Concatenated BOSS-NB-LDPC Codes for URLLC

Hyowon Lee, *Student Member, IEEE*, Geon Choi, *Student Member, IEEE*,
Chanho Park, *Student Member, IEEE* and Namyoon Lee, *Senior Member, IEEE*

***Abstract*—This paper proposes a concatenated coding architecture that integrates a non-binary low-density parity-check (LDPC) outer code with a block orthogonal sparse superposition (BOSS) inner code, designed for ultra-reliable low-latency communication in low-rate, moderate-blocklength regimes. The inner BOSS code, originally developed for real-valued channels, is extended to complex additive white Gaussian noise channels. Encoding maps each LDPC symbol to a BOSS codeword subvector via a structured unitary matrix with quadrature phase-shift keying modulation, providing diversity and energy efficiency. A low-latency three-stage approximate maximum a posteriori decoder exploits the block-orthogonal structure to separate index, symbol, and block detection. A list-based variant generates the top-K most reliable candidates with log-likelihood ratios for soft input to the LDPC decoder. The outer non-binary LDPC code is decoded using belief propagation with extended min-sum algorithm using the top-K selected message compression, significantly reducing check-node complexity for large Galois field sizes. This joint design achieves near-capacity performance, operating within 1.4 dB of the finite-blocklength limit at a block error rate of 10^{-3}. For a moderate blocklengths of 2048 and 4096 rates $1/4$ and $1/8$, it outperforms 5G LDPC and 5G cyclic redundancy check (CRC)-aided polar codes decoded with a successive cancellation list decoder.**

I. Introduction

Channel coding lies at the heart of reliable wireless communication. As emerging applications demand ultra-reliable low-latency communication (URLLC) for mission-critical services such as industrial automation, autonomous driving, and remote healthcare [1]–[4], its role has never been more crucial. URLLC pushes the limits of both reliability and delay, often requiring short-packet transmissions to meet stringent latency budgets. Yet short packets come at a cost: the error correction capability of any channel code degrades as blocklength shrinks, making the design problem fundamentally harder. The channel codes deployed in today's networks—turbo codes [5] and LDPC codes [6]–[8]—were built for moderate-to-high rates and long blocks, and thus are ill-suited for these new constraints. Beyond reliability, next-generation systems must also drive down decoding latency and improve energy efficiency, especially for mobile and IoT devices. With 5G fully deployed and 5G-Advanced imminent, the question now is how to evolve coding schemes to meet the key performance indicators (KPIs) expected in 6G—reliability, latency, spectral and energy efficiency, connection density, and capacity—while operating in the short or moderate-blocklength URLLC regime [9].

A. Related Work

Over the past decade, finite blocklength information theory has made remarkable progress, with sharp bounds now available on the tradeoff between error probability, code length, and rate [10]. Yet, despite these advances, the principles for designing optimal codes in the finite blocklength regime remain elusive. Among recent developments, transformed polar codes—most notably polarization-adjusted convolutional (PAC) codes [11] and deep polar codes [12]–[14]—have attracted considerable attention. Through tree search-based sequential decoding or successive cancellation list (SCL) decoding [15], these codes achieve performance close to the normal approximation bound at very short blocklengths (e.g., below 256). However, their decoding latency, even with small list sizes (e.g. 8 or 16 list sizes), grows rapidly with blocklength. This latency bottleneck makes them ill-suited for URLLC in the moderate-blocklength regime (e.g., 2048–4096), where both high reliability and low delay are essential. This gap motivates the need for low-rate codes that deliver URLLC-grade reliability with significantly reduced decoding latency at moderate blocklengths.

Sparse regression codes (SPARCs) [16] have been shown to asymptotically achieve Shannon capacity under polynomial-time decoding using adaptive successive decoding or approximate message passing (AMP) [17]. However, their practical performance deteriorates significantly in the short- and moderate-blocklength regimes, which are critical for URLLC. To address this limitation, block orthogonal sparse superposition (BOSS) codes [18]–[20] were proposed as a variant of sparse superposition codes optimized for short blocklengths. BOSS codes employ structured unitary matrices and successive index bit mapping, enabling ordered statistics decoding to achieve maximum likelihood (ML) performance at low rates. Despite these advantages, their specialization for short blocklengths and low rates poses challenges when extending to moderate blocklengths or when soft-output information is required for concatenation with powerful outer codes.

Non-binary low-density parity-check (LDPC) codes [21]–[24] are well known for their strong error-correction capability at short and moderate blocklengths. However, their high decoding complexity, dominated by check-node processing in the belief-propagation (BP) algorithm, is a barrier to low-latency applications. Recent research has introduced low-complexity non-binary LDPC decoders, such as extended min-sum (EMS) [25], adaptive truncated message passing [**?**], and FFT-based check-node updates [**?**], which significantly reduce complexity by operating on compressed message representations.

While BOSS codes excel in low-rate and very short-blocklength scenarios and non-binary LDPC codes provide

H. Lee, G. Choi, C. Park, and N. Lee are with the Department of Electrical Engineering, POSTECH, Pohang, South Korea, (e-mail: {hyowon, geon.choi, chanho26, nylee}@postech.ac.kr).

excellent performance in moderate-blocklength regimes, there has been no prior work that integrates these two paradigms into a single, latency-optimized concatenated design for URLLC. In particular, existing BOSS decoding methods are tailored to hard-decision ML or ordered-statistics detection, with no efficient mechanism to generate the high-quality soft-output values required for non-binary LDPC decoding. This gap motivates the present work.

B. Contributions

This paper presents a new concatenated coding architecture that integrates a non-binary LDPC outer code with a BOSS inner code, specifically optimized for URLLC in low-rate, moderate-blocklength regimes. The goal is to jointly meet three critical design objectives—high reliability, low decoding latency, and manageable complexity—under the stringent requirements of short-packet communications.

On the inner layer, we extend the original BOSS code construction, previously formulated for real-valued channels, to complex additive white Gaussian noise (AWGN) channels. The encoding maps each non-binary LDPC symbol to a BOSS codeword subvector using a structured unitary matrix and QPSK modulation, enabling energy efficiency. For decoding, we propose a low-latency, three-stage approximate maximum a posteriori (MAP) algorithm that exploits the block-orthogonal structure to decouple detection into sequential index, symbol, and block decisions. To interface with the non-binary LDPC decoder, we further develop a list-based BOSS decoding strategy that generates the top-L most reliable symbol candidates along with their log-likelihood ratios (LLRs), ensuring effective soft-information transfer.

On the outer layer, the non-binary LDPC code is decoded via belief propagation (BP) on its factor graph. To reduce decoding computational complexity, we employ compressed message representations in the BP updates leveraging extended min-sum (EMS) [24], which significantly lower the complexity of non-binary check-node processing, particularly for large field sizes.

Simulation results show that the proposed design operates within 1.4 dB of the finite blocklength capacity at a block error rate (BLER) of 10^{-3}. For a blocklength of $N = 2048$ and code rates of $1/4$ and $1/8$, it consistently outperforms state-of-the-art 5G binary LDPC codes and 5G CRC-aided polar codes decoded with a successive cancellation list (SCL) of size 8. The performance gain stems from the synergy between structural exploitation in the list BOSS decoder and message compression in the LDPC belief propagation (BP) decoder, which together yield superior coding performance at moderate blocklengths while maintaining low decoding latency.

II. Preliminaries and Notations

A. Channel Model

We consider a point-to-point communication system in which a transmitter sends a signal $\mathbf{x} \in \mathbb{C}^n$ to a receiver over a memoryless additive white Gaussian noise (AWGN) channel. The received signal $\mathbf{y} \in \mathbb{C}^n$ is given by

$$\mathbf{y} = \mathbf{x} + \mathbf{z}, \tag{1}$$

where $\mathbf{z} \in \mathbb{C}^n$ denotes AWGN whose elements are independent and identically distributed as $\mathbf{z} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$.

B. Sum-Product Decoder for the NB-LDPC

[Geon] 이 subsection 점검 필요 [Geon] Decoding stage 삭제, γ를 channel LLR로 정의, In the second decoding stage, the decoder utilizes a multi-dimensional belief propagation method through the message passing algorithm, leveraging the previously computed LLR values $\gamma_n(\beta)$ for $n \in [N']$.

To explain the conventional belief propagation algorithm, we introduce a set of notations. We denote the message from variable node v_ℓ to check node r_p by $\boldsymbol{\alpha}_{v_\ell \to r_p}$. In the reverse direction, the message is defined as $\boldsymbol{\alpha}_{r_p \to v_\ell}$. We also let $\mathcal{N}_c(v_\ell)$ ($\mathcal{N}_v(r_p)$) be the set of check (variable) nodes connected to variable (check) node $v_\ell(r_p)$. Let $h_{i,j}$ be (i,j) entry of the parity check matrix $\mathbf{H}$. For the message vector update, we define a configuration set for the finite field element s_{v_j} associated with index $v_j \in \mathcal{N}_v(r_p) \backslash v_\ell$ such that [Geon] 원소 s_{v_j}와 조건의 합산 인덱스 s_{v_j}가 겹쳐서 이해가 안됨.

$$\mathcal{A}(r_p | s_{v_\ell} = \beta) = \left\{ s_{v_j} \in \mathbb{F}_q : \sum_{v_j \in \mathcal{N}_v(r_p) \backslash v_\ell} h_{r_p, v_j} s_{v_j} = h_{r_p, v_\ell} \beta \right\}. \tag{2}$$

At iteration k, for each variable node n, each $m \in \mathcal{N}(n)$, each $\beta \in \mathbb{F}_q$[Geon] for each $(n, \mathcal{N}(n), \beta) \in X$, the check node update, which calculates a message going from check node r_p to variable node $v_\ell \in \mathcal{N}_v(r_p)$, is performed by [Geon] variable node n, v_ℓ 표기법 통일. 개인적으로 v_ℓ 표기가 아랫첨자때문에 불편함. 근데 $\mathcal{N}(n)$이 뭐지?

$$\boldsymbol{\alpha}^{(k)}_{r_p \to v_\ell}(\beta) = \sum_{s_{v_j} \in \mathcal{A}(r_p | s_{v_\ell} = \beta)} \left(\prod_{v_j \in \mathcal{N}_v(r_p) \backslash v_\ell} \boldsymbol{\alpha}^{(k-1)}_{v_\ell \to r_p}(s_{v_j}) \right). \tag{3}$$

[Geon] update 식 이상함, 역시 안쪽 곱과 바깥쪽 합의 v_j 표기의 scope 이슈, 왜 v_ℓ에서 온 메시지만 사용? 뭔가 틀린거같음 The variable node update that computes the message vector passed from variable node v_ℓ to check node r_p, is performed as

$$\boldsymbol{\alpha}^{(k)}_{v_\ell \to r_p} = \gamma_{v_\ell}(\beta) \prod_{c_\ell \in \mathcal{N}_c(v_\ell) \backslash r_p} \boldsymbol{\alpha}^{(k)}_{c_\ell \to v_\ell}(\beta). \tag{4}$$

The variable node update and subsequent message computation in NB-LDPC BP are natural extensions of their counterparts in binary BP. Nonetheless, in the case of a code with row weight d_c, the configuration set's cardinality in (2) expands to $\mathcal{O}\left(q^{d_c - 1}\right)$. Consequently, the check node update in NB-LDPC decoding entails the summation of a significantly greater number of products, rendering it a notably more intricate task.

C. Notations

III. BOSS-LDPC Codes

We explain the encoding method of a BOSS-LDPC concatenation code. The proposed BOSS-LDPC encoder comprises a two-stage encoding process: i) NB-LDPC for an outer encoding and ii) BOSS for an inner encoding.

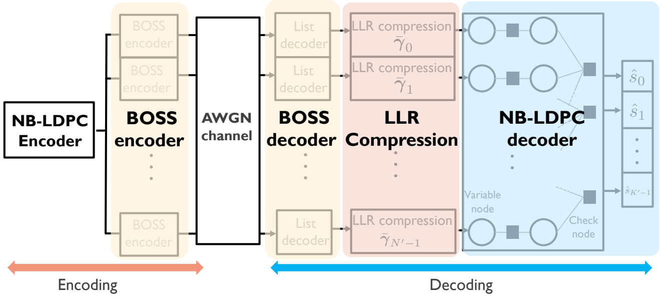

Fig. 1. An illustration of the encoder and decoder for a BOSS-LDPC code.

A. Outer Encoder

The NB-LDPC encoder maps a binary message vector $\mathbf{w} \in \mathbb{F}_2^K$ to a q-ary codeword vector $\mathbf{c} = [s_1, s_2, \ldots, s_{N'}] \in \mathbb{F}_q^{N'}$, where q is the cardinality of the Galois' field. The encoder first maps a binary message vector $\mathbf{w} \in \mathbb{F}_2^K$ to a q-ary non-binary message vector $\mathbf{u} \in \mathbb{F}_q^{K'}$ where $K' = \frac{K}{\log_2(q)}$ by using a bijection ϕ: $\mathbb{F}_q \to [q]$, which maps the elements of $\mathbb{F}_q$ to the integers $[q] = \{1, 2, \ldots, q\}$ [Geon] $\mathbf{w} \mapsto \mathbf{u}$ undefined. ϕ는 갈루아필드 원소와 정수를 identify한 것. 사실 message subvector와 그의 이진수 표현, 그의 십진수 표현, 갈루아 필드를 서로 identify하여 혼용하겠다고 말로 표현하고 넘어가거나, 굳이 표현할거면 해당 노테이션을 모두 도입해야할듯. 개인적으로는 웬만하면 전자를 선호.. Let $\mathbf{G} \in \mathbb{F}_q^{K' \times N'}$ be a generator matrix of the NB-LDPC. Then, the NB-LDPC encoder $\mathcal{E}_{\text{NB-LDPC}} : \mathbb{F}_q^{K'} \to \mathbb{F}_q^{N'}$ generates the codeword as [Geon] $\mathbf{u} = \phi(\mathbf{w})$는 type-check 실패하는 표현

$$\mathcal{E}_{\text{out}} : \phi\left(\mathbf{w}\right)\mathbf{G} = \mathbf{u}\mathbf{G} = \mathbf{c}. \tag{5}$$

The code rate of the outer encoder is $R_1 = \frac{K'}{N'}$.

B. Inner Encoder

The BOSS encoder transforms each non-binary symbol $c_n \in \mathbb{F}_q$ independently into an M-dimensional complex vector $\mathbf{x}_n \in \mathbb{C}^M$, for $n \in \{1, 2, \ldots, N'\}$. Consequently, the overall codeword vector produced by the inner encoder is

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^\top & \mathbf{x}_2^\top & \cdots & \mathbf{x}_{N'}^\top \end{bmatrix}^\top \in \mathbb{C}^{MN'}.$$

Each non-binary symbol c_n represents $\log_2 q$ bits, partitioned into three distinct components, satisfying $q = G \times M \times 4$. Specifically, the bits associated with each non-binary LDPC-coded symbol c_n are grouped into three segments: B_g, B_m, and B_s, such that $B_g + B_m + B_s = \log_2 q$. The encoding of these segments is carried out independently and in parallel for efficiency.

The first group of bits, $B_g = \log_2(G)$, selects one unitary matrix from among G unitary matrix groups $\mathbf{U}_g \in \mathbb{C}^{M \times M}$ for $g \in [G]$, each satisfying $\mathbf{U}_g^{\mathsf{H}}\mathbf{U}_g = \mathbf{I}$. Next, the second group of bits, $B_m = \log_2(M)$, determines a specific column of the chosen matrix. Finally, the remaining two bits, $B_s = 2$, select the phase of the chosen entry from the quadrature phase shift keying (QPSK) constellation $\mathcal{S} = \{+1, -1, +j, -j\}$.

For example, when $q = 256$, possible encoder parameters include $(G, M) = (4, 16)$ or $(2, 32)$. By appropriately selecting the values of M, G, and q, various overall code rates can be achieved, providing flexibility in meeting different communication requirements.

Leveraging the above bit mapping principle, we explain how to generate a BOSS codeword $\mathbf{x}_n$ for c_n. Suppose the mth column of the unitary matrix $\mathbf{U}_g$ is selected and modulated by a QPSK symbol $s \in \mathcal{S}$. The resulting codeword subvector $\mathbf{x}_n \in \mathbb{C}^M$ is then given by

$$\mathbf{x}_n = \mathbf{u}_{g,m} s, \tag{6}$$

where $\mathbf{u}_{g,m}$ denotes the mth column of $\mathbf{U}_g$. This simple mapping generates a BOSS codeword for c_n for $n \in [N']$: its low computational complexity—each subvector is generated by, a unitary matrix selection, a single column selection and scalar multiplication.

C. Code Rate

It is instructive to observe how the BOSS-LDPC code leverages the strengths of both its inner and outer components. The inner encoder takes each non-binary symbol c_n, which carries $\log_2(q)$ bits, and maps it to an M-dimensional complex vector in $\mathbb{C}^M$. This mapping defines the inner code rate as $R_2 = \frac{\log_2(q)}{M}$. By construction, the dimension M relates to the overall code blocklengths as $M = \frac{N}{N'}$, where N is the total channel blocklength and N' is the number of non-binary symbols. This setup yields a remarkably simple expression for the concatenated code rate:

$$R = \frac{K}{N' \log_2 q} \cdot \frac{\log_2 q}{M} = \frac{K}{N}. \tag{7}$$

Thus, regardless of how the bits are partitioned between the inner and outer codes, the overall rate depends only on the information payload K and the channel blocklength N.

The interplay between the inner and outer code rates is central to the overall performance. The BOSS encoder, with its structured mapping, is well suited for scenarios requiring low code rates and short blocklengths, as demonstrated in [19]. In contrast, non-binary LDPC codes excel at higher code rates and moderate blocklengths [26]. By appropriately choosing the rates—typically with $R_1 \geq 1/2$ for the outer LDPC and $R_2 \leq 1/2$ for the inner BOSS encoder—we are able to harness the distinct advantages of each code. This careful balance ensures that the concatenated code achieves high efficiency and robust performance across a range of operating conditions. [Geon] 참고: BOSS의 soft output을 내놓는 과정까지를 새로운 AWGN으로 보고 NB-LDPC의 EXIT chart(?)를 그려볼 수도 있음.

D. Remarks

It is instructive to highlight several remarks that illuminate the key features of the proposed encoding method.

Remark (Unitary Matrix Construction): The choice of unitary matrices $\{\mathbf{U}_g\}$ plays a pivotal role in both performance and complexity. While one could use randomly generated unitary matrices, a more practical approach leverages the structure of the Fourier matrix $\mathbf{F}_M$. Specifically, consider

$$\mathbf{P} = \left[\mathbf{e}_{\pi(1)}, \mathbf{e}_{\pi(2)}, \ldots, \mathbf{e}_{\pi(M)}\right]^\top,$$

where $\mathbf{e}_i$ is the ith standard basis vector and π is a permutation of row indices. By composing the Fourier matrix with different row permutation matrices $\{\mathbf{P}_i\}$, we obtain the family of unitary matrices:

$$\mathbf{U}_i = \mathbf{P}_i \mathbf{F}_M, \tag{8}$$

which satisfy the unitary property:

$$\mathbf{U}_i^{\mathsf{H}} \mathbf{U}_i = \left(\mathbf{P}_i \mathbf{F}_M\right)^{\mathsf{H}} \left(\mathbf{P}_i \mathbf{F}_M\right) = \mathbf{I}. \tag{9}$$

This construction yields an ensemble of unitary matrices that not only preserve mathematical rigor but also enable fast transform operations—an essential feature for efficient decoding with complexity $\mathcal{O}(M \log M)$.

Remark (Connection to OFDM): When the BOSS encoder adopts a row-permuted Fourier matrix, the codeword generation takes the form

$$\mathbf{x}_n = \mathbf{P}_g \mathbf{F}_M \mathbf{e}_m s, \tag{10}$$

where $\mathbf{e}_m$ is the one-hot vector indicating the selected column. This construction can be viewed as a generalization of OFDM, with the added twist that the information bits not only select subcarriers (columns) but also permute the time-frequency structure via $\mathbf{P}_g$. Thus, BOSS encoding naturally integrates with conventional OFDM, while offering additional degrees of freedom for embedding coded information in the OFDM waveform.

Remark (Comparison to SPARC): Unlike the conventional SPARC encoder [16], which employs an unstructured dictionary, the BOSS encoder introduces a structured dictionary matrix with a group orthogonal property:

$$\mathbf{A} = [\mathbf{U}_1,\ \mathbf{U}_2,\ \ldots,\ \mathbf{U}_G] \in \mathbb{C}^{M \times q'}, \tag{11}$$

where $G = \frac{q'}{M}$ is the group size and each $\mathbf{U}_g$ is an $M \times M$ unitary matrix. Explicitly,

$$\mathbf{U}_g = [\mathbf{a}_{(g-1)M+1},\ \mathbf{a}_{(g-1)M+2},\ \ldots,\ \mathbf{a}_{gM}] \in \mathbb{C}^{M \times M}. \tag{12}$$

This group structure is fundamental to the BOSS code's efficient encoding and decoding, as well as its enhanced performance relative to unstructured approaches.

Remark (Difference with Prior Work [19], [20]): Our BOSS encoding method extends the framework of prior work [19], [20], which was originally developed for real-valued spaces $\mathbb{R}^M$, to the complex domain $\mathbb{C}^M$. The key difference in our approach lies in the construction of a set of unitary matrices using the Fourier matrix, which distinguishes our encoding scheme from previous designs. This use of Fourier-based unitary matrices not only enables natural integration with complex modulation formats but also enhances the flexibility and efficiency of the encoding process.

IV. Low-Latency Concentrated Decoder

In this section, we develop a three-stage decoding architecture that bridges the BOSS front-end with the non-binary LDPC back-end [Geon] 이전섹션에서는 inner, outer라고 했으니 통일. We begin with an approximate MAP decoder for the BOSS code, exploiting its block-orthogonal structure to prune the search space. We then elevate this detector[Geon] ‘this‘ detector? to a soft-output list decoder, producing a ranked set of symbol hypotheses along with their LLRs for each non-binary code symbol. These LLRs form the interface to the outer non-binary LDPC decoder, which operates under an extended min-sum algorithm. To further reduce computational burden, we employ a compressed LLR representation in the check-node processing, preserving decoding performance while significantly lowering complexity.

A. Approximate MAP BOSS Decoding

Suppose a codeword subvector $\mathbf{x}_n$, corresponding to the NB-LDPC codeword symbol $c_n \in \mathbb{F}_q$, is generated by selecting the mth column of a unitary matrix $\mathbf{U}_g$ and modulating it with a QPSK symbol $x_s \in \mathcal{S}$ [Geon] 앞에선 s였으니 통일, as follows:

$$\mathbf{x}_n = \mathbf{u}_{g,m} x_s, \tag{13}$$

where $\mathbf{u}_{g,m}$ denotes the mth column of $\mathbf{U}_g$.

Given the observation $\mathbf{y}_n$, the maximum a posteriori (MAP) decoding can be naturally decomposed using the chain rule of probability:

$$\begin{aligned}
&\hat{\mathbf{x}}_n^{\mathrm{MAP}} \\
&= \arg \max_{\mathbf{x}_n \in \mathbb{C}^M} \mathbb{P}(\mathbf{x}_n \mid \mathbf{y}_n) \\
&= \arg \max_{g \in [G],\, m \in [M],\, s \in [S]} \mathbb{P}(g, m, s \mid \mathbf{y}_n) \\
&= \arg \max_{g \in [G],\, m \in [M],\, s \in [S]} \mathbb{P}(s \mid \mathbf{y}_n, m, g)\, \mathbb{P}(m \mid \mathbf{y}_n, g)\, \mathbb{P}(g \mid \mathbf{y}_n).
\end{aligned} \tag{14}$$

Solving this MAP decoding problem jointly requires a complexity of order $\mathcal{O}(q)$, which becomes prohibitive as the field size q grows, which typically can be chosen as $q \in \{256, 512, 1024\}$. To address this, we propose a three-stage approximate MAP decoding strategy that exploits the problem structure and reduces the complexity.

- **Stage 1:** For each $g \in [G]$, identify the most likely $m \in [M]$:
$$\hat{m} = \arg \max_{m \in [M]} \mathbb{P}(m \mid \mathbf{y}_n, g). \tag{15}$$
- **Stage 2:** For each pair (g, m), identify the most likely $s \in \mathcal{S}$:
$$\hat{s} = \arg \max_{s \in [S]} \mathbb{P}(s \mid \mathbf{y}_n, g, m). \tag{16}$$
- **Stage 3:** Finally, select the best $g \in [G]$ based on the previous selections:
$$\hat{g} = \arg \max_{g \in [G]} \mathbb{P}(g \mid \mathbf{y}_n). \tag{17}$$

By decoupling the original MAP problem into these three coordinated stages, the overall decoding complexity is dramatically reduced. This approach highlights how careful exploitation of code structure transforms an intractable problem into a practical and efficient decoding algorithm.

Stage 1 Decoding: In the first stage, the decoder identifies the index of column assuming $\mathbf{U}_g$ is fixed. The following theorem shows that the ordered statistic decoding is the MAP decoding.

The decoder first computes the proxy vector by multiplying $\mathbf{U}_{g'}^{\mathsf{H}}$ to $\mathbf{y}_n$ for $g' \in [G]$ as

$$\bar{\mathbf{y}}_{n,g'} = \mathbf{U}_{g'}^{\mathsf{H}} \mathbf{y}_n. \tag{18}$$

Suppose $g' = g$. Then, the proxiy vector becomes

$$\bar{\mathbf{y}}_{n,g} = \mathbf{x}_{n,g} + \bar{\mathbf{z}}_{n,g}, \tag{19}$$

where $\bar{\mathbf{z}}_{n,g} = \mathbf{U}_g^{\mathsf{H}} \mathbf{z}_n$ denotes the effective noise. Since the noise distribution is invariant over the unitary transformation, it follows $\bar{\mathbf{z}}_{n,g} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$.

Assuming a uniform prior over the possible indices m, the MAP decision rule simplifies to a maximization over the likelihood function:

$$\hat{m} = \arg \max_{m \in \{1,\ldots,M\}} \mathbb{P}(\bar{y}_{n,g,m} \mid \text{signal at } m). \tag{20}$$

Under the hypothesis that the signal is present at position m, the likelihood function, marginalized over the unknown QPSK symbol s, can be written as:

$$\mathbb{P}(\bar{y}_{n,g,m} \mid \text{signal at } m) = \frac{1}{S} \sum_{x_s \in \mathcal{S}} \frac{1}{\pi \sigma^2} \exp\left(-\frac{|\bar{y}_{n,g,m} - x_s|^2}{\sigma^2} \right), \tag{21}$$

where $\mathcal{S} = \{1, -1, +j, -j\}$ denotes the set of QPSK constellation points. For all other indices $m' \neq m$, the observation $\bar{y}_{n,g,m'}$ contains only noise, so its likelihood is

$$\mathbb{P}(\bar{y}_{n,g,m'} \mid \text{signal not at } m') = \frac{1}{\pi \sigma^2} \exp\left(-\frac{|\bar{y}_{n,g,m'}|^2}{\sigma^2} \right). \tag{22}$$

Since the noise-only terms are statistically identical and independent of the signal location, they do not influence the outcome of the maximization. Thus, the optimal decision rule reduces to:

$$\hat{m} = \arg \max_{m \in \{1,\ldots,M\}} \sum_{x_s \in \mathcal{S}} \exp\left(-\frac{|\bar{y}_{n,g,m} - x_s|^2}{\sigma^2} \right). \tag{23}$$

For each candidate index m, we evaluate how well the observation $\bar{y}_{n,g,m}$ matches each possible QPSK symbol by summing the corresponding likelihoods. The index yielding the highest sum is selected as the most likely signal location. This detection rule not only exploits the inherent symmetry of the QPSK constellation but also achieves optimal performance under the uniform prior. Notably, this approach can be viewed as a form of ordered statistics decoding (OSD), as the decision for m is made solely based on the local observation $\bar{y}_{n,g,m}$ via (23).

Stage 2 Decoding: [Geon] State 1 + 2를 해서 $\bar{\mathbf{y}}_{n,g'}$에 $1, -1, i, -i$를 곱해서 나온 후보들 중에서 확률이 제일 큰거 고르면 안되나? 이렇게 하면 진짜 MAP이고, 나눠서 하면 approximate MAP일텐데 어차피 계산복잡도는 동일하고. 예를 들어.. 어떤 column에 대해 각 symbol이 pdf $0.1, 0.1, 0.1, 10$을 주는 경우와 $3, 3, 3, 3$을 주는 경우가 있을때 안나누면 10짜리가, 나누면 $3, 3, 3, 3$ 컬럼이 골라져서 최종으로 3이 골라질테니까. In the second stage, the decoder determines the transmitted QPSK symbol $s \in \mathcal{S}$, given the previously selected group g and column index $\hat{m}$. Because each QPSK symbol is equally likely, the optimal decision rule reduces to a minimum-distance criterion:

$$\hat{s} = \arg \min_{s \in \mathcal{S}} |\bar{y}_{n,g,\hat{m}} - x_s|^2. \tag{24}$$

In other words, the decoder selects the QPSK symbol closest in Euclidean distance to the observed value $\bar{y}_{n,g,\hat{m}}$.

Stage 3 Decoding: In the final stage, the decoder aims to identify the true block index g among all candidates in $[G]$. Leveraging the results of the previous stages, the decoder constructs a tentative codeword for each g, given by $\hat{\mathbf{x}}_{n,g} = \mathbf{u}_{g,\hat{m}} \hat{s}$. Assuming that the noise vector $\mathbf{z}_n$ is complex Gaussian and the block index was selected uniformly at random, the MAP hypothesis test reduces to a minimum-distance search:

$$\hat{g} = \arg \min_{g \in [G]} \|\mathbf{y}_n - \hat{\mathbf{x}}_{n,g}\|_2^2. \tag{25}$$

That is, the decoder selects the block index whose candidate codeword is closest to the received signal in Euclidean distance.

Decoder Complexity: In Stage 1, the receiver first computes the proxy vector $\bar{\mathbf{y}}_{n,g} = \mathbf{U}_g^{\mathsf{H}} \mathbf{y}_n$. For arbitrary unitary matrices, this operation requires $\mathcal{O}(M^2)$ computations. However, if the set of unitary matrices is constructed as row-permuted Fourier matrices, i.e., $\mathbf{U}_g = \mathbf{P}_g \mathbf{F}_M$, the complexity can be reduced to $\mathcal{O}(M \log M)$ using the fast Fourier transform.

Identifying the optimal indices $\hat{m}$ and $\hat{s}$ requires $\mathcal{O}(SM)$ computations, where S is the size of the signal constellation. For the final stage, searching over all block indices incurs a complexity of $\mathcal{O}(GM)$. Summing up all stages, the overall decoding complexity is

$$\mathcal{O}\left(GM \log M + GSM + GM \right),$$

which is dominated by $\mathcal{O}(GM \log M)$ when the unitary matrices are based on Fourier constructions.

B. *Soft-Output List Decoding for BOSS Codes*

The MAP detection of a BOSS-encoded symbol

$$\mathbf{x}_n = \mathbf{u}_{g,m} x_s \tag{26}$$

amounts to solving

$$\max_{g \in [G],\ m \in [M],\ s \in [S]} \mathbb{P}(g, m, s \mid \mathbf{y}_n). \tag{27}$$

A brute-force evaluation requires $\mathcal{O}(q)$ operations, where $q = GMS$ and $S = |\mathcal{S}|$ (e.g., $S = 4$ for QPSK). For $q \in \{256, 512, 1024\}$, this becomes prohibitive. The block-orthogonal structure of BOSS allows us to decompose the search into three stages, keeping only the L most likely hypotheses for soft-output generation.

[Geon] 여기서 column-symbol joint니까 앞 서브섹션도 그게 더 자연스러울듯. 그래야 $L_g = 1$인게 앞 디코더로 환원되기도 할듯..?

Stage 1: Joint column–symbol ranking within each group. For each group g, project the received vector:

$$\bar{\mathbf{y}}_{n,g} = \mathbf{U}_g^{\mathsf{H}} \mathbf{y}_n. \tag{28}$$

For each $(m, s) \in [M] \times [S]$, compute the score

$$\Lambda_g(m, s) = \exp\left(-\frac{|\bar{y}_{n,g,m} - x_s|^2}{\sigma^2} \right), \tag{29}$$

and retain the L_g best (m, s) pairs for that g. Let

$$\mathcal{I}_g = \{(g, m_k, s_k)\}_{k=1}^{L_g}. \tag{30}$$

Stage 2: Candidate set construction across groups. Form the union

$$\mathcal{I} = \bigcup_{g=1}^{G} \mathcal{I}_g, \tag{31}$$

with total size $L = \sum_{g=1}^{G} L_g < q$. Each element of $\mathcal{I}$ is a triple (g, m, s).

Stage 3: Mapping to non-binary symbols. The BOSS encoder defines a bijection [Geon] III-A와 표기 ϕ 중복

$$\phi : [G] \times [M] \times [S] \longrightarrow \{1, \ldots, q\}, \tag{32}$$

given by

$$i = \phi(g, m, s) = (g-1)(MS) + (m-1)S + s. \tag{33}$$

The inverse mapping $\phi^{-1}(i) = (g, m, s)$ is [Geon] 갈루아필드 원소 0부터 표현하면 $i-1$이 사라져서 더 깔끔할지도?

$$g = \left\lfloor \frac{i-1}{MS} \right\rfloor + 1, \tag{34}$$

$$m = \left\lfloor \frac{i - 1 - (g-1)MS}{S} \right\rfloor + 1, \tag{35}$$

$$s = i - (g-1)MS - (m-1)S. \tag{36}$$

Thus, each retained $(g, m, s) \in \mathcal{I}$ corresponds to a unique non-binary symbol $\alpha_i \in \mathbb{F}_q$.

Stage 4: Ordered list and soft metrics. Define the ordered index set

$$\mathcal{L} = \{i_1, i_2, \ldots, i_L\} \subset \{1, \ldots, q\}, \tag{37}$$

sorted so that

$$\mathbb{P}(\mathbf{y}_n \mid c_n = \alpha_{i_\ell}) > \mathbb{P}(\mathbf{y}_n \mid c_n = \alpha_{i_j}), \quad \ell < j. \tag{38}$$

Under the AWGN model,

$$\mathbb{P}(\mathbf{y}_n \mid c_n = \alpha_{i_\ell}) = \exp\left(-\frac{\|\mathbf{y}_n - \mathbf{u}_{g_{i_\ell}, m_{i_\ell}} x_{s_{i_\ell}}\|_2^2}{\sigma^2} \right). \tag{39}$$

Stage 5: LLR computation for NB-LDPC decoding. Let $\phi(1, 1, 1) = 1$ be the reference symbol. For each retained index i_ℓ, the log-likelihood ratio is

$$\lambda_n(\alpha_{i_\ell}) = \log \frac{\mathbb{P}(\mathbf{y}_n \mid c_n = \alpha_{i_\ell})}{\mathbb{P}(\mathbf{y}_n \mid c_n = \alpha_1)} \tag{40}$$

$$= \log \frac{\exp\left(-\frac{\|\mathbf{y}_n - \mathbf{u}_{g_{i_\ell}, m_{i_\ell}} x_{s_{i_\ell}}\|_2^2}{\sigma^2} \right)}{\exp\left(-\frac{\|\mathbf{y}_n - \mathbf{u}_{1,1} x_1\|_2^2}{\sigma^2} \right)}. \tag{41}$$

The LLR vector $\boldsymbol{\lambda}_n$ formed from these values feeds directly into the non-binary LDPC decoder.

For $j \notin \mathcal{L} = \{i_1, \ldots, i_L\}$ and $j \in [q]$, the decoder assigns zeros in the corresponding LLRs. As a result, the resulting LLR vector

$$\boldsymbol{\lambda}_n = \left[\lambda_n(\alpha_1), \lambda_n(\alpha_2), \ldots, \lambda_n(\alpha_q) \right]$$

is sparse since only L entries correspond to retained candidates. This sparsity is exploited in the subsequent NB-LDPC decoding stage via the EMS algorithm.

C. Extended Min-Sum Decoder

In the EMS framework, each message vector $\boldsymbol{\lambda}_n \in \mathbb{R}^q$ passed between nodes in the LDPC Tanner graph is stored in a compressed form:

$$\boldsymbol{\lambda}_n^L = [\lambda_n(\alpha_{i_1}), \lambda_n(\alpha_2), \ldots, \lambda_n(\alpha_{i_L})] \in \mathbb{R}^L \tag{42}$$

for $i_\ell \in \mathcal{L}$. The BOSS soft-output stage naturally provides such a truncated likelihood profile, allowing $L \ll q$ at the cost of a negligible decoding performance loss.

Check-node update: For each check node r_p and connected variable node v_ℓ, the EMS update computes

$$\mathcal{M}_{r_p \to v_\ell}^{(k)} = \mathrm{Trunc}_L \left(\min_{\substack{\beta_j \in \mathbb{F}_q \\ \sum h_{r_p, v_j} \beta_j = h_{r_p, v_\ell} \alpha}} \sum_{v_j \in \mathcal{N}_v(r_p) \setminus v_\ell} \lambda_{v_j \to r_p}^{(k-1)}(\beta_j) \right), \tag{43}$$

where $\mathrm{Trunc}_L(\cdot)$ retains only the L smallest entries. [Geon] SP랑 같은 표기이슈

Variable-node update: For each variable node v_ℓ and check node r_p,

$$\mathcal{M}_{v_\ell \to r_p}^{(k)} = \mathrm{Trunc}_L \left(\lambda_{v_\ell}^{\mathrm{BOSS}}(\alpha) + \sum_{c' \in \mathcal{N}_c(v_\ell) \setminus r_p} \lambda_{c' \to v_\ell}^{(k)}(\alpha) \right), \tag{44}$$

where $\lambda_{v_\ell}^{\mathrm{BOSS}}(\alpha)$ are the initial LLRs from Stage 5. After $I_{\max}$ iterations,

$$\hat{c}_{v_\ell} = \arg\min_{\alpha \in \mathbb{F}_q} \left[\lambda_{v_\ell}^{\mathrm{BOSS}}(\alpha) + \sum_{c' \in \mathcal{N}_c(v_\ell)} \lambda_{c' \to v_\ell}^{(I_{\max})}(\alpha) \right]. \tag{45}$$

By tightly coupling the top-L likelihoods α_{i_ℓ} selected via $\phi(g, m, s)$ with EMS message compression, the overall decoding complexity scales with L rather than q, enabling low-latency, high-reliability NB-LDPC decoding in the short-packet regime.

D. Remarks

Decoding Complexity: The proposed receiver front-end reduces the MAP detection search space from $\mathcal{O}(q)$ to $\mathcal{O}(L)$ symbol hypotheses per codeword position, where $L \ll q$. This truncation directly carries over to the EMS decoding stage: the check-node updates now scale as

$$\mathcal{O}(d_c L^2)$$

TABLE I
Comparison of complexity and latency

	BOSS-LDPC	Polar
Complexity	$\mathcal{O}(N'\cdot\max(GlM^2,d_cN_m\log_2 N_mI))$	$\mathcal{O}(LN\log N)$
Latency	$\mathcal{O}(GlM^2+d_cN_m\log_2 N_mI)$	$\mathcal{O}(N\log N)$

instead of $\mathcal{O}(d_cq^2)$, and the variable-node updates scale as

$$\mathcal{O}(d_vL)$$

instead of $\mathcal{O}(d_vq)$, where d_c and d_v denote the check and variable node degrees, respectively. When $q\in\{256,512,1024\}$ and L is on the order of tens, this reduction yields orders-of-magnitude savings in both arithmetic operations and memory access. The combination of BOSS-based soft-output generation and EMS decoding thus achieves near-MAP error performance while retaining the computational tractability and low latency required for short-packet, high-throughput URLLC scenarios.

Decoding Latency: When ample parallel computing resources are available, both the symbol-wise BOSS front-end and the non-binary LDPC back-end admit fine-grained parallelization. In this regime, the BOSS detection for all N' symbols can be executed concurrently, resulting in a per-symbol latency scaling as

$$\mathcal{O}(G\,L\,M^2),$$

where G is the number of unitary groups, M the block size, and L the list size. Similarly, NB-LDPC belief propagation can be parallelized across all check-node updates, yielding a decoding latency of

$$\mathcal{O}(L\log L\,I),$$

where L is the number of messages per iteration and I the iteration count.

Table I contrasts the latency of our scheme with state-of-the-art coding methods. While the use of non-binary LDPC codes precludes outperforming binary LDPC in absolute complexity or latency, our design operates with lower latency than polar codes at the moderate blocklengths of interest. This competitive regime—low latency, high reliability, and moderate complexity—makes the proposed scheme well-suited for short-packet URLLC deployments.

V. List BOSS Decoding Analysis

In this section, we present an exact analytical expression for the block error rate (BLER) of single-layered BOSS codes under the proposed three-stage maximum a posteriori (MAP) list decoding scheme. Our analysis reveals how the BLER fundamentally depends on key system parameters, including the blocklength M, the number of unitary matrices G, and the signal-to-noise ratio E_b/N_0. The following theorem summarizes the main result of this section.

VI. Simulation Results

This section compares the BLERs of proposed codes with existing codes under AWGN channel. We consider the following codes and benchmarks. The detailed information about the construction is listed in Table II.

- **BOSS-LDPC codes:** The decoding performance of proposed codes depends on the construction of outer NB-LDPC codes and inner BOSS codes. For the outer codes, we employed the progressive edge growth (PEG) algorithm to construct a Tanner graph with variable-node degree $d_v=2$ and average check-node degree $\bar{d}_c=4$, thereby achieving a code rate of $1/2$, and we randomly assinged nonzero weights over $\mathbb{F}_q$. For the inner code, targeting overall rates of $1/4$ and $1/8$, we chose the BOSS code parameters $(G,M)=(16,16)$ and $(8,32)$, respectively. To trade-off complexity and performance in the concatenated scheme, we set the BOSS list size to 4 and the outer-code compression size to 30 for the low-complexity design, and increased them to list size 16 and compression size 64 as a near-optimal approach.
- **CA-polar codes:** The CA-polar codes serve as the baseline methods. The information sets are chosen according to the channel-independent polarization weight method in [27]. The 24-bit CRC polynomial is used, which are adopted in 3GPP standards [28]. For decoding, the standard SCL decoders with the list size of 2 and 4 are used [29].
- **Binary LDPC codes:** The binary LDPC codes are considered as the baseline methods. We generate the parity check matrix using the base graph defined in 5G standard [28]. The CRC polynomial $g_{\text{CRC16}}(x)=x^{16}+x^{12}+x^9+x^5+1$ is used. The maximum iteration of the decoder is 12.
- **SR-LDPC codes:** The SR-LDPC code is a decoding method that concatenates LDPC with the AMP-based decoder [30]. As in this study, we set the rate of the outer code to 0.9, and the number of channel uses is 2048, consistent with other codes for performance comparison. The LDPC channel was constructed using the PEG algorithm. The number of iterations for the AMP decoder was set to 25, and the maximum number of iterations for the LDPC decoder is set to 30.
- **Theoretical bound [10]:** We use normal approximation bound as a theoretical bound to assess the gaps between the achievable performance of our methods and the finite-blocklength capacity.

Fig. 2 shows the BLER performance comparison among BOSS-LDPC, binary LDPC, and CA-polar codes for $N=2048$ and $R=\frac{1}{4}$, with the normal approximation bound as the benchmark. Under the near-optimal configuration of the proposed scheme—where the inner code employs full search and the outer code uses a large compression size ($N_m=64$), resulting in high computational complexity—we observe an approximate 1 dB performance gain over existing coding schemes. Conversely, in the lower-complexity variant of our algorithm, the inner code's performance in the complex domain is markedly superior to that in the real domain;

TABLE II
SIMULATION PARAMETERS

	Fig.	(N', K', q)	CRC	(G, M, ℓ)	N_m
NAB	2	(2048,512,ϕ)	ϕ	ϕ	ϕ
Binary-LDPC	2	(2048,512,2)	16	ϕ	ϕ
SR-LDPC	2	(71,64,256)	ϕ	ϕ	ϕ
CA-Polar + SCL-2	2	(2048,512,ϕ)	24	ϕ	ϕ
CA-Polar + SCL-4	2	(2048,512,ϕ)	24	ϕ	ϕ
CA-Polar + SCL-8	2	(2048,512,ϕ)	24	ϕ	ϕ
BOSS-LDPC	2	(128,64,256)	ϕ	(16,16,16)	64
NAB	**??**	(2048,256,ϕ)	ϕ	ϕ	ϕ
SR-LDPC	**??**	(71,64,256)	ϕ	ϕ	ϕ
CA-Polar + SCL-2	**??**	(2048,256ϕ)	24	ϕ	ϕ
CA-Polar + SCL-4	**??**	(2048,256,ϕ)	24	ϕ	ϕ
CA-Polar + SCL-8	**??**	(2048,256,ϕ)	24	ϕ	ϕ
BOSS-LDPC in $\mathbb{C}$	**??**	(128,64,256)	ϕ	(8,32,16)	30

** NAB: normal approximation bound

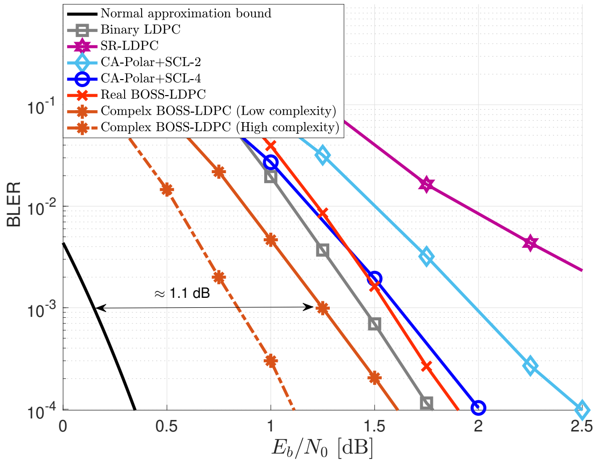

Fig. 2. BLER performance of binary LDPC, CA-polar, and BOSS-LDPC codes. The parameters are $N = 2048$ and $R = 1/4$.

this improvement is attributable to the dimensionality gain. In addition, BOSS-LDPC exhibits superior performance over CA-polar and 5G binary LDPC codes, achieving a coding gain of approximately 0.2 dB at 10^{-3} BLER. Notably, the gap to the nomal-approximation bound at 10^{-3} under practical parameter setting is 1.1 dB.

Fig. **??** shows the BLER performance of BOSS-LDPC, binary LDPC, non-binary LDPC, and CA-polar codes, evaluated at blocklength $N = 2048$ and code rate $R = \frac{1}{3}$. The results elucidate that the BOSS-LDPC code outperforms the binary LDPC and CA polar codes on the SCL decoder with list size two and the NB-LDPC code at a code rate of $1/4$. A performance crossover point, however, occurs at 2 dB with CA-polar using the SCL decoder with size 4.

VII. CONCLUSION

This letter proposes a novel concatenated code combining the BOSS inner code with NB-LDPC outer codes. This concatenated code exhibits structural compatibility between the soft information decoded in the BOSS code and the compressed LLR vector of NB-LDPC. We demonstrate superior performance from simulations compared to benchmark schemes such as 5G CA-polar and LDPC codes at a moderate blocklength of 2048 in supporting low code rates.

REFERENCES

[1] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, “5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view,” *IEEE Access*, vol. 6, pp. 55 765–55 779, 2018.

[2] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas, J. S. Thompson, E. G. Larsson, M. D. Renzo, W. Tong, P. Zhu, X. Shen, H. V. Poor, and L. Hanzo, “On the road to 6G: Visions, requirements, key technologies, and testbeds,” *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 905–974, 2nd Quart. 2023.

[3] K. David and H. Berndt, “6G vision and requirements: Is there any need for beyond 5G?” vol. 13, no. 3, pp. 72–80, Sep. 2018.

[4] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li, S. Johnson, and B. Vucetic, “Short block-length codes for ultra-reliable low latency communications,” *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 130–137, 2019.

[5] S. Benedetto and G. Montorsi, “Unveiling turbo codes: Some results on parallel concatenated coding schemes,” *IEEE Trans. Inf. Theory*, vol. 42, no. 2, pp. 409–428, 1996.

[6] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[7] D. J. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[8] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Improved low-density parity-check codes using irregular graphs,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 585–598, 2001.

[9] M. Rowshan, M. Qiu, Y. Xie, X. Gu, and J. Yuan, “Channel coding toward 6G: Technical overview and outlook,” *IEEE Open Journal of the Communications Society*, vol. 5, pp. 2585–2685, 2024.

[10] Y. Polyanskiy, H. V. Poor, and S. Verdu, “Channel coding rate in the finite blocklength regime,” *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[11] E. Arıkan, “From sequential decoding to channel polarization and back again,” *arXiv:1908.09594*, 2019, [Online]. Available: https://arxiv.org/abs/1908.09594.

[12] G. Choi and N. Lee, “Deep polar codes,” *IEEE Transactions on Communications*, vol. 72, no. 7, pp. 3842–3855, 2024.

[13] ——, “Sparsely pre-transformed polar codes for low-latency scl decoding,” *IEEE Transactions on Communications*, pp. 1–1, 2025.

[14] ——, “Rate-matching deep polar codes via polar coded extension,” *arXiv:2505.06867*, 2025, [Online]. Available: https://arxiv.org/abs/2505.06867.

[15] H. Yao, A. Fazeli, and A. Vardy, “List decoding of arıkan’s PAC codes,” *Entropy*, vol. 23, no. 7, p. 841, 2021.

[16] A. Joseph and A. R. Barron, “Least squares superposition codes of moderate dictionary size are reliable at rates up to capacity,” *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2541–2557, 2012.

[17] C. Rush, A. Greig, and R. Venkataramanan, “Capacity-achieving sparse regression codes via approximate message passing decoding,” in *2015 IEEE Int. Symp. Inf. Theory (ISIT)*, 2015, pp. 2016–2020.

[18] J. Park, J. Choi, W. Shin, and N. Lee, “Block orthogonal sparse superposition codes,” in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2021.

[19] D. Han, J. Park, Y. Lee, H. V. Poor, and N. Lee, “Block orthogonal sparse superposition codes for ultra-reliable low-latency communications,” *IEEE Trans. Commun.*, vol. 71, no. 12, pp. 6884–6897, Dec. 2023.

[20] D. Han, B. Lee, M. Jang, D. Lee, S. Myung, and N. Lee, “Block orthogonal sparse superposition codes for L^3 communications: Low error rate, low latency, and low transmission power,” *IEEE Journal on Selected Areas in Communications*, vol. 43, no. 4, pp. 1183–1199, April 2025.

[21] M. C. Davey and D. J. MacKay, “Low density parity check codes over GF (q),” in *Proc. IEEE Inf. Theory Workshop (ITW)*, 1998, pp. 70–71.

[22] M. Davey and D. MacKay, “Low density parity check codes over gf(q),” in *1998 Information Theory Workshop (Cat. No.98EX131)*, 1998, pp. 70–71.

[23] D. Declercq and M. Fossorier, “Decoding algorithms for nonbinary LDPC codes over GF(q),” *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, 2007.

[24] D. Declercq, M. Fossorier, and E. Biglieri, “Error control coding for non-binary ldpc codes,” in *Wiley Encyclopedia of Electrical and Electronics Engineering*, J. G. Webster, Ed. Hoboken, NJ, USA: Wiley, 2015.

[25] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, “Low-complexity decoding for non-binary LDPC codes in high order fields,” *IEEE Transactions on Communications*, vol. 58, no. 5, pp. 1365–1375, 2010.

[26] ——, “Low-complexity decoding for non-binary LDPC codes in high order fields,” *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1365–1375, 2010.

[27] G. He, J.-C. Belfiore, I. Land, G. Yang, X. Liu, Y. Chen, R. Li, J. Wang, Y. Ge, R. Zhang, and W. Tong, “Beta-expansion: A theoretical framework for fast and recursive construction of polar codes,” in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2017.

[28] 3GPP, “NR; multiplexing and channel coding,” *TS 38.212, Rel. 16*, 2020.

[29] I. Tal and A. Vardy, “List decoding of polar codes,” *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[30] J. R. Ebert, J.-F. Chamberland, and K. R. Narayanan, “On sparse regression LDPC codes,” *arXiv:2301.01899*, 2023, [Online]. Available: https://arxiv.org/abs/2301.01899.