Received: August 5, 2025. Revised: November 4, 2025.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Codes Achieving Reliable Beam Training in 6G Mobile Communications

Hawra'a Albusalih[1]* **Haider Al kim[1,2]**

[1]*Department of Electronics and Communications Engineering, Faculty of Engineering, University of Kufa, Iraq*
[2]*Former Doctoral Researcher at Coding and Cryptography Group, the Technical University of Munich, Germany*
* Corresponding author's Email: hawraaz.albusalih@student.uokufa.edu.iq

Abstract: This paper addresses the problem of reliable beam training in 6G mobile communication by considering two critical challenges: isolating early-stage errors in directional search and reducing feedback overhead. Our method utilizes a 2^2-ary hierarchical beam training structure, where a fixed set of four candidate beams is evaluated per stage, enabling faster convergence compared to conventional binary hierarchies. To confine error effects and prevent their propagation across angular layers, our method applies a specialized encoding scheme with dual error-and-erasure correction capabilities. This allows early misdetections to be isolated and corrected without affecting deeper search levels. Theoretical analysis and comparisons demonstrate that our 2^2-ary hierarchical fixed beam training attains a total training time of $2log_2N_T - 1$, delivering a low-latency, feedback-efficient framework for high-mobility, low-SNR 6G deployments. Our scheme effectively reduces error propagation in low-SNR environments and shortens training time by 25% compared to adaptive coded beam training, and by 10% compared to fixed coded beam training.

Keywords: 6G, Massive MIMO, Coded beam training, Q-ary codes, Error correction code, Erasure correction code, Hierarchical beam training, Fixed beam training, Channel coding, Extended hamming codes.

1. Introduction

Sixth-generation (6G) communications promise ultra-fast connectivity and high efficiency. However, these technologies have drawbacks, such as poor signal strength for users who are far away. Communication systems usually estimate the channel condition either explicitly by channel estimates throughout the channel state information (CSI) or implicitly by beam training. Explicit channel state information (CSI) estimation, which directly gets channel information via pilot signals, guarantees high precision in channel state representation. However, this approach may significantly affect spectral efficiency because of the substantial pilot overhead. It is also less feasible for large MIMO systems as it becomes more difficult to scale as the number of antennas rises. Therefore, small to medium antenna systems are more suited for implicit CSI estimation. Implicit channel state information (CSI) estimation, obtaining channel state information (CSI) through indirect or more complex methods reduces pilot overhead, making it a more resource-efficient approach. While this method is easier to scale for large antenna systems, such as extra-large MIMO (XL-MIMO), it requires optimized beam sweeping to ensure accurate channel estimation. Our work benefits from the implicit method to anticipate the channel status. Increasing directional accuracy and reducing errors improve beamforming efficiency and ensure reliable communications even in situations with low interference or signal-to-noise ratio [1]. To achieve the prior advancement, the system can use channel coding techniques to optimize the beamforming process, and this is known as Coded Beamforming. Applying coded beam can make the signal transmission more flexible and efficient, especially in situations where feedback is limited or channel state information (CSI) is not optimal [2]. Thus, "coded beamforming" has been developed to improve beamforming accuracy in wide-area MIMO (XL-MIMO) systems.

Due to the low signal-to-noise ratio of remote users, it is difficult to accurately estimate the channel state. Thus, there would be an increased probability of selecting suboptimal beams due to interference and noise, which negatively affects the efficiency of the system. The major challenge is the phenomenon of error propagation as the system uses hierarchical binary beam training, and as follows. This method relies on dividing the angular space into several layers (see an illustration in Fig. 1), starting by defining a wide space and then gradually narrowing it until the optimal angle is reached. Each stage depends on the previous results, making it vulnerable to initial errors in low-signal environments. Inaccurate decisions may cause significant errors that occur in the early stages of the experiment to propagate to further stages, resulting in reduced guidance accuracy and difficulty in correcting errors in these later stages.

Another method for conducting beam training is exhaustive training. It depends on fully examining all possible packets to obtain the best matching case with the channel, but it requires longer time and consumes more resources, which is not desirable in low signal-to-noise environments where time and resources are limited. Balancing training time with ensuring beam selection accuracy is one of the biggest challenges we face in beam training. Therefore, increasing the training time means reducing the time available for data transfer, which leads to a decrease in the overall efficiency of the system. Inaccurate training also leads to the signal being oriented at the wrong angle, which reduces efficiency and increases noise. The optimal scenario is to find a technique that allows training time to be reduced while maintaining high accuracy, so this requires careful mathematical modeling and simulation experiments to determine the point at which the ideal balance is achieved between these two factors.

2. Related work

Various channel coding techniques are used to improve beamforming reliability by mitigating errors and improving signal integrity in complex channel environments, including techniques such as Polarization Adjusted convolutional (PAC). PAC codes are a new family of linear block codes that can perform close to the theoretical limits in the short block length regime [3]. These codes can fix errors almost as well as the dispersive approximation, which is the limit of non-convergent channel coding. The study in [3] employs PAC codes because of their superior performance over traditional polar and twist codes, for short code lengths. They also ensure reliable transmission of short packets with minimal latency, which is one of the primary aims of the next generation of wireless communication systems [4]. However, the researchers focused on only short code lengths.

The base station (BS) conducts beam training for each user sequentially using time-division multiple access (TDMA). However, training overhead can increase linearly with the number of users. To mitigate this, a simultaneous multiuser beam training scheme is introduced, in which each layer—except for the bottom layer—utilizes only two multi-mainlobe codewords, regardless of the number of users the BS serves [5].

On the other hand, traditional hierarchical beam training schemes typically require multiple feedbacks from user equipment (UE) to the base station (BS) to indicate the best codeword for the base station. This leads to high overhead, especially in multiuser scenarios [6]. In [7], the authors propose a beam training scheme that only reduces the feedback requirement to only two feedbacks in total, regardless of the number of layers in the hierarchical codebook.

The method that is described in [8] improves beam training by integrating a channel attention module that selectively trains a specific subset of broad beams depending on signals received from prior sessions. By efficiently extracting broad beam features, the model sustains superior narrow beam prediction accuracy despite diminished observations, thereby reducing training overhead. Additionally, the method in [10] adaptively allocates training resources to various beams, assigning more symbols to those with greater beamforming gains. By focusing resources where they are most needed, the method improves beam training accuracy and reduces the impact of error propagation in beam alignment.

3. Our contribution

As the work in [3] considers only codes with short block lengths, our code schemes have no limitation on the lengths of the codes. Unlike [5], we propose 2^2 multi-mainlobe codewords such that we relay on the finite field $\mathbb{F}_{2^2}$ of the elements $\{0, \alpha, 1, \alpha+1\}$ to introduce four higher-level angular directions; see Fig. 1. Our approach relies on the fact that we only need one bit, namely the most significant bit to define which symbol from $\{0, \alpha\}$ or $\{1, \alpha+1\}$, and consequently we never get significant error propagation, i.e., either BS targets to the upper direction or to the lower direction.

Extending the work proposed by [7] which already improves upon [6], we incorporate the channel coding techniques not only to reduce the feedback overhead and reduce the training time but also to kill the error propagation at its early stages,

Received: August 5, 2025. Revised: November 4, 2025.

i.e., the super-resolution angular direction where its existence has a significant effect.

For comparison, our method differs from the approach in [8] (which is an uncoded method) that enhances alignment via channel feature weighting. Their method deals with the significant measurements without incorporating error correction within the feedback bits. This should prevent errors from changing the correct channel weighting. In contrast, our dual error–erasure coding scheme (described in Section 6) directly protects the most important feedback bit against errors, leading to more precise beam selection in low-SNR regimes.

While the algorithm of [10] is adaptive in training resource allocation and has high angular resolution, it neither employs feedback, nor error protection, as well as lacking an apparent training bound, the system unfairly allocates resources. It assigns more symbols to users who have better beamforming gains, which complicates the scheme. Our method, however, achieves low-complexity coded beam training in only $2\ log_2\ N_T + 1$ (fixed) or $2.5\ log_2\ N_T - 1$ (adaptive) steps (cf. Table 1) and is thus more resilient and suitable for low-SNR and constrained systems. We summaries our main achievements as follows:

1. Unlike other papers that are using coded beam training (only error correction codes) [2, 3, 4, 5], we incorporate erasure correction with error correction codes to stop the (significant) error (in hierarchical schemes) from propagating in its early stages.
2. Generalizing from [3, 4] that focus on codes of short lengths, our coding scheme works for any code family with any code length.
3. Improving upon [7] and (inherently) [6] (cf. Remark 3), we reduce the feedback overhead (to one only) and the training time (by at least $0.5\ log_2\ N_T - 1$ in the adaptive training and by at least 1 (cf. Table 1), comparing with other training techniques).

Our framework is specifically optimized for high-mobility, low-to-moderate SNR 6G deployments where reducing the training time to a minimum is paramount to fit within the short channel coherence time (T_c). The total training time of $2\ log_2\ N_T + 1$or $2.5\ log_2\ N_T - 1$ is significantly shorter than the exhaustive approach, enabling reliable training even when T_c is severely limited by user velocity.

In Section 4 we give our main notations and definitions. In next section, Section 5, our beam training model and channel model are defined and described. Section 6 gives our coding scheme for reliable feedback in beam training, including an example based on extended Hamming codes. In Section 7, we review current beam training methods and compare their performance with our method, and also includes the performance evaluation of the success rate of our scheme. Section 8 gives the complexity analysis of our scheme compared with others. Lastly, Section 9 concludes the paper.

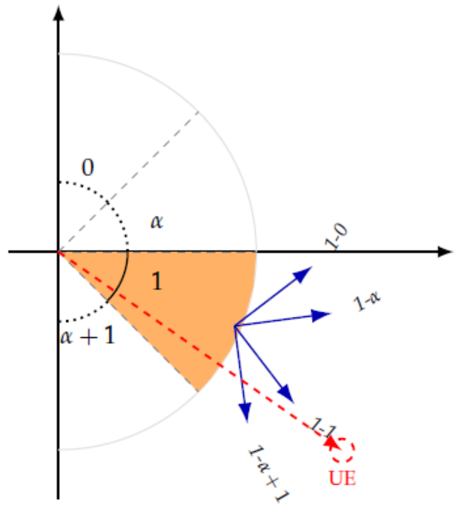

Figure. 1 Expected Beams Directions. The figure illustrates a two-level system: upper and lower layers. The spatial direction is initially divided into four primary angular regions: with $\theta 0 \in \{0, \alpha, 1, \alpha + 1\}$ matching the employed field elements (cf. Section 4-A). For example, the wide orange beam covering angle $1 = -45^\circ$ can be subdivided into narrower lower-layer angles, such as 1-0, ..., 1-N_T. The dashed red arrow indicates a narrow beam targeted at a user (UE) at angle 1-130

4. Preliminaries

4.1 Notations

Let $\mathbb{F}_{2^2}$ denote a finite (extension) field with four elements, $\{0, 1, \alpha, \alpha + 1\}$, where α is a root of a primitive irreducible polynomial in $\mathbb{F}_2$. Denote by N_T the number of transmit antennas in the system. Let t denote the maximum number of bits that can be erroneous in a transmitted codeword. As usual, an $[n, k, d]$q C denotes a linear code over a field $\mathbb{F}_q^{n-k}$ (over the alphabet q) of the parameters n, k and d, where n is the code length, k is the number of information symbols, and d is the minimum Hamming distance between two different codewords. This distance defines the code's capability to detect and correct errors or erasures.

Use τ as the number of allowed erasures, i.e., the number of wiped symbols the erasure correction can recover, and their known positions are given by the set $\mathcal{E}$. Let e $\in \mathbb{F}_{2^2}^{n}$be the error vector, representing the errors or erroneous values that occur in the received symbol sequence. $C_{n-1} \subset \mathbb{F}_q^{n-1}$ is a one symbol smaller code than $C \subset \mathbb{F}_q^n$, where every word has $n-1$ symbols. Remember our focus is on $q = 2^2$, and our coding scheme can be generalized to any q. Let $m \in \mathbb{F}_q^k$ be the message vector consisting of k information symbols before the inclusion of the redundant symbols (in our scheme an $n-k$ sequence of two bits as stated next), and let $p \in \mathbb{F}_q^{n-k}$ be the parity vector with $n-k$ symbols.

4.2. Definitions

4.2.1. Definition IV.1 (Beam training)

Beam training is an essential process in millimeter-wave (mmWave) and terahertz (THz) communication systems. In this process, the transmitter and receiver work together to find the best beam pair offering the highest directional link. The main issue in conventional training mechanisms is error propagation, with the errors occurring earlier at beam selection causing significant errors in the later layers. Thus, in Definition IV.2 we define Coded Beam Training.

4.2.2. Definition IV.2 (Coded beam training)

Coded beam training methods are an implicit approach for estimating the CSI, where error-correcting codes are employed during the beam training process to enhance reliability [2].

The coded beam training in Definition IV.2 results in the following types of beam training.

4.2.3. Definition IV.3 (Hierarchical codebooks)

Hierarchical codebooks typically aid in the process of beam training. The codebooks arrange beamforming vectors in a tree structure to make searching easier and reduce training time [6]. Under challenging conditions, particularly with a low signal-to-noise ratio (SNR), accurate beam training is highly crucial, for which we require robust channel coding techniques.

4.2.4. Definition IV.4 (Binary hierarchical beam training)

Binary hierarchical beam training is a layered beam search method that iteratively narrows down the beam direction by dividing the angular space into $M = \log_2 N_T$ layers, transmitting two candidate beams per layer. It requires one feedback per layer, resulting in a total training time of $2 \log_2 N_T$ slots. This approach significantly reduces training overhead compared to exhaustive sweeping (defined next in Definition IV.5) but involves higher implementation complexity due to its hierarchical structure.

4.2.5. Definition IV.5 (Exhaustive beam training)

Exhaustive beam training is a beam training technique that sequentially scans all N_T possible transmit beam directions to identify the optimal beamforming path. It requires a single feedback message from the user equipment (UE) after all beam directions are tested. While it achieves high beam selection accuracy, it incurs a high training overhead with a total time cost of $N_T + 1$ slots.

5. System model

This section describes the system architecture, including beam training protocols, coding schemes, and antenna configurations.

5.1 Semi-hierarchical beam training structure

To leverage the spatial sparsity of the channel while maintaining robust performance under uncertainty, we propose a semi-hierarchical coded beam training approach for 6G wireless systems. This architecture integrates hierarchical search with error-resilient channel coding over field extensions, thereby enhancing both training efficiency and resilience in low-SNR or highly stable channel conditions.

5.1.1. Definition V.1 (Semi-hierarchical beam training)

Semi-hierarchical beam training is a layered method that partitions the angular domain into multiple hierarchical levels. Initially, the space is segmented into L_1 coarse sectors (primary beams), each of which is further subdivided into $L2$ finer beams (sub-beams), resulting in a total of $N\mathrm{T} = L_1 \cdot L_2$ angular directions − matching the number of antennas (cf. Section 4.1). At each level, directional pilot signals are transmitted using error correction codes to improve sector identification accuracy, balancing rapid exploration with high angular resolution.

5.1.2. Definition V.2 (22−ary hierarchical beam training)

2^2-ary hierarchical beam training is a special type of semi-hierarchical beam training in which the angular domain is divided into four candidate beams per layer. Assuming N_T is a power of 2, the total number of layers is $M = 2 \log_2 N_T$. At each layer, four beams are probed sequentially, and a single feedback message (cf. Remark 1) is used to select the correct subregion for refinement in the next stage. This process continues recursively until the final beam is identified.

Remark 1. The UE observes all 4 candidate beams per layer. It returns 1 feedback, typically 2 bits of information (e.g., 00, 01, 10, or 11) to indicate which of the 4 beams is best.

Compared to binary hierarchical beam training (see Definition IV.4), 2^2 -ary hierarchical beam training halves the number of layers and overall training latency.

5.2 6G communications channel model

While the main contribution of this work lies in coding design for 6G systems, we briefly summarise the underlying channel model for completeness. A detailed formulation is provided in [2, Section 2], where the model follows the well-known Saleh Valenzuela representation widely adopted in mmWave and THz communications [9]. This model captures the sparse and directional characteristics of high-frequency propagation, offering a physically grounded basis for robust coding and beam training design.

We consider a downlink XL-MIMO system operating at mmWave/THz frequencies. The base station (BS) is equipped with a uniform linear array (ULA) of NT antennas spaced at half the carrier wavelength ($\lambda/2$), each connected to a dedicated RF chain, enabling fully-digital precoding. Although full-digital architecture is assumed for analytical clarity, the proposed scheme generalizes to hybrid precoding as discussed in [2].

Let $s_0 \in$ C denote the power-normalized transmit symbol. The received signal y at the user equipment (UE) is:

$$y = \sqrt{P}\, hws_0 + n \tag{1}$$

where $P > 0$ is the transmit power, $h \in \mathbb{C}^{1\times N_T}$ is the downlink channel vector, $w \in \mathbb{C}^{N_T\times 1}$ is the unit-norm beamforming vector, and $n \sim$ CN (0, $\sigma 2$) denotes complex Gaussian noise.

Due to the dominance of the line-of-sight (LoS) component (denoted by a subscript 0 in the terms of (1)) at high frequencies, we consider a simplified single-path channel model, which also determines the angular direction from the BS to the UE [11]:

$$h = \sqrt{N_T}\, \beta_0\, \alpha\,(\varphi_0) \tag{1}$$

where $\varphi 0 = \sin(\theta 0) \in [-1,\ 1]$ is the spatial direction from the BS to the UE with $\theta 0 \in [-\pi/2, \pi/2]$ representing the physical angle of departure (refer to Fig. 1), and $\alpha(\varphi 0)$ is the array steering vector defined as:

$$\alpha(\varphi_0) = \frac{1}{N_T}\left[1, e^{-j\pi\varphi_0}, \dots, e^{-j(N_T)\pi\varphi_0}\right] \tag{2}$$

The path gain $\beta 0$ follows the free-space path loss model:

$$\beta_0 = \frac{\lambda_0}{4\pi r} \tag{3}$$

where λ_0 is the carrier wavelength and r is the distance between the UE and the center of the antenna array. When the received signal y (stated in (1)) is corrupted by noise, an uncoded training scheme may misclassify the beam direction, leading to degraded communication. In contrast, our proposed 2^2-ary coded beam training architecture mitigates such errors by encoding k received pilot symbols into a codeword of length n, adding redundancy in the form of additional protection symbols, e.g., an $n-\ k$ consecutive two-bit symbols (refer to Definition V.2) to protect against errors or erasures.

5.3 Problem description

5.3.1. Channel model

Our channel model in (1) is seen as an ideal channel; however, it does not compromise the training process. The inherent estimation inaccuracies are implicitly embedded in the channel vector $\boldsymbol{h}$, and their impact can be reasonably mitigated as discussed next. The primary goal of beam training is to steer the beamformer $\boldsymbol{w}$ toward the angle of departure (AoD) of the dominant, typically line-of-sight (LoS), propagation path. Given the array steering vector structure in (2), we define the discrete Fourier transform (DFT)-based codebook, W, also referred to as the exhaustive codebook (see Definition IV.5), as

$$W = \left\{\alpha^H(\varphi) | \varphi = -1 + \frac{2n-1}{N_T}, n \in \{1,2,\cdots,N_T\}\right\}$$

The codebook-based beam training then aims to select the optimal codeword from W that maximizes the received signal power, i.e.,

$$\max_{w} |\boldsymbol{hw}| \quad s.t. \quad w \in W \tag{4}$$

A straightforward solution to problem (4) is exhaustive beam sweeping [12]. In this approach, the base station (BS) sequentially transmits using all codewords from W , while the user equipment (UE) identifies the codeword yielding the highest received power and feeds back its index to the BS. Although this method achieves near-optimal beam alignment performance, it requires N_T training time slots—equal to the number of BS antennas—which results in prohibitive training overhead, particularly in extremely large-scale MIMO (XL-MIMO) systems.

To mitigate this excessive overhead, hierarchical beam training (see Definition IV.4) has been widely adopted. This method employs a binary-search-based hierarchical codebook that enables coarse-to-fine beam alignment. A typical hierarchical codebook, denoted as $C_{hier}^{L_i}$, consists of 2^{L_i} codewords at the L_i-th layer, where $i \in \{1,2,\dots,log_2 N_T\}$. The authors in [2] denote the b-th codeword in the L_i-th layer as $C_{L_i,b}^{hier}$. Each codeword at layer L_i covers two narrower, higher-resolution beams at the subsequent layer L_{i+1}, enabling a progressively refined search for the dominant path. In this way, hierarchical beam training significantly reduces the required number of training slots while maintaining satisfactory beam alignment performance.

Unlike the conventional binary hierarchical codebook described above and in [2], our proposed 2^2-ary hierarchical codebook (see Definition V.2) further improves training efficiency. Specifically, the proposed codebook $C_{hier}^{L_i}$ reduces the total number of layer L_i by half, such that each L_i-th layer contains 2^{2L_i} ccodewords, for $i \in \{1,2,\dots,log_{2^2} N_T\}$. In other words, each layer L_i includes four codewords, denoted as $C_{L_i,b}^{hier}$, where $b \in \{1,2,3,4\}$.

To illustrate, consider four candidate signals, each represented by two bits. The first two signals correspond to the group with the most significant bit (MSB) equal to 0, while the remaining two belong to the group with MSB equal to 1. During the training procedure, the received powers of these two groups are first compared, effectively testing the MSB. The group yielding the higher received power is retained. Subsequently, within the selected group, the signals are compared according to their least significant bit (LSB) to determine the best candidate beam.

This 2^2 -ary hierarchical selection strategy enables efficient identification of the optimal beam while significantly reducing the search depth from $log_2 N_T$ (in binary search) to $log_{2^2} N_T$, thereby achieving faster convergence and lower training overhead without sacrificing accuracy.

Fig. 1 illustrates the 2^2-ary hierarchical beam training approach in which the BS divides the spatial region into multiple portions to enhance precision and reduce training overhead. The first spatial area of the beam training process is divided into four sub-regions and they are identified by four symbols: $0, \alpha, 1$, and $\alpha + 1$. These symbols represent distinct initial training directions (mapping to angles) based on an algebraic coding approach. The training algorithm selects the appropriate two-bits (symbol per layer) corresponding to the feedback from the user, guiding the subsequent layers in refining the search space. Once the initial section is determined, the beam training process proceeds by dividing the selected layer into smaller segments of the next four symbols: $0, \alpha, 1$, and $\alpha + 1$. This hierarchical refinement continues until the most accurate beam direction is identified and until a full codeword is defined in which an n symbols are selected. The angular division, as shown in Fig.1, progressively narrows down the search space, improving the alignment accuracy while minimizing the training time.

5.3.2. Error propagation

Hierarchical beam training is limited by the well-known issue of error propagation, which prevents reliable training in low-SNR scenarios, particularly for remote users, thereby restricting the coverage area. This drawback arises because codewords at the upper layers have wider beamwidths and lower beamforming gains, rendering them more susceptible to noise. Moreover, since the hierarchical search proceeds sequentially along a binary (or 2^2-ary in our scheme) tree, an incorrect decision at any intermediate layer inevitably results in unrecoverable training failure. To overcome this limitation, we propose a novel 2^2 beam training approach that leverages the erasure-error-correcting capability of channel coding. The proposed method not only mitigates error propagation but also reduces the training overhead while preserving high success rates in low-SNR environments.

While our primary channel model (Section 5.2) assumes a noiseless channel for analytical clarity, the robust correction capability of our code, $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors and $\tau = d - 1$ erasures, inherently provides resilience against practical system non-idealities. Random errors (t) are not exclusively caused by thermal noise; they also model the effects of:

- Quantization errors: Imperfect analog-to-digital conversion at the UE.

- Hardware impairments: Non-linearities in the RF chain.
- Imperfect synchronization: Small timing or frequency offsets.

By tolerating up to t random errors in addition to the deliberately induced erasure, our scheme maintains high beam selection accuracy under these practical, non-ideal conditions, confirming the superiority over uncoded methods.

6. Error-erasure correcting code

We propose a generalized construction of error-erasure correction codes by extending a base linear code $C_{n-1} \subseteq \mathbb{F}_q^{n-1}$ to a longer code $C_n \subseteq \mathbb{F}_q^{n}$ (cf. Section 4-A). The extended code can correct both random errors and erasures using a combination of syndrome decoding and linear systems of equations. Our scheme provides an uncomplicated but effective method to extend classical error-correcting codes to handle mixed errors and erasures with minimal modification to the encoder.

6.1 Construction 1

Let $C_{n-1} \subseteq \mathbb{F}_{2^2}^{n-1}$ be a linear code with parameters $[n-1, k, d]$ capable of correcting up to t=$\left\lfloor\frac{d-1}{2}\right\rfloor$ random errors. Let $G_{n-1} \in F_{2^2}^{k\times(n-1)}$ be its generator matrix. We construct an extended code $C_n \subseteq \mathbb{F}_{2^2}^{n}$ with a one symbol larger generator matrix:

$$\boldsymbol{G_n} = [\boldsymbol{p} \mid \boldsymbol{G_{n-1}}] \in \mathbb{F}_{2^2}^{k\times n},$$
$$\boldsymbol{G_n} = [\boldsymbol{p} \mid \boldsymbol{I_k} \mid \boldsymbol{P_{k\times(n-k-1)}}] \in \mathbb{F}_{2^2}^{k\times n},$$

where $p \in \mathbb{F}_{2^2}^{k-1}$ is a parity column vector, $\boldsymbol{P_{k\times(n-k-1)}}$ is the parity sub-matrix of the matrix G_{n-1}, and Ik is the identity matrix of size k. Encoder and decoder are shown in Algorithm 1 and Algorithm 2.

6.2 Theorem 1

The coding scheme in Construction 1 can obtain a hybrid (t, τ) error-erasure correction scheme, where:

- up to $t = \left\lfloor\frac{d-1}{2}\right\rfloor$ random errors in the first $n-1$ symbols can be corrected using an $[n-1, k, d]_{2^2}$ code C_{n-1},
- and up to $\tau = d-1$ known erasures anywhere in an n length codeword are recovered using $[n, k, d]_{2^2}$ code C_n. Proof.

We prove the hybrid correction capability by separating the encoding and decoding procedures.

6.2.1. Encoding

Given a message $m \in \mathbb{F}_{2^2}^{1\times k}$ (cf. Algorithm 1), the encoder first generates a codeword $c_{n-1} = m \cdot G_{n-1}$ using the base generator matrix G_{n-1}. It then computes a parity symbol $c_n = m \cdot p$ using a parity vector $p \in \mathbb{F}_{2^2}^{k\times 1}$, chosen to extend the code while preserving or increasing (by one) its minimum distance, typically constructed such that the resulting parity-check matrix enforces even-weight rows and linear independence. The final codeword is $c_n = (c_n, c_{n-1}) \in \mathbb{F}_{2^2}^{n}$, as described in Algorithm 1.

6.2.2. Decoding

We prove the decoding capability in two phases.

Step 1: Error Correction in C_{n-1}

The base code C_{n-1} has minimum distance d, which allows correction of any t= errors. Thus, any errors that occur within $n-1$ symbols (ignoring the first symbol) can be decoded by the decoder of C_{n-1} via, e.g., syndrome decoding.

Step 2: Erasure Correction in C_n

After correcting up to t errors, suppose $|\mathcal{E}| = \tau$ erasures exist in known positions given by the set $\mathcal{E}$ (including an intentionally erased first symbol where error propagation begins). Since the code C_n has minimum distance $\geq d$, it can recover at most $\tau \leq d$ erasures by solving a linear system derived from the parity-check equations. These known-position erasures can be uniquely recovered[1] as long as the number of erasures does not exceed d, due to the rank and distance properties of the code.

1. Known-position erasures can be uniquely recovered because a linear $[n, k, d]$ code can correct up to $d-1$ erasures in known positions, i.e., the submatrix of the parity-check matrix formed by columns corresponding to the erased positions has full rank, and the remaining coordinates provide sufficient constraints to solve for the missing values. This follows directly from the minimum distance d, which ensures that any $d-1$ columns of the parity-check matrix are linearly independent.

Received: August 5, 2025. Revised: November 4, 2025.

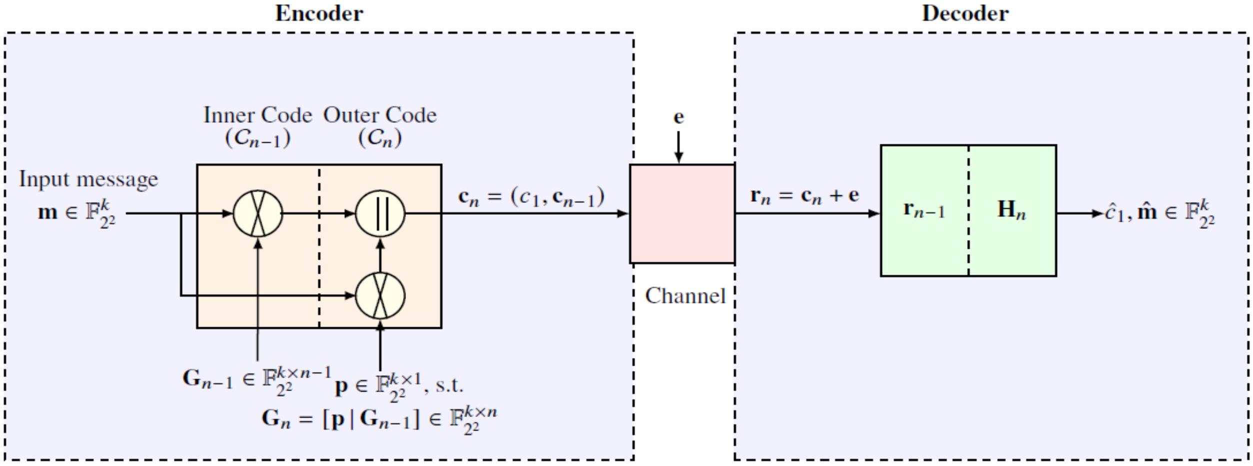

Figure. 2 An illustration of the whole coding and decoding procedure. The diagram shows an input message m which is first encoded to a codeword c_{n-1}, and then extended by c_1 to a codeword c_n. The codeword that is transmitted is modified by the channel to produce the received signal $r_n = c_n + e$, where e is the non-zero channel- added error vector of Hamming weight at most d. The word received is then processed by the error and erasure correction blocks, where it locates and repairs both corrupted and erased symbols to reconstruct the estimated codeword ĉ from which the message $\hat{\boldsymbol{m}}$ and the first erased symbol $\hat{c}_1$ can be recovered.

The decoder solves the linear system given by the parity-check equations of Cn, treating the positions of the erasures as unknowns, while using the known positions and values of the correct (or the "corrected" as stated in Step 1) symbols as constraints. The system is a full-rank due to the minimum distance property, ensuring a unique solution and preventing error propagation by finding the deliberately erased 1st symbol.

Remark 2. The decoder first corrects up to t random errors in the first $n-1$ symbols using C_{n-1}, and then reconstructs up to $\tau \leq d$ remaining known positions erasures in an n length codeword by solving a linear system using the parity-check matrix of C_n. These operations are sequentially conducted, as illustrated in Fig. 2, and composable since error and erasure positions are assumed disjoint (or error positions are first corrected).

We can further elaborate on the two steps (error and erasure correction) of encoding and decoding in Algorithm 1 and Algorithm 2 throughout Section 6-A and Appendix C.

The end-to-end signalling operation of our two proposed 2^2-ary coded beam training schemes are illustrated in Fig. 3 (hierarchal plus adaptive) and Fig. 4 (hierarchal plus fixed), and the precise decoding procedure is captured in Algorithm 2. The core integration step is the Base Station's (BS) application of the decoder (Algorithm 2, steps 6-9) upon receiving the single coded feedback symbol. The decoder's ability to first correct random channel errors in the remaining n-1 symbols and then solve for the intentionally erased first symbol (the MSB/coarse direction) is the key to decoupling the sequential dependence of the hierarchical search and preventing error propagation.

Algorithm 1: Encoding Algorithm (User Side)

Require: Message $\mathbf{m} \in \mathbb{F}_{2^2}^{1\times k}$, generator matrix G_{n-1}, parity vector $\mathbf{p}$
Ensure: Codeword $\mathbf{c}_n \in \mathbb{F}_{2^2}^{n}$

1: $\mathbf{c}_{n-1} \leftarrow \mathbf{m}\mathbf{G}_{n-1}$
2: $\mathbf{c}_n \leftarrow \mathbf{m} \cdot \mathbf{p}$
3: $\mathbf{c}_n \leftarrow (\mathbf{c}_n, \mathbf{c}_{n-1})$
4. **return** $\mathbf{c_n}$

Algorithm 2: Decoding Algorithm (BS Side)

Require: Received corrupted vector $\mathbf{r} = (\mathbf{c_n} + \mathbf{e}) \in \mathbb{F}_{2^2}^{n}$, code C_{n-1}, parity-check matrix $\mathbf{H_n}$, erasure positions $\mathcal{E}$
Ensure: Decoded message $\hat{\boldsymbol{m}} \in \mathbb{F}_{2^2}^{1\times k}$

1: Extract $r_{n-1} \leftarrow (r_1, \ldots, r_{n-1})$
2: Use syndrome decoding on r_{n-1} to correct up to $t$errors and obtain mˆ from the first k positions due to systematic encoding
3: $\hat{c}_{\boldsymbol{n-1}} \leftarrow \hat{\boldsymbol{m}} \cdot \boldsymbol{p}$
4: $\hat{c}_{\boldsymbol{n}} \leftarrow (\hat{c}_{\mathbf{1}}, \hat{\boldsymbol{m}}\, \boldsymbol{G}_{\boldsymbol{n-1}})$
5: **if** $|\mathcal{E}| = \tau \leq d - 1$ **then**
6: **Erase** $1st$ position
7: **Solve** $\hat{c}_{\boldsymbol{n}}\, \boldsymbol{H}_n^T = 0$ for unknowns $\hat{c}_{\mathbf{1}}$ and other erasures
8: **Return** Reconstructed codeword ĉ n and $\hat{c}_1$ tells the BS the first angle (direction)

 DOI: 10.22266/ijies2026.0131.07

Received: August 5, 2025. Revised: November 4, 2025.

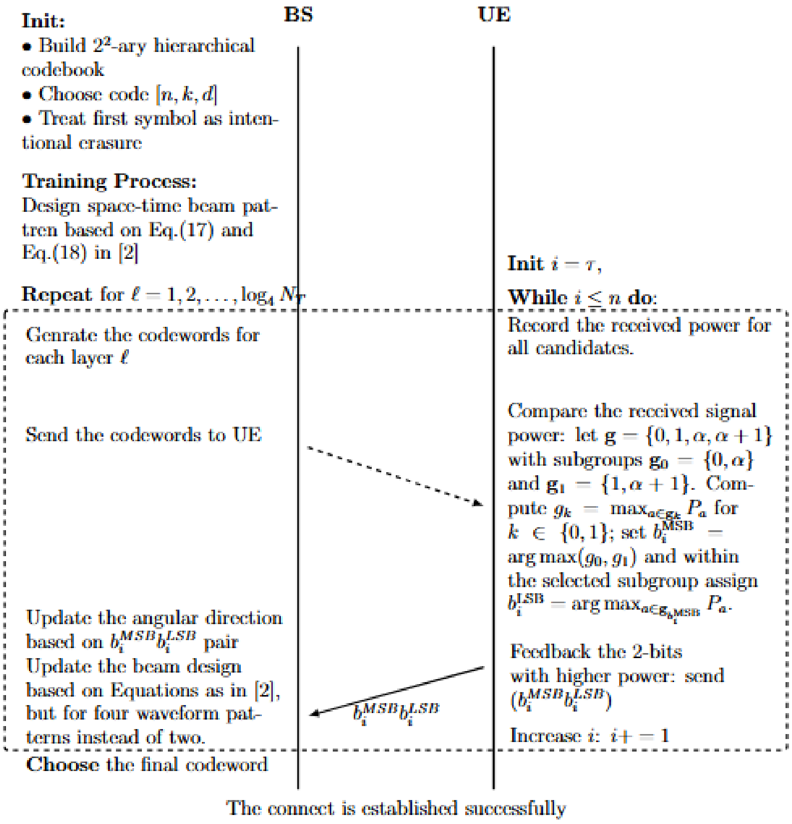

Figure. 3 Flow diagram of the Hierarchical Plus Adaptive Coded Beam Training pipeline for beam probing and selection

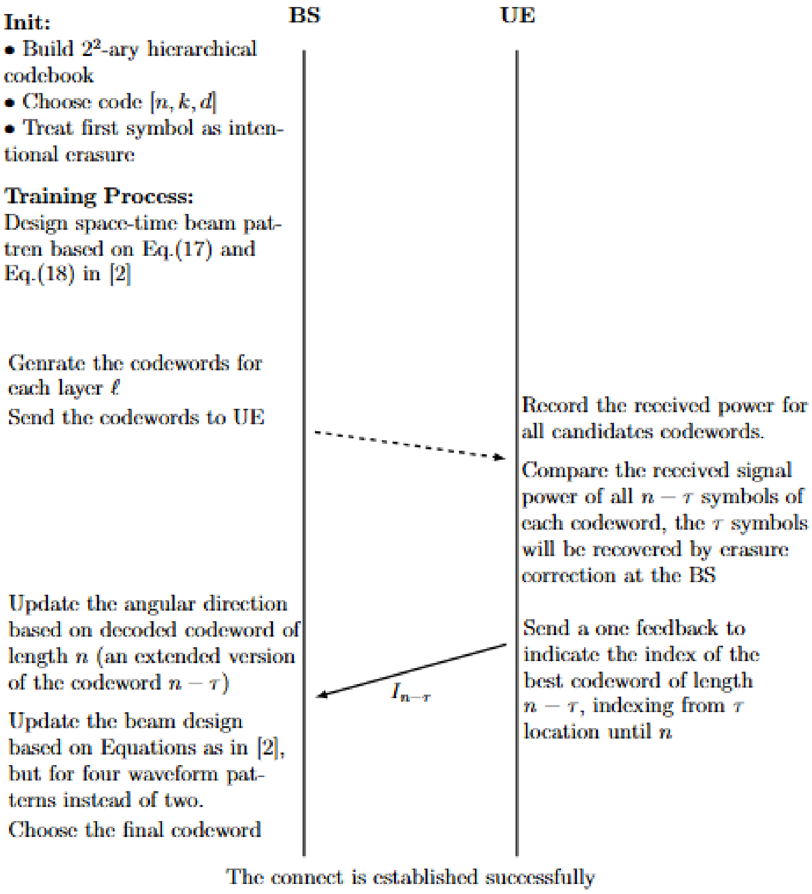

Figure. 4 Flow diagram of the Hierarchical Plus Fixed Coded Beam Training pipeline for beam probing and selection

6.2 Example of coded beam training using extended hamming codes

The example in Appendix C directly illustrates how Construction 1 and Theorem 1 enable error–erasure protection in practical beam training scenarios. To enhance feedback reliability for 6G beam training under low-SNR, we employ an extended Hamming code with parameters $[8,4,4]_{2^2}$. Theorem 1 implies correcting any combination of $t = \lfloor \frac{d-1}{2} \rfloor$ errors and $\tau = d - 1$ erasures. Thus, for the Hamming $[7,4,3]_{2^2}$ code (see Appendix C), $t = 1$. Furthermore, since the final codeword is of length n belongs the extended code $[8,4,4]_{2^2}$. , $\tau = 3$ erasures (one intentionally considered at the first received symbol) can be treated. This allows for a robust recovery of the most significant bit (MSB) in hierarchical feedback (e.g, the $1st$ bit in the 1st received symbol). Therefore, our coding scheme over $\mathbb{F}_{2^2}$ effectively limits error propagation throughout the beam refinement stages at the expense of using only one feedback as explained in Remark 3.

Remark 3. We cut the search space in half thanks to the quaternary field. Unlike the method in [7], which requires two feedback symbols—where the first carries the most significant bit to split the codebook into an "upper" region $[1, 0]$ and a "lower" region $[0, -1]$, we avoid that requirement. Instead, we deliberately erase the first symbol, relying on erasure correction to recover it, thus removing the need for that initial feedback. As a result, only one actual feedback symbol, I_{n-1}, is needed to refine the direction within the selected beam space region. The error-correcting capability of the extended code guarantees that the remaining $n - 1$ symbols are decoded correctly, even under low SNR conditions.

The decision to intentionally erase the first symbol and recover it via erasure correction is a strategic trade-off. This single erasure symbol (or τ symbols) captures the crucial, coarse-grained angular information. For scenarios with high mobility and short channel coherence, reducing the feedback overhead from $log_2 N_T$ (for binary hierarchical) to a single symbol (I_{n-1}) dramatically reduces the overall time the UE must remain silent awaiting a feedback opportunity. The robust τ -erasure recovery mechanism ensures that the critical, broad beam direction is successfully recovered, making the scheme resilient even when the remaining $n - 1$ symbols are highly corrupted due to channel variation over the training period.

7. Review of beam training techniques and comparison with the proposed semi-hierarchical coded approach

This section provides a concise review of the most prominent existing beam training techniques, highlighting their advantages and limitations, and their comparison with our proposed semi-hierarchical (2^2-ary hierarchical) coded strategy.

7.1 Techniques Overview

7.1.1. Uncoded hierarchical beam training

Training is divided into M layers, with two signals transmitted per layer. The training overhead is $\log_2 N_T$, requiring M feedbacks. The total training time is $2 \cdot \log_2 N_T$. This approach has a lower training cost compared to exhaustive search, but with increased complexity.

7.1.2. Exhaustive beam sweeping

This approach explores all possible beam directions (N_T), ensuring optimal beam selection. It requires a training overhead of N_T and only one feedback. The total training time is $N_T + 1$. While it offers high accuracy, it comes at a high cost, i.e., the highest total training time of $N_T + 1$.

7.1.3. Fixed coded beam training

This method is similar to hierarchical training, with a training overhead of $2 \cdot \log_2 N_T$. It requires two feedback messages, resulting in a total training time of $2 \cdot \log_2 N_T + 2$. However, it does not support dynamic adjustment during training.

7.1.4. Adaptive coded beam training

This technique adjusts the beam direction at each layer based on feedback. The training overhead is $2 \cdot \log_2 N_T$, and the feedback overhead is $\log_2 N_T$, resulting in a total training time of $3 \cdot \log_2 N_T$. It introduces higher complexity in codebook design [2].

7.2 Our proposed techniques

7.2.1. 22−ary hierarchical with fixed coded beam training

It is a hybrid strategy that combines the hierarchical and fixed training methods. It uses one fixed feedback from the user for the $n - 1$ symbols, the total training will be mixed between hierarchical and fixed training methods. Thus, in total it needs

2 · $\log_2 N_T$ + 1 training time. Note that for N_T = 16 as stated in Corollary 2,

$$2 \cdot \log_2 N_T + 1 = 2.5 \cdot \log_2 N_T - 1$$

where $\log_2 N_T - 1$ is our 22−ary hierarchical and adaptive training time as stated next (see Item 2), and for NT> 1

$$2 \cdot \log_2 N_T + 1 < 2.5 \cdot \log_2 N_T - 1,$$

implying that the mixed hierarchical-fixed scheme achieves lower training time as compared in Table 1.

7.2.2. 22−ary hierarchical and adaptive coded beam training

In another consideration, we use adaptive and hierarchical. It sends four signals per layer. Thus, it needs in total (2.5 · $\log_2 N_T$) training time as follows. Training overhead: 2 · $\log_2 N_T$ and feedback overhead: 0.5 · $\log_2 N_T$.

Moreover, as we do not need one feedback to adjust the first layer decision, as we will ultimately recover it as an erasure (see Remark 3), we subtracts 1 from the term 2.5 · $\log_2 N_T$. For all $N_T \leq 16$,

$$2.5 \cdot \log_2 N_T - 1 \leq 2 \cdot \log_2 N_T + 1,$$

holds (check Corollary 1). Thus, the inequality is satisfied for $N_T \leq 16$, and is lower overall training time. However, for $N_T > 16$, 2 · $\log_2 N_T$ + 1 is lower meaning fixed and hierarchical training method is the best case.

In our work, we first set the semi-hierarchical coded beam training scheme as a baseline method. However, upon closer examination and investigation, we found that the hierarchical coded beam training method provides superior performance in both adaptive and fixed training scenarios. To better show the performance, we divided hierarchical scheme into two cases adaptive and fixed and added the original semi-hierarchical scheme for comparison. This helps to understand how beam training methods have advanced and improved in this research. Therefore, the table below compares the three techniques to indicate their difference in training costs and reliability.

The issue of scalability for large N_T is critical. As detailed in Appendix D, our scheme provides an advantage in computational complexity over Viterbi-based approaches. While our general scheme works

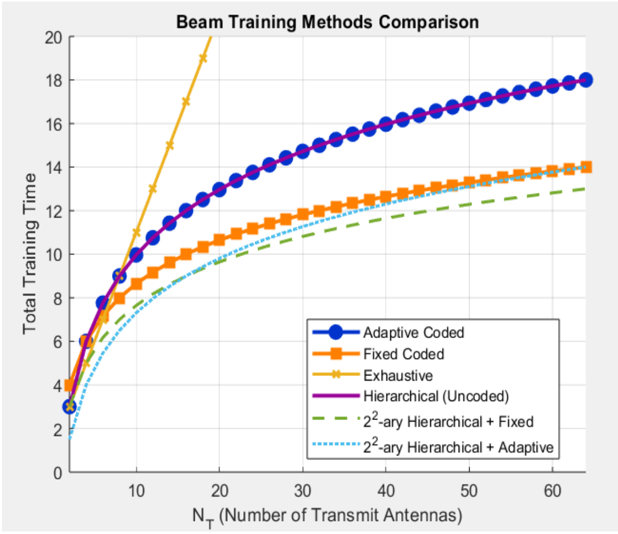

Figure. 5 Comparison of Beam Training Methods

for any linear block code family (e.g., Extended Hamming, BCH, etc.) and length, its practical implementation focuses on codes with short block lengths($n \ll N_T$)to keep the decoding complexity low.The decoding complexity of our block ECC scheme scales polynomially,

$$O(n^2)$$

where n is the codeword length (e.g., $n = 8$ in the Hamming code example). This is a significant advantage over the exponential complexity of Viterbi decoding in the baseline method,

$$O\left(L2^k(N - 1)\right)$$

Furthermore, the total complexity is reduced by the τ layers that are recovered via erasure correction, decreasing the number of active training layers from $(M-1)$ to $(M-1-\tau)$.

For a large system with (N_T= 1024) and (τ = 1),this elimination of (τ =1) layer of feedback and processing reduces latency in both training time and decoding time, ensuring the scheme remains feasible for massive MIMO deployments.

7.3. Comparison Table

We provide Table 1 which compares our different proposed schemes with beam training schemes. Note that the term $\log_2 N_T$ refers to the base-2 logarithm of N_T, which relates to the complexity of the training time.

Received: August 5, 2025. Revised: November 4, 2025.

Table 1. Comparison of Overheads for Different Schemes

Schemes	Type	Training Overheads (BS-UE)	Feedback Overheads (UE-BS)	Total Training Time	Complexity
Adaptive coded beam training	Coded	$2\log_2 N_T$	$\text{Log}_2 N_T$	$3\log_2 N_{TY}$	High
Fixed coded beam training	Coded	$2\log_2 N_T$	2	$2\log_2 N_T+2$	Low
Exhaustive beam sweeping	Uncoded	N_T	1	N_T+1	Very low
Binary search-based hierarchical beam training	Both	$2\log_2 N_T$	$\text{Log}_2 N_T$	$3\log_2 N_T$	High
Our proposed Hierarchical + Fixed (hybrid)	Coded	$2\log_2 N_T$	1	$2\log_2 N_T+1$	Low
Our proposed Hierarchical + Adaptive (hybrid)	Coded	$2\log_2 N_T$	$0.5\log_2 N_T$	$2.5\log_2 N_T-1$	Medium

7.3.1. Compared to adaptive coded beam training and Binary search-based hierarchical beam training

It is obvious from Table 1 that our scheme has a lower total training time, exactly $0.5\log_2 N_T$ less than the adaptive coded beam training and the binary search-based hierarchical beam training.

7.3.2. Compared to fixed coded beam training

Our method is at most the total training time of the fixed coded beam training as shown in Table 1, given $N_T \leq 16$ as follows.

Corollary 1 (Total training time in mixed hierarchical– fixed beam training). Let N_T be a positive real number such that $N_T \leq 16$, then the following inequality holds:

$$\frac{5}{2}\log_2(N_T) \leq 2 \cdot \log_2(N_T) + 2 \tag{4}$$

The proof of Corollary 1 has been moved to the Appendix A.

Thus, for $N_T < 16$, we achieve lower total training time than fixed coded beam training, and consequently better performance.

7.3.3. Compared to exhaustive beam sweeping

The exhaustive beam sweeping requires N_T +1 total training time, which is the highest among all schemes including ours.

7.3.4. Compared to hierarchical fixed coded beam training

The hybrid hierarchical-fixed training method, which has a total training time of $2 \cdot \log_2 N_T + 1$, outperforms hierarchical-adaptive training when the number of antennas exceeds 16, as shown in Corollary 2. This means that this method achieves shorter training time in systems with a large number of antennas, making it best suited for reducing training time in large communications environments.

Corollary 2 (Comparison between fixed and adaptive hierarchical coded beam training in total training time). Let N_T be a positive real number such that $N_T > 16$, then the following inequality holds:

$$2 \cdot \log_1(N_T) + 1 < \frac{5}{2}\log_2(N_T) - 1 \tag{5}$$

The proof of Corollary 2 has been moved to the Appendix B.

Thus, for $N_T > 16$, we achieve lower total training time using the mixed hierarchical-fixed beam training compared to the adaptive-hierarchical approach.

Our 2^2-ary hierarchical coded beam training scheme offers two optimized modes, making it suitable for distinct 6G deployment scenarios:

Large-Scale, Stationary/Low-Mobility Deployments ($\boldsymbol{N_T > 16}$): The Hierarchical + Fixed mode (Total Training Time2 $\log_2 N_T$+1) is superior (Corollary 2). This mode, with its lower complexity and shorter training time for large antenna arrays, is ideal for fixed wireless access or indoor massive MIMO where channel conditions are relatively stable, and large N_T is used for high capacity.

Small/Medium-Scale, High-Mobility Deployments ($\boldsymbol{N_T \leq 16}$): The Hierarchical + Adaptive mode (Total Training Time: 2.5 $\log_2 N_T$-1) is preferred (Corollary 1). This mode's slightly faster convergence for smaller N_T makes it perfectly suited for high-mobility urban or vehicular environments where low-latency beam tracking and re-acquisition is mandatory, and the channel coherence time is short.

7.4 Performance analysis

To validate the efficacy of the proposed scheme beyond the training time analysis, we conducted Monte Carlo simulations of the beam training process. We compare the Beam Training Success Probability (the probability of selecting the optimal beam) versus the Signal-to-Noise Ratio (SNR) for $N_T = 64$ antennas. The simulations incorporate realistic noise and interference, aligning with the channel model in Section 5.2. As shown in Fig. 6, our proposed 2^2-ary Hierarchical + Fixed/Adaptive schemes significantly outperform the Uncoded Hierarchical method, especially in the low-SNR regime (SNR < 0 dB). This gain is a direct consequence of the dual error-erase correction capability (Theorem 1), which isolates and corrects errors that would otherwise lead to catastrophic failure due to error propagation in the conventional uncoded hierarchy.

The performance gain confirms that the latency reduction is not achieved at the expense of selection accuracy in challenging channel conditions.

8. Complexity analysis

This section analysis complexity in terms of training and feedback overheads, implementation effort, and computational cost. We present a clear comparison among exhaustive beam training, binary search-based hierarchical training, adaptive/fixed coded training [2], and our hybrid hierarchical plus (fixed or adaptive) schemes, summarised in Table 1.

8.1 Training overhead

In our scheme, across the upper $M-2$ layers of the hierarchical beam codebook, the BS transmits one codeword per layer (one time slot each). At the bottom layer, the BS sends two codewords to finalize the selection, consuming two time slots. Hence, the total downlink training overhead is $M = 2log_2 N_T$. We assume $R = 0.5$ by default, but any $R = k/n$ is admissible as long as the code that used give us enough the minimum distance redandancy. It's possible to generalize to many/most coding scheme and families that satisfied the condition which is give us enough redundancy to have enough minimum distance to correct t errors and tau erasure. Note that, partially use our algorithm if we assume for simplicity there's no errors $t = 0$. Thus, at less one erasure it's possible to aviod the error propagation. The training overheads at $N_T = 1024$ are $20, 1024, 20$, and 20 time slots for our proposed scheme, exhaustive sweeping, traditional binary hierarchical training, and adaptive/fixed coded

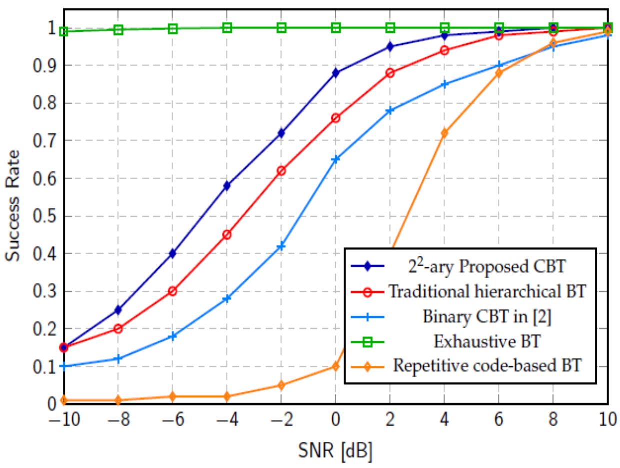

Figure. 6 Comparison of success rate for different beam training methods (N_T = 64)

beam training, respectively. Thus, our method matches the binary hierarchical and adaptive/fixed coded beam training overhead while reducing the cost relative to exhaustive search by $\frac{1024-20}{1024} \times 100\% \approx 98.05\%$.

8.2 Feedback overheads

The feedback overhead from the UEs to the BS is also compared. In our hybrid strategy that combines the hierarchical and fixed training methods, the BS requires one fixed feedback from the user for the $n-\tau$ symbols. The τ-layers is intentionally erased and later recovered by the channel code, so no feedback is required for these erasure symbol. Thus, the feedback payload remains fixed and independent of the number of layers. Also, for our hierarchal and adaptive coded beam training scheme, as in hierarchal fixed, our method skips the first feedback by treating the first-layer MSB τ-layers as a known erasure location and recovering it via the channel code the erasure correction code. Consequently, feedback is gathered only every two layers. The single layer is divided into four sections, unlike binary search-based hierarchical beam training, this reduces feedback by half $0.5\, log_2 N_T - \tau$. This outperforms all other types of beam training. In this work, we focused on $\tau = 1$, which is the least value we can consider so that we show the comparison in Table 1.

8.3 Implementation complexity

The codebook generation is considered first. In our hierarchical adaptive coded beam training, the beam patterns in the lower layers are determined both by the encoded algorithm and feedback of the upper

Table 2. Comparison of Computational Complexity

Aspect	**Viterbi-Based**	**Proposed**
Code Type	Convolutional	Block ECC
Decoder	Viterbi	Syndrome/Direct
Decoding Complexity	$O(L2^k(N - 1))$	$O(n^2)$
Codebook Generation	$O(I_{max}N_T \, log \, N_T)$	Same
Layers	$(M - 1)$	$(M - 1 - \tau)$
Total	$O((M-1)\,(I_{max}N_T \log N_T + 2^k\,(N-1)))$	$O((M-1-\tau)\,(I_{max}N_T \log N_T + n^2))$

layers, except for τ layers that can be determined mathematically through the erasure correction capability of the code. Therefore, the codebook is supposed to be generated adaptively only for $M - 1 - \tau$ with less complexity since the τ layers do not introduce extra burden on defining the codebook. In contrast, the codebooks of adaptive coded beam training stated in [2] that generated adaptively for all $M - 1$ layers. In contrast, for the other methods: hierarchical fixed, non-adaptive coded beam training and exhaustive sweeping beam training use predetermined search paths and thus can be executed with fully pre-generated codebooks, avoiding real-time UE-to-BS feedback and eliminating complex signalling control. On the other hand, our hierarchical fixed scheme enables fully pre-generated codebook generation and uses single fixed feedback; hence, its implementation complexity is very low, comparable to non-adaptive coded beam training in [2] and close to exhaustive sweeping training. Our hierarchical adaptive coded beam training scheme maintains lower implementation complexity than traditional hierarchical adaptive coded beam training and binary search–based hierarchical training. We can observe that throughout Table 1.

8.4 Computational complexity analysis

We compare the computational complexity of the proposed coded beam training with the conventional Viterbi-based approach [2]. In both schemes, the main complexity contributors are codebook generation and beam decoding.

In the baseline [2], convolutional codes are employed and the optimal beam index is recovered using a Viterbi decoder. For a code of parameters (n, k, N) and an information sequence of length L, the decoding complexity scales as

$$O\left(L2^k(N-1)\right) \tag{6}$$

which grows linearly with L but exponentially with k. Each layer also requires generating candidate beams via a GS-based design method with complexity $O(I_{max}\, N_T \log N_T)$.

In contrast, the proposed scheme employs a block error-correction code (ECC) instead of a convolutional one. The decoding is performed on short coded feedback symbols, eliminating the need for Viterbi decoding. For a block code $[n, k, d]$, the decoding complexity is polynomial, typically at most $O(n^2)$ for syndrome-based algorithms. Since n is small, this term dominates. Moreover, the number of active training layers decreases from $(M - 1)$ to $(M - 1 - \tau)$ due to the unnecessary τ feedback, and thus directly reduces both training and computational load. The total complexity can be expressed as

$$O\big((M-1-\tau)(I_{max}N_T \log N_T + n^2)\big) \tag{7}$$

where the first term represents online codebook generation and the second accounts for block decoding.

Usually, $n \ll L2^k$ in practical feedback systems (e.g., short block codes of 8‑16 symbols). Viterbi decoding is exponential in k, while block code decoding is polynomial ($O(n^2)$), and for practical parameters,

$$O(n^2) \ll O\left(L2^k(N - 1)\right). \tag{8}$$

Hence, our proposed method reduces complexity both by using polynomial block decoding and by decreasing the number of active layers by τ, enabling faster beam training.

9. Conclusion

This paper proposed Semi-Hierarchical Beam Training, specifically the 2^2-ary hierarchical beam training. In this 2^2-ary scheme, the UE sends one symbol consisting of 2-bits feedback (cf. Remark 1) in the hierarchical mixed fixed scheme or $0.5\, log_2 N_T$ in hierarchical mixed adaptive training method, reducing overall training time to $2.5\, log_2 N_T - 1$ (hierarchical mixed adaptive training) or to $2\, log_2 N_T + 1$ (hierarchical mixed fixed), respectfully. We included a dual-layer error-erasure correction mechanism. This system corrects up to t errors in any erroneous $n-1$ symbols and up to τ erasures (one intentionally introduced to stop premature error propagation), sequentially as stated in (cf. Remark 2) to combat error propagation. Compared to other systems, such as in [2], our coding scheme gives significant error propagation reduction in low-SNR and high reliability of the beam selection procedure without adding much feedback overhead, e.g., with a cost of one symbol, we achieve 1.1x faster in (hierarchical plus fixed) method and 1.33x in (hierarchical plus adaptive) scheme.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Hawra'a Albusalih and Haider Alkim; methodology, Hawra'a Albusalih and Haider Alkim; formal analysis, Hawra'a Albusalih; investigation, Hawra'a Albusalih; writing—original draft preparation, Hawra'a Albusalih; review and editing, Haider Alkim; supervision, Haider Alkim.

Acknowledgments

This work was supported by the College of Engineering, University of Kufa, Iraq. And we would like to thank Hayder Al-mosa for his advice on how to meet the reviewer requirements regarding the simulation and the comparisons, etc.

References

[1] E. Björnson, L. Sanguinetti, and M. Kountouris, "Deploying Dense Networks for Maximal Energy Efficiency: Small Cells Meet Massive MIMO", *IEEE J. Sel. Areas Commun.*, Vol. 39, No. 4, pp. 819–831, 2021.

[2] T. Zheng, J. Zhu, Q. Yu, Y. Yan, and L. Dai, "Coded Beam Training", *IEEE J. Sel. Areas Commun.*, Vol. 43, No. 3, pp. 928–943, 2025, doi: 10.1109/JSAC.2025.3531550.

[3] S. Madhusudan, C. Madapatha, B. Makki, H. Guo, and T. Svensson, "Beamforming in Wireless Coded-Caching Systems", *arXiv Preprint*, arXiv:2309.05276, 2023.

[4] M. Moradi, "Polarization-Adjusted Convolutional (PAC) Codes as a Concatenation of Inner Cyclic and Outer Polar- and Reed-Muller-like Codes", *arXiv Preprint*, arXiv:2304.01463v2, 2023.

[5] K. Chen, C. Qi, O. A. Dobre, and G. Y. Li, "Simultaneous multiuser beam training using adaptive hierarchical codebook for mmWave massive MIMO", In: *Proc. of IEEE Glob. Commun. Conf. (GLOBECOM)*, pp. 1–6, 2019.

[6] 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, 2016, doi: 10.1109/TWC.2016.2552040.

[7] Z. Li and C. Qi, "Beam training with limited feedback for multiuser mmWave massive MIMO", In: *Proc. of IEEE Glob. Commun. Conf. (GLOBECOM)*, pp. 1–6, 2020, doi: 10.1109/GLOBECOM42002.2020.9348205.

[8] J. Kim and J. Kim, "Augmenting Beam Alignment for mmWave Communication Systems via Channel Attention", *Electronics*, Vol. 12, No. 20, p. 4318, 2023.

[9] 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, 2014.

[10] Z. Tang, J. Wang, J. Wang, and J. Song, "A High-Accuracy Adaptive Beam Training Algorithm for MmWave Communication", *arXiv preprint*, arXiv:1807.05660v2, 2018.

[11] 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, 2021.

[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, 2015.

[13] Y. Chen and L. Dai, "Coded Beam Training for RIS Assisted Wireless Communications", *IEEE Trans. Wireless Commun.*, Vol. 23, No. 1, pp. 123–135, 2024.

[14] H. Pishro-Nik, "Introduction to Probability, Statistics, and Random Processes", *Kappa Research LLC*, 2014.

[15] M. Moradi and H. Mahdavifar, "PAC codes with bounded-complexity sequential decoding:

Received: August 5, 2025. Revised: November 4, 2025.

Pareto distribution and code design" *arXiv Preprint*, arXiv:2412.06072, 2024.
[16] Code Tables. [Online]. Available: https://www.codetables.de/ [Accessed: Jul. 24, 2025].

Appendix

A. Proof of Corollary 1
Proof. We start with the given inequality:

$$\frac{5}{2}\log_2(N_T) \le 2 \cdot \log_2(N_T) + 2.$$

Simplifying:

$$\frac{1}{2}\log_2(N_T) \le 2 \qquad \Rightarrow \ \log_2(N_T) \le 4.$$

Therefore:

$$N_T \le 2^4 = 16.$$

B. Proof of Corollary 2
Proof. We start with the given inequality:

$$2 \cdot \log_2(N_T) + 1 \ < \ \frac{5}{2}\log_2(N_T) - 1.$$

Rearrange:

$$-\frac{1}{2}\log_2(N_T) \ < -2.$$

Multiply both sides by -2:

$$\log_2(N_T) > 4. \quad \Rightarrow \ N_T > 2^4.$$
$$N_T > 16.$$

C. Hybrid Code: Encoding and Decoding Scheme
Let $\mathbb{F}_{2^2} = \{0, 1, \alpha, \alpha + 1\}$ where $\alpha^2 = \alpha + 1$. We analyze error and erasure decoding in two related codes:
• **C**$_7$: a $[7,4,3]_{2^2}$ Hamming code.
• **C**$_8$: its extended version, a $[8,4,4]_{2^2}$ code.
The generator matrix $\mathbf{G}_7 \in \mathbb{F}_{2^2}^{4\times 7}$ is:

$$G_7 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & \alpha & \alpha \\ 0 & 1 & 0 & 0 & \alpha & 1 & \alpha \\ 0 & 0 & 1 & 0 & \alpha & \alpha & 1 \\ 0 & 0 & 0 & 1 & 1 & \alpha & \alpha+1 \end{bmatrix}$$

Let $\mathbf{m} = (1, \alpha, 0, \alpha + 1) \in \mathbb{F}_{2^2}^{4}$ be the message. Then the encoded codeword is:

$\mathbf{c}_7 = \mathbf{m} \cdot \mathbf{G}_7 = (1, \alpha, 0, \alpha + 1, \alpha + 1, \alpha, \alpha + 1)$

We extend **C**$_7$ by appending one parity bit. The generator matrix $\mathbf{G}_8 \in \mathbb{F}_{2^2}^{4\times 8}$ becomes:

$$\boldsymbol{G_n} = [\,\boldsymbol{I_k}\ |\boldsymbol{P_{k\times(n-k-1)}}|\ \boldsymbol{p}] \in \mathbb{F}_{2^2}^{k\times n},$$
$$\boldsymbol{G_8} = [\,\boldsymbol{I_4}\ |\boldsymbol{P_{4\times 3}}|\ \boldsymbol{p}] \in \mathbb{F}_{2^2}^{4\times 8},$$

$$G_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & \alpha & \alpha & 0 \\ 0 & 1 & 0 & 0 & \alpha & 1 & \alpha & 0 \\ 0 & 0 & 1 & 0 & \alpha & \alpha & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & \alpha & \alpha+1 & 1 \end{bmatrix}$$

Thus:

$\mathbf{c}_8 = \mathbf{m} \cdot \mathbf{G}_8 = [1, \alpha, 0, \alpha + 1, \alpha + 1, \alpha, \alpha + 1, \alpha]$

Suppose the received message contains:
- An error at position 3.
- Intentionally the decoder erases position 0.

Let the error vector be:

$\mathbf{e} = [0, 0, 0, 1, 0, 0, 0, 0]$
$\mathbf{r} = \mathbf{c_8} + \mathbf{e} = [1, \alpha, 0, \alpha, \alpha + 1, \alpha, \alpha + 1, \alpha]$

The syndrome vector is computed as

$\mathbf{s} = \mathbf{r} \cdot \boldsymbol{H}^{\tau}$
$\mathbf{s} = (\mathbf{c} + \mathbf{e}) \cdot \boldsymbol{H}^{\tau}$
$\mathbf{s} = \mathbf{c} \cdot \boldsymbol{H}^{\tau} + \mathbf{e} \cdot \boldsymbol{H}^{\tau}$

Since $\mathbf{c} \cdot \boldsymbol{H}^{\tau} = 0$ for any valid codeword, then:

$\mathbf{s} = \mathbf{e} \cdot \boldsymbol{H}^{\tau}$

Let $\boldsymbol{H_8^{\tau}}$ be the transpose of the parity-check matrix:

$$\boldsymbol{H_8^{\tau}} = \begin{bmatrix} 1 & \alpha & \alpha & 1 \\ \alpha & 1 & \alpha & \alpha \\ \alpha & \alpha & 1 & \alpha+1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{F}_{2^2}^{8\times 4}$$

And
$\mathbf{e} = [0, 0, 0, 1, 0, 0, 0, 0]$
The syndrome is:

$$\boldsymbol{s} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

This indicates an error in position 3.
After correcting the error, we have:

$\mathbf{r}' = (x, \alpha, 0, \alpha + 1, \alpha + 1, \alpha, \alpha + 1, \alpha)$

To solve for the intentionally erased at position 0, define x and use:

$$x \cdot \boldsymbol{h}_1 + \sum_{i=2}^{8} \boldsymbol{r}'_{\boldsymbol{i}} \cdot \boldsymbol{h}_{\boldsymbol{i}} = 0$$

Given:

$$\boldsymbol{h}_{\mathbf{1}} = \begin{bmatrix} 1 \\ \alpha \\ \alpha \\ 1 \end{bmatrix}, \qquad \sum_{\boldsymbol{i}=\mathbf{2}}^{\mathbf{8}} \boldsymbol{r}'_{\boldsymbol{i}} \cdot \boldsymbol{h}_{\boldsymbol{i}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \alpha + 1 \end{bmatrix}$$

To solve exactly, we take only one none-zero equation (i.e., one component) and solve it. We choose the fourth component:

$x \cdot 1 = \alpha + 1 \Rightarrow x = \alpha + 1$

After identifying and correcting the error, the original codeword:

$\hat{\mathbf{c}}_{\mathbf{8}} = (1, \alpha, 0, \alpha + 1, \alpha + 1, \alpha, \alpha + 1, \alpha)$

Remark 4. Assume the first three symbols are erased, i.e., the received word is:

$\mathbf{r} = (x0, x1, x2, \alpha + 1, \alpha + 1, \alpha, \alpha + 1, \alpha)$

Where E = {0, 1, 2} is the set of erasures positions of size $\tau = 3$. Since $d = 4$, we have $\tau \leq d - 1$, so recovery is guaranteed[2]. By the same argument as above (for τ = 3).

$$\sum_{j \in \mathcal{E}} h_j x_j = -\sum_{j \notin \mathcal{E}} h_j r_j ,$$

where $\mathbf{h_j}$ refers to the j-th coordinate (or component) of **h,** x_j is the unknown at position j if that symbol was erased, and r_j is the known received symbol at position j.

Remark 5. The first three symbols are the most significant, as the base station uses $x0$, $x1$, and $x2$ to sequentially determine the primary, secondary, and tertiary directions toward the user. Even if later symbols are incorrect, this coarse localization allows the beam to be directed toward the approximate transmission zone where the user is located or moving. The finer direction can then be inferred probabilistically without requiring additional feedback.