

# Unsourced Random Massive Access with Beam-Space Tree Decoding

Jingze Che, *Student Member, IEEE*, Zhaoyang Zhang, *Senior Member, IEEE*, Zhaohui Yang, *Member, IEEE*, Xiaoming Chen, *Senior Member, IEEE*, Caijun Zhong, *Senior Member, IEEE*, and Derrick Wing Kwan Ng, *Fellow, IEEE*

**Abstract**—The core requirement of massive Machine-Type Communication (mMTC) is to support reliable and fast access for an enormous number of machine-type devices (MTDs). In many practical applications, the base station (BS) only concerns the list of received messages instead of the source information, introducing the emerging concept of unsourced random access (URA). Although some massive multiple-input multiple-output (MIMO) URA schemes have been proposed recently, the unique propagation properties of millimeter-wave (mmWave) massive MIMO systems are not fully exploited in conventional URA schemes. In grant-free random access, the BS cannot perform receive beamforming independently as the identities of active users are unknown to the BS. Therefore, only the intrinsic beam division property can be exploited to improve the decoding performance. In this paper, a URA scheme based on beam-space tree decoding is proposed for mmWave massive MIMO system. Specifically, two beam-space tree decoders are designed based on hard decision and soft decision, respectively, to utilize the beam division property. They both leverage the beam division property to assist in discriminating the sub-blocks transmitted from different users. Besides, the first decoder can reduce the searching space, enjoying a low complexity. The second decoder exploits the advantage of list decoding to recover the missed packets. Simulation results verify the superiority of the proposed URA schemes compared to the conventional URA schemes in terms of error probability.

**Index Terms**—Unsourced random access, massive access, beam-space tree decoder, Machine-Type Communication (mMTC).

## I. INTRODUCTION

### A. Motivation

ONE imminent demand for the next generation wireless mobile communication systems is to provide instant and

reliable access for an increasingly large number of machine-type devices (MTDs) [1], [2]. Different from human-centric communication, the resultant Massive Machine-Type Communication (mMTC) has two distinct features. In particular, only a small number of devices are active in each communication round due to the sporadic activity in mMTC [3]. Besides, MTDs usually transmit small data payloads adopting short-packet signaling [4]. These make traditional grant-based random access schemes generally not very suitable for the mMTC scenario because of their low spectral efficiency and exceedingly long latency [5]. Therefore, the design of reliable and efficient grant-free random access schemes has attracted significant attention recently, where active users transmit pilots and data to the base station (BS) directly without permission granted [6], [7]. In most grant-free random access schemes, a set of pilot sequences that are designated to the users are used for the BS to ensure its accurate user activity detection and channel estimation [8]–[10]. However, this is neither affordable nor feasible in the next generation multiple access (NGMA) scenarios due to the high density, the large number of connections therein, and the frequent collisions that may occur. To tackle the issues, a special type of grant-free random access, the so-called unsourced random access (URA), is introduced in [11], in which users do not transmit preambles, all the potential users share a common codebook, and the BS only needs to decode a list of messages instead of the identities of active users. This scheme can avoid the huge cost of preambles and the extra protocol of collision resolution, thus well meeting the requirements of next generation massive access.

On the other hand, massive or super MIMO technology, in combination with the millimeter-wave (mmWave) technology, have been promoted as two core technological features for the next generation wireless communication system with a witnessed potential to boost the capacity and efficiency. These two underlying key technologies jointly bring additional spatial-domain signal dimension with their excellent intrinsic directivity and proper beamforming, and also result in salient beam-space sparsity due to the lack of scattering in a mmWave MIMO channel [12]–[14]. To further increase the efficiency of the future massive access systems, such spatial-domain resources and properties should be fully explored and exploited. Various multi-user transmission schemes have been proposed to unleash the potential and properties of the beam-space resources, such as the typical works on beam division multiple access (BDMA), which simultaneously serves multiple users via different beams [15]–[17].

The work of J. Che, Z. Zhang, X. Chen and C. Zhong was supported in part by National Key R&D Program of China under Grant 2018YFB1801104 and 2020YFB1807101, and National Natural Science Foundation of China under Grant 61725104, 61922071 and U20A20158. The work of D. W. K. Ng was supported in part by funding from the UNSW Digital Grid Futures Institute, UNSW, Sydney, under a cross-disciplinary fund scheme and by the Australian Research Council's Discovery Project (DP210102169).

J. Che, Z. Zhang (Corresponding Author), X. Chen and C. Zhong (e-mails: {jzche, ning\_ming, chen\_xiaoming, caijunzhong}@zju.edu.cn) are with 1) College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, and 2) International Joint Innovation Center, Zhejiang University, Haining 314400, China, and also 3) Zhejiang Provincial Key Laboratory of Info. Proc., Commun. & Netw. (IPCAN), Hangzhou 310027, China.

Z. Yang (e-mail: zhaohui.yang@ucl.ac.uk) is with the Department of Electronic and Electrical Engineering, University College London, UK, and he is also a visiting scholar at Zhejiang University.

D. W. K. Ng (e-mail: w.k.ng@unsw.edu.au) is with the School of Electrical Engineering and Telecommunications, UNSW, Sydney, Australia.

However, for grant-free random access, even if the information of the location of all potential users is stored at the BS, the identities of active users are unknown to the BS. Therefore, the beam dimension cannot be directly exploited in the design of the encoding process of an unsourced grant-free random access, which is different from previous works [15]–[17]. Moreover, in a general URA system, the messages of active users are divided into several sub-blocks and transmitted in consecutive sub-slots. As the signals transmitted from active users often experience deep fading, some sub-blocks may be missed by the decoder at the receiver. The loss of any sub-block in any sub-slot finally leads to the failure of recovering the corresponding original message. The problem of packet loss is also needed to be solved as it can cause severe decoding performance degradation. Note that the user location information can generally serve as a hint of the indices of the messages originated from it. Therefore, the intrinsic beam division property and salient spatial sparsity in a mmWave MIMO system can provide extra extrinsic information for both multi-user signal separation and multi-sub-block message stitching. This motivates us to exploit the beam space properties to design new URA schemes for next generation multiple access to help the entire system accommodate more active users and improve the decoding performance.

### B. Related Works

Y. Polyanskiy first introduced a framework named URA [11]. Specifically, in URA, all the users share a common codebook, and the decoder only needs to decode a list of messages transmitted from the active users. The error probability is defined as the average fraction of mis-decoded messages over the number of active users, including both missed detection and false alarm. It is obvious that the message recovery at the BS can be formulated as a compressed sensing (CS) problem due to the sporadic activity in mMTC, which is similar to the conventional grant-free random access schemes [18], [19]. However, the size of the common codebook grows exponentially with the number of information bits. In practice, even if a short packet is transmitted, the size of information messages is typically at the order of 100 bits, which makes the CS algorithms computationally intractable. In this context, V. K. Amaladinne *et al.* proposed a coded compressed sensing (CCS) scheme for URA communication [20]. In particular, the messages from active users are first divided into several sub-blocks. Then, a systematic linear code adds redundancy to those sub-blocks. Once this is achieved, each sub-block is mapped into a codeword in a common codebook and transmitted in a certain sub-slot. Then a standard CS algorithm implements the detection of the sub-blocks. Finally, the sub-blocks transmitted in different sub-slots are stitched together to obtain the original messages. Build upon the findings in [20] and the structure of sparse regression codes (SPARCs), A. Fengler *et al.* provided an improved inner decoder, and a complete asymptotic error analysis [21].

Apart from the above works, the study of massive multiple-input multiple-output (MIMO) URA has also attracted much attention. A. Fengler *et al.* extended the URA model of

[20] to a block-fading MIMO channel by using a low-complexity covariance-based CS (CB-CS) recovery algorithm [22]. Considering the low code rate and spectral efficiency of the CCS scheme, V. Shyianov *et al.* proposed a new algorithmic solution to solve the massive URA problem by leveraging the rich spatial dimensionality offered by large-scale antenna arrays [23]. Besides, without requiring a separate activity detection or channel estimation step, A. Decurninge *et al.* introduced a structure that allows the receiver to separate the users using a classical tensor decomposition [24]. As URA is a special scheme of grant-free random access, A. Fengler *et al.* presented a conceptually simple algorithm based on pilot transmission, activity detection, channel estimation, Maximum Ratio Combining (MRC), and single-user decoding [25], which is similar to the existing grant-free random access schemes [3], [18]. The difference is that they use a pool of non-orthogonal pilots where every active user picks one of them pseudo-randomly. Furthermore, X. Shao *et al.* proposed a unified cooperative activity detection framework for sourced and unsourced random access based on the covariance of the received signals for the sixth generation (6G) cell-free wireless networks [26].

### C. Main Contributions

In this paper, we propose a URA scheme with beam-space tree decoding. Specifically, we adopt the CCS scheme [20] suitably to our case and design two beam-space tree decoders, which are based on hard decision and soft decision, respectively. By leveraging the beam division property to assist in distinguishing the sub-blocks transmitted from different users, both decoders can help the system serve more active users. As the discriminating power is improved, the searching space of the solution in the decoding process is reduced, such that the first decoder has low complexity. In addition, notice that any sub-block missed by the CS decoder would finally lead to missed detection, which degrades the decoding performance. To tackle this issue, the second decoder establishes factor graphs at each stage during the decoding process and implements message passing algorithm (MPA) to give each candidate sub-block drawn from the checking relationship a log-likelihood ratio (LLR) value. Then the reliability of every candidate path is calculated by a path metric (PM). At every stage, some reliable paths are kept, and finally, the surviving path is output as the valid message. Even if a sub-block is missed by the CS decoder, it is possible that the path of the original message is reliable and kept. The main contributions of this paper are summarized as follows:

- A URA scheme with beam-space tree decoding is proposed for mmWave communication systems in mMTC to accommodate more active users and to improve the system performance.
- Two beam-space tree decoders are designed. Both of them can exploit the intrinsic beam division property to improve the decoding performance of the tree decoder by enhancing the discriminating power and helping the system serve more active users. Besides, the first decoder is based on hard decision with low complexity. The

second one is based on soft decision and exploits the advantage of list decoding to recover the packet loss, which is the key of the proposed URA scheme.

- Simulation results verify that our URA schemes have significantly better performances than existing works.

#### D. Paper Organization and Notations

The rest of this paper is organized as follows: Section II provides a brief introduction of the considered massive URA system. Section III provides the encoding and decoding process of the considered system. Section IV proposes a beam-space tree decoder with hard decision. Then, Section V designs a beam-space tree decoder with soft decision. Next, Section VI analyzes the performance of the proposed URA scheme. Afterward, Section VII provides extensive simulation results to validate the effectiveness of the proposed algorithm. Finally, Section VIII concludes the paper.

Throughout this paper, we use bold letters to denote matrices or vectors and non-bold letters to denote scalars. We denote the  $i$ -th row and the  $j$ -th column of a matrix  $\mathbf{X}$  with the row-vector  $\mathbf{X}_{i,:}$  and the column-vector  $\mathbf{X}_{:,j}$  respectively. We denote  $\mathbb{C}^{A \times B}$  by the space of complex matrices of size  $A \times B$ . We use  $|\cdot|$  to denote the absolute value of a complex number,  $(\cdot)^H$  and  $(\cdot)^T$  to denote conjugate transpose and transpose, respectively. The  $l_i$ -norm of an input vector is denoted by  $\|\cdot\|_i$ .  $|\mathcal{K}|_c$  denotes the number of elements of set  $\mathcal{K}$ . The notation  $x \sim \mathcal{CN}(\mu, \delta^2)$  denotes that the random variable (r.v.)  $x$  follows the circular symmetric complex Gaussian distribution.  $\mathcal{O}(\cdot)$  stands for the big-O notation.

## II. SYSTEM MODEL

Consider an uplink single-cell cellular network consisting of  $K_{\text{total}}$  single-antenna users. The BS is equipped with  $N_r$  antennas and  $N_{\text{RF}}$  radio frequency (RF) chains such that  $N_{\text{RF}} < N_r$ , as shown in Fig. 1. Due to the sporadic user activity of mMTC, only a small number of  $K_a$  users are active in a transmission process, i.e.,  $K_a \ll K_{\text{total}}$ . Each active user has  $B$  bits of information to be transmitted in a block-fading channel. According to [12], [14], the channel vector  $\mathbf{h}_k$  from user  $k$  to the BS can be written as

$$\mathbf{h}_k = \sum_{p=1}^{P_c} \sum_{q=1}^{Q_p} \beta_{p,q} \mathbf{e}(\theta_{p,q}), \quad (1)$$

where  $P_c$  denotes the total number of clusters and within the  $p$ -th cluster there are  $Q_p$  sub-paths.  $\beta_{p,q}$  and  $\theta_{p,q}$  denote the gain and the angle of arrival (AOA) of the  $q$ -th sub-path within the  $p$ -th cluster. For the uniform linear array (ULA), the  $N_r \times 1$  array steering vector  $\mathbf{e}(\theta)$  can be expressed as

$$\mathbf{e}(\theta) = \left[ e^{-\frac{j2\pi d \sin(\theta)m}{\lambda}} \right]_{m \in J(N_r)}, \quad (2)$$

where  $J(N_r) = \{i - \frac{N_r-1}{2}, i = 0, 1, 2, \dots, N_r - 1\}$ ,  $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ ,  $\lambda$  is the signal wavelength, and  $d$  is the antenna spacing which is usually half of the signal wavelength.

To overcome the strong path loss in mmWave channels, a beamforming technique should be adopted. However, the BS cannot focus in any specific direction in grant-free random

access. The reason is that even if the location of all potential users is stored at the BS, which users are active is not prior information known to the BS. Besides, due to the constraint of hardware implementations and large energy consumption of RF chains, we have  $N_{\text{RF}} < N_r$ . Therefore, many beamforming methods in existing works [27], [28] cannot be applied in our system directly as  $N_{\text{RF}}$  narrow beams cannot cover the whole beam space. In this paper, we give a beamforming method based on the widely used Discrete Fourier Transform (DFT) based beamforming codebook [27], [28], to overcome the strong path loss of mmWave channels in grant-free random access. Specifically, the DFT based beamforming codebook, which is denoted by  $\mathbf{W}$ , can be written as

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{N_r}] \in \mathbb{C}^{N_r \times N_r}, \quad (3)$$

where

$$\begin{aligned} \mathbf{w}_i &= \frac{1}{\sqrt{N_r}} \mathbf{e}(\theta_i), \\ \theta_i &= \arcsin\left(\frac{2i-1}{N_r} - 1\right), \quad i = 1, 2, \dots, N_r. \end{aligned} \quad (4)$$

Consider the process of hardware implementations, the number of antennas is usually a multiple of the number of RF chains. Therefore, the  $\frac{N_r}{N_{\text{RF}}}$  consecutive beamforming vectors can be grouped and summed together to form a new beamforming vector  $\bar{\mathbf{w}}_i$ ,  $i = 1, 2, \dots, N_{\text{RF}}$ .  $\bar{\mathbf{w}}_i$  is expressed as

$$\bar{\mathbf{w}}_i = \gamma(\mathbf{w}_{1+\frac{N_r}{N_{\text{RF}}}(i-1)} + \mathbf{w}_{2+\frac{N_r}{N_{\text{RF}}}(i-1)} + \dots + \mathbf{w}_{\frac{N_r}{N_{\text{RF}}}i}), \quad (5)$$

where the parameter  $\gamma$  is set to constrain the power of receive beamforming, i.e.,  $\|\bar{\mathbf{w}}_i\|_2^2 = 1$ . Then the beamforming matrix  $\bar{\mathbf{W}}$  is obtained, where  $\bar{\mathbf{W}}$  is written as  $\bar{\mathbf{W}} = [\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \dots, \bar{\mathbf{w}}_{N_{\text{RF}}}] \in \mathbb{C}^{N_r \times N_{\text{RF}}}$ . By applying this beamforming method, the width of every beam is  $\frac{\pi}{N_{\text{RF}}}$ , thus the  $N_{\text{RF}}$  beams can cover the whole beam space, which means that the signals coming from all directions can be received by the BS.

In a typical URA scenario, all the users share a common codebook  $\mathbf{A}$ , which is denoted by  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N] \in \mathbb{C}^{L_p \times N}$ . The power of each codeword  $\mathbf{a}_n \in \mathbb{C}^{L_p \times 1}$  is constrained to 1, i.e.,  $\|\mathbf{a}_n\|_2^2 = 1$ . Let  $\delta_{n,k} \in \{0, 1\}$  denote whether user  $k$  transmits the codeword  $\mathbf{a}_n$ .  $\delta_{n,k}$  can be written as

$$\delta_{n,k} = \begin{cases} 1, & \text{active user } k \text{ transmits codeword } \mathbf{a}_n, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

After receive beamforming at the receiver, the beam domain channel vector of the active user  $k$  is denoted by  $\bar{\mathbf{h}}_k = \bar{\mathbf{W}}^H \mathbf{h}_k = [\bar{h}_{k,1}, \bar{h}_{k,2}, \dots, \bar{h}_{k,N_{\text{RF}}}]^T$ . Also, the random noise vector is denoted by  $\bar{\mathbf{z}} = \bar{\mathbf{W}}^H \mathbf{z} = [\bar{z}_1, \bar{z}_2, \dots, \bar{z}_{N_{\text{RF}}}]^T$ , where  $\mathbf{z}$  is modeled by a complex circular Gaussian random vector with i.i.d. components, i.e.,  $\mathbf{z} \sim \mathcal{CN}(0, \sigma_z^2 \mathbf{I})$ . Then the received signal on the  $b$ -th beam can be written as

$$y_b = \sum_{k=1}^{K_{\text{total}}} \sum_{n=1}^N \bar{h}_{k,b} \delta_{n,k} \mathbf{a}_n^T + \bar{z}_b, \quad (7)$$

By summarizing all the  $N_{\text{RF}}$  samples in a transmission

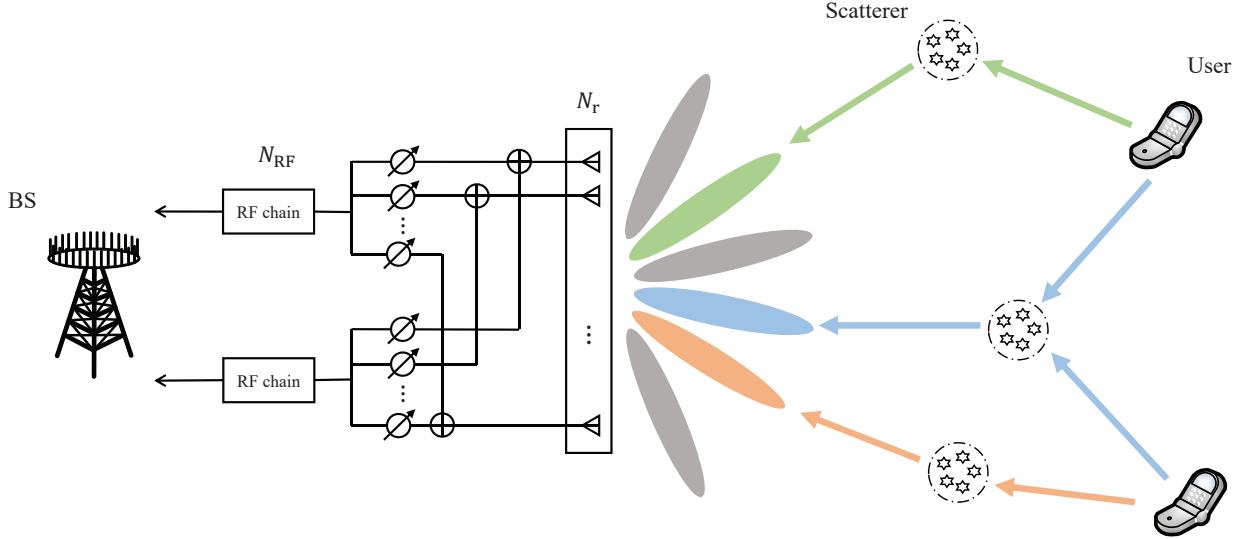


Fig. 1. The system model of our proposed scheme.

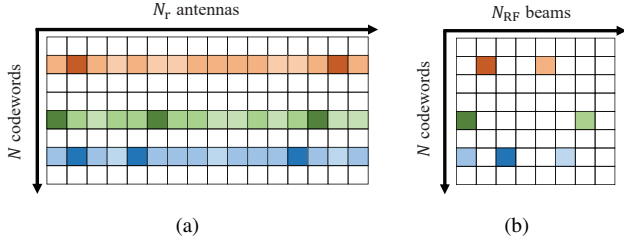


Fig. 2. (a) The matrix  $\mathbf{X}$  exhibits the sparsity of user activity; (b) The matrix  $\bar{\mathbf{X}}$  exhibits the sparsity of beam domain channel in mmWave bands.

block, the received signal can be recast as

$$\mathbf{Y} = \mathbf{A}\Delta\bar{\mathbf{H}} + \bar{\mathbf{Z}} = \mathbf{A}\bar{\mathbf{X}} + \bar{\mathbf{Z}}, \quad (8)$$

where  $\mathbf{Y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \dots, \mathbf{y}_{N_{\text{RF}}}^T] \in \mathbb{C}^{L_p \times N_{\text{RF}}}$ ,  $\bar{\mathbf{H}} = [\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \dots, \bar{\mathbf{h}}_{K_{\text{total}}}]^T \in \mathbb{C}^{K_{\text{total}} \times N_{\text{RF}}}$ ,  $\bar{\mathbf{X}} = \Delta\bar{\mathbf{H}} \in \mathbb{C}^{N \times N_{\text{RF}}}$  and  $\Delta = \{0, 1\}^{N \times K_{\text{total}}}$ . The matrix  $\Delta$  contains only  $K_a$  non-zero columns each of which having a non-zero entry.

For the matrix  $\mathbf{X} = \Delta\mathbf{H}$ , where  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{K_{\text{total}}}]^T$ , the  $i$ -th row of such matrix is given as

$$\mathbf{X}_{i,:} = \sum_{k=1}^{K_{\text{total}}} \delta_{i,k} \mathbf{h}_k^T. \quad (9)$$

The probability that  $\mathbf{X}_{i,:}$  is identically zero is given by  $(1 - 2^{-J})^{K_a}$ . Since  $2^J$  is significantly larger than  $K_a$ , the matrix  $\mathbf{X}$  is row-sparse, which is shown in Fig. 2(a). The reason is that only a small number of users are active due to the sporadic traffic of users, i.e.,  $K_a \ll K_{\text{total}}$ . For the same reason, the matrix  $\bar{\mathbf{X}}$  is also row-sparse. Moreover, due to the lack of scattering in mmWave bands, the signal propagates from the transmitter to the receiver through a small number of path clusters. This leads to the sparsity of mmWave massive MIMO channels in the beam domain as well, i.e., the channel vector  $\bar{\mathbf{h}}_k$  is sparse. Therefore, for the matrix  $\bar{\mathbf{X}}$ , the number of non-

zero entries of its columns is less than that of  $\mathbf{X}$ , which is shown in Fig. 2(b). Note that the total number of users  $K_{\text{total}}$  plays no role in the matrix  $\bar{\mathbf{X}}$ . This means that if the matrix  $\bar{\mathbf{X}}$  is recovered, only the codeword index is known to the BS, instead of the user's ID, which leads to the so-called unsourced property.

Let  $\mathcal{L}$  and  $\mathcal{K}_a$  denote the set of the recovered messages at the BS and the set of the active users, respectively. Each active user  $k \in \mathcal{K}_a$  expects to transmit  $B$  bits of information, i.e.,  $\mathbf{u}_k = \{0, 1\}^B$ . The performance in URA is evaluated by the probability of missed detection and false alarm, denoted by  $p_{\text{md}}$  and  $p_{\text{fa}}$  respectively, which can be given by:

$$p_{\text{md}} = \frac{1}{K_a} \sum_{k \in \mathcal{K}_a} \Pr(\mathbf{u}_k \notin \mathcal{L}), \quad (10)$$

$$p_{\text{fa}} = \frac{|\mathcal{L} \setminus \{\mathbf{u}_k | k \in \mathcal{K}_a\}|_c}{|\mathcal{L}|_c}, \quad (11)$$

and the error probability of the system is defined as

$$p_{\text{err}} = p_{\text{md}} + p_{\text{fa}}. \quad (12)$$

### III. PROPOSED URA SCHEME

In this section, we first review the studies of the CCS scheme in [20] and then propose a URA scheme. In the CCS scheme, each active user partitions the message into several sub-blocks and adds parity bits. The CS techniques detect the sub-blocks transmitted by active users in all sub-slots. A tree-based algorithm then stitched these sub-blocks to recover the original messages.

#### A. Encoding Process

The transmission strategy includes two encoders: tree encoder and CS encoder. The tree encoder uses a systematic linear block code based on random parity checks to add parity bits to every sub-block. The CS encoder maps each sub-block into a codeword in the common codebook.

1) *Tree Encoder*: Divide  $B$  bits message into  $S$  sub-blocks of size  $b_1, b_2, \dots, b_S$ , where  $\sum_{i=1}^S b_i = B$ . Let  $b_1 = J$  and  $b_s < J$ ,  $s = 2, 3, \dots, S$ . Each sub-block  $s$  is resized to length  $J$  by appending  $l_s = J - b_s$  parity bits, which is obtained by linear combinations of the information bits of the previous sub-blocks. Mathematically, define  $\mathbf{m}$  as a coded message, then we have  $\mathbf{m} = [\mathbf{m}(1), \mathbf{m}(2), \dots, \mathbf{m}(S)] = [\mathbf{b}(1), \mathbf{l}(1), \mathbf{b}(2), \mathbf{l}(2), \dots, \mathbf{b}(S), \mathbf{l}(S)] \in \{0, 1\}^{1 \times JS}$ . Herein,  $\mathbf{l}(s)$  is obtained by

$$\mathbf{l}(s) = \sum_{i=1}^{s-1} \mathbf{b}(i) \mathbf{G}_{i,s-1}, \quad (13)$$

where  $\mathbf{G}_{i,s-1} \in \{0, 1\}^{b_i \times l_s}$  is a binary matrix. Parity bits are computed using modulo-2 arithmetic and, as such, they remain binary. Every sub-block has the same size  $b_s + l_s = J$ , and the code rate  $R_{\text{tree}}$  is fixed as  $R_{\text{tree}} = \frac{B}{JS}$ .

2) *CS Encoder*: For each active user  $k$ ,  $\mathbf{m}^{(k)} = [\mathbf{m}^{(k)}(1), \mathbf{m}^{(k)}(2), \dots, \mathbf{m}^{(k)}(S)]$  is the coded message output by the tree encoder.  $\mathbf{m}^{(k)}(1), \mathbf{m}^{(k)}(2), \dots, \mathbf{m}^{(k)}(S)$  are mapped in to  $i_k(1), i_k(2), \dots, i_k(S)$ , which denote the indices of the codewords in the common codebook  $\mathbf{A} \in \mathbb{C}^{L_p \times N}$ , where  $N = 2^J$ . Then the active user  $k$  transmits the consecutive codewords of length  $L_p$ , i.e.,  $\mathbf{a}_{i_k(1)}, \mathbf{a}_{i_k(2)}, \dots, \mathbf{a}_{i_k(S)}$ .

## B. Decoding Process

The input to the decoder is the sum of the signals transmitted by active users plus noise after receive beamforming. The decoding process also consists of a CS decoder and tree decoder. The conventional CS decoder exploits CS techniques to recover the sub-blocks transmitted from all active users. The tree decoder forms code trees to piece these sub-blocks together to obtain the original messages.

1) *CS Decoder*: For each sub-slot  $s$ , the received signal can be expressed as

$$\mathbf{Y}_s = \mathbf{A} \Delta \bar{\mathbf{H}}_s + \bar{\mathbf{Z}}_s = \mathbf{A} \bar{\mathbf{X}}_s + \bar{\mathbf{Z}}_s, \quad (14)$$

$\bar{\mathbf{X}}_s$  is a row sparse matrix and can be recovered by CS techniques such as Approximate Message Passing (AMP) [29].

For rich-scattering environments, an accurate and widely used statistical model for the actual channel coefficients is the Gaussian model. However, in mmWave communications, the entries  $\bar{\mathbf{H}}_s$  cannot be approximated by a Gaussian distribution due to the lack of scatterers. Thus, we design a special activity detector for our considered scenario. Specifically, we approximate the unknown prior distribution with Gaussian mixture (GM) [13] and EM-GM-AMP [30] models for activity detection and channel estimation. The coefficients in the  $i$ -th column of  $\bar{\mathbf{X}}_s = [\bar{x}_{s,1}, \bar{x}_{s,2}, \dots, \bar{x}_{s,N_{\text{RF}}}]$  are approximated to be i.i.d with marginal pdf

$$p_X(x; \rho, \omega, \boldsymbol{\mu}, \boldsymbol{\nu}) = (1 - \rho) \delta(x) + \rho \sum_{i=1}^I \omega_i \mathcal{N}(x; \mu_i, \nu_i), \quad (15)$$

where  $\delta(\cdot)$  is the Dirac delta,  $\rho$  is the sparsity rate, and for the  $k$ -th GM component,  $w_k, \mu_k, \nu_k$  are the weight, mean, and variance, respectively. The sparsity of the vector is captured by the sparsity rate  $\rho$ . The weights, means, and variances can

be iteratively learned by the Expectation-Maximization (EM) algorithm.

For each sub-slot  $s$ , the CS algorithm outputs the estimation of  $\bar{\mathbf{X}}_s$ , i.e.,  $\hat{\bar{\mathbf{X}}}_s$ . Via maximum-ratio-combining (MRC), the activity detector  $\tilde{a}_k(s)$  is defined as

$$\tilde{a}_k(s) = \begin{cases} 1, & \sum_{i=1}^{N_{\text{RF}}} \eta_i \left| \hat{x}_{k,i}^{(s)} \right| \geq \epsilon, \\ 0, & \sum_{i=1}^{N_{\text{RF}}} \eta_i \left| \hat{x}_{k,i}^{(s)} \right| < \epsilon, \end{cases} \quad (16)$$

where  $\epsilon$  is a threshold, and  $\eta_i$  is expressed as

$$\eta_i = \frac{\left| \hat{x}_{k,i}^{(s)} \right|}{\sqrt{\sum_{j=1}^{N_{\text{RF}}} \left| \hat{x}_{k,j}^{(s)} \right|^2}}. \quad (17)$$

Through the activity detector, the indices of the transmitted codewords in the common codebook  $\mathbf{A}$  are obtained and collected in the set  $\mathcal{K}_s$ , which is written as  $\mathcal{K}_s = \{k \mid \tilde{a}_k(s) = 1, k = 1, 2, \dots, N\}$ . As the relationship between a sub-block and the corresponding codeword is a one-to-one mapping, if a codeword is detected, then the corresponding sub-block can be recovered automatically. The CS Decoder finally outputs the set of the sub-blocks  $\mathcal{L}_s = \{\mathbf{m}_k(s) \mid k \in \mathcal{K}_s\}$  and the corresponding estimated channel vectors  $\mathcal{H}_s = \{\hat{\mathbf{h}}_k^{(s)} \mid k \in \mathcal{K}_s\}$ . Notice that the index  $k$  cannot represent the identity of the active user. The information that is known at the BS is that a sub-block  $\mathbf{m}_k(s)$  is transmitted, it comes from a certain user and the estimated channel gain of that user is  $\hat{\mathbf{h}}_k^{(s)}$ . Besides, let  $|\mathcal{L}_s|_c = K_s$ ,  $K_s$  means the number of the sub-blocks collected in sub-slot  $s$ .  $K_s$  is usually less than  $K_a$ , i.e.,  $K_s \leq K_a$ , due to the following two reasons:

- i) Since all users use a common codebook, the messages from different users may share some sub-blocks, which is defined as collision.
- ii) Due to the poor channel condition and the mistake of the CS decoder, some sub-blocks may be lost.

2) *Tree Decoder*: The traditional tree decoder in [20] aims to recover the original messages transmitted from all active users by piecing together valid sequences of the sub-blocks drawn from  $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_S$ . As an initial step, the decoder fixes a sub-block in  $\mathcal{L}_1$  as the root of a tree and gets the parity bits of the next sub-block by (13). All sub-blocks in  $\mathcal{L}_2$  matching the parity bits are attached to the root. This process then moves forward. For every candidate path at stage  $s$ , parity bits are computed, and the matching sub-blocks in  $\mathcal{L}_s$  are attached to this path, forming new branches. This continues until the last sub-slot is reached. At this point, every surviving path is output as a valid tree message.

However, the traditional tree decoder has the following two problems:

- i) The loss of a sub-block from a particular user by the CS decoder finally leads to missed detection of the original message from that user.
- ii) The parity bits to be attended in every sub-block are fixed, which restricts the maximum active users that the system

Notation	Parameter Description
$\mathbf{m}_i(s)$	The $i$ -th sub-block detected in the $s$ -th sub-slot
$c_i[l]$	The index of the sub-block at stage $i$ in the $l$ -th path
$\mathbf{f}_k^{(s)}$	The beam pattern of the sub-block $\mathbf{m}_k(s)$
$\mathcal{T}_s[l]$	A set that contains the indexes of the detected sub-blocks that meet the parity constraints at stage $s$ for the $l$ -th path
$\mathcal{M}_s[l]$	A set that contains the detected sub-blocks that meet the parity constraints at stage $s$ for the $l$ -th path
$\mathcal{M}_s^{\text{hd}}[l]$	A set that contains the detected sub-blocks that meet the parity and beam pattern matching constraints at stage $s$ for the $l$ -th path

TABLE I. This list contains the key parameters encountered in Section IV.

can serve.

To tackle the above problems, we propose two beam-space tree decoders, which are based on hard decision and soft decision, respectively. The beam-space tree decoder with hard decision has low complexity, which is suitable to the scenario of massive connectivity. The beam-space tree decoder with soft decision considers the problem of packet loss, which can be applied to the scenario with poor channel condition.

#### IV. BEAM-SPACE TREE DECODER WITH HARD DECISION

The traditional tree decoder exploits the discriminating power of parity bits to stitch the sub-blocks together to form a valid message instead of the erroneous one. At any stage of a path during the decoding process, the sub-blocks meeting the parity constraints are attached to the path. Besides the valid sub-block, other attached sub-blocks are the ones that cannot be distinguished by the parity bits. These invalid sub-blocks may finally lead to an erroneous message output by the tree decoder. Notice that in all sub-slots, the sub-blocks sent by different users are received by different beams at the BS according to the location of the users and scatterers. Therefore, the discriminating power of beams can be exploited to distinguish the invalid sub-blocks that meet the parity constraints. By leveraging the beam dimension, the decoding process can be formulated as a problem of path search in the three-dimensional space, which is shown in Fig. 3.

To better describe the decoding process of the beam-space tree decoder with hard decision, Table I is given to summarize the important parameters encountered in this Section. Specifically, define the beam pattern of a sub-block as a set that contains the indices of the beams that receive the sub-block. Then different sub-blocks can be distinguished by their beam patterns. The beam pattern  $\mathbf{f}_k^{(s)}$  is written as  $\mathbf{f}_k^{(s)} = [f_{k,1}^{(s)}, f_{k,2}^{(s)}, \dots, f_{k,N_{\text{RF}}}^{(s)}] \in \{0, 1\}^{1 \times N_{\text{RF}}}$ . To get accurate beam patterns, assume the gains of the active beams obey a prior known Gaussian distribution, i.e.,  $\mathcal{N}(\mu_1, \delta_1^2)$ , where an "active" beam means that at least the signal from one user is received by the beam. And the gains of the inactive beams

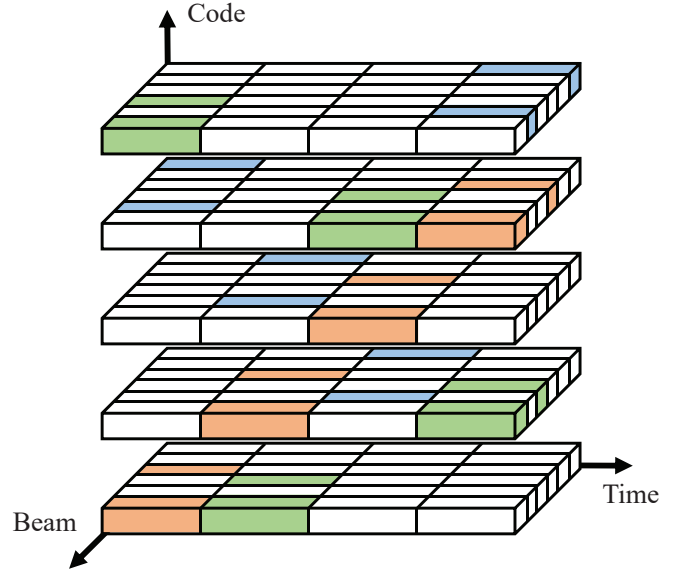


Fig. 3. Time-beam-code sparse pattern.

obey another known Gaussian distribution, i.e.,  $\mathcal{N}(\mu_2, \delta_2^2)$ , where  $\mu_1 > \mu_2$  and  $\delta_1 > \delta_2$ . For the gains of the inactive beams, as no signal is received or the signal experiences deep fading,  $\mu_2$  is close to zero. For the gains of the active beams,  $\delta_1$  is large as the signals experience random fading. Using these two prior distributions, the gains of the beams can be grouped into two classes. And for each class, the mean and variance of the samples are calculated and the prior distributions can be updated, i.e.,  $\mu_1 \rightarrow \hat{\mu}_1$ ,  $\mu_2 \rightarrow \hat{\mu}_2$ ,  $\delta_1 \rightarrow \hat{\delta}_1$  and  $\delta_2 \rightarrow \hat{\delta}_2$ . Then according to these updated distributions, we can give the decision rules of the beam patterns. Specifically,  $f_{k,m}^{(s)}$  is obtained by  $\hat{h}_{k,m}^{(s)}$ , which is expressed as

$$f_{k,m}^{(s)} = \begin{cases} 1, & \frac{P_1(|\hat{h}_{k,m}^{(s)}|)}{P_2(|\hat{h}_{k,m}^{(s)}|)} \geq 1, \\ 0, & \frac{P_1(|\hat{h}_{k,m}^{(s)}|)}{P_2(|\hat{h}_{k,m}^{(s)}|)} < 1. \end{cases} \quad (18)$$

where  $P_i(|\hat{h}_{k,m}^{(s)}|) = \frac{1}{\sqrt{2\pi}\delta_i} e^{-\frac{(|\hat{h}_{k,m}^{(s)}| - \mu_i)^2}{2\delta_i^2}}$ .

For this proposed beam-space tree decoder, take the decoding process of a certain user for example. At the first stage, a code tree is created and a detected sub-block in the first sub-slot becomes the root of the tree and forms the first path. The root sub-block is written as  $\mathbf{m}_{c_1[1]}(1)$  and its beam pattern is  $\mathbf{f}_{c_1[1]}^{(1)}$ . At later stages, the sub-blocks that meet the parity and the beam pattern matching constraints are kept. By meeting the parity constraints,  $\mathcal{T}_s[l]$  is written as

$$\mathcal{T}_s[l] = \{i \mid i \in \mathcal{K}_s, \mathbf{l}_i(s) = \sum_{j=1}^{s-1} \mathbf{b}_{c_j[l]}(j) \mathbf{G}_{j,s-1}\}. \quad (19)$$

And  $\mathcal{M}_s[l]$  is obtained by

$$\mathcal{M}_s[l] = \{\mathbf{m}_i(s) \mid i \in \mathcal{T}_s[l]\}. \quad (20)$$

After beam pattern matching, only the sub-blocks in  $\mathcal{M}_s^{\text{hd}}[l]$

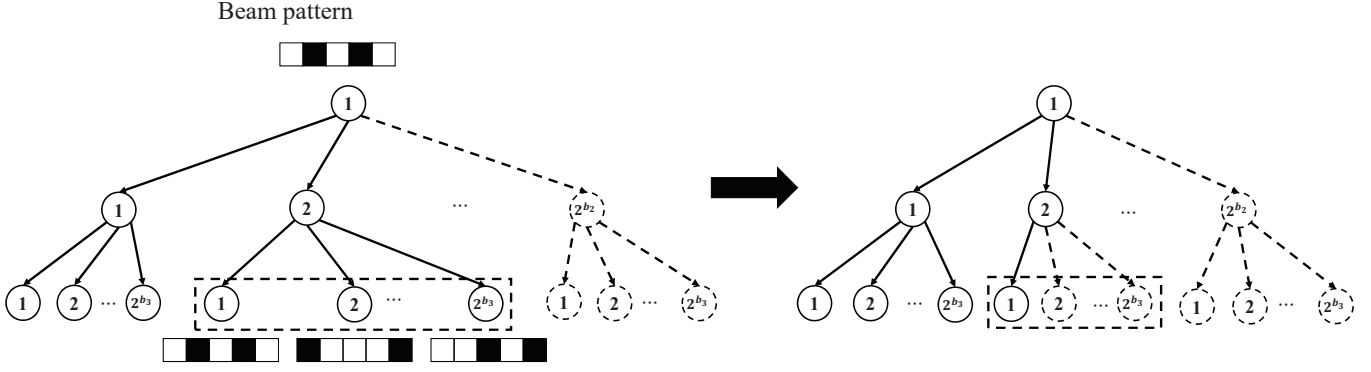


Fig. 4. The pruning process of the proposed beam-space tree decoder with hard decision.

---

**Algorithm 1 : Beam-Space Tree Decoder with Hard Decision**


---

**Input:**  $\mathcal{F}_i, \mathcal{L}_i, \mathcal{K}_i, i = 1, 2, \dots, S$

**Output:**  $\mathcal{L}$

- 1: **for**  $k \in \mathcal{K}_1$  **do**
  - 2:   **for**  $i = 2 : S$  **do**
  - 3:     For each path  $l$ :
  - 4:     Get  $\mathcal{M}_i^{\text{hd}}[l]$  according to (21).
  - 5:     **if**  $\mathcal{M}_i^{\text{hd}}[l]$  is empty **then**
  - 6:       Delete the path  $l$ .
  - 7:     **else**
  - 8:       Attach the sub-blocks in  $\mathcal{M}_i^{\text{hd}}[l]$  to path  $l$ , forming new branches.
  - 9:     **end if**
  - 10:   **end for**
  - 11:   Add the surviving paths rooted by  $\mathbf{m}_k(1)$  to  $\mathcal{L}$ .
  - 12: **end for**
- 

are survived.  $\mathcal{M}_s^{\text{hd}}[l]$  can be written as

$$\mathcal{M}_s^{\text{hd}}[l] = \{\mathbf{m}_i(s) \mid \mathbf{f}_{c_1[l]}^{(1)} \mathbf{f}_i^{(s)\text{T}} \neq 0, i \in \mathcal{T}_s[l]\}, \quad (21)$$

where  $\mathcal{M}_s^{\text{hd}}[l] \subseteq \mathcal{M}_s[l]$ . Also,  $\mathbf{f}_i^{(1)} \mathbf{f}_j^{(s)\text{T}} \neq 0$  means that the sub-blocks  $\mathbf{m}_i(1)$  and  $\mathbf{m}_j(s)$  are received by at least one same beam at the BS, then  $\mathbf{m}_i(1)$  and  $\mathbf{m}_j(s)$  have the probability to be transmitted by the same user. By beam pattern matching, the proposed beam-space tree decoder can reduce the number of surviving sub-blocks in each sub-slot, which improves the discriminating power of the decoder. A practical pruning process of this algorithm is shown in Fig. 4. The sub-block is received by several beams at the BS, which is shown in the beam pattern. For every candidate path at stage  $s$ , there are  $2^{b_s}$  candidate sub-blocks in the common codebook  $\mathbf{A}$  that meet the parity constraints according to the parity bits. A part of them are inactive, while another part of them are discriminated by the beam pattern matching. As shown in Fig. 4, the lines of the 2-nd and the  $2^{b_3}$ -th candidate sub-blocks change from solid lines to dashed lines, which means that they are deleted, as there is no overlap between their beam patterns and the beam pattern of the root sub-block at stage 1.

Finally, the proposed beam-space tree decoder with hard decision is summarized and given in Algorithm 1.  $\mathcal{F}_s$  is a set that contains the beam patterns of the sub-blocks detected in sub-slot  $s$ , where  $\mathcal{F}_s$  is expressed as  $\mathcal{F}_s = \{\mathbf{f}_k^{(s)} \mid k \in \mathcal{K}_s\}$ .

## V. BEAM-SPACE TREE DECODER WITH SOFT DECISION

As described above, the CS decoder cannot always detect all the transmitted sub-blocks because the received signals may experience deep fading. The loss of a sub-block by the CS decoder in any sub-slot finally leads to missed detection of the original message. This is because the traditional tree decoder and the beam-space tree decoder with hard decision just stitch the sub-blocks drawn from the output of the CS decoder. At any stage, according to the parity bits, the set of candidate sub-blocks can be obtained. At stage  $s$ , the tree decoder keeps the intersection between the candidate set and  $\mathcal{L}_s$ . In this proposed algorithm, we keep all the candidate sub-blocks and calculate the LLR values of them by implementing the MPA algorithm, which denotes the probability of whether the sub-blocks are transmitted. Then, we define a path metric to calculate the reliability of the consecutive sub-blocks and keep some reliable paths at every stage. Even if a sub-block in a sub-slot is missed, it is possible for the path to be reliable because the path metric measures the reliability of the entire path. Therefore, the purpose of packet loss recovery is achieved.

Specifically, take a user's decoding process for example. At stage  $s$  for the  $l$ -th path, the number of the candidate sub-blocks is  $2^{b_s}$ , and these sub-blocks are collected in the set  $\mathcal{M}'_s[l]$ .  $\mathcal{M}'_s[l]$  is expressed as

$$\mathcal{M}'_s[l] = \{\mathbf{m}_i(s) \mid i \in \mathcal{T}'_s[l]\}, \quad (22)$$

where

$$\mathcal{T}'_s[l] = \{i \mid \mathbf{l}_i(s) = \sum_{j=1}^{s-1} \mathbf{b}_{c_j[l]}(j) \mathbf{G}_{j,s-1}\}. \quad (23)$$

The difference between  $\mathcal{M}'_s[l]$  and  $\mathcal{M}_s[l]$  in (20) is that the candidate sub-blocks in  $\mathcal{M}'_s[l]$  are drawn according to the parity bits only, thus  $\mathcal{M}_s[l] \subseteq \mathcal{M}'_s[l]$ . To reduce interference, only the received signals of those candidate sub-blocks are kept, which is denoted by  $\tilde{\mathbf{Y}}_s[l] \in \mathbb{C}^{N_{\text{RF}} \times L_P}$ .  $\tilde{\mathbf{Y}}_s[l]$  is written as

$$\tilde{\mathbf{Y}}_s[l] = \mathbf{Y}_s - \sum_{k \in \{1, 2, \dots, N\} \setminus \mathcal{T}'_s[l]} \hat{\mathbf{h}}_k^{(s)} \mathbf{a}_k^{\text{T}}, \quad (24)$$

where  $k \in \{1, 2, \dots, N\} \setminus \mathcal{T}'_s[l]$  means that  $k$  is in the set  $\{1, 2, \dots, N\}$  instead of  $\mathcal{T}'_s[l]$ . Then a factor graph is formed, taking the corresponding codewords as variable nodes and

beam resources as resource nodes. Let  $K$  and  $T$  denote the number of variable nodes and resource nodes. To exploit the beam division property, the active beams in the beam pattern of the root sub-block form the resource nodes. Then the remaining received signal is defined as  $\tilde{\mathbf{Y}}_s^r[l] \in \mathbb{C}^{\|\mathbf{f}_{c_1[1]}^{(1)}\|_1 \times L_p}$ . A certain row of  $\tilde{\mathbf{Y}}_s^r[l]$  comes from the  $j$ -th row of  $\tilde{\mathbf{Y}}_s[l]$ , where  $j \in \{i \mid \mathbf{f}_{c_1[1]}^{(1)} = 1\}$ . For the sake of simplicity, define  $\mathbf{y}_t$  as the received signal at the  $t$ -th resource,  $\hat{h}_{k,t}$  as the estimated channel gain between the user transmitting the  $k$ -th codeword and the BS at the  $t$ -th resource,  $\mathbf{a}_k$  as the  $k$ -th possible codeword and  $x_k \in \{0, 1\}$  as a random variable that indicates whether the codeword  $\mathbf{a}_k$  is transmitted. Then  $\tilde{\mathbf{Y}}_s^r[l]$  is written as  $\tilde{\mathbf{Y}}_s^r[l] = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_T]$ , where  $\mathbf{y}_t$  can be expressed as

$$\mathbf{y}_t = \sum_{k=1}^K \hat{h}_{k,t} x_k \mathbf{a}_k + \mathbf{z}_t = \hat{h}_{k,t} x_k \mathbf{a}_k + \xi_{k,t}, \quad (25)$$

where

$$\xi_{k,t} = \mathbf{y}_t - \hat{h}_{k,t} x_k \mathbf{a}_k = \sum_{k' \neq k} \hat{h}_{k',t} x_{k'} \mathbf{a}_{k'} + \mathbf{z}_t, \quad (26)$$

and  $\xi_{k,t} = [\xi_{k,t,1}, \xi_{k,t,2}, \dots, \xi_{k,t,L_p}]$ . To reduce the computational complexity, we resort to Gaussian Approximation (GA) as in [31], and approximate  $\xi_{k,t,j}$  as a complex Gaussian-distributed random variable with mean  $\mu_{\xi_{k,t,j}}$  and variance  $\delta_{\xi_{k,t,j}}^2$ , i.e.,  $\xi_{k,t,j} \sim \mathcal{CN}(\mu_{\xi_{k,t,j}}, \delta_{\xi_{k,t,j}}^2)$ .  $\mu_{\xi_{k,t,j}}$  and  $\delta_{\xi_{k,t,j}}^2$  can be expressed as

$$\begin{aligned} \mu_{\xi_{k,t,j}} &= \sum_{k' \neq k} \hat{h}_{k',t} a_{k',j} P_{k' \rightarrow t}, \\ \delta_{\xi_{k,t,j}}^2 &= \sum_{k' \neq k} |\hat{h}_{k',t} a_{k',j}|^2 P_{k' \rightarrow t} (1 - P_{k' \rightarrow t}) + \delta_{z_{t,j}}^2, \end{aligned} \quad (27)$$

respectively, where

$$P_{k' \rightarrow t} = \frac{\exp(L_{k' \rightarrow t})}{1 + \exp(L_{k' \rightarrow t})}. \quad (28)$$

$L_{k \rightarrow t}$  is the log likelihood ratio (LLR) delivered from the  $k$ -th variable node to the  $t$ -th resource node. Also,  $L_{t \rightarrow k}$  denotes the LLR delivered from the  $t$ -th resource node to the  $k$ -th variable node, which is written as

$$L_{t \rightarrow k} = \ln \frac{p(\mathbf{y}_t | x_k = 1)}{p(\mathbf{y}_t | x_k = 0)}, \quad (29)$$

where

$$\begin{aligned} p(\mathbf{y}_t | x_k = 1) &= \prod_{j=1}^{L_p} \frac{1}{\sqrt{2\pi\delta_{\xi_{k,t,j}}^2}} \exp\left(-\frac{|y_{t,j} - \hat{h}_{k,t} a_{k,j} - \mu_{\xi_{k,t,j}}|^2}{2\delta_{\xi_{k,t,j}}^2}\right), \end{aligned} \quad (30)$$

$$p(\mathbf{y}_t | x_k = 0) = \prod_{j=1}^{L_p} \frac{1}{\sqrt{2\pi\delta_{\xi_{k,t,j}}^2}} \exp\left(-\frac{|y_{t,j} - \mu_{\xi_{k,t,j}}|^2}{2\delta_{\xi_{k,t,j}}^2}\right). \quad (31)$$

Besides,  $L_{k \rightarrow t}$  is given as

$$L_{k \rightarrow t} = \sum_{t' \neq t} L_{t' \rightarrow k}. \quad (32)$$

Finally,  $L_k$  can be expressed as

$$L_k = \sum_{t=1}^T L_{t \rightarrow k}. \quad (33)$$

Denote  $l^{(n)}$  as the new paths that are split from the  $l$ -th path. By implementing MPA, from path  $l$  at stage  $s$ , we can obtain the LLRs of the candidate sub-blocks in  $\mathcal{M}'_s[l]$ , which are written as  $L_s[l^{(n)}]$ ,  $n = 1, 2, \dots, 2^{b_s}$ . Learning from the way of list decoding [32], we define a path metric to calculate the reliability of the new branches from path  $l$  at stage  $s$ . Specifically, the PM of the new branch  $l^{(n)}$  is written as

$$\begin{aligned} \text{PM}_s[l^{(n)}] &= \sum_{i=1}^{s-1} \ln(1 + e^{-L_i[l]}) + \ln(1 + e^{-L_s[l^{(n)}]}) \\ &= \text{PM}_{s-1}[l] + \ln(1 + e^{-L_s[l^{(n)}]}), \end{aligned} \quad (34)$$

where  $L_i[l]$  denotes the LLR of the sub-block at stage  $i$  in the path  $l$ . The decoder calculates the PM of all the branches from every candidate path and keep some reliable paths at every stage. And at the last stage, the decoder outputs the most reliable path as the recovered message.

However, this scheme is not suitable in the case that collision occurs. As mentioned above, active users select codewords from a common codebook  $\mathbf{A}$ . Even if the dimension of  $\mathbf{A}$  is large, collisions may still occur. If the traditional tree decoder fixes a sub-block transmitted by two active users at the first stage, then the decoder finally outputs two valid tree messages. According to [25], we give  $\mathbf{E}\{C_{k,s}\}$  as the average number of collisions of  $k$  users on consistent  $s$  sub-blocks started from the first one, which is written as

$$\mathbf{E}\{C_{k,s}\} = \frac{\binom{K_a}{k}}{(N \prod_{i=2}^s 2^{b_s})^{k-1}}. \quad (35)$$

The collision of more than two users is ignored because the number is usually much smaller than 1. As  $s$  grows, the collision can be ignored when  $k = 2$ . In other words, it is impossible for the valid messages of the collision users to be the new branches of the same candidate path. However, when the depth of a code tree grows, the LLR of a sub-block has less impact on the PM of the entire path. Therefore, the remaining paths at stage  $s$  may all come from the new branches of the most reliable path at stage  $s - 1$ , leading to missed detection. Actually, there is no need to keep all new branches from a candidate path because the valid messages of collision users come from different candidate paths. Denote  $L_{\text{split}}$  as the number of splitting paths, which means that only  $L_{\text{split}}$  new branches from a candidate path are kept according to the PM. Then for the current stage, keep  $L_{\text{save}}$  most reliable paths, where  $L_{\text{save}} > L_{\text{split}}$ . This pruning process is shown in Fig. 5. The number of candidate sub-blocks at stage  $s$  is  $2^{b_s}$ , which is equivalent to the process in Fig. 4. At stage  $s$ , for the new branches of a path, a factor graph is created to implement the MPA algorithm and give the candidate sub-blocks LLRs. Then the most reliable  $L_{\text{split}}$  paths are kept, and others are deleted, which is shown in Fig. 5 that the lines of those deleted sub-blocks change from solid to dashed. Finally at stage  $s$ ,



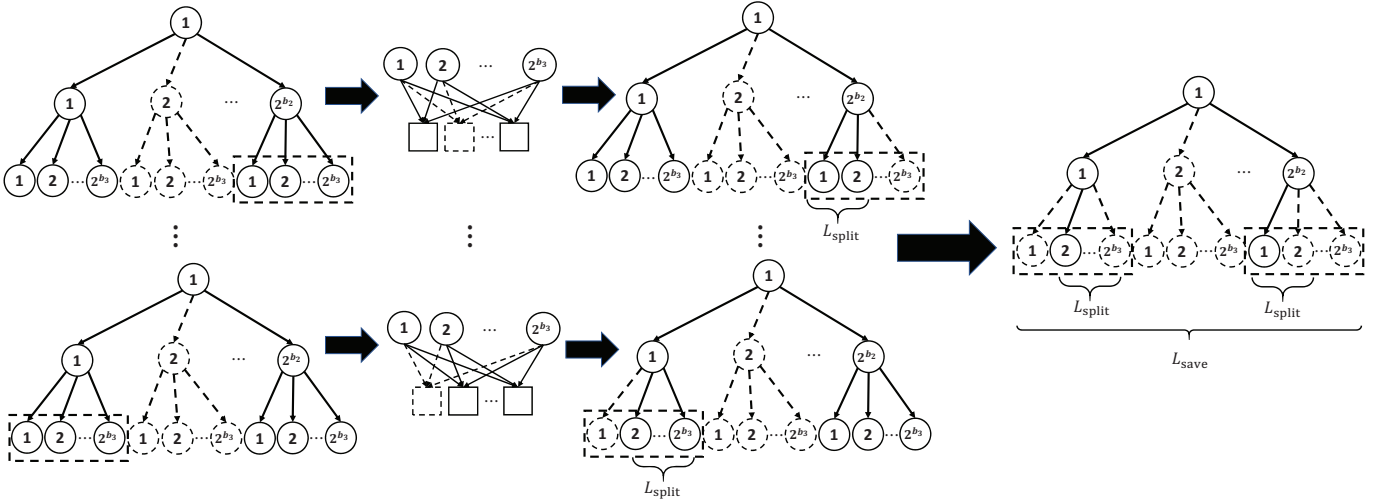


Fig. 5. The pruning process of the proposed beam-space tree decoder with soft decision.

---

**Algorithm 2 : Beam-Space Tree Decoder with Soft Decision**


---

**Input:**  $\mathcal{F}_i, \mathcal{L}_i, \mathcal{K}_i, i = 1, 2, \dots, S$

**Output:**  $\mathcal{L}$

```

1: for  $k \in \mathcal{K}_1$  do
2:   for  $i = 2 : S$  do
3:     if  $i \leq S'$  then
4:       For each path  $l$ :
5:         Get  $\mathcal{M}'_i[l]$  according to (22).
6:         Form a factor graph and implement MPA algorithm to
           get LLR values of the sub-blocks in  $\mathcal{M}'_i[l]$ .
7:         Calculate  $\text{PM}_i[l^{(n)}]$  by (34), keep the  $L_{\text{split}}$  most reli-
           able paths and delete others.
8:         For all the new paths:
9:           Keep the  $L_{\text{save}}$  most reliable paths and delete others.
10:      else
11:        For each path  $l$ :
12:          Get  $\mathcal{M}_i^{\text{hd}}[l]$  according to (21).
13:          if  $\mathcal{M}_i^{\text{hd}}[l]$  is empty then
14:            Delete the path  $l$ .
15:          else
16:            Attach the sub-blocks in  $\mathcal{M}_i^{\text{hd}}[l]$  to path  $l$ , forming
              new branches.
17:          end if
18:        end if
19:      end for
20:      For every pair  $\langle \mathbf{u}_m, \mathbf{u}_n \rangle$  in the surviving paths:
21:        Get  $P_s(\mathbf{u}_m, \mathbf{u}_n)$  according to (36).
22:        if  $P_s(\mathbf{u}_m, \mathbf{u}_n) > \tau$  then
23:          Keep the more reliable one according to (34) and delete the
            other.
24:        end if
25:      Add the remaining paths to  $\mathcal{L}$ .
26:    end for

```

---

for the  $2^{b_s-1}L_{\text{split}}$  paths, the most reliable  $L_{\text{save}}$  paths are kept and others are deleted.  $L_{\text{split}}$  is chosen according to the trade-off between the computational complexity and decoding performance of the decoder.

At the last stage, the number of messages output by the decoder cannot be determined because whether a collision occurs is unknown to the decoder. Notice that the traditional tree decoder outputs all the paths meeting the parity constraints

as valid messages. When a collision occurs, the traditional tree decoder outputs several paths as the recovered messages. Besides, more parity bits are pushed towards later stages to reduce the probability of error according to [20]. Thus, we exploit the discriminating power of the parity bits at later stages to output the results of the beam-space tree decoder with soft decision. To summarize, list decoding is implemented at the former  $S'$  stages to keep the missed sub-blocks, and the sub-blocks that are full of parity bits are exploited to prune the erroneous paths at the latter  $S - S'$  stages.

However, due to the loss packet recovery, the missed detection rate decreases while the false alarm rate increases. The reason is that some undetected sub-blocks may be kept in the decoding process. Some of them are not transmitted actually, which may lead to false alarm. As the valid messages from collision users are not similar with each other, the invalid messages can be discriminated. Specifically, define a similarity metric  $P_s(\mathbf{u}_i, \mathbf{u}_j)$ , which is denoted by

$$P_s(\mathbf{u}_i, \mathbf{u}_j) = \frac{\sum_{s=1}^S \mathbb{I}(\mathbf{b}_i(s) = \mathbf{b}_j(s))}{S}, \quad (36)$$

where  $\mathbb{I}(\cdot)$  is the indicator function,  $\mathbf{u}_i$  and  $\mathbf{u}_j$  are two messages output by a code tree. Calculate every pair of the outputs from a code tree, if  $P_s(\mathbf{u}_i, \mathbf{u}_j) > \tau$ , then keep the more reliable one according to the PM, otherwise keep both, where  $\tau$  is a threshold. By leveraging the similarity metric, the invalid messages meeting the parity constraints are deleted. The beam-space tree decoder with soft decision is summarized and given in Algorithm 2.

## VI. PERFORMANCE ANALYSIS

The performance of our proposed URA system is connected with the reliability of CS techniques in each sub-slot and the efficiency of message stitching across different sub-slots. In the remainder of this section, we ignore the collision that different active users share a sub-block in the first sub-slot.

Take the decoding process of user  $k$  for example. Let  $p_{cs}$  be the probability that at least one sub-block of user  $k$  is not output by the CS decoder,  $p_{tree}$  be the probability of error,  $p_{tree}^{md}$  be the probability of missed detection, and  $p_{tree}^{fa}$  be the probability of false alarm.  $p_{tree}$  is written as

$$\begin{aligned} p_{tree} &= p_{tree}^{md} + p_{tree}^{fa} \\ &= p_{cs} p_{tree|cs}^{md} + p_{\bar{cs}} p_{tree|\bar{cs}}^{md} + p_{tree}^{fa}, \end{aligned} \quad (37)$$

where  $p_{\bar{cs}} = 1 - p_{cs}$ , which denotes the probability that the CS decoder is error-free.  $p_{tree|cs}^{md}$  denotes the probability that the message of user  $k$  is not output by the tree decoder in the case that the CS decoder is error-free.

### A. Tree Code Analysis

In this subsection, we analyze the error probability of all the tree decoders. Denote  $p_{tree}$ ,  $\tilde{p}_{tree}$ ,  $\hat{p}_{tree}$  as the error probability of the traditional tree decoder, the beam-space tree decoder with hard decision and the beam-space tree decoder with soft decision.

For the traditional tree decoder and the beam-space tree decoder with hard decision, it is obvious that  $p_{tree|cs}^{md} = 1$ ,  $\tilde{p}_{tree|cs}^{md} = 1$  because these two decoders cannot deal with the problem of packet loss. In contrast,  $p_{tree|\bar{cs}} = 0$ ,  $\tilde{p}_{tree|\bar{cs}} = 0$  because there is no missed detection when the CS decoder is error-free. Thus, we use  $p_{tree}^{fa}$ ,  $\tilde{p}_{tree}^{fa}$  to analyze the performance of the two decoders, respectively.

Let  $L^{(j)}$ ,  $\tilde{L}^{(j)}$  be the number of erroneous paths of the traditional tree decoder and the beam-space tree decoder with hard decision at stage  $j$ , respectively. Define  $p_{match}$  as the probability of the event that the intersection between the beam patterns of two sub-blocks are not empty. Assume the signal transmitted from an active user is received by  $N_b$  beams at the BS. According to (21), the beam pattern matching is implemented by  $\mathbf{f}_i^{(1)} \mathbf{f}_j^{(s)T} \neq 0$ . Then  $p_{match}$  is given as

$$p_{match} = \frac{\sum_{i=1}^{N_b} \binom{N_b}{i} \binom{N_{RF} - N_b}{N_b - i}}{\binom{N_{RF}}{N_b}} = 1 - \frac{\binom{N_{RF} - N_b}{N_b}}{\binom{N_{RF}}{N_b}}. \quad (38)$$

$p_{match} \ll 1$  because of the lack of scatterers in the mmWave communication system, i.e.,  $N_b \ll N_{RF}$ .

For the beam-space tree decoder with hard decision, we give the expected values of  $\tilde{L}^{(j)}$  in Theorem 1.

**Theorem 1:** The expected values of  $\tilde{L}^{(j)}$ , which is denoted by  $\mathbb{E}[\tilde{L}^{(j)}]$ , can be expressed as

$$\mathbb{E}[\tilde{L}^{(j)}] = \sum_{q=2}^j p_{match}^{j-q+1} K^{j-q} (K-1) \prod_{s=q}^j 2^{-l_s}, \quad (39)$$

where  $K = K_a$ ,  $j = 2, \dots, S$ .

*Proof:* According to [20], for  $j \geq 3$ ,  $\mathbb{E}[L^{(j)}]$  is written as

$$\mathbb{E}[L^{(j)}] = 2^{-l_j} \mathbb{E}[L^{(j-1)}] + 2^{-l_j} (K-1). \quad (40)$$

For  $j = 2$ ,  $\mathbb{E}[L^{(2)}]$  is given as

$$\mathbb{E}[L^{(2)}] = (K-1) 2^{-l_2}. \quad (41)$$

By implementing beam pattern matching, the expected erroneous paths  $\mathbb{E}[\tilde{L}^{(j)}]$  at every stage are reduced. For  $j = 2$ ,  $\mathbb{E}[\tilde{L}^{(2)}]$  can be expressed as

$$\mathbb{E}[\tilde{L}^{(2)}] = p_{match} (K-1) 2^{-l_2}. \quad (42)$$

For  $j \geq 3$ ,  $\mathbb{E}[L^{(j)}]$  is written as

$$\mathbb{E}[\tilde{L}^{(j)}] = 2^{-l_j} p_{match} \mathbb{E}[\tilde{L}^{(j-1)}] + 2^{-l_j} p_{match} (K-1). \quad (43)$$

Using (42) as initial condition,  $\mathbb{E}[\tilde{L}^{(j)}]$  is rewritten as (39). ■

Based on Theorem 1, we give the upper bound of the false alarm rate of the beam-space tree decoder with hard decision in Corollary 1.

**Corollary 1:**  $\tilde{p}_{tree}^{fa}$  is bounded by

$$\tilde{p}_{tree}^{fa} \leq \mathbb{E}[\tilde{L}^{(S)}]. \quad (44)$$

*Proof:*  $\tilde{p}_{tree}^{fa}$  is the probability of false alarm, which means that the number of erroneous paths at the last stage is at least one. Therefore,  $\tilde{p}_{tree}^{fa}$  is written as

$$\tilde{p}_{tree}^{fa} = \Pr(\tilde{L}^{(S)} \geq 1) \leq \mathbb{E}[\tilde{L}^{(S)}]. \quad (45)$$

The inequality (45) is obtained by the application of Markov inequality. ■

For the beam-space tree decoder with soft decision,  $\hat{p}_{tree|cs}^{md} < 1$  because the decoder considers the problem of packet loss.  $\hat{p}_{tree|\bar{cs}}^{md} > 0$  because the decoder keeps limit reliable paths at every stage. As  $L_{save}$  grows,  $\hat{p}_{tree|\bar{cs}}^{md}$  decreases. Besides, notice that the decoder outputs the most reliable path at the last stage. If the recovered message is invalid, then missed detection and false alarm occur simultaneously, i.e.,  $\hat{p}_{tree}^{md} = \hat{p}_{tree}^{fa}$ .

### B. CS Analysis

A fundamental limitation of compressed sensing is that the required signal dimension  $L_p$  to reliably identify a subset of  $K_a$  transmitted codewords among a set consisting of  $N$  codewords in the common codebook scales as  $L_p = \mathcal{O}(K_a \log \frac{N}{K_a})$ .  $L_p$  is almost linearly with  $K_a$ .  $K_a$  is bounded by  $K_a = \mathcal{O}(L_p / \log \frac{N}{L_p})$ .

In our scenario, although  $K_a$  active users transmit codewords simultaneously, only a small number of codewords is received by a certain beam at the BS. The average number of codewords received by a beam is given as

$$\bar{K}_a = \frac{\binom{N_{RF}-1}{N_b-1}}{\binom{N_{RF}}{N_b}} K_a = \frac{N_b K_a}{N_{RF}}. \quad (46)$$

Denote  $p_b = \frac{N_b}{N_{RF}}$ , then  $K_a$  is bounded by

$$K_a = \mathcal{O}\left(\frac{L_p}{p_b \log \frac{N}{L_p}}\right). \quad (47)$$

Similarly,  $L_p$  is bounded by

$$L_p = \mathcal{O}\left(\bar{K}_a \log \frac{N}{\bar{K}_a}\right) = \mathcal{O}\left(p_b K_a \log \frac{N}{p_b K_a}\right). \quad (48)$$

Considering the limitation of both the CS decoder and the tree decoder, we give the upper bound of the number of active users  $K_a$  in Lemma 1.

**Lemma 1:** In our scenario, the number of active users, i.e.,  $K_a$ , is bounded by

$$K_a \leq \min \left( c_1 \frac{L_p}{p_b \log \frac{N}{L_p}}, \frac{2^{J(1-R_{\text{tree}})+1}}{p_b} \right), \quad (49)$$

where  $c_1$  is a constant.

*Proof:* Denote the average number of sub-blocks detected in a certain beam as  $K_v$ , where  $K_v = p_b K_a$ . According to the limitation of the tree decoder given in [33],  $K_v$  is bounded by  $K_v \leq 2^{J(1-R_{\text{tree}})+1}$ . Substituting  $K_v$  by  $K_a$  in the inequality, the bound of  $K_a$  is obtained, i.e.,  $K_a \leq \frac{2^{J(1-R_{\text{tree}})+1}}{p_b}$ . ■

### C. Asymptotic Analysis

In this subsection, we study the proposed algorithms in the context of large settings. The asymptotic analysis in [20] shows that the probability of false alarm goes to zero in the logarithmic regime with constant code rate  $R_{\text{tree}}$ . In this subsection, we analyze the impact of the beam division property on system performance in the asymptotic regime while fixing the number of parity bits.

**Lemma 2:** Fix  $N_{\text{RF}} = \alpha K_a$ , for some  $\alpha < 1$  and consider the number of active users  $K_a \rightarrow \infty$ . For the CS decoder, the required signal dimension  $L_p$  is given as

$$L_p = \mathcal{O} \left( \frac{1}{c_2} \log(c_2 N) \right), \quad (50)$$

where  $c_2$  is a constant.

*Proof:* According to (48),  $L_p$  is rewritten as

$$\begin{aligned} L_p &= \mathcal{O} \left( K_a p_b \log \frac{N}{K_a p_b} \right) = \mathcal{O} \left( \frac{N_b K_a}{N_{\text{RF}}} \log \frac{N}{\frac{N_b K_a}{N_{\text{RF}}}} \right) \\ &= \mathcal{O} \left( \frac{1}{c_2} \log(c_2 N) \right). \end{aligned} \quad (51)$$

This completes the proof. ■

Lemma 2 shows that as the number of RF chains and active users increases together while keeping a fixed ratio, the number of active users  $K_a$  plays no role in the required signal dimension  $L_p$ . This means that the decoding performance of the CS decoder is guaranteed by exploiting the beam division property while keeping the dimension of the common codebook  $\mathbf{A}$ .

**Theorem 2:** Fix  $N_{\text{RF}} = \alpha K_a$ ,  $J = \log \frac{K_a}{\varepsilon}$  for some  $0 < \alpha < 1$ ,  $\varepsilon \ll 1$ , consider the number of active users  $K_a \rightarrow \infty$  and the number of paths  $L_{\text{save}} \rightarrow \infty$ . The decoding performance of the beam-space tree decoder with hard decision and the beam-space tree decoder with soft decision, i.e.,  $\tilde{p}_{\text{tree}}^{\text{fa}}$  and  $\hat{p}_{\text{tree}}^{\text{fa}}$ , is close to zero.

*Proof:* For the beam-space tree decoder with hard decision, according to (38),  $p_{\text{match}} K_a = \left( 1 - \frac{\binom{N_{\text{RF}} - N_b}{N_b}}{\binom{N_{\text{RF}}}{N_b}} \right) \frac{1}{\alpha} N_{\text{RF}} \rightarrow$

$\frac{N_b^2}{\alpha}$ . For the sake of similarity, assume  $2^{-ls} = c_3 \ll 1$  for  $s = 2, 3, \dots, S$ . Then  $\mathbb{E}[\tilde{L}^S]$  is approximated as

$$\begin{aligned} \mathbb{E}[\tilde{L}^S] &\rightarrow \sum_{q=2}^S \left( \frac{N_b^2}{\alpha} \right)^{S-q+1} c_3^{S-q+1} = \sum_{q=2}^S \left( \frac{c_3 N_b^2}{\alpha} \right)^{S-q+1} \\ &= \frac{c_4 (1 - c_4^{S-1})}{1 - c_4} \approx c_4, \end{aligned} \quad (52)$$

where  $c_4$  is a constant.

According to Corollary 1,  $\tilde{p}_{\text{tree}}^{\text{fa}}$  is rewritten as

$$\tilde{p}_{\text{tree}}^{\text{fa}} \leq \mathbb{E}[\tilde{L}^{(S)}] \approx c_4 \ll 1. \quad (53)$$

For the beam-space tree decoder with soft decision, let the number of paths  $L_{\text{save}} \rightarrow \infty$ . At every stage, although  $2^{b_s} \rightarrow \infty$  sub-blocks are drawn by the checking relationship, most of the new paths are not reliable. Only  $p_{\text{match}} K_a + 1 \rightarrow \frac{N_b^2}{\alpha} + 1$  codewords are needed to be verified. Without the process of pruning,  $\left( \frac{N_b^2}{\alpha} + 1 \right)^{S-1}$  possible paths are kept at the last stage. As  $L_{\text{save}} \rightarrow \infty$ , the valid path is reliable and kept. At the last stage, the valid path has a high probability of being the most reliable path according to the PM and being output by the decoder as a result. ■

It can be seen from Theorem 2 that in the regime, the decoding performance of the beam-space tree decoder with hard decision and the beam-space tree decoder with soft decision is guaranteed by exploiting the beam division property.

### D. Computational Complexity Analysis

We denote  $C_{\text{tree}}$ ,  $\tilde{C}_{\text{tree}}$ ,  $\hat{C}_{\text{tree}}$  as the computational complexity of the traditional tree decoder, the beam-space tree decoder with hard decision and the beam-space tree decoder with soft decision. For the sake of simplicity, let  $K$  be the candidate sub-blocks in every sub-slot and ignore the collision. According to [20], the expected computational complexity of the traditional tree decoder, i.e.,  $\mathbb{E}[C_{\text{tree}}]$ , is given as

$$\mathbb{E}[C_{\text{tree}}] = (S-1)K + \sum_{j=2}^{S-1} \mathbb{E}[L^{(j)}]K. \quad (54)$$

The computational complexity of the beam-space tree decoder with hard decision, i.e.,  $\mathbb{E}[\tilde{C}_{\text{tree}}]$ , is written as

$$\mathbb{E}[\tilde{C}_{\text{tree}}] = (S-1)K + \sum_{j=2}^{S-1} \mathbb{E}[\tilde{L}^{(j)}]K. \quad (55)$$

Notice that  $\mathbb{E}[\tilde{C}_{\text{tree}}] \ll \mathbb{E}[C_{\text{tree}}]$  because  $\mathbb{E}[\tilde{L}^{(j)}] \ll \mathbb{E}[L^{(j)}]$ . Therefore, the beam-space tree decoder with hard decision has low computational complexity.

For the beam-space tree decoder with soft decision, the computational complexity  $\hat{C}_{\text{tree}}$  is given as

$$\hat{C}_{\text{tree}} = N_b 2^{b_2} I_{\text{max}} + \sum_{j=3}^S L_{\text{save}} N_b 2^{b_j} I_{\text{max}}, \quad (56)$$

where  $2^{b_j}$  and  $N_b$  are the number of variable nodes and resource nodes of the factor graph,  $I_{\text{max}}$  is the max number of iterations of MPA.

## VII. NUMERICAL RESULTS

In this section, we investigate the performance of our proposed scheme in terms of error probability. We consider a simulation scenario where the BS employs a ULA antenna array with  $N_r = 256$ , and the RF chains are set to  $N_{RF} = 16$ . The multipath channel consists of  $P_c = 3$  clusters each of which containing  $Q_p = 10$  sub-paths. It is assumed that the location of users obeys a two-dimensional Poisson distribution. The transmit power is fixed as  $P = 20$  dBm for all users. For the tree code scheme,  $B = 94$  bits of information are split into  $S = 32$  sub-blocks with length  $J = 10$  bits. Data profile and parity profile are given as  $[10, 3, \dots, 3, 0, 0, 0]$  and  $[0, 7, \dots, 7, 10, 10, 10]$ , respectively. For the beam-space tree decoder with soft decision,  $L_{save} = 24, L_{split} = 8$ . As a reference, we compare the proposed beam-space tree decoders with the traditional tree decoder [20] and the CB-CS algorithm [22]. For the sake of simplicity, in the following figures, TD means tree decoder, HD means hard decision, and SD means soft decision.

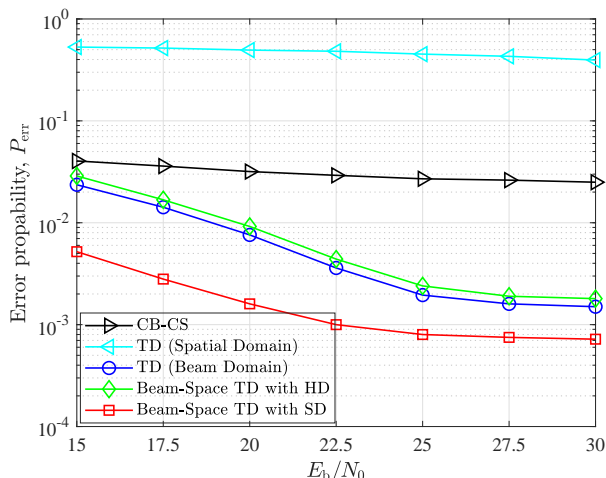


Fig. 6. The error probability of different URA schemes versus  $E_b/N_0$  when  $K_a = 50$ .

Fig. 6 depicts the error probability of different URA schemes versus  $E_b/N_0$  when  $K_a = 50$ . The difference between the tree decoder (beam domain) and the tree decoder (spatial domain) is whether the tree decoding process is performed after receive beamforming. Comparing these two algorithms, we can see that the decoding performance is significantly improved after receive beamforming because the detection performance of CS techniques is improved by exploiting the sparsity of beam domain channel and the beamforming gain. Besides, at the BS the signals are received after beamforming, which means that the dimension of the received signals is determined by the number of RF chains instead of the number of antennas. Therefore, the high dimension of antennas can not be exploited by the CB-CS algorithm in mmWave scenarios. In addition, there is a gap between the beam-space tree decoder with hard decision and the traditional tree decoder (beam domain). The reason is that getting accurate beam patterns of the sub-blocks, which means that whether the signal is received by every beam at the BS

should be accurately determined, is harder than the activity detection of these sub-blocks. Therefore, the beam patterns of the sub-blocks obtained by (18) may not always be accurate, resulting in a decrease in the decoding performance. Besides, it is observed that for the considered  $E_b/N_0$ , the beam-space tree decoder with soft decision achieves the best performance. Such advantages come from the fact that this algorithm considers the problem of packet loss.

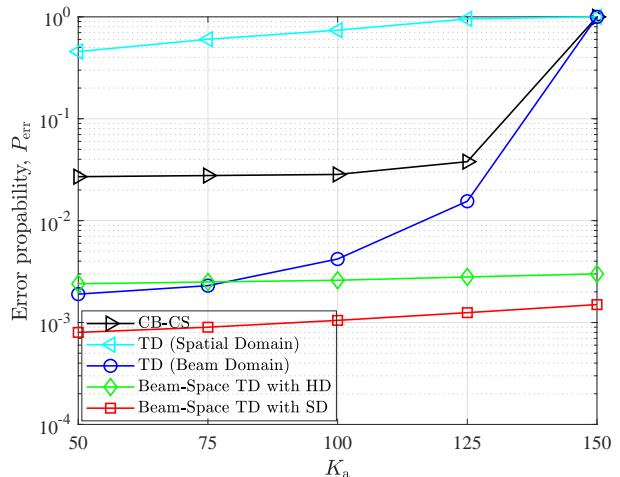
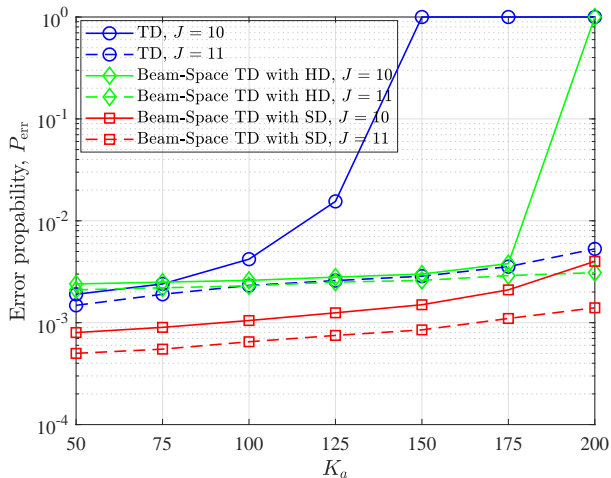


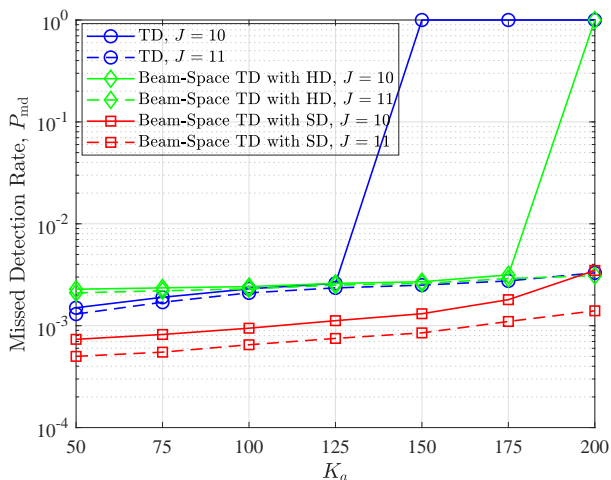
Fig. 7. The error probability of different URA schemes versus  $K_a$  when  $E_b/N_0 = 25$  dB.

Fig. 7 plots the decoding performance with different active users when  $E_b/N_0 = 25$  dB. Initially, in the regime with a few number of active users, the error probability of all algorithms increases slowly. However, when the number of active users continues to increase, the error probability of the tree decoder increases sharply to 1, which means that the tree decoder cannot discriminate and stitch the valid sub-blocks when the number of active users is large. Actually, allocating more parity bits can help the traditional tree decoder accommodate more active users while the computational complexity of the CS techniques and the tree decoders grows. It is observed that the error probability of the two beam-space tree decoders increases slowly when the number of active users is large. Such an advantage of the beam-space tree decoders mainly comes from enhancing the discriminating power of the decoder by exploiting beam resources.

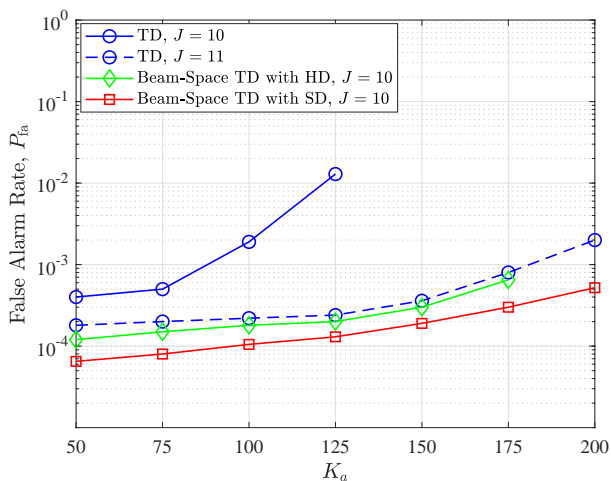
Then we study the impact of the number of parity bits on the system performance. For  $J = 11$ , data profile and parity profile are given as  $[11, 3, \dots, 3, 0, 0, 0]$  and  $[0, 8, \dots, 8, 11, 11, 11]$ , respectively. In Fig. 8, we compare the performance of the tree decoder (beam domain), the beam-space tree decoder with hard decision, and the beam-space tree decoder with soft decision. It is observed that allocating more parity bits can improve the decoding performance, which is seen in Fig. 8(a). Notice that as the number of active users grows, the error probability of the tree decoder and the beam-space tree decoder with hard decision increase sharply to 1. This is because the number of parity bits added to sub-blocks is fixed, the max number of active users the system can serve is limited. Once beyond the limit, the decoders cannot recover



(a) The error probability.



(b) The missed detection rate.



(c) The false alarm rate.

Fig. 8. The performance of different URA schemes with different sub-block lengths versus  $K_a$  when  $E_b/N_0 = 25$  dB.

any original message of the active users, thus  $p_{md} = 1$ ,  $p_{fa} = 0$ . As the beam-space tree decoder with hard decision exploits the beam division property, more active users can be served by the system. For the beam-space tree decoder with soft decision, the number of candidate sub-blocks to be kept is determined by the list numbers, not the parity bits. Therefore, the decoder is implemented successfully as the number of active users is sufficiently large. Besides, Fig. 8(b) shows that the missed detection rate of the tree decoder (beam domain) and the beam-space tree decoder with hard decision decreases slowly as the number of parity bits increases. This is because these two algorithms do not deal with the problem of packet loss. Besides, as the number of parity bits is increased, the discriminating power of the tree decoder is improved. Therefore, the false alarm rate of the tree decoder (beam domain) decreases, which is shown in Fig. 8(c). Since the beam-space tree decoders exploit beam resources to improve the discriminating power, there is no false alarm when  $J = 11$ .

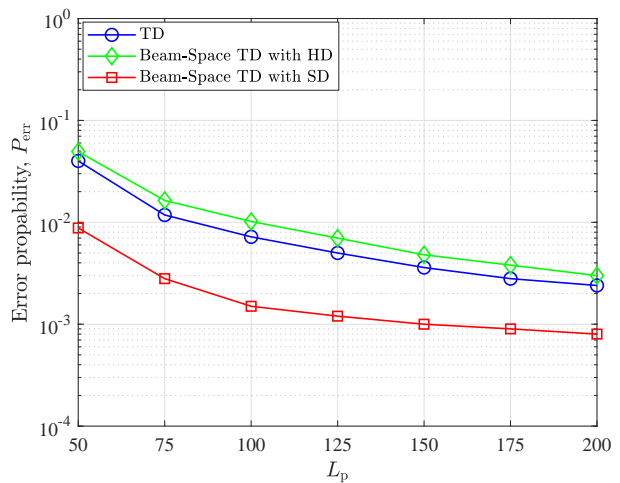


Fig. 9. The error probability of different URA schemes versus  $L_p$ .

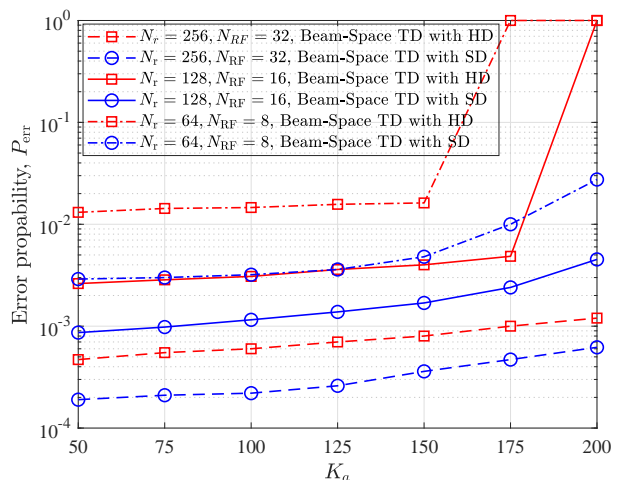


Fig. 10. The error probability of different URA schemes versus  $K_a$  when the BS is equipped with different antennas and RF chains and  $E_b/N_0 = 25$  dB.

Fig. 9 depicts the error probability of different URA schemes with different spectral efficiency when  $K_a = 50$ . The total spectral efficiency is calculated by  $\frac{K_a B}{L_p S}$ . As the number of observations controls the spectral efficiency, the code length  $L_p$  is set as the axis. It can be seen that as the spectral efficiency increases,  $L_p$  decreases, and the error probability increases. The reason is that as fewer observations are obtained, the performance of activity detection and channel estimation decreases.

Next, we evaluate the decoding performance with different active users when the BS is equipped with different antennas and RF chains in Fig. 10. It can be observed that as the number of antennas and RF chains grows, the error probability decreases, and more active users can be served. The reason is that the BS can generate more narrow beams for the receive beamforming simultaneously, which means that the spatial resolution of the beams is improved, and the beam division property can be exploited completely.

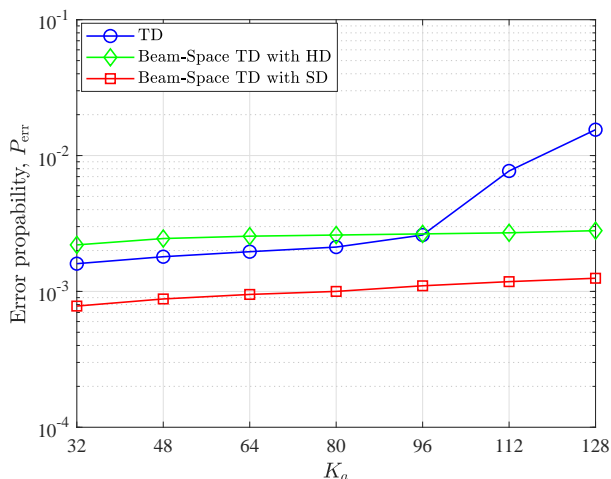


Fig. 11. The error probability of different URA schemes versus  $K_a$  while keeping a fixed ratio between antennas and users.

Besides, Fig. 11 plots the increasing number of users and antennas together while keep a fixed ratio, i.e.,  $\frac{K_a}{N_r} = \frac{1}{2}$ . It can be observed that as the number of users and antennas increases, the error probability increases slowly. The reason is that although the number of users increases, more antenna array gain is obtained, and the beam division property is exploited. For the conventional tree decoder, as the beam division property is not exploited, the error probability increases more rapidly.

## VIII. CONCLUSION

An URA scheme with beam-space tree decoding under the framework of bit partition and slotted transmission was proposed in this paper. Specifically, we designed two beam-space tree decoders, which are based on hard decision and soft decision, respectively. Both of them exploit the intrinsic beam division property to improve the system performance and help the system serve more active users. Besides, the first decoder can reduce the solution searching space and has low complexity, while the second decoder exploits the advantage of list decoding to recover the miss-detected packets. Simulation

results validated that our proposed URA scheme was superior with respect to error probability.

The beam division property is an intrinsic property in mmWave communication systems due to channel propagation. Therefore, this property exploited by our proposed decoders can also be exploited by other schemes under the scenarios of NGMA to help the system serve more users and improve the system performance.

## REFERENCES

- [1] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, 2016.
- [2] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, "Machine-type communications: current status and future perspectives toward 5G systems," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 10–17, 2015.
- [3] Z. Chen, F. Söhrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1890–1904, 2018.
- [4] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016.
- [5] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dahir, and R. Schober, "Massive access for 5G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 615–637, 2020.
- [6] Y. Liang, X. Li, J. Zhang, and Z. Ding, "Non-orthogonal random access for 5G networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4817–4831, 2017.
- [7] Z. Zhang, X. Wang, Y. Zhang, and Y. Chen, "Grant-free rateless multiple access: A novel massive access scheme for Internet of Things," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 2019–2022, 2016.
- [8] K. Senel and E. G. Larsson, "Grant-free massive MTC-enabled massive MIMO: A compressive sensing approach," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6164–6175, 2018.
- [9] S. Kim, H. Kim, H. Noh, Y. Kim, and D. Hong, "Novel transceiver architecture for an asynchronous grant-free IDMA system," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4491–4504, 2019.
- [10] J. Wang, Z. Zhang, and L. Hanzo, "Joint active user detection and channel estimation in massive access systems exploiting Reed–Muller sequences," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 3, pp. 739–752, 2019.
- [11] Y. Polyanskiy, "A perspective on massive random-access," in *IEEE Int. Symp. Inf. Theory (ISIT)*. IEEE, 2017, pp. 2523–2527.
- [12] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164–1179, 2014.
- [13] C.-K. Wen, S. Jin, K.-K. Wong, J.-C. Chen, and P. Ting, "Channel estimation for massive MIMO using Gaussian-mixture bayesian learning," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1356–1368, 2014.
- [14] F. Bellili, F. Söhrabi, and W. Yu, "Generalized approximate message passing for massive MIMO mmwave channel estimation with laplacian prior," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3205–3219, 2019.
- [15] C. Sun, X. Gao, S. Jin, M. Matthaiou, Z. Ding, and C. Xiao, "Beam division multiple access transmission for massive MIMO communications," *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 2170–2184, 2015.
- [16] L. You, X. Gao, G. Y. Li, X.-G. Xia, and N. Ma, "BDMA for millimeter-wave/terahertz massive MIMO transmission with per-beam synchronization," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1550–1563, 2017.
- [17] R. Jia, X. Chen, Q. Qi, and H. Lin, "Massive beam-division multiple access for B5G cellular Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2386–2396, 2019.
- [18] L. Liu and W. Yu, "Massive connectivity with massive MIMO—Part I: Device activity detection and channel estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2933–2946, 2018.
- [19] L. Liu and W. Yu, "Massive connectivity with massive MIMO—Part II: Achievable rate characterization," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2947–2959, 2018.
- [20] V. K. Amalladinne, J.-F. Chamberland, and K. R. Narayanan, "A coded compressed sensing scheme for unsourced multiple access," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6509–6533, 2020.

- [21] A. Fengler, P. Jung, and G. Caire, "SPARCs for unsourced random access," *IEEE Trans. on Inf. Theory*, pp. 1–1, 2021.
- [22] A. Fengler, S. Haghghatshoar, P. Jung, and G. Caire, "Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2925–2951, 2021.
- [23] V. Shyianov, F. Bellili, A. Mezghani, and E. Hossain, "Massive unsourced random access based on uncoupled compressive sensing: Another blessing of massive MIMO," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 820–834, 2020.
- [24] A. Decurninge, I. Land, and M. Guillaud, "Tensor-based modulation for unsourced massive random access," *IEEE Wireless Commun. Lett.*, vol. 10, no. 3, pp. 552–556, 2020.
- [25] A. Fengler, P. Jung, and G. Caire, "Pilot-based unsourced random access with a massive MIMO receiver, MRC and polar codes," *arXiv preprint arXiv:2012.03277*, 2020.
- [26] X. Shao, X. Chen, D. W. K. Ng, C. Zhong, and Z. Zhang, "Cooperative activity detection: Sourced and unsourced massive random access paradigms," *IEEE Trans. Signal Process.*, vol. 68, pp. 6578–6593, 2020.
- [27] X. Gao, L. Dai, Z. Chen, Z. Wang, and Z. Zhang, "Near-optimal beam selection for beamspace mmwave massive MIMO systems," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1054–1057, 2016.
- [28] B. Wang, L. Dai, Z. Wang, N. Ge, and S. Zhou, "Spectrum and energy-efficient beamspace MIMO-NOMA for millimeter-wave communications using lens antenna array," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2370–2382, 2017.
- [29] D.L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18914–18919, 2009.
- [30] J.P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, 2013.
- [31] P. Li, L. Liu, K. Wu, and W.K. Leung, "Interleave division multiple-access," *IEEE Trans. Wireless Commun.*, vol. 5, no. 4, pp. 938–947, 2006.
- [32] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.
- [33] A. Fengler, G. Caire, P. Jung, and S. Haghghatshoar, "Massive MIMO unsourced random access," *arXiv preprint arXiv:1901.00828*, 2019.