Coded Beam Training

Tianyue Zheng, *Graduate Student Member, IEEE*, Jieao Zhu, *Graduate Student Member, IEEE*, Qiumo Yu, *Graduate Student Member, IEEE*, Yongli Yan, *Member, IEEE*, and Linglong Dai, *Fellow, IEEE*

***Abstract*— In extremely large-scale multiple-input-multiple-output (XL-MIMO) systems for future sixth-generation (6G) communications, codebook-based beam training stands out as a promising technology to acquire channel state information (CSI). Despite their effectiveness, existing beam training methods suffer from significant achievable rate degradation for remote users with low signal-to-noise ratio (SNR). To tackle this challenge, leveraging the error-correcting capability of channel codes, we incorporate channel coding theory into beam training to enhance the training accuracy, thereby extending the coverage area. Specifically, we establish the duality between hierarchical beam training and channel coding, and build on it to propose a general coded beam training framework. Then, we present two specific implementations exemplified by coded beam training methods based on Hamming codes and convolutional codes, during which the beam encoding and decoding processes are refined respectively to better accommodate to the beam training problem. Simulation results have demonstrated that, the proposed coded beam training method can enable reliable beam training performance for remote users with low SNR, while keeping training overhead low.**

***Index Terms*— Beam training, channel codes, hierarchical codebook, convolutional codes, Hamming codes.**

I. Introduction

MASSIVE multiple-input multiple-output (mMIMO) [1] has been considered as a technological enabler for current fifth-generation (5G) communications. To achieve spectral efficiency enhancement in mMIMO systems, accurate channel state information (CSI) at the transmitter is a prerequisite, which can be realized either by the explicit CSI acquisition (i.e., channel estimation) or the implicit CSI acquisition (i.e., beam training) [2]. To further increase spectral efficiency, future sixth-generation (6G) communication systems are expected to employ extremely large-scale MIMO (XL-MIMO) antenna arrays [3], [4]. Unfortunately, due to the high-dimensional XL-MIMO channels, the pilot overhead of channel estimation will increase dramatically, making explicit CSI acquisition challenging [5], [6], especially with passive antenna elements in XL-reconfigurable intelligent surface (RIS) systems [7], [8]. In such cases, implicit CSI acquisition, i.e., beam training, serves as a more practical way for CSI acquisition [9]. This implicit procedure is performed by transmitting several predefined directional beams (codewords) toward the users, and determining the users' direction from their received power [10], [11].

An important way to conduct beam training is exhaustive beam sweeping [12], [13], which sequentially tests the narrow beams predefined in the codebook, and selects the codeword with the highest received power. Thanks to the high beamforming gain of narrow beams, exhaustive beam sweeping by narrow beams ensures reliable beam training performance even for remote users (usually located in the cell-edge area) with low signal-to-noise ratio (SNR) [12]. Despite the considerable reliability of exhaustive beam sweeping, this exhaustive method brings unaffordable training overhead in XL-MIMO communication systems. Specifically, the size of a typical exhaustive codebook is equal to the number of BS antennas with each codeword corresponding to a narrow beam focused on a specific direction. In recent years, a polar-domain codebook [14] has been proposed to improve the XL-MIMO CSI acquisition accuracy by considering the physically spherical wave property in the near-field region. In this codebook, each codeword aligns with a sampled location in the angle-distance domain, thus the size of the codebook is even larger (i.e. the product of antenna number and sampled distance grid) [15].

To reduce the training overhead, hierarchical beam training methods have been proposed [16], [17], [18], [19]. In hierarchical beam training, the possible user directions are narrowed down in a layer-by-layer manner. Specifically, hierarchical beam training utilizes a hierarchical codebook comprising multi-layer codewords. In this codebook, the spatial region covered by a certain codeword at any layer of the codebook is partitioned into two smaller disjoint spatial regions in the next layer [16]. Then, applying this codebook, the BS can gradually narrow down the possible user direction by choosing the spatial region with larger received power based the user's feedback signal in each layer. Owing to the half reduction of uncertain region of the user's direction in each layer, this hierarchical scheme can exponentially speedup the beam training process compared with the exhaustive beam sweeping [17]. Thus, the idea of hierarchical beam training has triggered various improvement efforts for designing binary search-based hierarchical codebooks [16], [17], [18], [20], [21], [22]. Besides, the idea of hierarchical beam training

Received 24 February 2024; revised 1 September 2024; accepted 8 November 2024. Date of publication 20 January 2025; date of current version 27 February 2025. This work was supported in part by the National Natural Science Foundation for Distinguished Young Scholars under Grant 62325106, in part by the National Key Research and Development Program of China under Grant 2023YFB3811503, and in part by the National Natural Science Foundation of China under Grant 62031019. *(Corresponding author: Linglong Dai.)*

Tianyue Zheng, Jieao Zhu, Qiumo Yu, and Yongli Yan are with the State Key Laboratory of Space Network and Communications, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: zhengty22@mails.tsinghua.edu.cn; zja21@mails.tsinghua.edu.cn; yqm22@mails.tsinghua.edu.cn; yanyongli@tsinghua.edu.cn).

Linglong Dai is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China, and also with the Department of EECS, Massachussets Institute of Technology, Cambridge, MA 02139 USA (e-mail: daill@tsinghua.edu.cn).

Digital Object Identifier 10.1109/JSAC.2025.3531550

is also widely employed to perform efficient near-field beam training [15], [19], [23].

However, due to the "error propagation" phenomenon, hierarchical beam training methods suffer from serious performance deterioration for remote users with low SNR. The error propagation originates from low directional beam gain of wide beams in the upper layers. With reduced beam gain, these upper-layer beams are especially susceptible to errors, causing unrecoverable errors in the subsequent layers of the hierarchical process.

To our best knowledge, existing beam training methods, both exhaustive and hierarchical, can hardly resolve the conflict of reliability and efficiency in beam training for remote users with low SNR. To solve the problem, the authors of [24], [25] proposed a heuristic approach to introduce channel codes into beam training. Inspired by these works, in this paper, we propose a unified coded beam training framework by exploiting the error correction capability of channel codes. With elaborated adaptation for the beam training problem, the proposed framework enables reliable beam training performance with exponentially reduced pilot overhead, even for remote users. Specifically, the main contributions of this paper are summarized as follows.[1]

1) By analyzing the binary algebraic structure of hierarchical beam training, this paper reveals the duality of hierarchical beam training problem and channel coding problem, based on which a unified coded beam training framework is proposed. Leveraging this duality, almost all kinds of channel codes can be seamlessly integrated into the proposed coded beam training framework.
2) To perform coded beam training, we design the space-time coded beam patterns for generating the codewords and the transmitting beamformers during beam training, where different spatial directions are encoded into different time sequences based on the channel encoder to improve the tolerance to noise. Then, we utilize the sequence of received signal power to decode the spatial directions of the user by exploiting the error correction capability of channel codes, yielding the desired codeword for the user.
3) To better accommodate to the beam training problem, we improve the coding algorithms in two aspects. Firstly, existing channel coding algorithms are designed for Gaussian channel for data transmission, while the user's received power during beam training obeys a non-central χ^2 distribution. Therefore, we modify the log-likelihood ratio (LLR) calculator in the beam training decoder to better adapt to the probabilistic properties in the beam training problem. Secondly, we propose an adaptive encoding process where we dynamically adjust the coded beam pattern based on the outcomes of previous decoding iterations. The adaptive beam training encoder can exclude impossible directions, thus improve the real-time beam gain.
4) We employ classical Hamming codes and convolutional codes respectively as examples to illustrate our proposed coded beam training method. We provide simulation comparison of our proposed coded beam training method with other methods, demonstrating the proposed coded beam training method can enable reliable beam training for remote users with low SNR. Besides, simulation results also validate that the χ^2 decoder outperforms the traditional Gaussian decoder.

The rest of this paper is organized as follows. In Section II, the system as well as channel models are introduced, and the problem of beam training is formulated. Then, the principles and implementation of the proposed coded beam training are elaborated in Sections III and IV, respectively. Simulation results are provided in Section V. Finally, Section VI concludes this paper.

Notations: Lower-case and upper-case boldface letters represent vectors and matrices, respectively; $\|\cdot\|_p$ denotes the p-norm of a vector; $\mathbb{C}, \mathbb{R}$ denote the set of complex and real numbers, respectively; $[\cdot]^T, [\cdot]^H$ denote the transpose, and conjugate-transpose operations, respectively; $\bigcup, \bigcap$ denote the union and intersection operation of sets; $\mathcal{CN}(\mu, \Sigma)$ denotes the Gaussian distribution with mean μ and covariance Σ; $\oplus$ denotes the exclusive OR (XOR) operation; I_ν denotes the ν-th order modified Bessel function of the first kind.

II. System Model

In this section, the channel model of the XL-MIMO system used in this paper will be introduced first. Then, we will formulate the beam training problem.

A. System Model

We consider a mmWave/Terahertz (THz) XL-MIMO system with one base station (BS) and a single-antenna user equipment (UE). The BS employs a uniform linear array (ULA) equipped with N_{T} $\lambda/2$-spaced antennas, each being connected to one RF chain, i.e., we adopt the fully-digital precoding structure. It is worth emphasizing that our main technical contributions are not restricted to full-digital precoding and can be extended to arbitrary precoding architecture by applying corresponding beam design methods [16], [26], [27], [28], [29], [30], examples include the hybrid precoding elaborated in Section IV-E.

For the downlink transmission, let $s_0 \in \mathbb{C}$ be the power-normalized transmitted symbol, then the received signal y at the UE is given by

$$y = \sqrt{P}\boldsymbol{h}\boldsymbol{w}s_0 + n, \tag{1}$$

where $P > 0$ is the transmit power, $\boldsymbol{h} \in \mathbb{C}^{1\times N_{\mathrm{T}}}$ the downlink channel, $\boldsymbol{w} \in \mathbb{C}^{N_{\mathrm{T}}\times 1}$ the unit-norm transmit beamformer, and n the complex circularly-symmetric additive white Gaussian noise $n \sim \mathcal{CN}(0, \sigma^2)$ at the UE receiver.

According to the well-known Saleh-Valenzuela channel model [21], the channel $\boldsymbol{h}$ can be expressed as

$$\boldsymbol{h} = \sqrt{\frac{N_{\mathrm{T}}}{L_0}} \sum_{l=0}^{L_0-1} \beta_l \boldsymbol{\alpha}(\varphi_l), \tag{2}$$

[1]Simulation codes will be provided to reproduce the results in this paper: http://oa.ee.tsinghua.edu.cn/dailinglong/publications/publications.html

where L_0 is the number of multipath components, β_l and φ_l represent the complex gain and the angle-of-departure (AoD) of the l-th path. $\boldsymbol{\alpha}(\varphi) \in \mathbb{C}^{1\times N_{\mathrm{T}}}$ is the array steering vector, which is defined as

$$\boldsymbol{\alpha}(\varphi) = \frac{1}{\sqrt{N_{\mathrm{T}}}}[1, e^{-j\pi\varphi,}, \ldots, e^{-j(N_{\mathrm{T}}-1)\pi\varphi}], \tag{3}$$

where $\varphi \triangleq \sin\theta \in [-1, 1]$ denotes the spatial direction and $\theta \in [-\pi/2, \pi/2]$ the physical direction. The significant scattering attenuation at high frequencies makes the power of the LoS path considerably higher than its NLoS counterparts, rendering the former a dominant component for data transmission at mmWave/THz bands. Therefore, this paper mainly considers the channel with only LoS component, which also determines the pointing direction from the BS to the UE [6]. Thus, the channel model in (2) can be simplified into

$$\boldsymbol{h} = \sqrt{N_{\mathrm{T}}}\beta_0\boldsymbol{\alpha}(\varphi_0), \tag{4}$$

where the path gain of the LoS path is modeled as [3]

$$\beta_0 = \frac{\lambda_0}{4\pi r}, \tag{5}$$

with r denoting distance from the UE to the center of the antenna array.

B. Problem Description

The objective of beam training is to steer the beamformer $\boldsymbol{w}$ to the AoD of the dominate path (LoS path). Specifically, according to the structure of the array steering vector in (3), we define the DFT codebook, $\mathcal{W}$, also known as the exhaustive codebook, as

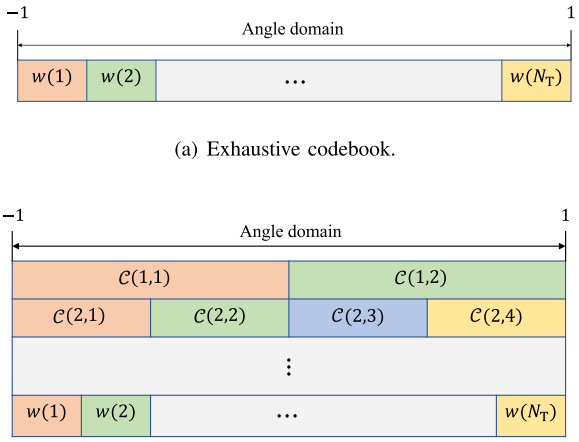
$$\mathcal{W} = \{\boldsymbol{\alpha}^H(\varphi) | \varphi = -1 + (2n-1)/N_{\mathrm{T}}, n \in \{1, 2, \cdots, N_{\mathrm{T}}\}\}. \tag{6}$$

the diagram of which is illustrated in Fig. 1(a). Codebook-based beam training attempts to select a codeword from $\mathcal{W}$ to maximize the received signal power, i.e.,

$$\begin{aligned} \max_{\boldsymbol{w}} \quad & |\boldsymbol{h}\boldsymbol{w}| \\ \text{s.t.} \quad & \boldsymbol{w} \in \mathcal{W}. \end{aligned} \tag{7}$$

To solve the problem (7), a straightforward way is to perform exhaustive beam sweeping [12]. The BS first sequentially sweeps all codewords from $\mathcal{W}$. Then, the UE selects the best codeword having the highest received signal power and feedbacks the selected codeword index. Clearly, the beam sweeping process occupies N_{T} time slots, equivalent to the number of the BS antennas. This fact means that although this exhaustive beam sweeping could achieve a good beam training performance, it inevitably consumes unaffordable training overhead, especially for XL-MIMO systems.

To avoid the unacceptable training overhead incurred by exhaustive beam sweeping, hierarchical beam training utilizing binary-search based codebook are widely adopted. As presented in Fig. 1(b), a typical hierarchical codebook [16], $\mathcal{C}^{\mathrm{hier}}$, has 2^t codewords at the t-th ($t \in \{1, 2, \cdots, \log_2 N_{\mathrm{T}}\}$) layer. We denote the b-th codeword in the t-th layer as $\mathcal{C}^{\mathrm{hier}}_{t,b}$, which covers two higher-resolution codewords with narrower coverage angle at the $t+1$-th layer. During the beam training process, we test the power of the received signal with two selective low-resolution codewords at the upper layer, choose the one with higher received power, and then narrow down the beam width in a layer-by-layer manner, until a specific codeword at the bottom layer is obtained. Through this way, the beam training overhead is reduced to $2\log_2 N_{\mathrm{T}}$ [16]. However, the performance of hierarchical beam training suffers from the "error propagation" phenomenon, and thus cannot ensure reliable beam training for remote users with low SNR, leading to a restricted coverage area. Specifically, the codewords at higher layers have wider beamwidth and lower beam gain, making it more vulnerable to noise. Since hierarchical beam training works on a binary tree in a sequential manner, any erroneous decision at some layer on the tree will lead to unrecoverable training failure.



(a) Exhaustive codebook.

(b) Hierarchical codebook.

Fig. 1. Illustration of the DFT codebook, $\mathcal{W}$, and the hierarchical codebook, $\mathcal{C}^{\mathrm{hier}}$, where the upscript "hier" in omitted in the figure.

In this paper, to alleviate the "error propagation" curse, we propose a new hierarchical beam training method utilizing the self-correcting capabilities of channel codes, which can reduce training overhead while maintaining the beam training success rate for remote users with low SNR.

III. Overview of Coded Beam Training

Channel codes are well-established error control techniques to protect the transmission data against channel noise by adding redundant bits. Compared to non-coded systems, coded systems are able to dramatically decrease the bit error rate (BER) under the same channel condition and data payload requirements, which is known as the waterfall effect of the BER. Inspired by channel coding, we develop an ultra-reliable hierarchical beam training framework, namely *coded beam training*. In the proposed framework, exploiting the error correction capability, channel codes are introduced to hierarchical beam training by adding extra layers of codewords to protect the hierarchical beam training process from channel noise.

This section elaborates on the principles of coded beam training. We first illustrate the fundamental idea of coded beam training using an introductory example of a $(7, 4)$ Hamming

code. Then, this idea is extended to a general framework of coded beam training.

A. An Introductory (7, 4) Hamming Code Example

To help understand of the proposed framework, we start from comparing the traditional binary-search based hierarchical beam training [16] with the coded beam training exploiting $(7,4)$ Hamming code in an $N_{\mathrm{T}}=16$-antenna system.

1) Motivation of Coded Beam Training: In traditional hierarchical beam training, the codebook $\mathcal{C}^{\mathrm{hier}}$ contains $T^{\mathrm{hier}}=\log_2 N_{\mathrm{T}}=4$ layers, the t-th layer of which has 2^t codewords. The detailed beam training procedure is carried out as follows. We divide the spatial direction into $N_{\mathrm{T}}=16$ segments uniformly and let the length-4 bitstream $\mathbf{u}\in\{0,1\}^4$ label the spatial direction of the user. At the first layer of codebook $\mathcal{C}^{\mathrm{hier}}$, the BS sequentially transmits $\mathcal{C}^{\mathrm{hier}}_{1,1}$ and $\mathcal{C}^{\mathrm{hier}}_{1,2}$ to the UE. Then, the UE compares the received signal power of $\mathcal{C}^{\mathrm{hier}}_{1,1}$ and $\mathcal{C}^{\mathrm{hier}}_{1,2}$, and set $\mathbf{u}(1)=0$ if the first codeword yields higher signal power, and $\mathbf{u}(1)=1$ otherwise. After that, the UE feeds back the bit $\mathbf{u}(1)$ to the BS, according to which the BS selects the two possible codewords in the second layer. We sequentially perform these steps until approaching the bottom layer of $\mathcal{C}^{\mathrm{hier}}$. The BS can finally recover the spatial direction and decide the optimal index according to the bitstream $\mathbf{u}=[\mathbf{u}(1),\mathbf{u}(2),\mathbf{u}(3),\mathbf{u}(4)]$. For example, if the received bitstream $\mathbf{u}=[0,0,1,0]$, the selected codeword index is $\mathrm{bintodec}([0,0,1,0])+1=3$. According to (6), beamformer $\boldsymbol{w}=\boldsymbol{\alpha}^H(-1+(2\times 3-1)/16)=\boldsymbol{\alpha}^H(-11/16)$ is adopted for data transmission. However, for a remote user with low SNR, if the first layer of codebook chooses the wrong index due to low directional gain of wide beam, the subsequent training process is invalid since we have missed the optimal index. Suppose the original $\mathbf{u}(1)=0$ is incorrectly decided as $\mathbf{u}(1)=1$, then the selected codeword index is 11 instead of 3. This issue is referred to the "error propagation" problem, which will be alleviated using the Hamming code to protect the index against incorrect bits.

2) Beam Encoding: To alleviate the error propagation issue, we consider to incorporate $(7,4)$ Hamming code to beam training. Our novelty lies in the design of a new codebook, say $\mathcal{C}^{\mathrm{ham}}$, comprising seven layers (four information layers and *three additional check layers*) as presented in Fig. 2. Compared with the classical hierarchical codebook, $\mathcal{C}^{\mathrm{hier}}$, the proposed codebook, $\mathcal{C}^{\mathrm{ham}}$, contains two complementary codewords in each layer, featuring sawtooth-shaped beam patterns, which makes it capable of encoding beams. The detailed construction of $\mathcal{C}^{\mathrm{ham}}$ involves two step: 1) space-time beam pattern design relying on the Hamming code, 2) codebook generation based on the beam pattern, which are elaborated below.

- **Space-time beam pattern design**: To begin with, it is necessary to determine the space-time 0-1 beam pattern, $\mathcal{V}^{\mathrm{ham}}$, for building the codebook $\mathcal{C}^{\mathrm{ham}}$. Each element of $\mathcal{V}^{\mathrm{ham}}$ reflects the desired beam pattern of one codeword of $\mathcal{C}^{\mathrm{ham}}$. Specifically, for the desired beam generated by the b-th codeword in t-th layer, $\mathcal{C}^{\mathrm{ham}}_{t,b}$, its beam pattern $\mathcal{V}^{\mathrm{ham}}_{t,b}$ is composed of $N_{\mathrm{T}}=16$ binary numbers, i.e., $\mathcal{V}^{\mathrm{ham}}_{t,b}=\{0,1\}^{16}$. By dividing the spatial region $[-1,1]$ into $N_{\mathrm{T}}=16$ segments uniformly, the s-th bit $\mathcal{V}^{\mathrm{ham}}_{t,b}(s)$ is set as 0 if the beam generated by codeword $\mathcal{C}^{\mathrm{ham}}_{t,b}$ is expected to have a high gain in the s-th segment ($s\in\{0,1,\cdots,15\}$) and $\mathcal{V}^{\mathrm{ham}}_{t,b}(s)=1$ otherwise.
  Applying this definition, we can build $\mathcal{V}^{\mathrm{ham}}$ using the Hamming codes. To be specific, we first index the s-th spatial segment, $s\in\{0,1,\cdots,15\}$, by a length-$k=4$ bits $\mathbf{u}_s$. According to the encoding operation of $(7,4)$ Hamming coding, the coded bitstream $\boldsymbol{x}_s$ corresponding to the s-th spatial direction is expressed as
  $$\boldsymbol{x}_s=\mathbf{u}_s\mathbf{G}\in\{0,1\}^7, \tag{8}$$
  where $\mathbf{G}$ is the generator matrix, which is denoted as
  $$\mathbf{G}=\left[\begin{array}{cccc|ccc}1&0&0&0&1&1&1\\0&1&0&0&1&1&0\\0&0&1&0&1&0&1\\0&0&0&1&0&1&1\end{array}\right]. \tag{9}$$
  In our codebook design, the s-th encoded space bitstream $\boldsymbol{x}_s\in\{0,1\}^7$ defines the beam pattern of the codewords $\mathcal{C}^{\mathrm{ham}}_{t,1},t=1,2,\cdots,7$ over the s-th segment, while its flip determines the beam pattern of $\mathcal{C}^{\mathrm{ham}}_{t,2}$. Therefore, the corresponding space-time beam pattern $\mathcal{V}^{\mathrm{ham}}_{t,b},t\in\{1,2,\cdots,7\},b\in\{1,2\}$ can be expressed as
  $$\left\{\begin{array}{ll}\mathcal{V}^{\mathrm{ham}}_{t,1}(s) & =\boldsymbol{x}_s(t)\\ \mathcal{V}^{\mathrm{ham}}_{t,2}(s) & =\bar{\boldsymbol{x}}_s(t)\end{array}\right.,\quad s\in\{0,1,\cdots,15\}, \tag{10}$$
  where $\bar{x}$ denotes the bit flip and $\boldsymbol{x}(s)$ denotes the s-th bit of a vector $\boldsymbol{x}$. The designed beam pattern by (10) is illustrated in the left part of Fig. 2. For example, for the $s=2$ direction, the spatial bitstream is $\mathbf{u}_2=[0,0,1,0]$ and the encoded bitstream can be calculated as $\boldsymbol{x}_2=[0,0,1,0,1,0,1]$. Thus, the beam pattern of $s=2$-th segment in the layer sequence is also $[0,0,1,0,1,0,1]$, which means that the $1,2,4,6$-th layers present high beam gains while the others present low beam gains, as shown in Fig. 2.
- **Codebook Generation**: After obtaining the space-time beam pattern, we are able to generate the codebook, $\mathcal{C}^{\mathrm{ham}}$, employing various beam design methods, such as the weighted sum of narrow beams [31] and the Gerchberg-Saxton (GS) algorithm [23].

As illustrated in Fig. 2, the first four layers of the beams are regular square beams and the extra three check layers are irregular multi-mainlobe wide beams. In each layer, the BS sequentially sends all codewords to the UEs. The UEs compare the received signal power of two codewords and feedback one bit to label the stronger codeword. In the same example, if the original spatial information bitstream is $\mathbf{u}=[0,0,1,0]$, then the desired feedback bit time sequence is $\hat{\boldsymbol{x}}=[0,0,1,0,1,0,1]$.

3) Beam Decoding: In the beam decoding part, we aim to decide the optimal codeword (i.e. space information) index according to the received bitstream. If we suppose the first layer is decided incorrectly due to the influence of noise again in this example, the received bitstream will change to $\hat{\boldsymbol{x}}=[1,0,1,0,1,0,1]$. Then we illustrate how we obtain the correct index with the error correction ability of Hamming code.

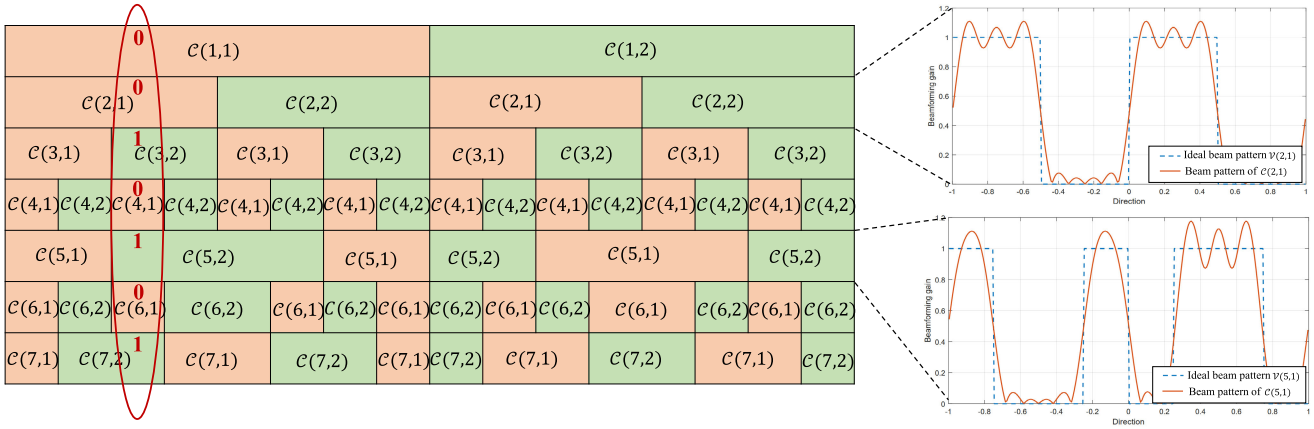


Fig. 2. The codebook $\mathcal{C}^{\text{ham}}$ designed by the (7,4) Hamming code (14 codewords in 7 layers in total), where the upscript is omitted in the figure.

TABLE I
ERROR PATTERNS AND THE CORRESPONDING SYNDROMES OF $(7,4)$HAMMING CODE

error bit	c	error bit	c
b_1	111	b_2	110
b_3	101	b_4	011
b_5	100	b_6	010
b_7	001	no	000

Based on the parity check matrix $\mathbf{H}$, Hamming decoder helps determine whether the received bitstream contains error bit and the specific position of the error bit. The syndrome is computed as

$$\mathbf{c} = \hat{\boldsymbol{x}}\mathbf{H}^{\mathbf{T}}, \tag{11}$$

where parity check matrix can be expressed as

$$\mathbf{H} = \left[\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right]. \tag{12}$$

Then we can decide the error pattern based on the syndrome as in Table. I.

Based on (12), in this example, we calculate the syndrome as $\mathbf{c} = [1,1,1]$, which means the first bit is wrong. Therefore, we correct the information bitstream as $\hat{\boldsymbol{x}} = [0,0,1,0,1,0,1]$. The selected codeword index is 3, which successfully correct the erroneous bit. In this way, by exploiting the "self-correcting" capabilities of channel coding, it is possible to obtain the correct angular index for beamforming even if the wide beam in the upper layer leads to incorrect decision. Thanks to the coding gain, it is promising that the proposed hierarchical beam training method can enable reliable beam training for remote users with low SNR.

B. Overall Idea Description

In this subsection, we generalize the specific example of $(7,4)$ Hamming code to a unified coded beam training framework. We will first present the theoretical foundations of the proposed coded beam training and then the general coded beam training framework is illustrated.

1) Theoretical Foundations: The theoretical foundations of coded beam training lies in *the duality of hierarchical beam training and channel coding*, which are elaborated below.

- **Channel coding**: Following Shannon's seminal paper [32], a channel code consists of an encoder function f and a decoder function g. The encoder f maps a message $u \in \mathcal{U}$ to a codeword $\boldsymbol{x} = f(u) \in \mathcal{X}^n$, where $\mathcal{X}$ is the output alphabet of the encoder (usually binary, i.e., $\mathcal{X} = \{0,1\}$), and n is the code length. The channel $W : \mathcal{X}^n \to \mathcal{Y}^n$ randomly maps a coded sequence $\boldsymbol{x}$ to a received sequence $\boldsymbol{y} \in \mathcal{Y}^n$, where $\mathcal{Y}$ is the received alphabet. Finally, the decoder g maps $\boldsymbol{y}$ to an estimated message $\hat{u} \in \mathcal{U}$, which is expected to equal u with high probability, i.e., $\mathbb{P}[u = \hat{u}]$ is close to 1. The number of different messages $|\mathcal{U}|$ determines the number of payload bits $k = \log_2(|\mathcal{U}|)$, and the code rate is defined as $R = k/n$. Thus, an (n,k)-code is a pair of encoder-decoder that takes k bits into n channel input symbols, and recovers the k payload bits from n output symbols.
- **Beam training**: As mentioned in Section II-B, the target of beam training is to determine the angular directions of the users from the received signals after the BS transmits a pre-designed beam training codebook $\mathcal{C}$. Generally, an (T, N_{T})-*beam training code (BTC)* is defined as a beam training procedure capable of distinguishing N_{T} different angular directions via an T-layer beam training codebook. We denote the codebook $\mathcal{C} = \{\mathcal{C}_{t,b} \in \mathbb{C}^{N_{\text{T}} \times 1}, t \in \{1,2,\cdots,T\}, b \in \{1,2,\cdots,b_t\}\}$ with b_t codewords for layer t. Besides, we denote the beam pattern corresponding to the codeword $\mathcal{C}_{t,b}$ as $\mathcal{V}_{t,b} \in \{0,1\}^{N_{\text{T}}}$, which describes the 0-1 pattern of the multi-mainlobe beam in the angular domain as is defined in Section III-A. An information-theoretic insight is that it is only possible to construct (T, N_{T})-BTC with $|T| = \Omega(\log N_{\text{T}})$, since during beam training the user can obtain at most one bit of information within each training slot.

The above comparison reveals that beam training is intrinsically an information transmission problem. In the beam

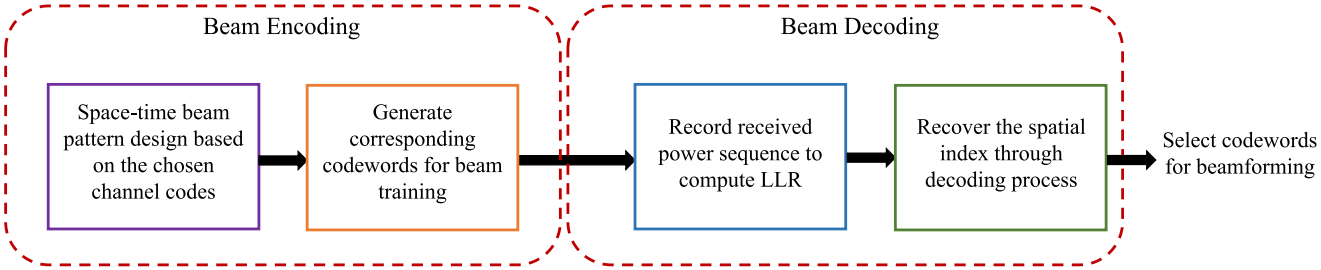


Fig. 3. Overall framework of channel codes-aided hierarchical beam training with adaptive convolutional codes as an example.

training problem, the *payload bits* are the unknown physical direction of the UE, the *channel* is the angular response function of the physical channel, and the *receiver* is the UE. In this context, the BTC plays the same role as the channel codes during data transmission.

From above description, we summarize the relationship between a BTC and a channel code: An (n, k)-channel code is equivalent to an $(n, 2^k)$-BTC. This relationship ensures that an arbitrary channel code can be converted to a reliable BTC to protect beam training against noisy channel conditions, which motivates the following channel code-BTC framework.

2) Framework Description: As illustrated in Fig. 3, the framework of coded beam training comprises two stages. They are the *beam encoding* for designing the BTC codebook, $\mathcal{C}$, and the *beam decoding* to recover the users' spatial directions, respectively. These two stages are detailed as follows.

2.1) Beam encoding: The target of beam encoding is to construct an n-layer BTC codebook, denoted as $\mathcal{C}$, from an (n, k)-channel code, which is capable of distinguishing 2^k angular directions. Similar to the design of $\mathcal{C}^{\text{ham}}$ in the Hamming code case, each layer of the general codebook, $\mathcal{C}$, contains two complementary codewords, which are built on the following two steps: 1) design the space-time beam pattern, $\mathcal{V}$, according to an arbitrary channel code, 2) generate the codebook $\mathcal{C}$ based on $\mathcal{V}$.

- **Space-time beam pattern design**: Recall that the set $\mathcal{V}$ records the ideal beam patterns of all codewords belonging to $\mathcal{C}$. For easy understanding, we provide an overall explanation for space-time beam pattern design. During beam training, we divide the spatial direction into N_{T} segments and attempt to recover the optimal direction. Each spatial information $s \in \{0, 1, 2, \cdots, N_{\text{T}} - 1\}$ is encoded to a time sequence $\boldsymbol{x}$. The time sequence determines the beam pattern of the s-th segment of the layers in the coded beam training codebook, which will sequentially transmit to the UE. And the received power sequence is employed to recover the direction later. To achieve it, we index all the possible angular directions $s \in \{0, 1, \cdots, 2^k - 1\}$ by a length-k spatial information bits $\mathbf{u}_s \in \{0, 1\}^k$, then the encoded bits $\boldsymbol{x}_s \in \{0, 1\}^n$ is given by

$$\boldsymbol{x}_s = f(\mathbf{u}_s) \in \{0, 1\}^n, \quad s \in \{0, 1, \cdots, 2^k - 1\}, \quad (13)$$

where $f(\cdot)$ denotes an arbitrary channel encoder. In this context, the space-time beam pattern $\mathcal{V}_{t,b}, t \in \{1, 2, \cdots, n\}, b \in \{1, 2\}$ can be established according to the encoded bits as

$$\begin{cases} \mathcal{V}_{t,1}(s) = \boldsymbol{x}_s(t) & t \in \{1, 2, \cdots, n\}, \\ \mathcal{V}_{t,2}(s) = \bar{\boldsymbol{x}}_s(t) & s \in \{0, 1, \cdots, 2^k - 1\} \end{cases} \quad (14)$$

Consequently, the different spatial directions are encoded into different time sequences (i.e. different beam gains in the layer sequences), which is illustrated in Fig. 4. The figure depicts the space-time beam pattern of the first codeword in all layers. Note that $|\mathcal{C}| = 2n$, i.e., the number of training time slots needed in the constructed BTC is $2n$.

- **Codebook generation**: Consider the generation of $\mathcal{C}$. Each codeword of $\mathcal{C}$, is optimized and generated by making its beam pattern as close to the ideal beam pattern, labeled by $\mathcal{V}$, as possible. This step can be efficiently performed using existing beam design methods [23], [31], [33], with the consideration of array structure constraints, such as the full-digital precoding and hybrid precoding.

2.2) Beam decoding: After acquiring the codebook $\mathcal{C}$, the beam decoding can be performed to decode the spatial directions of the user with the received power sequence of the transmitted codewords in $\mathcal{C}$.

Specifically, the BS sequentially assigns the beamformer $\boldsymbol{w}$ with $\mathcal{C}_{t,1}$ and $\mathcal{C}_{t,2}$ layer-by-layer and transmits pilots to the UE. Denote the UE's received signal power as $P_{t,1}$ and $P_{t,2}$. To perform **hard decoder**, the UE compares the received power pair and records the results in a bit sequence $\hat{\boldsymbol{x}}$, i.e., $\hat{\boldsymbol{x}}(t) = 0$ if $P_{t,1} > P_{t,2}$ and $\hat{\boldsymbol{x}}(t) = 1$ otherwise. After the training phase, UE feeds back the bit sequence $\hat{\boldsymbol{x}}$ to the BS for carrying out hard decoding. Once receiving the feedback bit sequence $\hat{\boldsymbol{x}}$, BS can obtain $\hat{\mathbf{u}} = g(\hat{\boldsymbol{x}})$ as the estimation of the user's angular direction $\hat{s}$, by invoking the channel decoder $g(\cdot)$. In contrast, if the **soft decoder** is implemented, the UE needs to compute and record the log likelihood ratio (LLR) based on signal power sequence $P_{t,1}$ (in this case, only the first codeword in each layer needs to be transmitted). And the estimation of the user's angular direction is calculated as $\hat{\mathbf{u}} = g(\text{LLR})$.

In this way, the above constructed space-time beam pattern can distinguish $N_{\text{T}} = 2^k$ different directions, which completes the beam decoding stage.

IV. Implementation of Coded Beam Training

The preceding example of $(7, 4)$ Hamming code is only suitable to 16-antenna systems. To this end, this section considers the practical implementation of coded beam training.

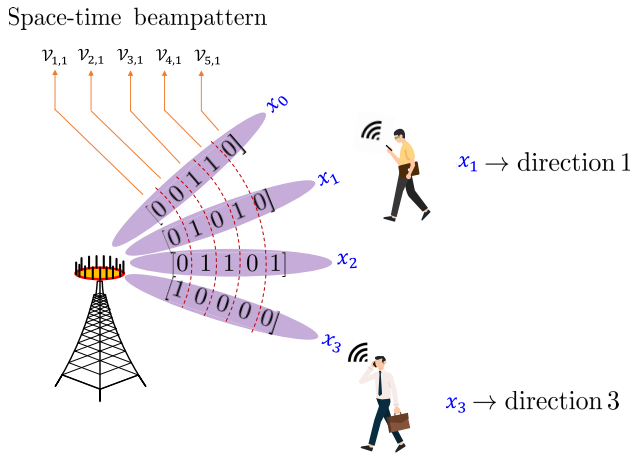


Fig. 4. Space-time beam pattern: directions are encoded into $\boldsymbol{x}$ and then beam pattern $\mathcal{V}$ is constructed based on it.

Specifically, we will first apply the convolutional channel code into the framework of coded beam training for communication systems with arbitrary number of antennas. Then, we also provide the extension of coded beam training to hybrid precoding architecture.

A. A Brief Introduction to Convolutional Codes

Invented by P. Elias in 1955, convolutional codes are efficient error-correcting codes that have already been widely adopted in 4G LTE control channel coding standards [34].

1) Convolutional Encoder: The (n, k, N) convolutional code is a coding scheme with memory that accepts a bitstream in blocks of length-k and outputs a bitstream in blocks of length-n. Each block of n output bits are determined by both the current input k bits and preceding $N-1$ blocks, where N represents the constraint length. Convolutional encoders are implemented by N shift registers with taps determined by the generator polynomials. Here we adopt a convolutional encoder of $N=3, k=1, n=2$, and the output bits $\boldsymbol{x}(2i-1), \boldsymbol{x}(2i)$ can be computed as

$$\boldsymbol{x}(2i-1) = \mathbf{u}(i) \oplus \mathbf{u}(i-1) \oplus \mathbf{u}(i-2), \tag{15}$$

$$\boldsymbol{x}(2i) = \mathbf{u}(i) \oplus \mathbf{u}(i-2). \tag{16}$$

The operation of the encoder proceeds as follows: Denote the bits $\mathbf{u}(i-1), \mathbf{u}(i-2)$ in the register $\mathrm{M1}, \mathrm{M2}$ as "state" and initialize the state as 00. Then the first input bit is fed into register M0 as $\mathbf{u}(i)$ and outputs $\boldsymbol{x}(2i-1), \boldsymbol{x}(2i)$ according to (15) and (16). Then, the next bit enters M0 while the previous bits in $\mathrm{M0}, \mathrm{M1}$ are shifted right to $\mathrm{M1}, \mathrm{M2}$. The process continues until eventually the last bit enters the register and the entire process is denoted as f_{conv}.

2) Viterbi Decoder: There are a variety of algorithms for decoding the received power sequence, among which Viterbi algorithm is an effective and practical technique [35]. Exploiting dynamic programming [35], the Viterbi decoder works in a sequential manner, where the output LLRs of the noisy channel are sequentially fed into a trellis graph. Then, sequential maximum a posteriori (MAP) estimators are applied to this trellis graph, in order to retrieve the most possible information sequence as survival paths and exclude impossible paths gradually. Fig. 5 illustrates the trellis of the utilized decoder with the initial state being 00. In the trellis, each node denotes one of $2^{N-1}=4$ states and each column corresponds to a decoding turn with 4 nodes.

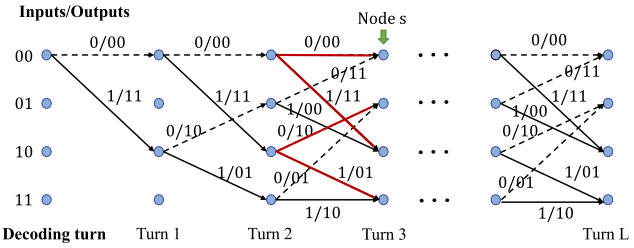


Fig. 5. Trellis of the convolutional decoder.

B. Convolutional Beam Encoding

As proposed in Section. III, to perform coded beam training, we divide the spatial direction into N_{T} segments and recover the direction with (n, k) $(k = \log_2 N_{\mathrm{T}})$ channel codes. To further improve the performance, inspired by [36], we propose an improved codebook. The codebook contains $T^{\mathrm{conv}} = 2\log_2 N_{\mathrm{T}} - 1$ layers, consisting of two parts: bottom layer and $T^{\mathrm{conv}} - 1$ remaining upper layers. The bottom layer can be designed according to (6), making use of the high direction gain of codebook $\mathcal{W}$ to improve beam training performance. For the upper layers, we only divide the spatial direction into $\frac{N_{\mathrm{T}}}{2}$ parts and select one of the $\frac{N_{\mathrm{T}}}{2}$ directions through coded beam training, which contains two codewords in the bottom layer. Then the two codewords will be tested later to determine the optimal codeword.

1) Space-Time Beam Pattern Design: First, we focus on the design of the space-time beam pattern in the upper layers based on the encoding algorithm of convolutional code f_{conv}. Each of upper layer only includes one codeword since we utilize a soft decoder instead of a hard decoder to improve the performance. Denote the codebook in the upper layers as $\mathcal{C}^{\mathrm{conv}}$ and corresponding space-time beam pattern as $\mathcal{V}^{\mathrm{conv}}$. To design the space-time beam pattern, we index all the possible angular directions with a bitstream of length $k = \log_2 N_{\mathrm{T}} - 1$ and obtain the coded bit sequence as

$$\boldsymbol{x}_s = f_{\mathrm{conv}}(\mathbf{u}_s) \in \{0,1\}^{T^{\mathrm{conv}}-1}, \quad s \in \{0, 1, \cdots, 2^k - 1\}. \tag{17}$$

Then, we construct the space-time beam pattern $\mathcal{V}_t^{\mathrm{conv}}, t \in \{1, 2, \ldots, T^{\mathrm{conv}} - 1\}$ in codebook $\mathcal{C}^{\mathrm{conv}}$ according to the encoded bits as

$$\mathcal{V}_t^{\mathrm{conv}}(s) = \boldsymbol{x}_s(t), \quad s \in \{0, 1, \cdots, 2^k - 1\}, \tag{18}$$

Here, since each layer only contains one codeword, we skip the subscript b.

2) Generate Corresponding Codewords: Next, we focus on generating the codewords corresponding to the designed space-time beam pattern. In general, the normalized codewords of t-th layer $t \in \{1, 2, \ldots, T^{\mathrm{conv}}\}$ is denoted as $\mathcal{C}_t^{\mathrm{conv}}$ and the

corresponding complex beam gain vector of a multi-mainlobe beam $\mathcal{C}_t^{\text{conv}}$ with beam coverage B_t is denoted as

$$\mathbf{g}_t = [g_t(\phi_1), g_t(\phi_2), \cdots, g_t(\phi_M)], \tag{19}$$

where M is the sampled angle number for beam generation. The beamforming gain can be presented as $g_t(\phi_m) = |g_t(\phi_m)|e^{\omega_m}$ where ω_m is phase information and the absolute beam gain $|g_t(\phi_m)|$ is predefined as [36] and [37]

$$|g_t(\phi_m)| = \begin{cases} \sqrt{2/|B_t|}, & \phi_m \in B_t \\ 0, & \phi_m \notin B_t \end{cases} \tag{20}$$

where $|B_t|$ is the coverage length of B_t. In convolutional coding aided hierarchical codebook, the B_t can be written as

$$B_t = \bigcup_s D_s, \text{ if } \mathcal{V}_t^{\text{conv}}(s) = 0, s \in \{0, 1, \cdots, 2^k - 1\}, \tag{21}$$

where $D_s = [-1 + s/2^{k-1}, -1 + (s+1)/2^{k-1}] \subset [-1, 1]$. Note that for the proposed method, each of the codeword corresponds to a sawtooth-shaped beam covering half of the entire angular range $[-1, 1]$. Thus, the $|B_t|$ equals 1 for all layers in the codebook. Therefore, the ideal beam gain can be calculated as $\sqrt{2/|B_t|} = \sqrt{2}$ for all layers.

Obtaining the absolute beam gain vector we can generate the codewords based on GS codeword design algorithm proposed in [23]. Employing the similarity of phase retrieval problem and codeword design [38], GS-based codeword design procedure is shown in **Algorithm 1**.

Algorithm 1 GS-Based Codeword Design

Input: $|\mathbf{g}|$, I_{max}, $\mathbf{A}$

1: Randomly initial phase information $w_m, m \in \{1, 2, \cdots, M\}$ to obtain $\mathbf{g}^0$
2: **for** each $i \in [1, I_{max}]$ **do**
3: calculate $\mathbf{v}^i$ based on $\mathbf{g}^{i-1}$ according to (22)
4: $\mathbf{g}^i = |\mathbf{g}| \angle \mathbf{A}^H \mathbf{v}^i$
5: **end for**
6: $\mathcal{C}_{\text{conv}} = (\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{A}\mathbf{g}^{I_{max}}$

Output: Designed codeword $\mathcal{C}_{\text{conv}}$

Firstly, for each codeword, we randomly initial the phase information $w_m, m \in \{1, 2, \cdots, M\}$ to obtain $\mathbf{g}^0$ (since the codeword generation is the same for each layer of codeword, we omit the subscripts of layer t here). Then in the i-th iteration, $\mathbf{v}^i$ is calculated by least square algorithm as

$$\mathbf{v}^i = (\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{A}\mathbf{g}^{i-1}, \tag{22}$$

where $\mathbf{A} \in \mathbb{C}^{N_{\text{T}} \times M}$ can be expressed as

$$\mathbf{A} = [\boldsymbol{\alpha}^H(\phi_1), \boldsymbol{\alpha}^H(\phi_2), \cdots, \boldsymbol{\alpha}^H(\phi_M)] \tag{23}$$

In this way, the current complex beam gain can be written as $\mathbf{A}^H\mathbf{v}^i$. In order to maintain the amplitude information of desired $\mathbf{g}$, we only utilize the phase information of current beam pattern $\mathbf{A}^H\mathbf{v}^i$ to update $\mathbf{g}^i$. After the iteration number reaches I_{max}, the designed codeword $\mathcal{C}^{\text{conv}}$ can be obtained. Futhermore, to fairly compare different codewords in each test, we usually normalize $\mathcal{C}_t^{\text{conv}}$ so that $\|\mathcal{C}_t^{\text{conv}}\|_2 = 1$.

In the beam training process, the BS sequentially transmits $\mathcal{C}_t^{\text{conv}}, t \in \{1, 2, \ldots, T^{\text{conv}} - 1\}$ to the UE. Then the UE records the received signal power sequence $\mathcal{P}_t$ for beam decoding in the following section.

C. Convolutional Beam Decoding

The objective of beam decoding is to select the optimal codeword in $\mathcal{W}$. Based on the received signal power sequence, we are capable of determining whether the UE is in the coverage B_t of $\mathcal{C}_t^{\text{conv}}$, and thus recover the spatial information to select codeword with the aid of convolutional decoding algorithm.

1) Calculate LLR: As discussed above, the critical component to the efficiency of convolutional decoder is the accurate computation of LLR. Therefore, in this paragraph, we will focus on the calculation of LLR.

In majority of typical research on channel codes, only Gaussian channel is employed. However, when it comes to beam training problem, only the power of the received signal, rather than the entire signal, can be measured. Since the received power obeys χ^2 distribution rather than Gaussian distribution for typical channel codes research, modifications are supposed to be made to the LLR calculator to adapt to the beam training problem. Specifically, according to (1), the received power can be calculated as $P_t = |\sqrt{P}\boldsymbol{h}\boldsymbol{w} + n|^2$ with transmitted symbol $s_0 = 1$, where n the complex circularly-symmetric additive white Gaussian noise $n \sim \mathcal{CN}(0, \sigma^2)$.

If the UE is in the coverage B_t of the beam in the t-th layer, the ideal received beam gain is $|\sqrt{P}\boldsymbol{h}\boldsymbol{w}| = A_t$, which is determined by the coverage length of B_t as (20). Thus, the power of received signal is $\mathcal{P}_t = |A_t + n|^2$. In such cases, the received power obeys non-central χ^2 distribution with degree of freedom $df = 2$. Therefore, the existing decoding algorithm can not be directly employed. The conditional probability density function of $\mathcal{P}_t$ should be recalculated as

$$p(\mathcal{P}_t = x | \theta_{\text{UE}} \in B_t) = \frac{1}{\sigma^2} e^{-\frac{x + A_t^2}{\sigma^2}} I_0\left(\frac{\sqrt{A_t^2 x}}{\sigma^2/2}\right), \tag{24}$$

where I_0 is zeroth order modified Bessel function of the first kind. Similarly, if the UE is *not* in the coverage B_t of the beam in the t-th layer, the received beam gain is $|\sqrt{P}\boldsymbol{h}\boldsymbol{w}| = 0$ and the signal power is $|n|^2$, correspondingly. Thus, received power obeys central χ^2 distribution with degree of freedom $df = 2$, i.e.

$$p(\mathcal{P}_t = x | \theta_{\text{UE}} \notin B_t) = \frac{1}{\sigma^2} e^{-\frac{x}{\sigma^2}}. \tag{25}$$

Therefore, the modified LLR can be expressed as

$$\begin{aligned} \text{LLR} &= \log \frac{p(\mathcal{P}_t = x | \theta_{\text{UE}} \in B_t)}{p(\mathcal{P}_t = x | \theta_{\text{UE}} \notin B_t)} \\ &= -\frac{A_t^2}{\sigma^2} + \log I_0\left(\frac{\sqrt{A_t^2 x}}{\sigma^2/2}\right) \end{aligned} \tag{26}$$

After obtaining LLR, beam decoding can be performed to recover the information bits through the Viterbi decoder, which is specified in next paragraph.

2) Recover the Spatial Index Through Decoding Process: The Viterbi algorithm-based beam decoding process proceeds in a step-by-step fashion as follows:

For **initialization** step, set survival paths as empty and initial loss of survival paths $\text{loss}_0 \in \mathbb{R}^{1\times 4}$ as $\mathbf{0}$ for all $2^{N-1}=4$ states (i.e. nodes in the trellis). In the **decoding** process, we divide the received power sequence into L groups of $n=2$ received powers $P_{2l-1}, P_{2l}, l \in \{1,2,\dots L\}$. Each group of powers P_{2l-1}, P_{2l} corresponds to a decoding turn. In the l-th decoding turn, firstly, the BS records the received powers P_{2l-1}, P_{2l}, and compute the LLR as $llr1_l$ and $llr2_l$ according to (26).

Then we attempt to calculate the survival paths for the nodes in the l-th decoding turn. Denote the two coded sequences (i.e. outputs of channel coding) of the paths entering the node s as $\boldsymbol{y_{s1}}$ and $\boldsymbol{y_{s2}}$ and the corresponding incoming nodes as node t_1 and t_2, respectively. For example, for the node s in Fig. 5, the incoming nodes are $t_1=1$ and $t_2=2$ while the coded sequences of entering paths are $\boldsymbol{y_{s1}}=00$ and $\boldsymbol{y_{s2}}=11$. Then UE can compute the "distance" for two paths entering each state of the trellis by adding the "distance" of incoming branches to the "distance" of the connecting survival path from incoming node t level $l-1$ as

$$l_1 = \text{loss}_{l-1}(t_1) + (-1)^{\mathbb{I}(\mathbf{y_{s1}}(1)=0)} llr1 + (-1)^{\mathbb{I}(\mathbf{y_{s1}}(2)=0)} llr2 \tag{27}$$

$$l_2 = \text{loss}_{l-1}(t_2) + (-1)^{\mathbb{I}(\mathbf{y_{s2}}(1)=0)} llr1 + (-1)^{\mathbb{I}(\mathbf{y_{s2}}(2)=0)} llr2 \tag{28}$$

where $\mathbb{I}(\cdot)$ is the indicator function. According to maximum a posteriori estimators, UE can select the lowest "distance" as the $\text{loss}_l(s)$ of survival path for node s in l-th turn, which can be presented as

$$\text{loss}_l(s) = \min\{l_1, l_2\}, \tag{29}$$

and the selected node is node t^*. Denote the input bit corresponding to the selected incoming path as $b(s) \in \{0,1\}$. Then the BS updates the survival paths as

$$\text{path}_l(s) = \text{append}(\text{path}_{l-1}(t^*), b(s)) \tag{30}$$

Continue the computation until the algorithm completes its forward search. Then the BS can select the node with lowest "distance" at the terminal decoding turn and the corresponding survival path as $\mathbf{q}$. Then the path $\mathbf{q}$ will be feedback to the BS. Through this way, BS can obtain a decimal index $\mathcal{T} = \text{bintodec}(\mathbf{q})$ which includes two codewords in the bottom layer. Lastly, in the bottom layer, we test two codewords of index $2\mathcal{T}+1$ and $2\mathcal{T}+2$ in codebook $\mathcal{W}$ to acquire the final selected codeword.

3) ML Decoder: Different decoding algorithms can result in different beam training performances, therefore to intuitively evaluate the effectiveness of our improved decoder we attempt to derive the performance bound of convolutional code. Maximum likelihood (ML) decoding is the optimal decoding method that minimizes the probability of decoding errors when each codeword is sent with an equal probability. The computational complexity of ML decoder prohibits it from practical employment since the required computation complexity grows exponentially with the input length. However, it serves as the performance bound of convolutional codes. For the first $T^{\text{conv}}-1$ layers, ML decoder selects the UE direction index $idx=s$ with the maximal probability of received signal $\boldsymbol{x}$, i.e.

$$s = \arg\max_j p(\boldsymbol{x}|j) \tag{31}$$

Let $\mathcal{N}_{0s} = \{t|\mathcal{V}_t(s)=0, t\in\{1,2,\dots,T^{\text{conv}}-1\}\}$ be the set where the beam pattern is 0, while $\mathcal{N}_{1s} = \{t|\mathcal{V}_t(s)=1, t\in\{1,2,\dots,T^{\text{conv}}-1\}\}$ be the set where the beam pattern is 1. Therefore, $p(\boldsymbol{x}|s)$ can be expressed as

$$\begin{aligned} p(\boldsymbol{x}|s) = &\prod_{t\in\mathcal{N}_{0s}} \frac{1}{\sigma^2} e^{-\frac{x_t}{\sigma^2}} \\ &\cdot \prod_{t\in\mathcal{N}_{1s}} \frac{1}{\sigma^2} e^{-\frac{x_t+A_t^2}{\sigma^2}} I_0\left(\frac{\sqrt{A_t^2 x_t}}{\sigma^2/2}\right) \end{aligned} \tag{32}$$

Thus, the log likelihood is

$$\begin{aligned} &\log p(\boldsymbol{x}|s) \\ &= -2N\log\sigma - \sum_{t\in\mathcal{N}_{0s}} \frac{x_t}{\sigma^2} - \sum_{t\in\mathcal{N}_{1s}} \frac{x_t + A_t^2}{\sigma^2} \\ &\quad + \sum_{t\in\mathcal{N}_{1s}} \log I_0\left(\frac{\sqrt{A_t^2 x_t}}{\sigma^2/2}\right) \\ &= -2N\log\sigma - \frac{N_{1s}A_t^2 + \sum x_t}{\sigma^2} + \sum_{t\in\mathcal{N}_{1s}} \log I_0\left(\frac{\sqrt{A_t^2 x_t}}{\sigma^2/2}\right) \end{aligned} \tag{33}$$

where $N_{1s} = |\mathcal{N}_{1s}|$ is the number of elements in the set $\mathcal{N}_{1s}$. Then the ML decoder then can be simplified as

$$s = \arg\max_j \sum_{t\in\mathcal{N}_{1j}} \left(-\frac{A_t^2}{\sigma^2} + \log I_0\left(\frac{\sqrt{A_t^2 x_t}}{\sigma^2/2}\right)\right) \tag{34}$$

Based on (34), we sequentially test all N_{T} direction indexes to select the optimal index s and then acquire the performance bound of convolutional decoder. The more the performance of a designed decoding algorithm approaches that of ML decoder, the more efficient the designed decoder is.

D. Adaptive Beam Encoding Based on Decoding Algorithm

According to the proposed method in the subsections above, although the error correction capabilities can help resolve the "error propagation" dilemma, the reliability of beam training performance for UEs may be limited by the low directional gain of wide beams. Specifically, for traditional binary-search-based hierarchical beam training, the spatial region covered by a certain codeword at any layer of the codebook is reduced to half in the next layer based on the feedback from the UE. In this way, the beamforming gain can gradually improve. On the contrary, the proposed method above is a **non-adaptive** method. The beams of the coded codebook are all wide beams or sawtooth-shaped covering half of the overall search angular space, which limits its performance.

To tackle this challenge, we address the problem by an adaptive method. Specifically, we exploit the "self-truncation" feature of the Viterbi decoder to narrow down the beam width layer-by-layer in the coded beam training procedure, just like the traditional hierarchical beam training does. To better illustrate the idea, we recall the "self-truncation" feature of the Viterbi decoder with a (n, k, N) convolutional codes. In each decoding turn of Viterbi decoder, sequential MAP estimators are employed to select the most possible input sequences from the 2^k incoming paths as the survival path and exclude the remaining $2^k - 1$ paths. Inspired by this feature, the adaptive coded beam training method attempts to focus the energy of the beam on the directions corresponding to the survival paths only, while allocating no beam energy to those referring to the $2^k - 1$ excluded paths. By this means, the beam width becomes narrow and the beam gain is gradually increased. In this way, we can employ the error correction capabilities of channel codes while still retaining the advantage of traditional hierarchical beam training to gradually narrow down the beam coverage. The main adjustments for the adaptive method lie in two steps: **space-time beam pattern design** and **calculation of the adaptive beamforming gain**, which is presented in detail as follows.

Step 1: Space-time beam pattern design. Compared to the predefined space-time beam pattern in non-adaptive method, the adaptive method requires previous decoding results, i.e. survival paths to adjust the space-time beam pattern in the following layers. Thus, it requires the UE's feedback the survival path to the BS after each decoding turn.

Consider the $(2, 1, 3)$ convolutional codes employed in this work. As shown in Fig. 5, from the third decoding turn of Viterbi decoder (i.e. receiving the power of the 5-th and 6-th layer of codewords), the decoder retains a survival path $\mathrm{path}_l(s), s \in \{1, 2, 3, 4\}$ and exclude another impossible path according to (27)-(30) for each node (i.e. state) in the l-th decoding turn. In such cases, for the $l+1$-th group of encoding layers, we only inject energy to the intersection direction of the original non-adaptive beam coverage and the direction range corresponding to the survival path. The original non-adaptive beam coverage can be acquired according to (17),(18) and (21), which is the same as Section. IV-B. Besides, the specific survival direction range can be obtained as follows: transfer the survival paths $\mathrm{path}_l(s), s \in \{1, 2, 3, 4\}$ in the l-th decoding turn to decimal index $d_l(s)$, and the indicated direction corresponding to the path is

$$\mathrm{Dir}_l(s) = [-1 + d_l(s)/2^{l-1}, -1 + (d_l(s) + 1)/2^{l-1}]. \quad (35)$$

Then the survival direction range can be expressed as

$$S_l = \bigcup_s \mathrm{Dir}_l(s), s \in \{1, 2, 3, 4\}. \quad (36)$$

Note that survival direction region length after the l-th ($l \geq 3$) decoding turn can be calculated as $|S_l| = 2^{3-l}$, which reduces to half of that in the last turn. It means that for the $l+1$-th group of encoding layers, the beam coverage is no more than 2^{3-l} (exponential decline as traditional hierarchical beam training). Based on it, coverage of adjusted beam pattern in layer $t = 2l + 1$ and $t = 2l + 2$ as

$$B\mathrm{new}_t = B_t \bigcap S_l, t = 2l + 1, 2l + 2. \quad (37)$$

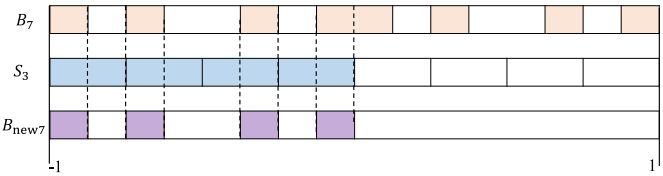


Fig. 6. Example of determining the beam pattern of the adaptive codebook.

To provide a better intuitive understanding of the proposed adaptive method, we take the space-time beam pattern design in the 7-th layer as an example. As illustrated in Fig. 6, the original beam coverage of 7-th layer codeword $\mathcal{C}_{\mathrm{conv}}(7)$ is B_7. If the survival paths are $000, 010, 001, 011$ (red lines in Fig. 5), the corresponding survival direction range is $[-1, 0]$. Thus, the adjusted beam pattern only covers the intersection of the above set. In this way, the beams are narrowed down and the beamforming gain is improved.

Second step: Calculate the adaptive beamforming gain. Different from the non-adaptive method, due to the different beam coverage length in different layers, beamforming gains also vary across different layers. Based on (20), the beamforming gain in each layer should be respectively calculated as $A_t = \sqrt{2/|B\mathrm{new}_t|}$.

Then we can employ the beamforming gain for codebook generation based on GS algorithm and beam decoding process, which are exactly the same with non-adaptive method. We summarize the adaptive coded beam training framework as Fig. 7.

E. Extension to Hybrid Precoding Structure

As we have clarified in Section. II-A, the proposed method is independent of the precoding architecture, so we utilize full-digital precoding scheme for concise representation. In this subsection, we will demonstrate how the proposed method can be conveniently transfered to hybrid precoding scheme, which is widely employed in exsiting communication systems.

Consider a typical mmWave/Terahertz massive MIMO system where the BS employs $N_{\mathrm{RF}}(N_{\mathrm{RF}} \ll N_{\mathrm{T}})$ RF chains to serve a single-antenna user. The BS employs hybrid precoding, and the optimization problem can be decomposed into two sub-problem: digital precoding optimization and analog precoding optimization. For analog precoding, the training process is the same with that of full-digital structure. The codewords chosen finally in coded beam training meet the requirements of constant envelop constraint due to phase shifters. The only difference lies in that the codewords required during beam training should be generated in hybrid structure insetead of full-digital structure. As for this issue, the authors in [27] have verified that the full-digital structure can be approximate by hybrid precoding with $N_{\mathrm{RF}} > 2N_s$ RF chain where N_s denotes the data stream number. Besides, several beamformers [16], [26], [28] have been proposed to generate required wide beams with hybrid structure. Therefore,

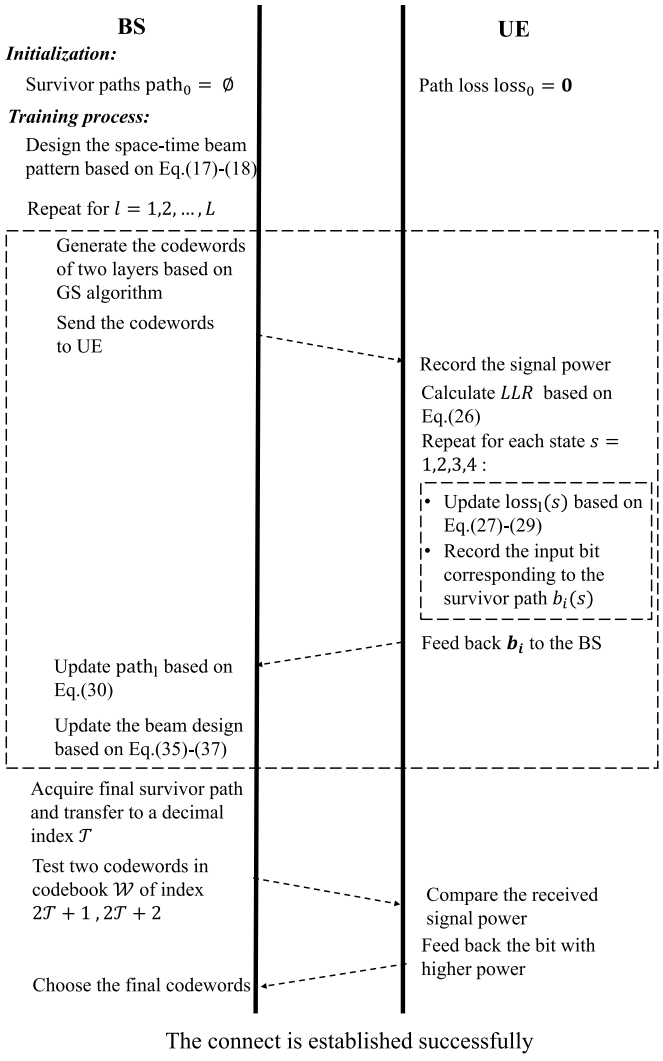


Fig. 7. Signaling diagram of the coded beam training procedure.

the proposed method can be directly transfered to analog beamformer design.

After obtaining the analog beamformer, we design the digital precoding based on the low complexity Zero Forcing (ZF) algorithm [1].

F. Extension to Multi-User Scenarios

The above proposed model supposes single-user communication scenarios for clear expression, and in this subsection we aim to reveal the scalability of proposed coded beam training framework to multi-user scenarios.

For non-adaptive method, the space-time beam pattern and codeword generation process described in Section IV-B is independent of UE, so the BS can send the same codewords to different users. Then each user performs Viterbi decoding to find the survival paths simultaneously by their own. As for the adaptive beam encoding where we adapt the beam design according to the feedback during the training process, the only difference is that we are supposed to inject energy to angular directions of the **union** of survival paths for each user.

V. Simulation Results

A. Complexity Analysis

In this subsection, we provide complexity analysis from the perspective of training overhead, feedback overhead, implementation complexity and computational complexity. We attempt to depict a clear comparison across exhaustive beam training, traditional binary search-based hierarchical beam training, and the proposed adaptive/non-adaptive coded beam training, as shown in Table. II.

1) Training Overheads: In the proposed method, the BS transmits a single codeword to the UE for upper $T^{\text{conv}}-1$ layers of $\mathcal{C}$, which occupies $T^{\text{conv}}-1$ time slots. At the bottom layer of $\mathcal{C}$, the BS sends two codewords to the UE, for the determination of the ultimate chosen codewords, which takes up 2 time slots. Therefore, the proposed beam training scheme takes up $T^{\text{conv}}+1 = 2\log_2 N_{\text{T}}$ time slots. Note that here we suppose the employed code rate $k/n = 0.5$. For an arbitrary code rate k/n, the coded layer number $T-1=\frac{n}{k}(\log_2 N_{\text{T}}-1)$. Suppose $N_{\text{T}}=1024$, the training overheads of our proposed method, exhaustive beam training and traditional hierarchical beam training are $20, 1024$ and 20 time slots, respectively. Our proposed method exhibits training overheads comparable to binary search-based hierarchical beam training and substantially curtails training overheads by 98% compared with exhaustive beam training.

2) Feedback Overheads: We also conduct a comparison of the feedback overhead from the UEs to the BS. In the proposed non-adaptive coded beam training method in Section. IV-B and Section. IV-C, the method requires one time slot to feedback the decoded angular direction after codewords in $T^{\text{conv}}-1$ layers are all transmitted. Then after the beam training for the bottom layer, an additional time slot is expended to feed the beam index back to BS. Therefore, the number of the cumulative feedback time slot is 2. In contrast, for the adaptive coded beam training method, BS necessitates the feedback from the UE to dynamically adjust the beam pattern every two layers. In such cases, the time slots needed amount to $\log_2 N_{\text{T}}-1$. Adding the time slot required in the bottom layer, there are $\log_2 N_{\text{T}}$ time slots required in total. It is consistent with binary search-based hierarchical beam training which also relies on the feedback from UE to choose codewords for the subsequent layer within the codebook of length $\log_2 N_{\text{T}}$. For exhaustive beam sweeping, UE only needs to feed back after receiving all codewords, which results in totally 1 times of feedback.

3) Implementation Complexity: First, we consider implementation complexity for codebook generation. For adaptive coded beam training, the beam patterns in the lower layers are determined both by the encoding algorithm and feedback of the upper layers. Therefore, the codebook is supposed to be generated online. In contrast, the codebooks of the other three methods can all be generated offline. Secondly, as discussed before, adaptive coded beam training, as well as traditional hierarchical beam training requires real-time feedback to determine the following codewords. While non-adaptive coded beam training and exhaustive beam training can avoid the need for real-time feedback and complex signaling control.

Finally, as for the scalability of multi-user beam training, the proposed non-adaptive method and exhaustive method enable simultaneous beam training to different users directly. The proposed adaptive method can also support multi-user beam training simultaneously with modification in Section. IV-F. However, the traditional beam training method has to perform beam training to different users in a time division way. In summary, the proposed non-adaptive coded beam training presents very low implementation complexity, comparable to simple exhaustive method. The implementation complexity of proposed adaptive method is similar to that of traditional hierarchical beam training (better scalability to multi-user beam training but requirement for online codebook generation).

4) Computational Complexity: The additional computational complexity introduced by coded beam training is mainly derived from online codebook generation and beam decoding. To perform beam decoding, the users are supposed to utilize Viterbi decoder to recover the optimal codeword index. The computational complexity of decoding a L-bit information bits with (n, k, N) Viterbi codes is $\mathcal{O}(L2^{k(N-1)})$. Thus, the complexity of Viterbi decoder grows linearly with the information bit length. Since the information bit length L in beam training is much lower than the bit length during data transmission in current 5G communications, the complexity of beam encoding is relatively low and acceptable in practice. Besides, adaptive coded beam training method will introduce the complexity of online codebook generation, which depends on the employed generation method. As for the GS-based codeword design utilized in this work, the algorithm is quite efficient only with complexity $\mathcal{O}(I_{max} N_{\mathrm{T}} \log N_{\mathrm{T}})$.

B. Performance Analysis

In this section, numerical results are presented to evaluate the performance of the designed coded beam training framework. We consider an XL-MIMO communication system, where a single-antenna user is served by a BS. The BS is equipped 1024-antenna ULA, with spacing between antennas equal to $\lambda/2$ in digital precoding system. The carrier frequency is 60 GHz and the corresponding wavelength is set as $\lambda = 0.005$ m. We adopt the Saleh-Valenzuela channel model described in (2) with LoS component being the dominant path. We further assume the UEs are uniformly distributed within physical direction $[0, 2\pi]$. The performances are all averaged on the instantaneous results of 1000 random Monte Carlo realizations of channel. For convenient compression, we use CBT to present coded beam training while BT to present beam training in this section.

Since the beamforming gains vary for different codebooks and layers, we utilize a pre-beamforming SNR, which can be calculated as $\mathrm{SNR} = \frac{P\beta_0^2}{\sigma^2}$. The large-scale fading and transmit power are assumed to be the same across the array, thus we just suppose $P\beta_0^2 = 1$ for Fig. 8 to Fig. 13.

First, we compare the average rate performance of the proposed adaptive/non-adaptive coded beam training with soft/hard decoder, respectively in Fig. 8. Specifically, [25] introduces the basic idea of channel codes-based beam training with a linear block code. However, it only employs a hard decoder and non-adaptive encoding process. Thus, we extend the method by utilizing convolutional codes as a benchmark. Besides the proposed adaptive method with soft decoder, we also depict the performance of adaptive method with hard decoder and non-adaptive method with soft decoder to demonstrate the source of gain better. Based on the simulation results in Fig. 8, the proposed adaptive method with soft decoder significantly outperforms other methods. Therefore, we can safely conclude that proposed improvement of adaptive encoding process and LLR calculator helps better accommodate to the beam training problem and thus significantly improves the performance.

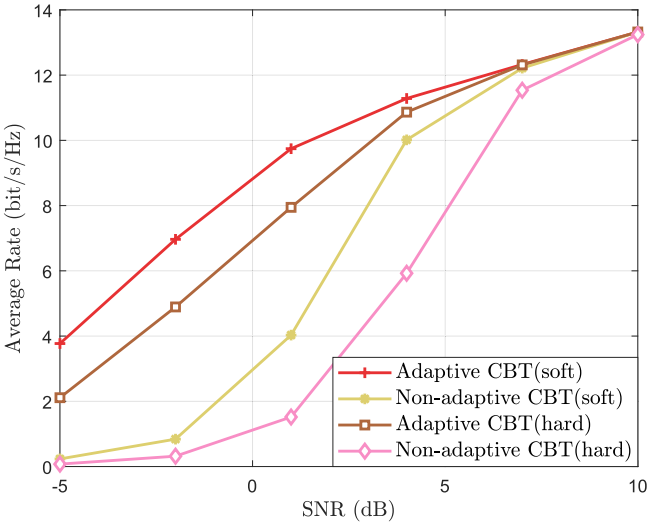


Fig. 8. Comparisons of the average rate for adaptive/non-adaptive CBT with soft/hard decoders.

Thus, the following simulations are mainly based on the adaptive coded beam training with soft decoder. Next, we draw a comparison of the proposed method with other beam training methods. Besides exhaustive beam sweeping and hierarchical beam training discussed above, we also add repetitive code-based beam training as another benchmark. In [36], the authors proposed a beam training method, which can simultaneously conduct beam training for multiple users. The method can be viewed as a special kind of coded beam training with an uncoded codebook, consisting of sawtooth-shaped beams of $\log_2 N_{\mathrm{T}}$ layers. For fair comparison, we are supposed to compare the performance of the proposed coded beam training with repetitive code-based beam training with the same pilot overhead, rather than the uncoded one. To achieve this, we increase the power of the transmitted signal by n/k times, which is equal to repeatedly transmitting the signals.

Fig. 9 depicts a comparison of the success rate of beam training for different schemes. For fair evaluation, we emphasize that the proposed coded beam training method shares equivalent training overheads with that of binary search-based hierarchical beam training. In such a case, the performance gain can be attributed to the coding gain facilitated by channel codes, rather than an increased utilization of pilot resources. It is evident that the scheme in [12] attains a superior performance than the other schemes, which lies in the fact that the

TABLE II
COMPARISONS OF COMPLEXITY FOR DIFFERENT SCHEMES

Schemes	Adaptive CBT	Non-adaptive CBT	Exhaustive BT	Traditional hierarchical BT
Training overheads	$2\log_2 N_{\mathrm{T}}$	$2\log_2 N_{\mathrm{T}}$	N_{T}	$2\log_2 N_{\mathrm{T}}$
Feedback overheads	$\log_2 N_{\mathrm{T}}$	2	1	$\log_2 N_{\mathrm{T}}$
Real-time feedback and control	yes	no	no	yes
Online codebook generation	yes	no	no	no
Scalability to multi-user	middle	good	good	bad
Beam decoding	required	required	not required	not required

• BT denotes beam training, CBT denotes coded beam training, for expression clarity.

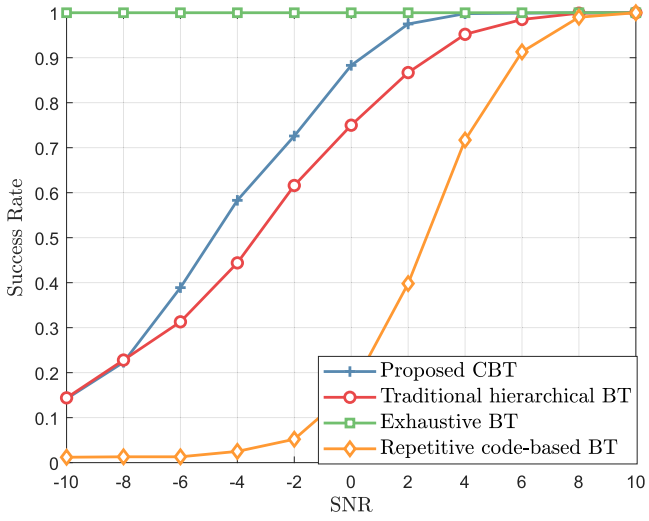


Fig. 9. Comparisons of the success rate for different beam training methods.

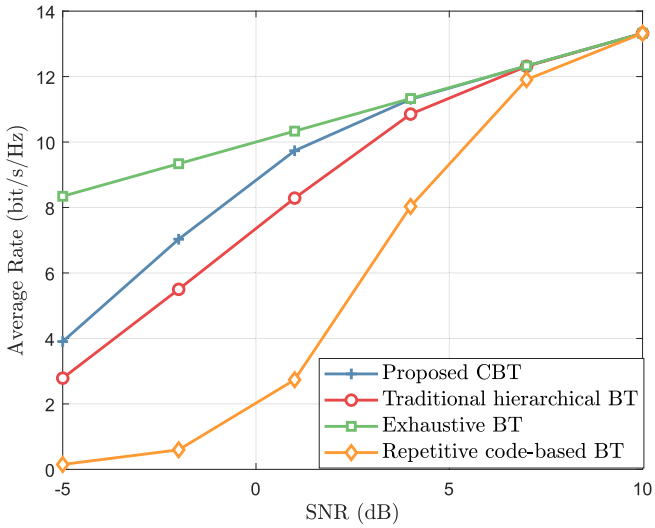


Fig. 10. Comparisons of the average rate for different beam training methods.

exhaustive beam sweeping, whose training overhead is much higher than the other schemes, inherently performs better at the expense of training efficiency. Existing hierarchical beam training method significantly reduces the training overheads, but it is not capable of obtaining reliable beam training performance for remote users with low SNR. This inadequacy arises from the "error propagation" phenomenon with the lower signal power of wide beam during beam training. Our proposed method significantly improves the success rate compared to existing beam training method, especially at low SNR while maintaining training overhead low, thanks to the leverage of the error correction capability of channel codes.

Fig. 10 offers a comparative view of the achievable rate for different beam training schemes. The graph clearly illustrates the performance of our proposed method outstands traditional hierarchical beam training method with comparative training overheads, especially at low SNR. Moreover, as the SNR increases, the performance gap between our scheme and the exhaustive beam sweeping scheme in [12] diminishes, where the curves of our scheme and the beam sweeping scheme almost coincide at $\mathrm{SNR} = 6\,\mathrm{dB}$. However, it's worth noting that the proposed method achieves considerable reduction (more than 98% reduction) in training overhead. As shown in Fig. 10, our proposed method outperforms the repetitive code-based beam training in all SNR regions. It means that the performance gain originates in the coding gain and adaptive encoding process, rather than transmitting more pilots. Based on the discussion above, we can conclude that our scheme strikes a remarkable balance between training overhead and beam training performance.

Furthermore, we verify the effectiveness of proposed enhanced convolutional decoding algorithm. Fig. 11 reveals the performance of proposed method and that with traditional decoder. The simulation results highlight the superior performance of our improved decoder in contrast to the traditional Gaussian distribution-based decoder, which substantiates the practicability of our modified decoding algorithm.

Besides, we have plotted the average rate performance under different BS antenna configurations, as shown in Fig. 12. The number of BS antennas increases from 64 to 2048 and the SNR is set as 5 dB. As illustrated in Fig. 12, since the increased number of antennas can acquire higher beamforming gain, the average rate increases with the growth of antenna number. The proposed method achieves near-optimal performance, comparable to exhaustive beam training. It outperforms the

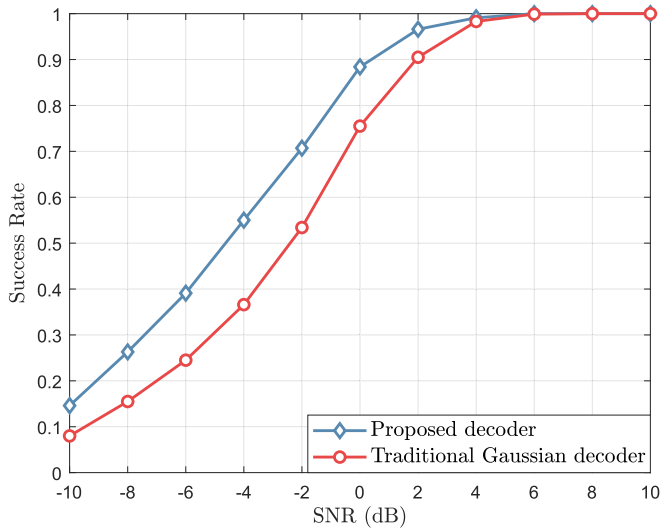


Fig. 11. Comparisons of the performance for convolutional decoders with different LLR calculators.

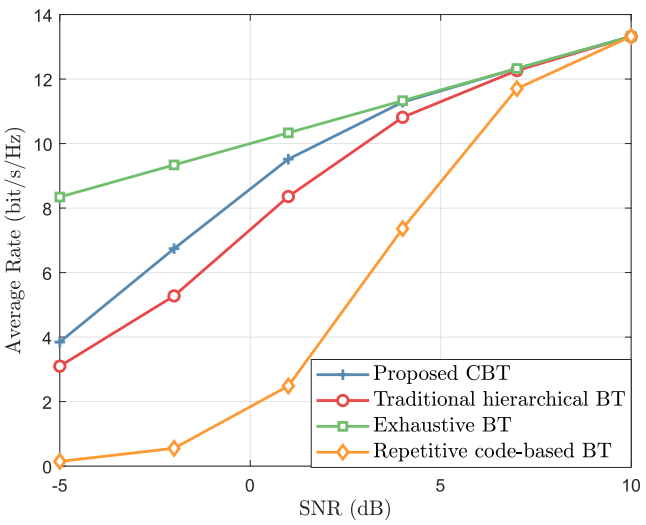


Fig. 13. Comparisons of the average rate for different beam training methods in mixed LoS and NLoS channel.

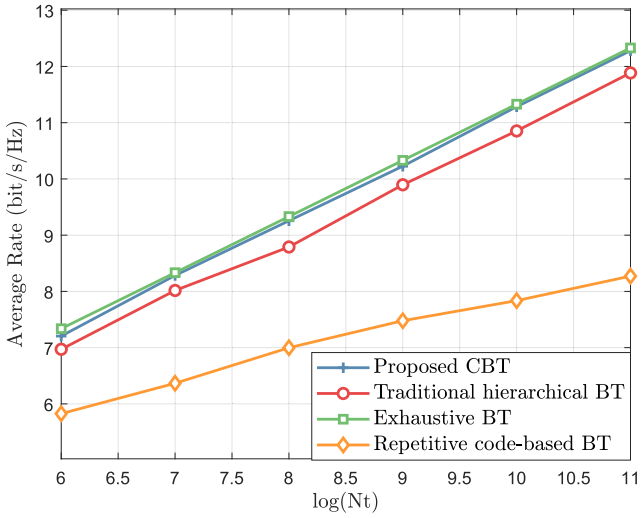


Fig. 12. Average rate performance under different BS antenna configurations.

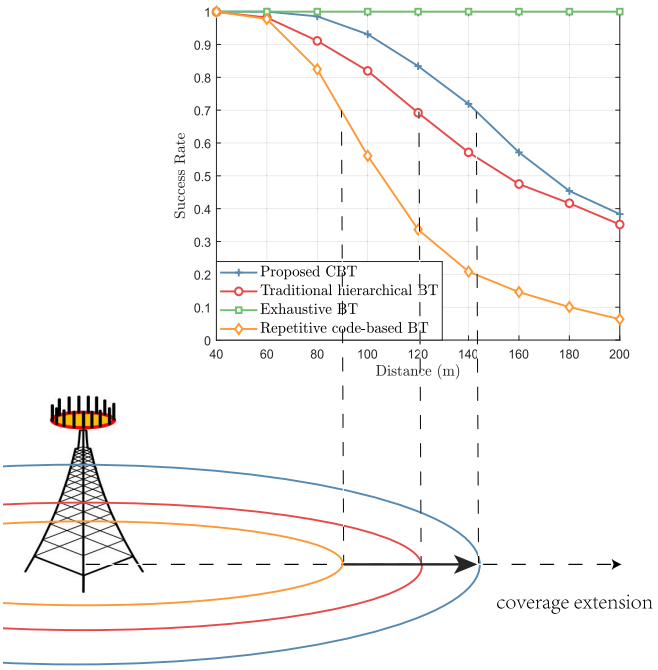


Fig. 14. Comparisons of success rate of different beam training methods with different distance, together with the illustration of the extended user coverage.

traditional hierarchical and repetitive code-based beam training in the entire SNR regime.

Then, we have considered multi-path mmWave channel model, which consists of one LoS path and three NLoS paths. The complex channel gain of the LoS path is set as β_0. To account for the scattering loss induced in the NLoS paths, their complex gains are generated by $\beta_i = \beta_0\alpha_i, \forall i \in \{1,2,3\}$, where α_i follow the Gaussian distribution $\mathcal{CN}(0, 0.1)$. Fig. 13 compares the average rate performance of different methods in such cases. The exhaustive beam training method still achieves the highest rate with N_T training overheads. The proposed method outperforms other methods with significantly reduced training overheads. Thus, the proposed method is promising to enable reliable beam training with high efficiency.

Besides, we present simulation experiments to illustrate the capability of the proposed coded beam training method to extend the coverage area. The simulation results are acquired under carrier frequency $f_c = 60\,\text{GHz}$ at mmWave frequency, transmit power of BS $P_t = 50\,\text{dBm}$, BS antenna number $N_\text{T} = 256$, subcarrier number $N_{sub} = 256$ and noise power $\sigma^2 = -110\,\text{dBm}$. As illustrated in Fig. 14, the proposed coded beam training framework can extend coverage area by more than $20\,\text{m}$ under the same success rate compared with traditional hierarchical beam training and more than $50\,\text{m}$ compared with repetitive code-based beam training. For instance, the proposed coded beam training achieves success rate 0.7 at $145\,\text{m}$ while traditional hierarchical beam training method can guarantee the same performance only at $120\,\text{m}$ and repetitive code-based beam training only $90\,\text{m}$. It is promising that the proposed coded beam training method can enable

beam training for remote users and thus extend the coverage area.

VI. Conclusion

In this paper, we introduce channel codes into hierarchical beam training to enable reliable implicit CSI acquisition for remote users in future 6G wireless communications. By proving the duality of hierarchical beam training and channel coding, we reveal that the hierarchical beam training problem can be transformed into designing channel codes, which enables the exploitation of the coding gain. We also demonstrate that the decoders need to be modified to fit the beam training problem. Simulation results have verified the effectiveness of the proposed method, which serves as a promising way to achieve reliable coverage of remote users. Future works can be focused on improving the multi-mainlobe beam generation algorithm to produce wide beams with better energy concentration. In addition, diverse channel coding methods can be utilized to further improve the coded beam training performance. The extension of the proposed coded beam training method to near-field communications [9], [39] and XL-RIS scenarios [7], [8] can be also considered in future works.

References

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[2] X. Wei, L. Dai, Y. Zhao, G. Yu, and X. Duan, "Codebook design and beam training for extremely large-scale RIS: Far-field or near-field?" *China Commun.*, vol. 19, no. 6, pp. 193–204, Jun. 2022.

[3] T. S. Rappaport et al., "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78729–78757, 2019.

[4] Z. Wang et al., "A tutorial on extremely large-scale MIMO for 6G: Fundamentals, signal processing, and applications," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 3, pp. 1560–1605, 3rd Quart., 2024.

[5] X. Ma, Z. Gao, F. Gao, and M. Di Renzo, "Model-driven deep learning based channel estimation and feedback for millimeter-wave massive hybrid MIMO systems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2388–2406, Aug. 2021.

[6] C. Han, L. Yan, and J. Yuan, "Hybrid beamforming for terahertz wireless communications: Challenges, architectures, and open problems," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 198–204, Aug. 2021.

[7] S. Yang, C. Xie, W. Lyu, B. Ning, Z. Zhang, and C. Yuen, "Near-field channel estimation for extremely large-scale reconfigurable intelligent surface (XL-RIS)-aided wideband mmWave systems," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 6, pp. 1567–1582, Jun. 2024.

[8] S. Yang, W. Lyu, Z. Hu, Z. Zhang, and C. Yuen, "Channel estimation for near-field XL-RIS-aided mmWave hybrid beamforming architectures," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 11029–11034, Aug. 2023.

[9] Y. Zhang, X. Wu, and C. You, "Fast near-field beam training for extremely large-scale array," *IEEE Wireless Commun. Lett.*, vol. 11, no. 12, pp. 2625–2629, Dec. 2022.

[10] W. Liu, C. Pan, H. Ren, F. Shu, S. Jin, and J. Wang, "Low-overhead beam training scheme for extremely large-scale RIS in near field," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4924–4940, Aug. 2023.

[11] 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, May 2016.

[12] A. Alkhateeb, G. Leus, and R. W. Heath Jr., "Limited feedback hybrid precoding for multi-user millimeter wave systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 6481–6494, Nov. 2015.

[13] X. Sun, C. Qi, and G. Y. Li, "Beam training and allocation for multiuser millimeter wave massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 1041–1053, Feb. 2019.

[14] M. Cui and L. Dai, "Channel estimation for extremely large-scale MIMO: Far-field or near-field?" *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2663–2677, Apr. 2022.

[15] C. You et al., "Near-field beam management for extremely large-scale array communications," 2023, *arXiv:2306.16206*.

[16] Z. Xiao, T. He, P. Xia, and X.-G. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.

[17] B. Ning, Z. Chen, Z. Tian, C. Han, and S. Li, "A unified 3D beam training and tracking procedure for terahertz communication," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2445–2461, Apr. 2022.

[18] J. Wang, W. Tang, S. Jin, C.-K. Wen, X. Li, and X. Hou, "Hierarchical codebook-based beam training for RIS-assisted mmWave communication systems," *IEEE Trans. Commun.*, vol. 71, no. 6, pp. 3650–3662, Jun. 2023.

[19] C. Wu, C. You, Y. Liu, L. Chen, and S. Shi, "Two-stage hierarchical beam training for near-field communications," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2032–2044, Feb. 2024.

[20] J. Wang et al., "Beam codebook based beamforming protocol for multi-gbps millimeter-wave WPAN systems," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1390–1399, Oct. 2009.

[21] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath Jr., "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.

[22] L. Chen, Y. Yang, X. Chen, and W. Wang, "Multi-stage beamforming codebook for 60GHz WPAN," in *Proc. 6th Int. ICST Conf. Commun. Netw. China (CHINACOM)*, Aug. 2011, pp. 361–365.

[23] Y. Lu, Z. Zhang, and L. Dai, "Hierarchical beam training for extremely large-scale MIMO: From far-field to near-field," *IEEE Trans. Commun.*, vol. 72, no. 4, pp. 2247–2259, Apr. 2024.

[24] Y. Shabara, C. E. Koksal, and E. Ekici, "Beam discovery using linear block codes for millimeter wave communication networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1446–1459, Aug. 2019.

[25] V. Suresh and D. J. Love, "Single-bit millimeter wave beam alignment using error control sounding strategies," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 1032–1045, Sep. 2019.

[26] Z. Xiao, H. Dong, L. Bai, P. Xia, and X.-G. Xia, "Enhanced channel estimation and codebook design for millimeter-wave communication," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9393–9405, Oct. 2018.

[27] E. Zhang and C. Huang, "On achieving optimal rate of digital precoder by RF-baseband codesign for MIMO systems," in *Proc. IEEE 80th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2014, pp. 1–5.

[28] S. Lyu, Z. Wang, Z. Gao, H. He, and L. Hanzo, "Lattice-based mmWave hybrid beamforming," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4907–4920, Jul. 2021.

[29] W. Xu, L. Gan, and C. Huang, "A robust deep learning-based beamforming design for RIS-assisted multiuser MISO communications with practical constraints," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 694–706, Jun. 2022.

[30] G. Sun, W. Yan, W. Hao, C. Huang, and C. Yuen, "Beamforming design for the distributed RISs-aided THz communications with double-layer true time delays," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3886–3900, Mar. 2024.

[31] K. Chen, C. Qi, and G. Y. Li, "Two-step codeword design for millimeter wave massive MIMO systems with quantized phase shifters," *IEEE Trans. Signal Process.*, vol. 68, pp. 170–180, 2020.

[32] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[33] B. Ning, T. Wang, C. Huang, Y. Zhang, and Z. Chen, "Wide-beam designs for terahertz massive MIMO: SCA-ATP and S-SARV," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10857–10869, Jun. 2023.

[34] P. Elias, "Coding for noisy channels," *IRE Conv. Rec.*, vol. 7, pp. 37–47, May 1955.

[35] G. D. Forney, "Convolutional codes II. Maximum-likelihood decoding," *Inf. Control*, vol. 25, no. 3, pp. 222–266, Jul. 1974.

[36] C. Qi, K. Chen, O. A. Dobre, and G. Y. Li, "Hierarchical codebook-based multiuser beam training for millimeter wave massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8142–8152, Dec. 2020.

[37] J. Song, J. Choi, and D. J. Love, "Common codebook millimeter wave beam design: Designing beams for both sounding and communication with uniform planar arrays," *IEEE Trans. Commun.*, vol. 65, no. 4, pp. 1859–1872, Apr. 2017.

[38] O. Bucci, G. Franceschetti, G. Mazzarella, and G. Panariello, "Intersecetion approach to array pattern synthesis," *IEEE Internet Things J.*, vol. 137, no. 6, pp. 349–357, Apr. 1990.

[39] M. Cui, Z. Wu, Y. Lu, X. Wei, and L. Dai, "Near-field MIMO communications for 6G: Fundamentals, challenges, potentials, and future directions," *IEEE Commun. Mag.*, vol. 61, no. 1, pp. 40–46, Jan. 2023.

Tianyue Zheng (Graduate Student Member, IEEE) received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2022. She is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Tsinghua University, Beijing, China. Her research interests include extremely large-scale MIMO (XL-MIMO), CSI acquisition, and AI for communications. She received the National Scholarship in 2019 and the Excellent Student of Jiangsu Province in 2021.

Jieao Zhu (Graduate Student Member, IEEE) received the B.E. degree in electronic engineering and the B.S. degree in applied mathematics from Tsinghua University, Beijing, China, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His research interests include electromagnetic information theory (EIT), coding theory, and quantum computing. He received the National Scholarship in 2018 and 2020 and the Excellent Graduates of Beijing in 2021.

Qiumo Yu (Graduate Student Member, IEEE) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2022, where he is currently pursuing the master's degree with the Department of Electronic Engineering. His research interests include extremely large-scale MIMO (XL-MIMO), mmWave/THz communications, and reconfigurable intelligent surfaces (RIS). He received the National Scholarship in 2024.

Yongli Yan (Member, IEEE) received the B.S. degree in electronic science and technology from Zhengzhou University, China, in 2014, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2019. From 2020 to 2023, he was a Senior Engineer with Huawei Technologies Company Ltd. He is currently a Post-Doctoral Research Fellow with the Department of Electronic Engineering, Tsinghua University. His research focused on key technologies in channel coding and VLSI implementation. His research interests include massive MIMO, millimeter-wave communications, channel coding, and AI-powered wireless communications. Dedicated to honest and innovative research, he aims to drive advances in wireless communications. He was honored with awards for Huawei Gold Medal Team and Outstanding Engineer from Huawei Technologies Company Ltd.

Linglong Dai (Fellow, IEEE) received the B.S. degree from Zhejiang University, Hangzhou, China, in 2003, the M.S. degree from China Academy of Telecommunications Technology, Beijing, China, in 2006, and the Ph.D. degree from Tsinghua University, Beijing, in 2011.

From 2011 to 2013, he was a Post-Doctoral Researcher with the Department of Electronic Engineering, Tsinghua University, where he was an Assistant Professor from 2013 to 2016, an Associate Professor from 2016 to 2022, and has been a Professor since 2022. He has co-authored the book *MmWave Massive MIMO: A Paradigm for 5G* (Academic Press, 2016). He has authored or co-authored over 100 IEEE journal articles and over 60 IEEE conference papers. He also holds over 20 granted patents. His current research interests include massive MIMO, reconfigurable intelligent surface (RIS), millimeter-wave and Terahertz communications, near-field communications, machine learning for wireless communications, and electromagnetic information theory. He has received five IEEE Best Paper Awards at IEEE ICC 2013, IEEE ICC 2014, IEEE ICC 2017, IEEE VTC 2017-Fall, IEEE ICC 2018, and IEEE GLOBECOM 2023. He has also received Tsinghua University Outstanding Ph.D. Graduate Award in 2011, Beijing Excellent Doctoral Dissertation Award in 2012, China National Excellent Doctoral Dissertation Nomination Award in 2013, the URSI Young Scientist Award in 2014, the IEEE Transactions on Broadcasting Best Paper Award in 2015, the Electronics Letters Best Paper Award in 2016, the National Natural Science Foundation of China for Outstanding Young Scholars in 2017, the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2017, the IEEE ComSoc Asia-Pacific Outstanding Paper Award in 2018, China Communications Best Paper Award in 2019, the IEEE Access Best Multimedia Award in 2020, the IEEE Communications Society Leonard G. Abraham Prize in 2020, the IEEE ComSoc Stephen O. Rice Prize in 2022, the IEEE ICC Best Demo Award in 2022, and the National Science Foundation for Distinguished Young Scholars in 2023. He was listed as a Highly Cited Researcher by Clarivate Analytics from 2020 to 2023.