel [1], the number of multipath components received by the system receiver is less than the amount of system antenna. According to the sparsity of the wireless channel [2], many channel estimation methods based on compressed sensing are widely used. The sparse linear inverse problem is the core of compressed sensing, which has been concerned in recent years[3]. For such channel estimation problems and based on the premise of spatial sparsity [4], more extensive Saleh-Valenzuela(SV) models are used [5].

Many ways have been put forward to solve this problem, such as the tracking matching algorithm and the greedy algorithm. Especially, there is an orthogonal matching pursuit algorithm that estimates sparse signal channels by reducing pilot overhead [6][7]. In addition, a channel estimation method based on a block sparse compressed sensing algorithm is proposed, by taking advantage of the feature that non-zero values in the angle domain matrix of mmWave channels are distributed in blocks [8]. However, these methods cannot achieve a high accuracy in massive MIMO systems. Zou proposed an approximate message passing (AMP) algorithm in [9], which adds Onsager correction to accelerate the convergence of the algorithm [10][11]. Moreover, the algorithm can achieve high estimation accuracy, but may diverge when the matrix deviates from the i.i.d.sub-Gaussian model. Rangan proposed a vector approximate message passing (VAMP) algorithm to solve this problem. For some large right orthogonal invariant matrices, the VAMP algorithm has higher stability, which is more appropriate for channel estimation in massive MIMO systems [12][13]. The derivation and linear model extension of the AMP algorithm and the VAMP

algorithm are also investigated in [14][15][16]. However, the main difficulty lies in determining the optimal shrinkage parameters, thus empirical parameters are usually used, which degrades the accuracy of channel estimation.

Recently, deep learning is developing rapidly in the field of wireless communication. Some intelligent schemes for channel estimation [17][18][19] have been proposed, including beamforming design in [20]. Combined with the proposed objective function, models generated by deep learning eliminate the need for prior knowledge of the sparsication and maximize the correlation between the receiver and base station signals like [17]. Similarly, [18] exploits a learned denoising-based approximate message passing(LDAMP) network to train channel data and structure. It effectively solves the problem of the limited number of base station RF chains, realizing the combination of fully convolutional denoising network with learned approximate message passing(LAMP) algorithm in [19] in the millimeter-wave massive MIMO system. On the basis of compressed sensing, a learning-based AMP network in [21][22] is proposed, which is suitable for channel estimation in massive MIMO. The linear transformation coefficients and nonlinear shrinkage parameters are jointly optimized to achieve higher channel estimation accuracy. Besides, a complex-valued Gaussian mixture LAMP is proposed for the channel distribution problem [23], where it uses the threshold parameters of the AMP algorithm as learning objects for hierarchical training [24][25]. Specifically, using the prior information constructs a GM-LAMP network because beamspace channel elements obey Gaussian mixture distribution, which improves the accuracy of beamspace channel estimation [23].

Based on this idea, we exploit Gaussian distribution to conduct prior information on the beamspace. However, due to the loss of channel data in the construction of Gaussian mixture distribution, the expectation maximum(EM) algorithm is added to recover the data, which greatly improves the accuracy of channel estimation. Simulation results demonstrate the performance gain of the proposed scheme. The main contributions of this article are summarized as follows:

- Based on the sparsity of the beamspace, we construct the channel models to analyze the AMP and VAMP algorithms in two antenna arrays (ULA, UPA). Based on this, deep learning is introduced into the VAMP algorithm and an LVAMP algorithm with high stability and accuracy is generated through hierarchical optimization.
- Considering that the beamspace elements obey the Gaussian mixture distribution, part of the channel information is missing after the construction of the new threshold function. Therefore, the EM algorithm is used to restore Gaussian mixture distribution data to improve the accuracy of channel information via deep learning.
- A new GMEM threshold function based on these algorithms is constructed, and replace the LVAMP threshold function to generate a new GMEM-LVAMP channel estimation algorithm.

Notation. : The notations in this paper are as follows: the uppercase boldface letters denote matrices, lowercase boldface letters are vectors and normal letters are scalar quantities. $(\cdot)^H$ denotes conjugate transpose. $\|\cdot\|_2$ is two-norm. $E\{\cdot\}$ denotes the expectation, $\mathcal{U}(-a, a)$ denotes the probability density function of uniform distribution on $(-a, a)$.

2. SYSTEM MODEL

In this section, we represent the beamspace channel estimation problem in uplink mmWave massive MIMO system, and then transform the channel estimation problem into a sparse signal recovery problem. The last, the VAMP algorithm and its improving LVAMP network are presented.

2.1. Array Model

Our work considers a time division duplex (TDD) based massive MIMO system. For convenience, we assume that one user only has a single antenna, and the total number of users is K . We describe the universal spatial channel

firstly and based on it to introduce other channel models. Based on the SV model in [1], the channel vector between the k th user and the N antenna base station under the lens antenna array channel is

$$\mathbf{h}_k = \sqrt{\frac{N}{L_k}} \sum_{i=0}^{L_k} \beta_k^{(i)} \mathbf{a}(\theta_k^{(i)}) = \sqrt{\frac{N}{L_k}} \sum_{i=0}^{L_k} \mathbf{c}_{k,i}, \quad (1)$$

where $\mathbf{c}_{k,i} = \beta_k^{(i)} \mathbf{a}(\theta_k^{(i)})$ is the i th path component with $\beta_k^{(i)}$ presents the complex gain and L_k is the number of resolvable paths [2]. $\theta_k^{(i)}$ denotes the spatial direction, which depends on the array geometry. $\mathbf{a}(\theta_k^{(i)})$ denotes the $N \times 1$ array steering vector. For uniform linear array (ULA) which has N antennas, we have

$$\mathbf{a}(\theta) = \frac{1}{\sqrt{N}} \left[e^{-j2\pi d \sin(\theta) \mathbf{n} / \lambda} \right], \quad (2)$$

where $\mathbf{n} = [0, 1, \dots, N-1]^T$. For uniform planar arrays (UPAs) which has $N_1 \times N_2$ ($N=N_1 \times N_2$) antennas [4], we have

$$\mathbf{a}(\theta) = \frac{1}{\sqrt{N}} \left[e^{-j2\pi d \sin(\theta_a) \sin(\theta_e) \pi \mathbf{n}_1 / \lambda} \right] \otimes \left[e^{-j2\pi d \cos(\theta_a) \pi \mathbf{n}_2 / \lambda} \right], \quad (3)$$

where $\mathbf{n}_1 = [0, 1, \dots, N_1-1]^T$, $\mathbf{n}_2 = [0, 1, \dots, N_2-1]^T$, λ denotes the wavelength of carrier, and d is the antenna spacing that usually satisfies $d = \lambda/2$ in mm-wave communications.

2.2. Beam-space Channel

For the spatial domain channel model in Formula (2), it can be converted into a beam spatial channel through a lens antenna array. Such lens antenna array is like a spatial DFT matrix \mathbf{P} of size $N \times N$, which contains the array steering vectors of N orthogonal directions covering the entire space as

$$\mathbf{P} = \left[\mathbf{a}(\bar{\psi}_1), \mathbf{a}(\bar{\psi}_2), \dots, \mathbf{a}(\bar{\psi}_N) \right]^H, \quad (4)$$

90 where $\bar{\psi}_n = \frac{1}{N} (n - \frac{N+1}{2})$, and $n=1, 2, \dots, N$ is the pre-defined spatial directions by the lens antenna array. As the same, UPA also can be expressed as

$$\mathbf{P} = \left[\mathbf{a}(\bar{\psi}_1^a, \bar{\psi}_1^e), \dots, \mathbf{a}(\bar{\psi}_1^a, \bar{\psi}_{N_2}^e), \dots, \mathbf{a}(\bar{\psi}_{N_1}^a, \bar{\psi}_1^e), \dots, \mathbf{a}(\bar{\psi}_{N_1}^a, \bar{\psi}_{N_2}^e) \right]^H, \quad (5)$$

where $\bar{\psi}_n^a = \frac{1}{N_1}(n - \frac{N_1+1}{2})$, $n = 1, 2, \dots, N_1$ and $\bar{\psi}_n^e = \frac{1}{N_2}(n - \frac{N_2+1}{2})$, $n = 1, 2, \dots, N_2$. They represent the spatial Angle of azimuth and elevation defined by the lens antenna array respectively. Therefore, through different antenna arrays and the derivation of Formula (4) (5), a beam channel space vector $\tilde{\mathbf{h}}_k$ with a size of $N \times 1$ between the k th user and N antennas is obtained, and its beamspace channel is expressed by the following [22]

$$[\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \dots, \tilde{\mathbf{h}}_k] = [\mathbf{P}\mathbf{h}_1, \mathbf{P}\mathbf{h}_2, \dots, \mathbf{P}\mathbf{h}_k]. \quad (6)$$

It is worth pointing out that only a few path signals are sent after passing through the lenticular line array, so our beam channel space is characterized by sparse signals. The massive MIMO with a lenticular antenna array only have a
95 small amount of RF chains with a little compromise on performance.

2.3. Problem Formulation

In the upward training stage, the received signal vector of the base station $\mathbf{y}_{k,q} \in N_{RF} \times 1$ can be expressed [5] as

$$\mathbf{y}_{k,q} = \mathbf{A}_{k,q} \tilde{\mathbf{h}}_k s_{k,q} + \bar{\mathbf{n}}_{k,q}, q = 1, 2, \dots, Q, \quad (7)$$

where $\tilde{\mathbf{h}}_k$ represents the channel vector, $\mathbf{A}_{k,q}$ is the lens beam selection network of $N_{RF} \times N$, $s_{k,q}$ is the pilot symbol, and $\bar{\mathbf{n}}_{k,q} = \mathbf{A}_{k,q} \mathbf{n}_{k,q}$, $\mathbf{n}_{k,q} \sim \mathcal{CN}(0, \sigma_n^2 I_N)$ is a gaussian noise of $N \times 1$. In the remaining parts, we set $s = 1$, because the pilot symbol s is usually known at the receiving end. We obtain an $M \times 1 (M = QN_{RF})$ measurement signal \mathbf{y}_k as:

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_{k,1} \\ \mathbf{y}_{k,2} \\ \vdots \\ \mathbf{y}_{k,Q} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{k,1} \\ \mathbf{A}_{k,2} \\ \vdots \\ \mathbf{A}_{k,Q} \end{bmatrix} \tilde{\mathbf{h}}_k + \begin{bmatrix} \bar{\mathbf{n}}_{k,1} \\ \bar{\mathbf{n}}_{k,2} \\ \vdots \\ \bar{\mathbf{n}}_{k,Q} \end{bmatrix} = \mathbf{A}_k \tilde{\mathbf{h}}_k + \mathbf{n}_k, \quad (8)$$

where \mathbf{A}_k is the beam selection matrix with input of $M \times N$, and \mathbf{n}_k is the noise vector of $M \times 1$. Due to the orthogonality of pilot frequency, we have unified

the estimation of users, that is to say, we can use the same channel estimation method for total K users, so we get

$$\mathbf{y} = \mathbf{A}\tilde{\mathbf{h}} + \mathbf{n}. \quad (9)$$

Since the beamspace channel is approximately sparse, we use the sparse signal recovery algorithm of compressed sensing knowledge to carry out the channel estimation of the beamspace. Matrix \mathbf{A} is the perception matrix.

100 Although the optimal solution can be obtained through such a method, it is a typical NP problem, the computational efficiency of the algorithm will be greatly reduced. In order to solve this problem efficiently, one of the most famous methods is convert such sparse problem to a convex optimization problem. Because the minimum l_0 norm combination is needed in the process of solving
 105 all non-zero values, which can lead to the complexity of the algorithm being improved, signal accuracy of reconstructing drops. Therefore, the non-convex problem is relaxed to a convex problem to solve, using the l_1 norm equivalent instead of l_0 norm [3]. Because the l_1 norm has an optimal solution, turning the problem into a linear programming problem

$$\begin{aligned} \hat{\mathbf{h}} &= \arg \min_h \left\| \tilde{\mathbf{h}} \right\|_1, \\ \text{s.t.} \\ \left\| \mathbf{A}\tilde{\mathbf{h}} - \mathbf{y} \right\|_2 &\leq \varepsilon. \end{aligned}$$

110 Because of the large computational complexity and time cost of the convex optimization algorithm, it is generally considered to approach the original signal indirectly by employing coefficient decomposition to realize signal reconstruction. Some traditional greedy algorithms can solve this problem, such as the orthogonal matching tracking algorithm, regularization orthogonal matching algorithm and generalized orthogonal matching tracking algorithm.
 115 However, the accuracy and effectiveness of these traditional algorithms cannot achieve satisfactory results.

2.4. LAMP and LVAMP

There is a large number of antenna array in massive MIMO systems, and the dimension of sparse signals through lens antenna array is very high. To solve this problem, an iterative AMP algorithm is proposed to restore sparse signals with low complexity, especially for high-dimensional sparse signals.

AMP algorithm decouples the input of each contraction function and accelerates convergence in [10] by introducing the Onsager correction. The Onsager correction ensures that the shrinking input is an undermined form of additive Gaussian white noise (AWGN) of the real signal \mathbf{h}_0 . The key part of this algorithm is the soft threshold contraction function

$$[\boldsymbol{\eta}(\mathbf{r}_t; \lambda_t, \sigma_t^2)]_i = \max(|r_{t,i}| - \lambda_t \sigma_t, 0) e^{j\omega_{t,i}} \quad (10)$$

where i represents the elements of the input vector, $\omega_{t,i}$ is the phase of complex-valued element $\mathbf{r}_{t,i}$, λ_t is the predefined and fixed parameter in the t th iteration, and σ_t^2 is updated via estimating the noise variance. Meanwhile, in Formula (10) we can see that the denoising threshold of the threshold function λ_t depends on the value of t , which is updated by estimating the noise variance [11]. The AMP algorithm can work well with the response problem of massive sparse signals, but it is hard to deal with large matrices. The VAMP algorithm is proposed by Rangan in [13] to solve the vulnerability of AMP relative to a special matrix (right orthogonal invariant matrix, a right orthogonal invariant matrix \mathbf{A} is a random matrix whose distribution remains the same after right multiplication by any fixed orthogonal matrix). The VAMP algorithm has the same characteristics as the original AMP algorithm, such as low complexity of iteration and fewer convergent iterations. For matrix \mathbf{A} . Assuming that

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \quad (11)$$

For the right orthogonal invariant \mathbf{A} , the decomposed matrix \mathbf{V} will include the first R columns of the matrix uniformly distributed over the set of orthogonal matrices of the same size. Note that the I.I.D. Gaussian matrix is a special case of right orthogonal invariance, where \mathbf{U} is randomly orthogonal and \mathbf{S} has

specific numerical values. Importantly, the VAMP algorithm is stable and suitable for massive MIMO, because when M and N dimensions are large enough, the algorithm works well under any orthogonal matrix \mathbf{U} and different singular values \mathbf{S} [28,29]. In order to facilitate the derivation, we have adopted a method of matrix extension, refer to [21]. Although we have assumed that \mathbf{A} is square to streamline the analysis, we make this assumption without loss of generality. Where \mathbf{U} and \mathbf{V} are $N \times N$ orthogonal matrices such that \mathbf{U} is deterministic and \mathbf{V} is Haar distributed. For example, by setting

$$\mathbf{S} = \text{diag}(\mathbf{s}), \mathbf{U} = \begin{bmatrix} \mathbf{U}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \mathbf{s} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{0} \end{bmatrix}$$

Corresponding to the AMP algorithm, the Onsager correction term for VAMP algorithm is

$$\tilde{\mathbf{r}}_t = \left(\hat{\mathbf{h}}_t - \alpha_t \mathbf{r}_t \right) / (1 - \alpha_t) \quad (12)$$

$$\alpha_t = \mathbf{g}'_1(\mathbf{r}_t, \gamma_t) \quad (13)$$

where α_k is the Onsager correction term, $\mathbf{g}_1(\cdot)$ is the threshold function of the VAMP algorithm and has the same mathematical form as Formula (12). We present the architecture of the algorithm in Fig.1.

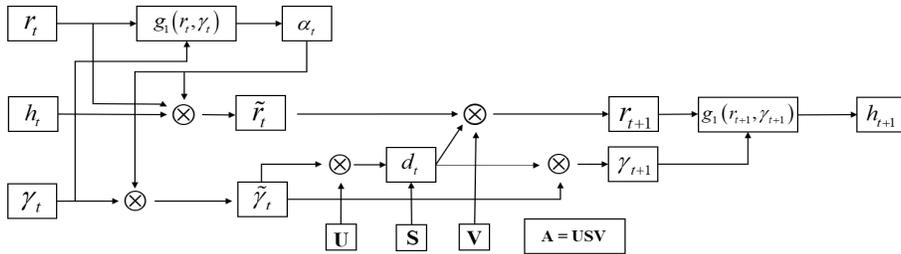


Figure 1: VAMP network of T layer.

125

It can be seen that the VAMP algorithm and the AMP algorithm have the same four basic steps, including estimation, divergence calculation, Onsager correction and variance calculation. For large random \mathbf{A} , the VAMP quantity

\mathbf{r}_t behaves like a white Gaussian noise-destroying version of the real signal h_0 :
 130 $\mathbf{r}_t = h^0 + \mathcal{N}(0, \tau_k I)$.

It is also proven that the VAMP algorithm has a wider application range than the AMP algorithm in [30]. However, it still has some problems in estimating the sparse beamspace. In the threshold function, the shrinkage parameters generally adopt the same empirical value for iteration, are not suitable for different massive MIMO beamspace channels. In addition, the prior information
 135 of beamspace channel in the VAMP algorithm is not effectively utilized, and the channel information is relatively independent of the algorithm.

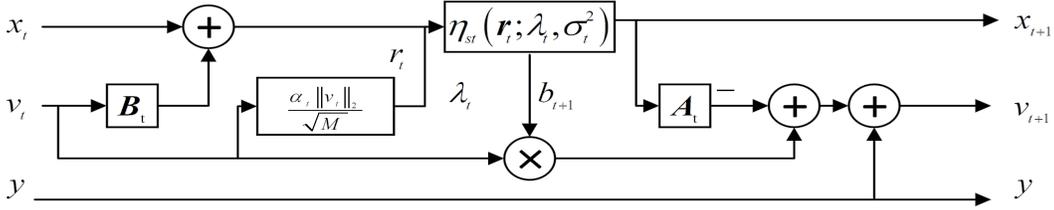


Figure 2: LAMP network of T layer.

Recently, due to the excellent channel approximation performance of machine learning, some deep learning networks were proposed based on the AMP
 140 algorithm to optimize its linear shrinkage parameter \mathbf{B}_t and nonlinear shrinkage parameters λ_t at each iteration. Finally, each iteration of the algorithm is mapped to each layer of the deep learning network. As illustrated in Fig.2, deep learning parameters can be introduced into the original AMP algorithm, realizing a linear transformation of the measured signal space to the original
 145 sparse signal space. Different linear coefficients are chosen at different layers of the iteration to improve the accuracy of channel information recovery. Referring to the LAMP algorithm threshold parameter, the threshold function $\mathbf{g}_1(\cdot)$ of the VAMP algorithm is decomposed during the iterative process of the LVAMP algorithm. Like Formula (10), γ_t is decomposed into two values λ_t and σ_t^2 , with
 150 λ_t being trained as a non-linear coefficient by machine learning in a hierarchical iteration, and σ_t^2 is updated by estimated noise variance. To accommodate

Formula (12) of the VAMP algorithm, linear coefficients are not constructed, which largely reduces the computation complexity.

$$\hat{\mathbf{h}}_t = \mathbf{g}_1(\mathbf{r}_t, \gamma_t) = \mathbf{g}_1(\mathbf{r}_t, \lambda_t, \sigma_t^2), \quad (14)$$

where $\sigma_t^2 = \frac{1}{M} \|\mathbf{r}_t\|_2^2$. The same supervised learning approach as the GMEM-LVAMP algorithm is used in the deep learning training of the LVAMP algorithm. The exact process is described uniformly in section 3.3. Here we give the loss function firstly for the LVAMP iterative process:

$$L_t(\lambda_t) = \frac{1}{D} \sum_{d=1}^D \left\| \hat{\mathbf{h}}_{t+1}^d(\mathbf{y}^d, \lambda_t) - \tilde{\mathbf{h}}^d \right\|_2^2, \quad (15)$$

The t-layer algorithm architecture of the LVAMP algorithm is illustrated in Fig. 3. The t-layer algorithm steps are mapped into the LVAMP-DNN network and the data generated using the SV channel model is iteratively computed.

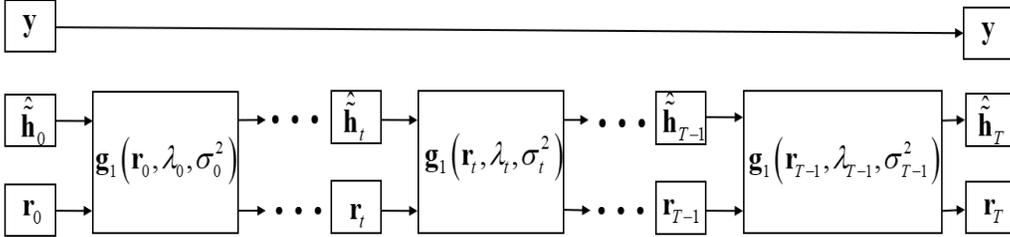


Figure 3: LVAMP network of T layer.

Algorithm 1 Learning VAMP

Input: compressed sensing matrix \mathbf{A} , measurements \mathbf{y} , denoiser g_1 , assumed noise precision γ_ω , number of iterations K_{it} ;

Output: $\hat{\mathbf{X}}_{K_{\text{it}}}$.

- 1: initial $\mathbf{A} = \mathbf{U} \text{Diag}(\bar{\mathbf{s}}) \mathbf{V}$ and $\bar{\mathbf{U}} \bar{\mathbf{U}}^T = \mathbf{I}_R, \bar{\mathbf{V}}^T \bar{\mathbf{V}} = \mathbf{I}_R, R = \text{rank}(\mathbf{A})$; Compute pre-conditioned $\mathbf{y} = \text{Diag}(\bar{\mathbf{s}})^{-1} \bar{\mathbf{U}}^T \mathbf{y}$
- 2: $\sigma_t^2 = \frac{1}{M} \|\mathbf{y}\|_2^2$
- 3: get λ_t from T layers

```

165 4:  $\gamma_t = \max(\text{abs}(\lambda_t \sigma_t), 0)$ 
5:  $\mathbf{h}_t = \mathbf{g}_1(\mathbf{r}_t; \lambda_t, \sigma_t^2)$ 
6:  $\alpha_t = \mathbf{g}'_1(\mathbf{r}_t; \lambda_t, \sigma_t^2)$ 
7:  $\mathbf{r}_t = (\mathbf{h}_t - \alpha_t \mathbf{r}_t) / (1 - \alpha_t)$  (Initial parameter calculation)
8: (see Section 2.4)
170 9: for  $t = 0, 1, \dots, K_{it}$  do
10:    $\sigma_t^2 = \frac{1}{M} \|\mathbf{r}_t\|_2^2$ 
11:    $\gamma_t = \max(\text{abs}(\lambda_t \sigma_t), 0)$ 
12:    $\mathbf{h}_t = \mathbf{g}_1(\mathbf{r}_t; \lambda_t, \sigma_t^2)$ 
13:    $\alpha_t = \mathbf{g}'_1(\mathbf{r}_t; \lambda_t, \sigma_t^2)$ 
175 14:    $\tilde{\mathbf{r}}_t = (\mathbf{h}_t - \alpha_t \mathbf{r}_t) / (1 - \alpha_t)$ 
15:    $\tilde{\gamma}_t = \gamma_t (1 - \alpha_t) / \alpha_t$ 
16:    $\mathbf{d}_t = \gamma_\omega \text{Diag}(\gamma_\omega \bar{\mathbf{s}}^2 + \tilde{\gamma}_t \mathbf{1})^{-1} \bar{\mathbf{s}}^2$ 
17:    $\gamma_{t+1} = \tilde{\gamma}_t \langle \mathbf{d}_t \rangle / (N/R - \langle \mathbf{d}_t \rangle)$ 
18:    $\mathbf{r}_{t+1} = \tilde{\mathbf{r}}_t + N/R \bar{\mathbf{V}} \text{Diag}(\mathbf{d}_t / \langle \mathbf{d}_t \rangle) (\tilde{\mathbf{y}} - \bar{\mathbf{V}}^T \tilde{\mathbf{r}}_t)$ 
180 19:    $\hat{\mathbf{X}}_K = \mathbf{g}_1(\mathbf{r}_t; \lambda_t, \sigma_t^2)$ 
20: end for
21: return  $\hat{\mathbf{X}}_{K_{it}}$ .

```

Referring to Algorithm1, the compressive sensing matrix is decomposed by singular value decomposition(SVD), and its characteristic matrix \mathbf{S} is used for subsequent calculation. In the basic VAMP algorithm process, the algorithm and the beamspace channel data are matched by updating parameters λ_t and σ_t^2 in step3. After initialization, we use the Onsager correction term to represent the variance of the shrinkage function in step10. Step12 is equivalent to the denoising process.

In the following simulation, the threshold of the shrinkage function in the VAMP algorithm has been improved. Through the improved threshold shrinkage function, the channel estimation performance has obvious enhancement and improvement. However, this method is based on signal sparse recovery algorithm. We improve the shrinkage function that does not play a good role in the specific beamspace channel. Then, on this basis, we derive a new contraction

function suitable for the beamspace based on the Gaussian mixture distribution, and propose a GMEM-LAMP channel estimation scheme.

3. A NEW SOFT THRESHOLD FUNCTION BY GMEM

3.1. Gaussian Mixture Model

200 When sparse signals are recovered from complex compressed linear measurements, the distribution of non-zero coefficients of the signal has an effect on the recovered mean square error (MSE). If this distribution is prior known, we can use the known prior distribution to construct the AMP algorithm [24], so that the AMP technique can achieve almost minimal MSE recovery.

Some previous work has verified that Gaussian mixture (GM) distribution is used to obtain a prior of the beamspace such as in [25]. We can obtain the probability density function of beamspace channel elements as

$$p(\tilde{h}; \boldsymbol{\theta}) = \sum_{k=0}^{N_c-1} p_k \mathcal{CN}(\tilde{h}; \mu_k, \sigma_k^2), \quad (16)$$

where $\boldsymbol{\theta} = \{p_0, \dots, p_{N_c-1}, \mu_0, \dots, \mu_{N_c-1}, \sigma_0^2, \dots, \sigma_{N_c-1}^2\}$ represents the set of all parameters, N_c denotes the number of Gaussian components in the Gaussian mixed distribution, p_k is the probability of the k th Gaussian component, equivalent to the weight of each single Gaussian distribution, μ_k and σ_k^2 are the mean and variance of the k th Gaussian component respectively. We can get the probability density function of the Gaussian distribution [26] as

$$\mathcal{CN}(\tilde{h}; \mu_k, \sigma_k^2) = \frac{1}{\pi \sigma_k^2} \exp\left(-\frac{(h - \mu_k)^* (\tilde{h} - \mu_k)}{\sigma_k^2}\right). \quad (17)$$

In this way, the Gaussian mixture distribution can be used to represent the channel distribution of the beamspace, from Formula (1), (4), and (8), we can get the n th element of the beamspace channel as

$$\tilde{h}_n = \sqrt{\frac{N}{L}} \sum_{l=1}^L \beta_l \text{sinc}(\Delta\psi_n), \quad (18)$$

205 where β_l is the complex gain, $\Delta\psi_n = \bar{\psi}_n - \psi_l$, $\bar{\psi}_n$ is the predefined spatial direction and ψ_l is the actual direction of the path. We know that at the transmitting end of the lens antenna array when the actual spatial direction $\bar{\psi}_n$ of the path approach the predefined spatial direction ψ_l , $\text{sinc}(\Delta\psi_n)$ can achieve a larger value, so \tilde{h}_n could have a large power. Due to the randomness of the direction ψ_l , the different \tilde{h}_n can be regarded as the different Gaussian component. We can use the gaussian mixture distribution to simulate the distribution of beamspace channel elements.

3.2. Expectation Maximum Algorithm

215 Based on gaussian mixture distribution, we obtain the prior of the beamspace channel. However, because of different paths, the beamspace channel will lose part of the information, when using gaussian mixture distribution to describe the channel space through machine learning, the distribution will be biased, and the accuracy will be reduced. And with different levels of training, the EM algorithm maybe helpful in solving certain problems, such as overfitting problems and under-learning rates that can arise from machine learning. Therefore, we consider the EM algorithm to improve the Gaussian mixture distribution like [26].

The EM algorithm adopts the iterative optimization method, and each iteration is divided into two steps, one is the expectation step (E step), and the other is the maximum step (M step). The starting point of the design of the EM algorithm is to restore the parameters in the case of some missing data [28]. After training with the gaussian mixture distribution parameters of beamspace channel through machine learning, we obtain the characteristics of the initial Gaussian mixture distribution like [31]. Then we consider the derivation of the EM algorithm based on Gaussian mixture distribution. The original form of GM is known as Formula (16), for this, we introduce a k dimensional random variable H , where $h_k = k$ represents the probability of the k th class being selected, and $p(h_k = k) = p_k, \sum_K p_k = 1$. Assuming that h_k is independent and identically distributed, we can write the joint probability distribution form of

H:

$$p(h) = p(h_1)p(h_2)\cdots p(h_k) = \prod_{k=1}^K p_k^{h_k}, \quad (19)$$

$$p(x|h_k = k) = \mathcal{N}(x|\mu_k, \sigma_k^2), \quad (20)$$

$$p(x|h) = \prod_{k=1}^K \mathcal{N}(x|\mu_k, \sigma_k^2)^{h_k}, \quad (21)$$

It can be seen that Formula (21) of GM model has the same form as Formula (14), and a new variable H is introduced into Formula (19), which is usually called the implicit variable. For the data in the channel, the implied meaning is that we know that the channel data can be divided into K categories, but when a data point is randomly selected, we do not know which category this data point belongs to, and its attribution cannot be observed. Therefore, an implied variable H is introduced to describe this phenomenon. This is consistent with the Gaussian mixture distribution. Then under Bayes, $p(h)$ is the prior probability and $p(x|h)$ is the likelihood probability, so the posterior probability is

$$\gamma(h_k) = p(h_k = k|x) = \frac{p_k \mathcal{N}(x|\mu_k, \sigma_k^2)}{\sum_{j=1}^K p_j \mathcal{N}(x|\mu_j, \sigma_j^2)}. \quad (22)$$

This is the expectation step (E step) of the Expectation maximum algorithm. Then we can update and expand the parameters based on a posterior probability (M step)

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(h_{nk}) x_n, \quad (23)$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_{n=1}^N \gamma(h_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T, \quad (24)$$

$$p_k = \frac{N_k}{N}, \quad (25)$$

where $N_k = \sum_{n=1}^N \gamma(h_{nk})$. Then the steps are iterated until convergence completes the calculation of maximum expectation. We can see that in step12 to

16 of Algorithm2. In this way, we obtain a gaussian mixture distribution of the beamspace with information recovery, which can accurately represent the distribution of the beamspace channel. On this basis, the Bayesian minimum mean square error (MMSE) is used to deduce the contraction function

$$\boldsymbol{\eta}_{gmem} = \mathbb{E} \left\{ \tilde{h} \mid r; \boldsymbol{\theta}, \sigma^2 \right\} = \frac{\int \tilde{h} p \left(r \mid \tilde{h}; \sigma^2 \right) p(\tilde{h}; \boldsymbol{\theta}) d\tilde{h}}{\int p \left(r \mid \tilde{h}; \sigma^2 \right) p(\tilde{h}; \boldsymbol{\theta}) d\tilde{h}}, \quad (26)$$

where $r = \tilde{h} + n$ considering that n is additive Gaussian noise obeying $\mathcal{CN}(0, \sigma^2)$. For the specific derivation of the NMSE results, we can refer to the reference in [23]. For simplicity, we can obtain

$$\boldsymbol{\eta}_{gmem} (r; \boldsymbol{\theta}, \sigma^2) = \frac{\sum_{k=0}^{N_c-1} p_k \tilde{\mu}_k (r) \mathcal{CN} (r; \mu_k, \sigma^2 + \sigma_k^2)}{\sum_{k=0}^{N_c-1} p_k \mathcal{CN} (r; \mu_k, \sigma^2 + \sigma_k^2)}. \quad (27)$$

The distributed parameters $\boldsymbol{\theta}$ is also known as contraction parameters. It includes the probabilities, means and variances accounted for by the different Gaussian distributions $\boldsymbol{\theta}$, which are generated by machine learning and the EM algorithm training. We can see the detailed construction process of the GMEM shrinkage function in Algorithm2. The initialization of the algorithm values is performed first. In step7 to 10, the parameter $\boldsymbol{\theta}_t$ is trained by GMEM-DNN network firstly. The derivation of the threshold function NMSE based on a Gaussian mixed distribution is performed. The contraction function is introduced into the GMEM-DNN structure, which is equivalent to the effect of the activation function. $\boldsymbol{\theta}_t$ is obtained through the EM algorithm via step13 to 17 to receive the best Gaussian mixture distribution, which more accurately describes the beamspace channel information. Step19 constructs the final shrinkage function $\boldsymbol{g}_2 ()$ to replace the shrinkage function $\boldsymbol{g}_1 ()$ of the LVAMP algorithm. We can see the detailed construction process of GMEM shrinkage function in the Algorithm2. Compared with simple LVAMP soft threshold contraction function, this method can be estimated for a specific beamspace channel. After designing the new shrink function, the next step is to build the GMEM-LVAMP network for beamspace channels.

Algorithm 2 GMEM-LVAMP

Input: compressed sensing matrix \mathbf{A} , measurements \mathbf{y} , denoiser \mathbf{g}_2 , assumed noise precision γ_ω , number of iterations K_{it} ;

Output: $\hat{\mathbf{X}}_{K_{\text{it}}}$.

245 1: initial $\mathbf{A}=\mathbf{U}\text{Diag}(\bar{\mathbf{s}})\mathbf{V}$ and $\bar{\mathbf{U}}\bar{\mathbf{U}}^T=\mathbf{I}_R, \bar{\mathbf{V}}^T\bar{\mathbf{V}}=\mathbf{I}_R, R = \text{rank}(\mathbf{A})$;

2: Computepreconditioned $\mathbf{y}=\text{Diag}(\bar{\mathbf{s}})^{-1}\bar{\mathbf{U}}^T\mathbf{y}$.

3: Selectinitial r_0 and $\gamma_0 \geq 0$.

4: $\mathbf{g}_2(\mathbf{r}_t; \lambda_t, \sigma_t^2) = \max(|\mathbf{r}_{t,i}| - \lambda_t\sigma_t, 0)e^{j\omega t, i}$

5: $\mathbf{r}_t = (\mathbf{h}_t - \alpha_t\mathbf{r}_t) / (1 - \alpha_t)$

250 6: **for** $t = 0, 1, \dots, K_{\text{it}}$ **do**

7: $\mathbf{g}_2 = E \left\{ \tilde{h} \mid r; \boldsymbol{\theta}_t, \sigma^2 \right\} = \frac{\int \tilde{h} p(r \mid \tilde{h}; \sigma^2) p(\tilde{h}; \boldsymbol{\theta}_t) d\tilde{h}}{\int p(r \mid \tilde{h}; \sigma^2) p(\tilde{h}; \boldsymbol{\theta}_t) d\tilde{h}}$

8: $p(r \mid \tilde{h}; \sigma^2) p(\tilde{h}; \boldsymbol{\theta}_t) = \sum_{m=0}^{N_c-1} p_m \mathcal{CN}(r; \mu_m, \sigma^2 + \sigma_m^2) \mathcal{CN}(\tilde{h}; \tilde{\mu}_m(r), \tilde{\sigma}_m^2)$

9: $\tilde{\mu}_m(r) = \frac{\sigma^2 \mu_m + \sigma_m^2 r}{\sigma^2 + \sigma_m^2}, \tilde{\sigma}_m^2(r) = \frac{\sigma^2 \sigma_m^2}{\sigma^2 + \sigma_m^2}$

10: $\mathbf{g}_2(r; \boldsymbol{\theta}_t, \sigma^2) = \frac{\sum_{k=0}^{N_c-1} p_k \tilde{\mu}_k(r) \mathcal{CN}(r; \mu_k, \sigma^2 + \sigma_k^2)}{\sum_{k=0}^{N_c-1} p_k \mathcal{CN}(r; \mu_k, \sigma^2 + \sigma_k^2)}$

255 11: (see Refer [23])

12: **for** $k = 0, 1, \dots, T_{\text{it}}$ **do**

13: $\gamma(h_k) = \frac{p_t \mathcal{N}(x \mid \mu_k, \sigma_k^2)}{\sum_{j=1}^K p_j \mathcal{N}(x \mid \mu_j, \sigma_j^2)}$

14: (see Section 3.2)

15: get new μ_k, σ_k^2, p_k

260 16: **end for**

17: Return μ_k, σ_k^2, p_k

18: $\mathbf{g}_2(r; \boldsymbol{\theta}_t, \sigma^2) = \frac{\sum_{k=0}^{N_c-1} p_k \tilde{\mu}_k(r) \mathcal{CN}(r; \mu_k, \sigma^2 + \sigma_k^2)}{\sum_{k=0}^{N_c-1} p_k \mathcal{CN}(r; \mu_k, \sigma^2 + \sigma_k^2)}$

19: $\hat{\mathbf{X}}_K = \mathbf{g}_2(\mathbf{r}_t; \boldsymbol{\theta}_t, \sigma^2)$

20: (see Algorithm 1)

265 21: **end for**

22: **return** $\hat{\mathbf{X}}_{K_{\text{it}}}$.

3.3. GMEM-LVAMP Networks

In order to improve the precision of channel estimation, the machine learning method is used in the VAMP algorithm, making the choice of shrinkage coefficient more accurate. At the same time, building a new threshold function can provide the prior distribution of the beamspace channel, so the VAMP algorithm

architecture is the basis of the GMEM-LVAMP algorithm. GMEM-LVAMP network is divided into T layers for training [31]. Different from LVAMP, the new shrink function includes training the data distribution in the beamspace after the machine learning training and obtains new data containing the Gaussian mixture distribution. Next, we specifically describe how the two algorithms estimate the beamspace of massive MIMO systems.

The parameters of the threshold function are constructed based on Fig.3. For the training process of the GMEM-LVAMP algorithm, the derived shrinkage function with the beamspace channel data is used. Distinguishing from the linear systolic coefficient λ_t of Formula (14), we construct a new systolic function, where $\boldsymbol{\theta}_t$ is used as the object of training for each layer. For the GMEM-DNN training, a supervised learning approach is used in the offline training phase [23]. The training dataset can be represented as $\left\{ \mathbf{y}^d, \tilde{\mathbf{h}}^d \right\}_{d=1}^D$, where \mathbf{y}^d is the input to the training network, $\tilde{\mathbf{h}}^d$ is the corresponding label, and D represents the training number. Also using the layer-by-layer training method, we train the parameters for each layer in the GMEM network, and different loss functions for each layer can be expressed as

$$L_t(\boldsymbol{\theta}_t) = \frac{1}{D} \sum_{d=1}^D \left\| \hat{\mathbf{h}}_{t+1}^d(\mathbf{y}^d, \boldsymbol{\theta}_t) - \tilde{\mathbf{h}}^d \right\|_2^2, \quad (28)$$

where $\hat{\mathbf{h}}_{t+1}^d$ is the output of the non-linear shrinkage operation of the t th layer. After t -layer training and optimization, the optimization coefficients $\boldsymbol{\theta}_t$ are added to the algorithm. Based on the coefficients constructed by machine learning optimization, the algorithm is iteratively updated with the maximum expectation, and the new parameters are constructed to form the GMEM training network. We introduce the detail of algorithm steps, as shown in Algorithm2. Next NMSE is employed to evaluate the performance of the algorithm.

3.4. Computational Complexity Analysis

In this subsection, we discuss the complexity of the proposed algorithm and compare it with the current algorithms. We can know that both the LAMP

algorithm and the GM-LAMP algorithm are constructed on the AMP algorithm, which has a complexity of $\mathcal{O}(TMN)$. Similarly, the complexity of the LVAMP algorithm and the GMEM-LVAMP algorithm is $\mathcal{O}(TMN)$. And we take into account that the OMP algorithm complexity is $\mathcal{O}(SMN) + \mathcal{O}(S^3M)$ in comparison, where S is the sparsity level of the beamspace channel vector.

3.5. Discussion

From the above discussion, the algorithm is only used to restore the signal data sparseness in some present LAMP networks. The Gaussian mixture distribution threshold function is used to accurately estimate the channel prior distribution of the beamspace. It ignores the part data loss in the process of machine learning. Therefore, we adopt the VAMP algorithm and add the EM algorithm to make the GMEM-LVAMP algorithm more accurate and stable, which is suitable for a wider range of beamspace channel estimation problems.

In addition, the existing DNN training networks, such as the all-Unicom network architecture, are universal to many typical application scenarios, but there are a few changes to the algorithm. Therefore, in order to better fit with the algorithm, a special DNN architecture is designed, or even the algorithm can be directly presented with the DNN architecture for direct channel estimation.

4. NUMERICAL STUDIES

In this section, we conduct simulations of the LVAMP algorithm and the GMEM-LVAMP algorithm to compare the effectiveness of the existing algorithms for beamspace channel estimation. For channel modeling, the number of base station antennas and RF chains are $N = 256$ and $N_{RF} = 16$. Similarly, the number of single-antenna users and instants is set as $K = 16$, $Q = 8$. In order to facilitate computation generally, we expand the long matrix into a square matrix. So the number of measurements $M = N = 256$, random matrix $\mathbf{A} \in \mathcal{R}^{M \times N}$. To facilitate comparison between the LAMP algorithm and the LVAMP algorithm, for the construction of the random matrix \mathbf{A} , Ideal

derivative matrix is i.i.d random Gaussian matrix, in which both perform the best algorithmic performance. But the assumption of using Gaussian random matrix is inconsistent with the basic system constraint. so we onstructing a random matrix \mathbf{A}_0 swticher based by value of 1 and -1. $\mathbf{A} = \frac{1}{\sqrt{M}}\mathbf{A}_0$. In addition, our beamspace channel is based on the SV model and adopts three different antenna arrays to set the same channel parameters for K users: path component $L_K = 4$, $\beta_{k,l} \sim \mathcal{CN}(0, 1)$, where $l = 1, 2, 3, 4$. $\theta_{k,l} \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ $\theta_a \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ $\theta_e \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$. In the uplink channel construction, we set the frequency to 30 GHz. For the Gaussian mixture distribution construction part, we set four Gaussian components. Considering fairness, we set their initial probability to 0.25 and set the mean and variance to 0 based on sparsity. Therefore, the threshold contraction function $\boldsymbol{\theta}_t$ contains twelve elements in the training process of each layer and $\boldsymbol{\theta}_0 = \{0.25, 0.25, 0.25, 0.25, 0, 0, 0, 0, 0, 0, 0, 0\}$. For the OMP channel estimation algorithm, we set the channel sparsity $S = 24$, and the empirical shrinkage parameter of the AMP algorithm as $\lambda_t = 1.14$ for each iteration t based on [21].

4.1. SV Model Simulation Results

For the network testing, we generate channel data sets based on the SV channel model refer to Formula (1)(3)(5)(6). we generate 80000, 2000 and 2000 samples for training, validation, and testing, respectively for both algorithms in 256×1 ULA and 16×16 UPA, the number of training layers is set as 8, where the number of nodes in the LVAMP-DNN network and GMEM-DNN is related to the measured number N and channel size M . The signal-to-noise ratio(SNR) of the channel estimation is set to 0-30dB.

In order to reflect the performance of our algorithm, we compare the OMP algorithm, the AMP algorithm, and the VAMP algorithm with the algorithm added with machine learning in different Signal-to-noise ratio. In Fig.4, we can see that the normalized mean squared error(NMSE) of the original OMP algorithm and AMP algorithm only achieve -15dB around. The NMSE of the LVAMP algorithm is smaller than that of the VAMP algorithm while the ma-

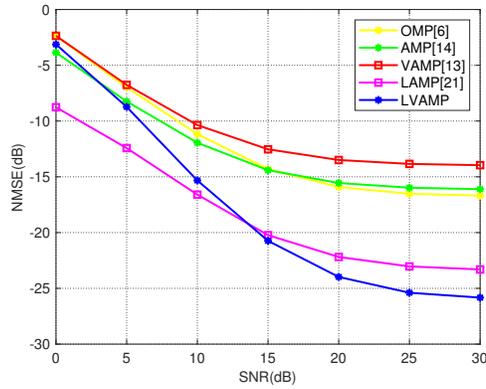


Figure 4: NMSE performance comparison for ULAs.

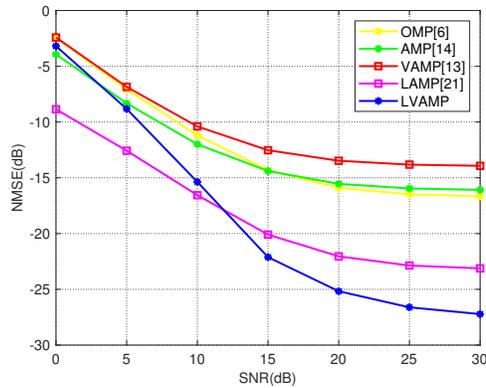


Figure 5: NMSE performance comparison for UPAs.

chine learning training parameters are utilized. which can achieve better channel estimation performance due to the accurate contraction threshold of the algorithm. In Fig.5, we replace the antenna array in order to achieve algorithm performance in different antenna array beamspace. By comparing the NMSE performance of the five algorithms at 16×16 UPA, it can be seen that the two schemes based on machine learning achieve similar performance. After 15dB, the performance of the LVAMP algorithm is better than the LAMP algorithm. But the proposed methods work well with high SNRs and worse than existing methods at low SNRs. Only the non-linear systolic parameters of the LVAMP

355 algorithm are constructed. In contrast, the LAMP algorithm underwent joint training of the linear coefficients and the non-linear parameters. LVAMP algorithm is therefore more significantly affected by changes in SNR.

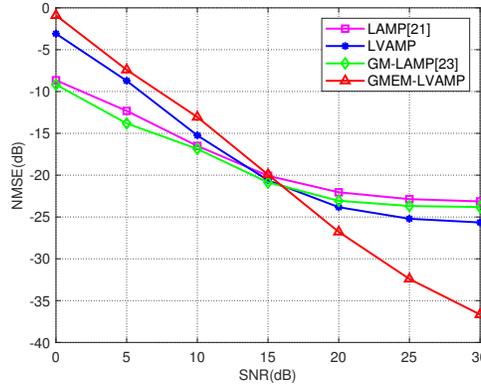


Figure 6: NMSE performance comparison for ULAs.

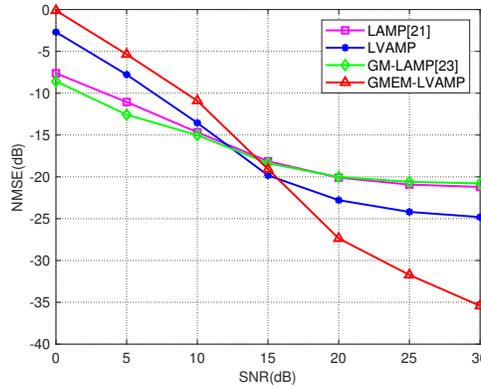


Figure 7: NMSE performance comparison for UPAs.

Based on the new soft threshold shrinkage function algorithm GMEM-LVAMP and the GM-LAMP networks in [23], we also give ULA and UPA arrays in Fig.6 and Fig.7, respectively. We conduct the comparison of the four algorithms, where two algorithms only add machine learning training network(i.e., the LAMP algorithm and the LVAMP algorithm), while the other two replace

360

the threshold function on this basis. In this way, we can clearly see the improvement effect of the threshold function. For instance, the GM-LAMP algorithm has a slight improvement under the overall SNR compared with the LAMP algorithm. Considering the recovery of data, the GMEM-LVAMP algorithm is better than the LVAMP algorithm in two types of antenna arrays. Especially after 15dB of SNR, the NMSE can achieve under 25dB.

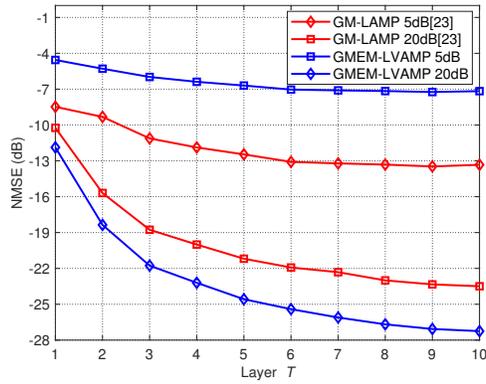


Figure 8: NMSE performance against the number of layers for the GMEM-LVAMP.

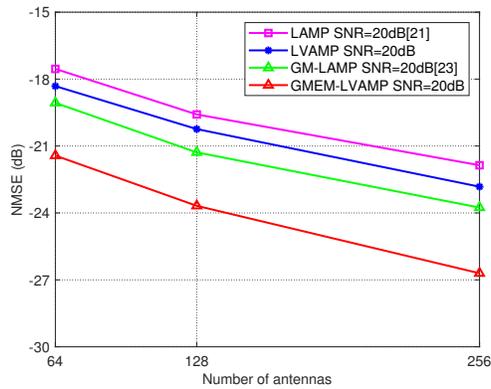


Figure 9: NMSE performance against the number of antennas for the GMEM-LVAMP.

In order to show the convergence of the algorithm under the condition of maximum expected iteration, we give different simulation results with differ-

ent SNRs in the ULA array based on the SV model in Fig.8. We can observe that the convergence is approximately reached in the number of layers $T=8$ when SNR=20dB. Besides, in Fig.9 we give the NMSE performance for different number of antennas in the massive MIMO case, with the number of antennas including 64, 128 and 256. For comparison, the SNR is set to 20 dB. We can see that the performance of the four algorithms is improved as the number of antennas increases, thanks to the spatial multiplexing function of the number of antennas. thanks to considering the prior distribution of the beamspace channel, both the GM-LAMP algorithm in [23] and our proposed GMEM-LVAMP algorithm are better than the LAMP algorithm and the LVAMP algorithm, respectively.

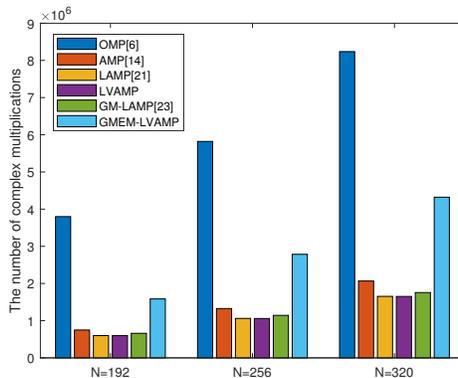


Figure 10: The number of complex multiplications against the number of antennas N.

In terms of complexity, we also calculate the number of antennas at the base station (BS) $N=256$ and the number of measurements $M=256$, so the complexity of five benchmark algorithms can be calculated as shown in Fig.10. Since we only train the shrinkage threshold in the machine learning network, the calculation complexity is lower in the LAMP algorithm and the LVAMP algorithm. The GM network in [23] considering the prior distribution of the beamspace channel. So the complexity of the GM-LAMP algorithm has been increased slightly. GMEM network both considering the channel distribution and

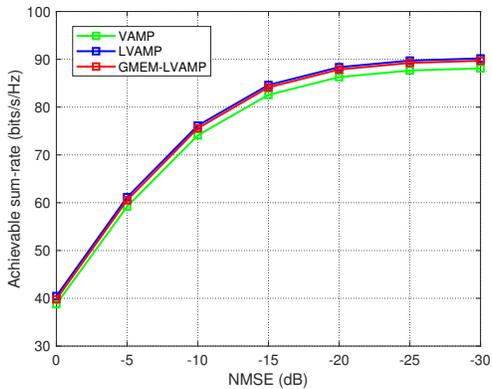


Figure 11: Sum-rate against different NMSE.

390 beamspace channel dates recovery, computational complexity is also improved.
 But it can be seen in Fig.6, we have greatly improved NMSE performance, with
 a small increase in complexity, the accuracy of channel estimation is greatly im-
 proved. Next, we evaluate the impact of the NMSE for the beamspace channel
 estimation by sum-rate. In the simulation of Fig9, the construction scheme in
 395 [32] was used, The estimated beamspace channel was modeled (imperfect CSI)
 as

$$\hat{\mathbf{H}} = \tilde{\mathbf{H}} + \mathbf{E}, \quad (29)$$

where $\tilde{\mathbf{H}}$ represents perfect CSI for K users, \mathbf{E} is the error matrix with
 entries following the distribution independent and identically distributed.

Specifically we continue to choose the SV model and the ULA antenna ar-
 ray to construct the channel. Fig11 shows the sum-rate achieved by the same
 400 channel model against the NMSE for the beamspace channel estimation. In our
 simulations, we can see that different algorithms have similar sum rates when
 the NMSE is the same. The highest value converges gradually at NMSE = -25.
 As seen in Fig4-7 at sufficient SNR, our channel estimation algorithms are able
 405 to achieve this rate.

5. CONCLUSION

In this paper, we consider a channel estimation method based on beamspace sparsity in massive MIMO. We adopt a widely used basic AMP algorithm and VAMP algorithm. In order to further improve the accuracy of the algorithm, referring to LAMP, we use machine learning to train the nonlinear shrinkage parameter of the VAMP algorithm, then obtain nearly a two-fold increase in channel estimation accuracy and make the algorithm more stable. On this basis, inspired by the GM-LAMP algorithm, our work reconstructs the threshold function of the algorithm, which is constructed by using the prior distribution of channel data and Gaussian mixture distribution. Considering the similarity between VAMP and AMP, we reconstruct the contraction function of LVAMP by combining Gaussian mixture distribution and EM algorithm. The contraction function consider the loss of channel data caused by machine learning, use the EM algorithm to recover the lost data a priori. In this regard, we evaluate it under the SV model. Simulation results show that the proposed GMEM-LVAMP algorithm can significantly improve the accuracy of Gaussian mixture distribution. When SNR reaches about 15dB to 30dB, its NMSE can still improve by 10dB.

Acknowledgments

This work was supported in part by the National Natural Science Funding of China under Grant 61601414, and in part by the Central University Basic Research Fund of China under Grant CUC210B032.

References

- [1] A. Meijerink and A. F. Molisch, "On the Physical Interpretation of the Saleh–Valenzuela Model and the Definition of Its Power Delay Profiles," in *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 9, pp. 4780-4793, Sept. 2014, doi: 10.1109/TAP.2014.2335812.

- [2] X. Gao, L. Dai, S. Han, C. I and X. Wang, "Reliable Beamspace Channel Estimation for Millimeter-Wave Massive MIMO Systems with Lens Antenna Array," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 6010-6021, Sept. 2017, doi: 10.1109/TWC.2017.2718502.
- [3] D. Ito, S. Takabe and T. Wadayama, "Trainable ISTA for Sparse Signal Recovery," in *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3113-3125, 15 June 15, 2019, doi: 10.1109/TSP.2019.2912879.
- [4] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi and R. W. Heath, "Spatially Sparse Precoding in Millimeter Wave MIMO Systems," in *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1499-1513, March 2014, doi: 10.1109/TWC.2014.011714.130846.
- [5] M. Ke, Z. Gao, Y. Wu, X. Gao and R. Schober, "Compressive Sensing-Based Adaptive Active User Detection and Channel Estimation: Massive Access Meets Massive MIMO," in *IEEE Transactions on Signal Processing*, vol. 68, pp. 764-779, 2020, doi: 10.1109/TSP.2020.2967175.
- [6] A. Alkhateeb, O. El Ayach, G. Leus and R. W. Heath, "Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 831-846, Oct. 2014, doi: 10.1109/JSTSP.2014.2334278.
- [7] N. Rashmi and M. Sarvagya, "Sparse channel estimation using orthogonal matching Pursuit algorithm for SCM-OFDM system," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 1224-1227, doi: 10.1109/ICACCI.2016.7732212.
- [8] X. He, R. Song and W. -P. Zhu, "Pilot Allocation for Distributed-Compressed-Sensing-Based Sparse Channel Estimation in MIMO-OFDM Systems," in *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 2990-3004, May 2016, doi: 10.1109/TVT.2015.2441743.
- [9] X. Zou, F. Li, J. Fang and H. Li, "Computationally efficient sparse Bayesian learning via generalized approximate message passing," 2016 IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), 2016, pp. 1-4, doi: 10.1109/ICUWB.2016.7790383.

- [10] M. Borgerding and P. Schniter, "Onsager-corrected deep learning for
465 sparse linear inverse problems," 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2016, pp. 227-231, doi: 10.1109/GlobalSIP.2016.7905837.
- [11] X. Wang and J. Liang, "Multi-resolution compressed sensing reconstruction via approximate message passing," 2015 IEEE International Conference on Image Processing (ICIP), 2015, pp. 4352-4356, doi: 10.1109/ICIP.2-
470 015.7351628.
- [12] S. Rangan, P. Schniter and A. K. Fletcher, "Vector approximate message passing," 2017 IEEE International Symposium on Information Theory (ISIT), 2017, pp. 1588-1592, doi: 10.1109/ISIT.2017.8006797.
- [13] S. Rangan, P. Schniter and A. K. Fletcher, "Vector Approximate Message
475 Passing," in IEEE Transactions on Information Theory, vol. 65, no. 10, pp. 6664-6684, Oct. 2019, doi: 10.1109/TIT.2019.2916359.
- [14] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in Proc. Inf.
480 Theory Workshop (ITW), Cairo, Egypt, Jan. 2010, pp. 1-5.
- [15] X. Meng, S. Wu and J. Zhu, "A Unified Bayesian Inference Framework for Generalized Linear Models," in IEEE Signal Processing Letters, vol. 25, no. 3, pp. 398-402, March 2018, doi: 10.1109/LSP.2017.2789163.
- [16] X. Meng and J. Zhu, "Bilinear Adaptive Generalized Vector Approximate
485 Message Passing," in IEEE Access, vol. 7, pp. 4807-4815, 2019, doi: 10.1109/ACCESS.2018.2887261.
- [17] E. Balevi, A. Doshi, A. Jalal, A. Dimakis and J. G. Andrews, "High Dimensional Channel Estimation Using Deep Generative Networks," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 1, pp.
490 18-30, Jan. 2021, doi: 10.1109/JSAC.2020.3036947.
- [18] H. He, C. Wen, S. Jin and G. Y. Li, "Deep Learning-Based Channel Estimation for Beamspace mmWave Massive MIMO Systems," in IEEE Wireless Communications Letters, vol. 7, no. 5, pp. 852-855, Oct. 2018, doi: 10.1109/LWC.2018.2832128.

- 495 [19] Y. Zhang, Y. Mu, Y. Liu, T. Zhang and Y. Qian, "Deep Learning-Based
Beamspace Channel Estimation in mmWave Massive MIMO Systems," in
IEEE Wireless Communications Letters, vol. 9, no. 12, pp. 2212-2215,
Dec. 2020, doi: 10.1109/LWC.2020.3019321.
- [20] T. Lin and Y. Zhu, "Beamforming Design for Large-Scale Antenna Arrays
500 Using Deep Learning," in IEEE Wireless Communications Letters, vol. 9,
no. 1, pp. 103-107, Jan. 2020, doi: 10.1109/LWC.2019.2943466.
- [21] M. Borgerding, P. Schniter and S. Rangan, "AMP-Inspired Deep Net-
works for Sparse Linear Inverse Problems," in IEEE Transactions on Sig-
nal Processing, vol. 65, no. 16, pp. 4293-4308, 15 Aug.15, 2017, doi:
505 10.1109/TSP.2017.2708040.
- [22] C. Huang, L. Liu, C. Yuen and S. Sun, "Iterative Channel Estimation
Using LSE and Sparse Message Passing for MmWave MIMO Systems," in
IEEE Transactions on Signal Processing, vol. 67, no. 1, pp. 245-259, 1
Jan.1, 2019, doi: 10.1109/TSP.2018.2879620.
- 510 [23] X. Wei, C. Hu and L. Dai, "Deep Learning for Beamspace Channel Es-
timation in Millimeter-Wave Massive MIMO Systems," in IEEE Transac-
tions on Communications, vol. 69, no. 1, pp. 182-193, Jan. 2021, doi:
10.1109/TCOMM.2020.3027027.
- [24] P. Sun, Z. Wang and P. Schniter, "Joint Channel-Estimation and Equal-
515 ization of Single-Carrier Systems via Bilinear AMP," in IEEE Transactions
on Signal Processing, vol. 66, no. 10, pp. 2772-2785, 15 May15, 2018, doi:
10.1109/TSP.2018.2812720.
- [25] C. Huang, L. Liu, C. Yuen and S. Sun, "Iterative Channel Estimation
Using LSE and Sparse Message Passing for MmWave MIMO Systems," in
520 IEEE Transactions on Signal Processing, vol. 67, no. 1, pp. 245-259, 1
Jan.1, 2019, doi: 10.1109/TSP.2018.2879620.
- [26] J. P. Vila and P. Schniter, "Expectation-Maximization Gaussian-Mixture
Approximate Message Passing," in IEEE Transactions on Signal Process-
ing, vol. 61, no. 19, pp. 4658-4672, Oct.1, 2013, doi: 10.1109/TSP.2013.2272287.

- 525 [27] A. Mohammed, S. Zhang, H. Li, N. Zhao and X. Wang, "Capturing the Sparsity and Tracking the Channels for Massive MIMO Networks," in IEEE Transactions on Vehicular Technology, vol. 69, no. 1, pp. 685-699, Jan. 2020, doi: 10.1109/TVT.2019.2953281.
- [28] S. Wu, H. Yao, C. Jiang, X. Chen, L. Kuang and L. Hanzo, "Downlink
530 Channel Estimation for Massive MIMO Systems Relying on Vector Approximate Message Passing," in IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 5145-5148, May 2019, doi: 10.1109/TVT.2019.2904405.
- [29] J. Mo, P. Schniter and R. W. Heath, "Channel Estimation in Broadband Millimeter Wave MIMO Systems With Few-Bit ADCs," in IEEE Transactions on Signal Processing, vol. 66, no. 5, pp. 1141-1154, 1 March1,
535 2018, doi: 10.1109/TSP.2017.2781644.
- [30] N. Skuratovs and M. Davies, "Upscaling Vector Approximate Message Passing," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 4757-4761, doi:
540 10.1109/ICASSP40776.2020.9053799.
- [31] Y. Wei, M. -M. Zhao, M. Zhao, M. Lei and Q. Yu, "An AMP-Based Network With Deep Residual Learning for mmWave Beamspace Channel Estimation," in IEEE Wireless Communications Letters, vol. 8, no. 4, pp. 1289-1292, Aug. 2019, doi: 10.1109/LWC.2019.2916786.
- 545 [32] X. Gao, L. Dai, S. Han, C.-L. I, and R. W. Heath, Jr., "Energy-efficient hybrid analog and digital precoding for mmWave MIMO systems with large antenna arrays," IEEE J. Sel. Areas Commun., vol. 34, no. 4, pp. 998-1009, Apr. 2016.