Deep Polar Codes

Geon Choi, *Student Member, IEEE* and Namyoon Lee, *Senior Member, IEEE*

Abstract—**In this paper, we introduce a novel class of pre-transformed polar codes, termed as *deep polar codes.* We first present a deep polar encoder that harnesses a series of multi-layered polar transformations with varying sizes. Our approach to encoding enables a low-complexity implementation while significantly enhancing the weight distribution of the code. Moreover, our encoding method offers flexibility in rate-profiling, embracing a wide range of code rates and blocklengths. Next, we put forth a low-complexity decoding algorithm called successive cancellation list with *back-propagation parity checks* (SCL-BPC). This decoding algorithm leverages the parity check equations in the reverse process of the multi-layered pre-transformed encoding for SCL decoding. Additionally, we present a low-latency decoding algorithm that employs parallel-SCL decoding by treating partially pre-transformed bit patterns as additional frozen bits. Through simulations, we demonstrate that deep polar codes outperform existing pre-transformed polar codes in terms of block error rates across various code rates under short block lengths, while maintaining low encoding and decoding complexity. Furthermore, we show that concatenating deep polar codes with cyclic-redundancy-check codes can achieve the meta-converse bound of the finite block length capacity within 0.4 dB in some instances.**

I. Introduction

The quest for ultra-reliable low-latency communications (URLLC) in next-generation wireless systems remains relentless [?], [?], [?], [?]. The concept of URLLC revolves around achieving lightning-fast data packet delivery within an incredibly short timeframe, such as 1 millisecond, while ensuring exceptional reliability, with packet error rates ranging from 10^{-3} to 10^{-6} [?]. The pursuit of URLLC remains unwavering for the next generation of wireless systems. The success of achieving ultra-low latency and high reliability is contingent on developing a cutting-edge channel coding technique capable of fast decoding and near-optimal performance in the finite-blocklength regime [?], [?]. While significant progress has been made in understanding finite-blocklength information theory [?], [?], [?], designing the optimal code in this context remains a formidable challenge due to the finite-blocklength constraint. In this paper, we make progress toward designing a new class of pre-transformed polar codes that can achieve the finite-blocklength capacity closely with a low-complexity encoder and decoder.

A. Related Work

Polar codes, invented by Arikan, represent a significant milestone as the first explicitly constructed error-correcting codes over binary input memoryless channels that achieve provably asymptotic capacity [?]. These codes utilize an explicit encoder and a successive cancellation (SC) decoder, boasting low complexity, approximately $\mathcal{O}(N \log N)$, where N denotes the code blocklength [?]. However, it is essential to note that standard polar codes do not exhibit outstanding performance at short-to-moderate blocklengths compared to low-density parity-check (LDPC) or turbo codes [?]. The reason is due to their inferior minimum distance and incomplete polarization. In order to address the challenges posed by incomplete polarization and enhance decoding robustness, SC-list (SCL) decoding was introduced in [?], [?]. The SCL decoding has demonstrated superior performance compared to the standard SC decoder by increasing the list sizes of a decoder. Remarkably, as the list size grows, the performance of SCL decoding approaches that of maximum likelihood (ML) decoding. However, despite the utilization of ML decoding, the weight spectrum of polar codes is relatively inferior when compared to that of Reed-Muller (RM) codes [?], [?], [?].

Recently, there have been significant advancements in improving the weight spectrum of polar codes through pre-transform techniques at short-to-moderate blocklengths. One such approach involves concatenating polar codes with cyclic-redundancy-check (CRC) [?], [?] and parity-check (PC) [?], [?], [?], [?]. In addition, Arikan's PAC codes [?] leverage convolutional precoding as a pre-transformation. In the binary input additive white Gaussian noise (BI-AWGN) channel, it turns out that the PAC codes with RM rate-profiling can achieve dispersion bound very closely under Fano decoding or SCL decoding using large list sizes [?], [?], [?]. Notwithstanding the quasi-optimal performance, the major drawback of the PAC codes lies in their high decoding complexity. For instance, the tree-search-based sequential decoders (e.g., Fano or stack decoders) can result in considerable decoding delay at a low signal-to-noise ratio (SNR), which is not applicable for low latency scenarios. Also, the SCL decoding with a large list size increases hardware complexity considerably. To reduce the decoding complexity, some prior studies in [?], [?] have put forth various extensions of the sequential decoding using improved metrics. Some notable prior studies in [?], [?], [?], [?], [?] have proposed methods for constructing PAC codes with enhanced rate-profiling

Geon Choi is with the Department of Electrical Engineering, POSTECH, Pohang 37673, South Korea (e-mail: simon03062@postech.ac.kr).

Namyoon Lee is with the School of Electrical Engineering, Korea University, Seoul 02841, South Korea (e-mail: namyoon@korea.ac.kr).

methods. Despite these efforts, finding the optimal rate-profiling method and the convolutional pre-transform at various rates and blocklengths remains an open problem.

A significant finding has recently emerged in pre-transformed polar codes, demonstrating that any upper-triangular pre-transformation applied to the polar codes does not reduce the minimum distance of the original polar codes [?], [?]. This finding offers an unified perspective on pre-transformed polar codes, encompassing various instances such as CRC-Aided (CA) polar (CA-polar), PC-polar, and PAC codes, all of which utilize upper-triangular transformation matrices for pre-transformation [?], [?], [?], [?], [?], [?], [?], [?], [?]. This result sheds new light on the efficient construction of pre-transformed polar codes under an upper triangular structure of the precoding matrix. Despite remarkable recent advances in the pre-transformed polar codes, finding the optimal yet pragmatic pre-transformed polar codes remains exceptionally challenging. Such codes require (i) operating at rates close to finite-blocklength capacity, (ii) having low-complexity pre-transformation with an efficient rate-profiling, and (iii) having a low-complexity decoding algorithm.

B. Contributions

In this paper, we put forth a novel pre-transformed polar codes, referred to as *deep polar codes.* The main contributions of this work are summarized as follows:

- We first present a novel successive encoding method for constructing deep polar codes. The deep polar encoder comprises a sequence of $L-1$ multi-layered pre-transformations, each with different sizes, followed by a regular polar transformation in the final layer. In each layer, a part of the information bits is encoded using the polar kernel matrix. The resulting transformed output is then fed into the partial input of the subsequent layer's encoder, along with additional information bits for the following layer encoding. Upon reaching the last layer, the encoder utilizes a regular polar transformation matrix to generate the final codewords. The proposed multi-layered pre-transformation method using the polar kernel matrices enhances the weight distribution of the resulting codes due to their upper triangular structures while enabling low-complexity encoding through a fast polar transformation algorithm.
- We then propose a flexible rate-profiling method for deep polar encoding. By harnessing the polar transform matrix used in each layer encoding, the encoder independently constructs three index sets: i) information, ii) connection, and iii) frozen sets. The information set per layer consists of highly reliable (almost perfectly polarized to the capacity of one) bit-channel indices. Then, the connection set composes of less polarized bit-channel indices while maintaining a pre-defined minimum Hamming distance. The proposed rate-profiling method is flexible with respect to i) the number of layers, ii) code rates, and iii) blocklengths, because of its independent rate-profiling structure over the layers.
- We present some construction examples of deep polar codes under symmetric binary erasure channels (BECs) with a blocklength of 32 and operating at two different rates. Through these examples, we provide an intuitive explanation of how the adoption of partial pre-transformation is sufficient to enhance the weight distribution of polar codes while enabling the design of a low-complexity decoder.
- We introduce two computationally-efficient decoding algorithms for deep polar codes. The first algorithm is called SCL with a backpropagation parity check (SCL-BPC) decoder. The main idea of the SCL-BPC decoder is to leverage bit-wise parity check equations in the reverse process of successive deep polar encoding. This BPC mechanism plays a crucial role in pruning the list that fails BPC equations when performing SCL decoding. We also introduce a low-latency decoding algorithm called parallel-SCL decoder. The main idea of the parallel-SCL decoder is to perform multiple SCL decoders in parallel by treating different hypotheses of connection bit patterns as additional frozen bits.
- We present simulation results evaluating the effectiveness of deep polar codes under BI-AWGN channels. When employing the SCL-BPC decoder with sufficient list sizes, deep polar codes are comparable with the PAC codes; both achieve the normal approximation bound tightly at rates of $R \in \{\frac{29}{128}, \frac{64}{128}\}$. Remarkably, when using a list size of 8 for the SCL-type decoders, which is a more practically relevant scenario, the deep polar codes outperform both PAC and CA-polar codes across various code rates at blocklength of 128. This result advocates their superiority and potential for use in short packet transmissions. Our findings further indicate that deep polar codes exhibit superior decoding performance when employing the parallel-SCL decoder, making them a promising candidate for low-latency applications. Furthermore, we demonstrate that the CRC-aided deep polar (CA-deep polar) code with ML decoding achieves the meta-converse bound within 0.4 dB for blocklengths of 128 and 256 when the information size is 16. These results demonstrate that the CRC precoding can further boost the performance of the deep polar codes, leading to the finite-blocklength capacity closely.

II. Preliminaries

In this section, we describe the preliminaries that are relevant to this work.

A. Channel Coding System

Consider an information vector $\mathbf{d} = [d_1, d_2, \ldots, d_K] \in \{0,1\}^K$, where d_i represents independent and uniformly distributed bits over $\{0,1\}$ for $i \in [K]$. An encoder $\mathcal{E} : \{0,1\}^K \to \mathcal{X}^N$ maps the information vector $\mathbf{d}$ to a codeword $\mathbf{x} = [x_1, x_2 \ldots, x_N]$ of length N, where x_i is the ith element of the codeword, which is chosen from a binary alphabet set $\mathcal{X}$, i.e., $x_i \in \mathcal{X}$. The code rate R is defined as the ratio of transmitted information bits to code blocklength, i.e., $R = \frac{K}{N}$.

Let $W : \mathcal{X} \to \mathcal{Y}$ be a binary input discrete memoryless channel (B-DMC) with output alphabet set $\mathcal{Y}$. The codeword $\mathbf{x}$ is transmitted over this B-DMC, resulting in an output sequence $\mathbf{y} = [y_1, y_2 \ldots, y_N]$ of length N. A decoder $\mathcal{D} : \mathcal{Y}^N \to \{0,1\}^K$ produces an estimate of the message bits $\widehat{\mathbf{d}}$.

Symmetric channel capacity: The symmetric capacity of B-DMC is defined as

$$I(W) = \sum_{x_i \in \{0,1\}} \sum_{y_i \in \mathcal{Y}} \frac{1}{2} W(y_i|x_i) \log_2 \frac{W(y_i|x_i)}{\frac{1}{2}W(y_i|0) + \frac{1}{2}W(y_i|1)}.$$

Further, the cutoff rate is computed as

$$R_0(W) = 1 - \log_2\left(1 + Z(W)\right), \tag{1}$$

where $Z(W)$ is the Bhattacharyya parameter defined as

$$Z(W) = \sum_{y_i \in \mathcal{Y}} \sqrt{W(y_i|0)W(y_i|1)}. \tag{2}$$

Block error probability: The block error rate (BLER) is defined as

$$P(E) = \mathbb{P}[\mathbf{d} \neq \widehat{\mathbf{d}}]. \tag{3}$$

For a linear block code with the minimum distance of $\mathsf{d}^{\min}$, the ML decoding performance over BI-AWGN channel is approximately given by [?]:

$$P_{\mathsf{ML}}(E) \approx A_{\mathsf{d}^{\min}} Q\left(\sqrt{\mathsf{d}^{\min}\mathsf{SNR}}\right), \tag{4}$$

where $A_{\mathsf{d}^{\min}}$ is the number of codewords with the minimum weight (or the nearest neighbors) and SNR is the SNR of BI-AWGN channel, and

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-y^2/2} \mathrm{d}y \tag{5}$$

is the standard Q-function that characterizes the tail probability of a Gaussian random variable. The two parameters, $\mathsf{d}^{\min}$ and $A_{\mathsf{d}^{\min}}$, play a crucial role in determining the performance of a code under ML decoding.

B. Polar Codes

A polar code with parameters $(N, K, \mathcal{I})$ is characterized by a polar transform matrix of size $N = 2^n$ and an index set $\mathcal{I} \subseteq [N]$. The polar transform matrix of size $N = 2^n$ is obtained through the nth Kronecker power of a binary kernel matrix $\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ as

$$\mathbf{G}_N = \mathbf{G}_2^{\otimes n}. \tag{6}$$

The input vector of the encoder, $\mathbf{u} = [u_1, u_2, \ldots, u_N] \in \mathbb{F}_2^N$, is generated based on the given information set $\mathcal{I}$. In this process, the data vector carrying K information bits, $\mathbf{d}$, is allocated to $\mathbf{u}_{\mathcal{I}}$. The remaining elements of $\mathbf{u}$, $\mathbf{u}_{\mathcal{I}^c}$, are assigned to zeros. Here, $\mathcal{I}^c = [N]/\mathcal{I}$ is referred to as the frozen bit set. This data assignment procedure is commonly known as rate-profiling. Finally, a polar codeword is constructed by multiplying $\mathbf{u}$ with $\mathbf{G}_N$ as

$$\mathbf{x}^{\mathsf{Polar}} = \mathbf{u}\mathbf{G}_N = \sum_{i \in \mathcal{I}} u_i \mathbf{g}_{N,i}, \tag{7}$$

where $\mathbf{g}_{N,i}$ is the ith row vector of $\mathbf{G}_N$. According to the channel combining and splitting principle [?], the ith bit-channel $W_N^{(i)} : \mathcal{X} \to \mathcal{Y}^N \times \mathcal{X}^{i-1}$, where $i \in [N]$, is defined as follows:

$$W_N^{(i)}\left(\mathbf{y}, \mathbf{u}_{1:i-1}|u_i\right) = \sum_{\mathbf{u}_{i+1:N} \in \mathbb{F}_2^{N-i}} \frac{1}{2^{N-i}} W^N\left(\mathbf{y}|\mathbf{x}\right), \tag{8}$$

where $W^N\left(\mathbf{y}|\mathbf{x}\right)$ is the N copies of B-DMCs and $\mathbf{u}_{a:b} = [u_a, u_{a+1}, \ldots, u_b]$ for $a, b \in [N]$ and $a < b$. When N is sufficiently large enough, the optimal rate-profiling is to select the indices having the capacity closed to one as

$$\mathcal{I} = \left\{ i \in [N] : I\left(W_N^{(i)}\right) = 1 - \epsilon \right\}, \tag{9}$$

for small $\epsilon > 0$. This rate-profiling is sufficient to achieve the capacity under simple SC decoding [?]. For a short blocklength regime, the information set consists of the indices that provide the K most reliable bit-channels or the K least bit-channel Bhattacharyya values.

C. RM Codes

Similar to polar codes, RM codes are constructed using $\mathbf{G}_N$, but with a different selection of rows. Instead of choosing rows based on the most reliable bit-channel capacity, RM codes simply select rows with the large weights. Let $\mathsf{wt}\left(\mathbf{g}_{N,i}\right)$ represent the Hamming weight of the ith row of $\mathbf{G}_N$, where $i \in [N]$. For the rth-order RM code with length, $N = 2^m$ and $0 \leq r \leq m$, the information set is constructed by choosing the rows with a Hamming weight greater than or equal to 2^{m-r} as

$$\mathcal{I}_{\mathsf{RM}} = \{i \in [n] : \mathsf{wt}\left(\mathbf{g}_{N,i}\right) \geq 2^{m-r}\}. \tag{10}$$

The encoder input vector $\mathbf{u} \in \mathbb{F}_2^N$ is formed by assigning the information vector $\mathbf{d}$ to $\mathbf{u}_{\mathcal{I}_{\mathsf{RM}}}$, while the remaining inputs $u_i = 0$ for $i \notin \mathcal{I}_{\mathsf{RM}}$. Subsequently, a RM codeword is generated as

$$\mathbf{x}^{\mathsf{RM}} = \mathbf{u}\mathbf{G}_N = \sum_{i \in \mathcal{I}_{\mathsf{RM}}} u_i \mathbf{g}_{N,i}. \tag{11}$$

The RM code has a minimum distance of 2^{m-r} [?].

D. Pre-transformed Polar Codes

Polar and RM codes can be enhanced for error correction by utilizing a pre-transformation technique. A $(N, K, \mathcal{I}, \mathbf{T})$ pre-transformed polar code consists of a binary pre-transformation matrix $\mathbf{T} \in \mathbb{F}_2^{N \times N}$ and an information set $\mathcal{I}$ for rate-profiling. The construction of pre-transformed polar codes involves a two-stage encoding process. In the first stage, the information vector $\mathbf{d} \in \mathbb{F}^K$, carrying K information bits, is inserted into the input vector $\mathbf{v} \in \mathbb{F}^N$ of the pre-transformation matrix. After assigning $\mathbf{v}_{\mathcal{I}} = \mathbf{d}$ and $\mathbf{v}_{\mathcal{I}^c} = \mathbf{0}$, the first-stage encoding is performed by multiplying $\mathbf{v}$ with $\mathbf{T}$ as

$$\mathbf{u} = \mathbf{v}\mathbf{T}. \tag{12}$$

In the second stage encoding, the codeword $\mathbf{x}$ is generated by transforming the output of the first-stage encoding using $\mathbf{G}_N$ as

$$\mathbf{x} = \mathbf{u}\mathbf{G}_N = \mathbf{v}\mathbf{T}\mathbf{G}_N. \tag{13}$$

The pre-transform matrix $\mathbf{T}$ and the rate-profiling index set $\mathcal{I}$ are required to be jointly optimized to enhance the decoding performance of finite-length polar codes. In a recent study **[?]**, it was demonstrated that using an upper-triangular matrix $\mathbf{T} \in \mathbb{F}_2^{N \times N}$ with non-zero diagonal elements for the pre-transform guarantees to generate codewords with a minimum distance at least as large as that of polar codes (i.e., $\mathbf{T} = \mathbf{I}$). The PAC codes are representative examples of such pre-transformed polar codes, utilizing the upper-triangular Toeplitz matrix $\mathbf{T}$ as a pre-transformed matrix involving a convolution operation. However, determining the optimal information set $\mathcal{I}$ for the given $\mathbf{T}$ remains an unresolved challenge.

III. deep polar Codes

In this section, we present deep polar codes, a family of pre-transformed polar codes.

A. Encoding

A $\left(N, K, \{\mathcal{I}_\ell\}_{\ell=1}^L, \{\mathcal{A}_\ell\}_{\ell=1}^L, \{\mathbf{T}_\ell\}_{\ell=1}^L\right)$ deep polar code is defined with the following parameters:

- i) L transformation matrices $\mathbf{T}_\ell \in \mathbb{F}_2^{N_\ell \times N_\ell}$,
- ii) L information sets $\{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_L\}$, and
- iii) L connection sets $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_L\}$.

For given these parameters, the deep polar encoder performs information bit splitting and successive encoding.

Information bit splitting and mapping: The information vector $\mathbf{d} \in \mathbb{F}_2^K$ carrying K information bits is splitted into L information sub-vectors $\mathbf{d}_\ell$, each with size of $K_\ell = |\mathcal{I}_\ell| (< N_\ell)$ for $\ell \in [L]$ and $\sum_{\ell=1}^L K_\ell = K$.

Let $\mathbf{u}_\ell = [u_{\ell,1}, u_{\ell,2}, \ldots, u_{\ell,N_\ell}] \in \mathbb{F}_2^{N_\ell}$ be the input vector of layer ℓ with length N_ℓ for $\ell \in [L]$. The index set of the ℓth layer is partitioned into three non-overlapped sub-index sets as

$$[N_\ell] = \mathcal{I}_\ell \cup \mathcal{A}_\ell \cup \mathcal{F}_\ell, \tag{14}$$

where $\mathcal{F}_\ell = [N_\ell]/\{\mathcal{I}_\ell \cup \mathcal{A}_\ell\}$ is the frozen bit set of layer ℓ and $\mathcal{I}_\ell \cap \mathcal{A}_\ell = \phi$. The information vector of the ℓth layer, $\mathbf{d}_\ell \in \mathbb{F}_2^{K_\ell}$, is assigned to $\mathbf{u}_{\mathcal{I}_\ell}$. Meanwhile, the frozen bits are assigned to $\mathbf{u}_{\mathcal{F}_\ell} = \mathbf{0}$, where $\mathcal{F}_\ell = [N_\ell]/\{\mathcal{I}_\ell \cup \mathcal{A}_\ell\}$.

Successive encoding: As depicted in Fig. **??**, an L-layered deep polar code is constructed by L-stage successive encoding procedures. Let $\mathbf{T}_\ell \in \mathbb{F}_2^{N_\ell \times N_\ell}$ be the ℓth layered pre-transformation matrix where $N_1 < N_2 < \cdots < N_{L-1} < N_L$. Unlike the PAC or other PAC-like codes, our deep polar codes adopt the pre-transform matrix by multiplying the transpose of the polar transform matrix with length $N_\ell = 2^{n_\ell}$ for $n_\ell \in \mathbb{Z}^+$ as

$$\mathbf{T}_\ell = \mathbf{G}_{N_\ell}^\top. \tag{15}$$

The primary feature is that the proposed pre-transformation matrix $\mathbf{T}_\ell = \mathbf{G}_{N_\ell}^\top$ not only facilitates easy calculations using fast polar transform but also has an upper triangular matrix structure. This upper triangular structure allows for improvement in the weight distribution of the code.

In the first layer, the encoder generates the input vector of the first layer encoding, $\mathbf{u}_1 = [\mathbf{u}_{\mathcal{I}_1},\ \mathbf{u}_{\mathcal{F}_1}]$ where $\mathcal{A}_1 = \phi$. The encoder output of the first layer is generated by $\mathbf{G}_{N_1}^\top$ as

$$\mathbf{v}_1 = \mathbf{u}_1 \mathbf{G}_{N_1}^\top. \tag{16}$$

In the second layer, the output vector of the first layer encoding is assigned to the input of the second layer to the connection index set $\mathcal{A}_2$, i.e., $\mathbf{u}_{\mathcal{A}_2} = \mathbf{v}_1$. Since $\mathbf{u}_{\mathcal{I}_2} = \mathbf{d}_2$ and $\mathbf{u}_{\mathcal{F}_2} = \mathbf{0}$, the input vector of the second layer encoding is

$$\mathbf{u}_2 = [\mathbf{u}_{\mathcal{I}_2},\ \mathbf{u}_{\mathcal{A}_2},\ \mathbf{u}_{\mathcal{F}_2}]. \tag{17}$$

By multiplying $\mathbf{u}_2$ with $\mathbf{G}_{N_2}^\top$, the output vector of the second layer encoding is obtained as

$$\mathbf{v}_2 = \mathbf{u}_2 \mathbf{G}_{N_2}^\top. \tag{18}$$

Similar to the second layer encoding, the ℓth layer encoder for $2 < \ell \leq L$ takes the input vector

$$\mathbf{u}_\ell = [\mathbf{u}_{\mathcal{I}_\ell},\ \mathbf{u}_{\mathcal{A}_\ell},\ \mathbf{u}_{\mathcal{F}_\ell}], \tag{19}$$

where $\mathbf{u}_{\mathcal{A}_\ell} = \mathbf{u}_{\ell-1}\mathbf{G}_{N_{\ell-1}}^\top$, and generates the corresponding output vector by multiplying $\mathbf{G}_{N_\ell}^\top$ or $\mathbf{G}_{N_L}$ as

$$\mathbf{v}_\ell = \begin{cases} \mathbf{u}_\ell \mathbf{G}_{N_\ell}^\top & 2 < \ell < L, \\ \mathbf{u}_L \mathbf{G}_{N_L} & \ell = L. \end{cases} \tag{20}$$

For notational simplicity, we denote the output of the last layer encoder as the channel input $\mathbf{x} = \mathbf{v}_L \in \mathbb{F}_2^{N_L}$.

B. Rate-Profiling

The selection of information and connection sets in all layers, $\mathcal{I}_\ell$ and $\mathcal{A}_\ell$, determines the error-correcting performance of deep polar codes. Unfortunately, finding the optimal set selection algorithm is very challenging. In this paper, we propose a flexible rate-profiling method

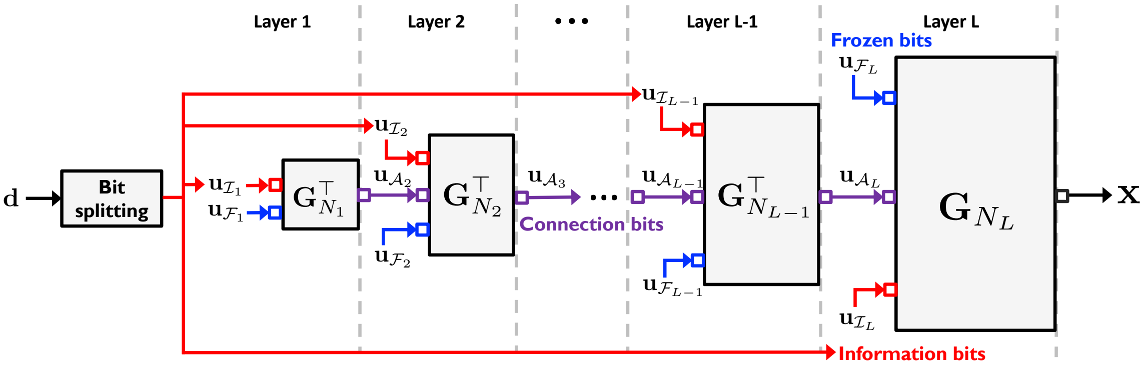

Layer 1
Layer 2
• • •
Layer L-1
Layer L
Frozen bits
$\mathbf{u}_{\mathcal{F}_L}$
$\mathbf{u}_{\mathcal{I}_{L-1}}$
$\mathbf{u}_{\mathcal{I}_2}$
$\mathbf{d}$
Bit splitting
$\mathbf{u}_{\mathcal{I}_1}$
$\mathbf{u}_{\mathcal{F}_1}$
$\mathbf{G}_{N_1}^{\top}$
$\mathbf{u}_{\mathcal{A}_2}$
$\mathbf{G}_{N_2}^{\top}$
$\mathbf{u}_{\mathcal{A}_3}$
• • •
Connection bits
$\mathbf{u}_{\mathcal{A}_{L-1}}$
$\mathbf{G}_{N_{L-1}}^{\top}$
$\mathbf{u}_{\mathcal{A}_L}$
$\mathbf{G}_{N_L}$
$\mathbf{x}$
$\mathbf{u}_{\mathcal{F}_2}$
$\mathbf{u}_{\mathcal{F}_{L-1}}$
$\mathbf{u}_{\mathcal{I}_L}$
Information bits

Fig. 1. An illustration of the proposed deep polar encoder with L layers.

that can support versatile code rates while guaranteeing an improved code weight distribution. The proposed rate-profiling method is to construct $\mathcal{I}_\ell$ and $\mathcal{A}_\ell$ for $\ell \in [L]$ individually across the layers.

Selection of $\mathcal{I}_\ell$ and $\mathcal{A}_\ell$: We explain how to select information and connection sets for the ℓth layer for $\ell \in \{1,\ldots,L\}$. To construct $\mathcal{I}_\ell$ and $\mathcal{A}_\ell$ independently over the layers, suppose the channel input $\mathbf{x}_\ell \in \{0,1\}^{N_\ell}$ is formed by the transpose of polar transform $\mathbf{x}_\ell = \mathbf{u}_\ell \mathbf{G}_{N_\ell}^{\top}$ for $\ell \in \{1,2,\ldots,L-1\}$ and the forward polar transform $\mathbf{x}_L = \mathbf{u}_L \mathbf{G}_{N_L}$ for the last layer. Then, the codeword generated by layer ℓ encoder, $\mathbf{x}_\ell$, is assumed to be transmitted over an B-DMC, $W : \mathcal{X} \rightarrow \mathcal{Y}$, which produces the channel output $\mathbf{y}_\ell$ for $\ell \in [L]$. The ith bit-channel when using layer ℓ encoder is defined as

$$W_{N_\ell}^{(i)}\left(\mathbf{y}_\ell, \mathbf{u}_{\ell,1:i-1} | u_{\ell,i}\right) = \sum_{\mathbf{u}_{\ell,i+1:N} \in \mathbb{F}_2^{N_\ell - i}} \frac{1}{2^{N_\ell - i}} W^{N_\ell}\left(\mathbf{y}_\ell | \mathbf{x}_\ell\right), \tag{21}$$

where $i \in [N_\ell]$, $\mathbf{u}_{\ell,a:b} = [u_{\ell,a}, u_{\ell,a+1}, \ldots, u_{\ell,b}]$ for $a, b \in [N_\ell]$ and $a < b$. Let $I\left(W_{N_\ell}^{(i)}\right)$ be the ith bit-channel capacity.

To construct $\mathcal{I}_\ell$ and $\mathcal{A}_\ell$, we first define an index set $\mathcal{R}_\ell$ according to the RM rate-profiling by selecting $i \in [N_\ell]$ such that the weight of the rows $\mathbf{G}_{N_\ell}$ is equal to or larger than the pre-determined $\mathsf{d}_\ell^{\min}$, i.e.,

$$\mathcal{R}_\ell = \{i \in [N_\ell] : \mathsf{wt}\left(\mathbf{g}_{N_\ell,i}\right) \geq \mathsf{d}_\ell^{\min}\}, \tag{22}$$

where $\mathsf{d}_\ell^{\min}$ is a target minimum distance for the ℓth layer encoding. Next, we define an ordered index set of $\mathcal{R}_\ell$ as

$$\bar{\mathcal{R}}_\ell = \left\{i_1, i_2, \ldots, i_{|\mathcal{R}_\ell|}\right\}, \tag{23}$$

where i_1 is the index of the most reliable synthetic channel, i.e., $I\left(W_{N_\ell}^{(i_1)}\right) \geq I\left(W_{N_\ell}^{(i_2)}\right) \geq \cdots \geq I\left(W_{N_\ell}^{(i_{|\mathcal{R}_\ell|})}\right)$. This ordered set can be constructed using the ordered Bhattacharyya values, i.e., $Z\left(W_{N_\ell}^{(i_1)}\right) \leq Z\left(W_{N_\ell}^{(i_2)}\right), \ldots \leq Z\left(W_{N_\ell}^{(i_{|\mathcal{B}_\ell|})}\right)$ or the values obtained from the density evolution with Gaussian approximation technique in [?], [?]. Using the ordered index set, the information set $\mathcal{I}_\ell$ is generated by selecting the bit-channel indices that are approximately polarized to the capacity of one for given code length N_ℓ:

$$\mathcal{I}_\ell = \left\{i \in \bar{\mathcal{R}}_\ell : I\left(W_{N_\ell}^{(i)}\right) \geq 1 - \delta_\ell\right\}, \tag{24}$$

where $\delta_\ell > 0$ is chosen arbitrary small and $|\mathcal{I}_\ell| = K_\ell$.

Next, we construct the connection set $\mathcal{A}_\ell$ as a subset of $\bar{\mathcal{R}}_\ell / \mathcal{I}_\ell$. Let $j_1, j_2, \ldots, j_{N_{\ell-1}} \in \mathcal{R}_\ell / \mathcal{I}_\ell$ be the indices that provide the highest $N_{\ell-1}$ bit-channel capacities in $\mathcal{R}_\ell / \mathcal{I}_\ell$ such that $I\left(W_{N_\ell}^{(j_1)}\right) \geq I\left(W_{N_\ell}^{(j_2)}\right) \geq \cdots \geq I\left(W_{N_\ell}^{(j_{N_{\ell-1}})}\right)$. Then,

$$\mathcal{A}_\ell = \left\{j_1, j_2, \ldots, j_{N_{\ell-1}}\right\}. \tag{25}$$

The frozen set of layer ℓ is defined as the collection of indices that are excluded in both the information and connection sets as

$$\mathcal{F}_\ell = [N_\ell] / \{\mathcal{I}_\ell \cup \mathcal{A}_\ell\}. \tag{26}$$

This rate-profiling method is performed independently over the layers.

C. Remarks

We provide some remarks on the encoding complexity, the superposition property, the minimum distance of the code, the concatenation with CRC codes, and the rate-compatibility.

Encoding complexity: The proposed successive encoder composed of L layers requires to take L polar transformations, each with N_ℓ size. Since the computation of the polar transform with size N_ℓ needs the complexity of $\mathcal{O}(N_\ell \log_2 N_\ell)$, the total encoding complexity boils down to

$$\mathcal{O}\left(\sum_{\ell=1}^{L} N_\ell \log_2 N_\ell\right). \tag{27}$$

It is worth mentioning that the encoding complexity can be comparable to that of the standard polar codes when N_L is sufficiently larger than N_ℓ for $\ell \in [L-1]$. Choosing a small size of N_ℓ for $\ell \in [L-1]$ is a practical and effective strategy for reducing the encoding complexity.

Superposition codes : The deep polar code can be viewed with a lens through a superposition code of both the polar and the transformed polar codes as

$$\mathbf{x} = \underbrace{\sum_{j \in \mathcal{I}_L} u_{L,j}\mathbf{g}_{N_L,j}}_{\mathsf{x}_{\mathsf{P}}} + \underbrace{\sum_{i \in \mathcal{A}_L} u_{L,i}\mathbf{g}_{N_L,i}}_{\mathsf{x}_{\mathsf{TP}}}, \tag{28}$$

where x_{P} and x_{TP} represent polar and pre-transformed polar subcodewords, respectively. This superposition code interpretation provides a useful guideline for designing a low-complexity decoder while improving the weight distribution. Specifically, the pre-transformed subcodewords play a crucial role in improving the weight distribution of the deep polar code. The polar subcodewords facilitate to use a simple SC decoding because the information bits $\mathbf{u}_{\mathcal{I}_L}$ are sent through almost perfectly polarized bit-channels when the connection bits $\mathbf{u}_{\mathcal{A}_L}$ are treating as additional frozen bits. As a result, the deep polar codes offer a desirable balance between code performance and decoding complexity by strategically allocating information bit sizes between $\mathbf{u}_{\mathcal{I}_L}$ and $\mathbf{u}_{\mathcal{A}_L}$. For instance, when the blocklength goes to infinity, the encoder allocates all information bits to $\mathbf{u}_{\mathcal{I}_L}$ while $\mathcal{A}_L = \phi$; the deep polar codes boil down to a standard polar code.

The minimum distance and weight distribution: The minimum distance of the deep polar codes is larger or equal to the minimum distance of layer L, $\mathsf{d}_L^{\min}$. By the construction, the encoder chooses the row vectors of $\mathbf{G}_{N_L}$ with the weight larger than $\mathsf{d}_L^{\min}$, namely,

$$\mathsf{wt}\left(\mathbf{g}_{N_L,j}\right) \geq \mathsf{d}_L^{\min} \tag{29}$$

for $j \in \mathcal{I}_L \cup \mathcal{A}_L$. Since the pre-transform matrix of layer $L-1$ holds the upper triangular structure, the minimum distance of the pre-transformed subcodewords $\mathsf{x}_{\mathsf{TP}} = \sum_{i \in \mathcal{A}_L} u_{L,i}\mathbf{g}_{N_L,i}$ cannot be less than $\mathsf{d}_L^{\min}$.

CRC-aided (CA) deep polar codes: One possible extension of deep polar codes is to concatenate them with CRC codes. In this approach, the K-bit information vector $\mathbf{d}$ is precoded by using CRC generator polynomials, adding CRC bits. The length of the resulting information sequence becomes $K + K_{\mathrm{CRC}}$, where K_{CRC} represents the number of CRC bits. Alternatively, multiple CRC bits, each with short length, can be appended to the information bits of each layer, i.e., $\mathbf{u}_{\mathcal{I}_\ell}$ for $\ell \in [L]$ to improve the decoding performance. The effect of CRC code concatenation with the deep polar codes will be discussed in the simulation section.

Rate-compatibility: A design of rate-compatible codes suitable for hybrid automatic repeat request (HARQ) is an important feature for practical communication systems. Our deep polar code possesses a rate-compatible property, thanks to its multi-layered successive encoding structure. Suppose a transmitter sends a codeword $\mathbf{x}$ and a receiver fails to decode the message bits $\mathbf{d}$. Then, in the next round of the transmission, the transmitter sends only the output of $L-1$ layer, $\mathbf{u}_{\mathcal{A}_L}$, with blocklength of N_{L-1} for enabling HARQ communications. The receiver attempts to decode the message by using both the noisy connection bits received in the second round and the received signal in the first round. This re-transmission and decoding protocol can be iteratively applied to the first layer, enhancing the decoding performance while reducing the code rates from $R = \frac{K}{N_L}$ to $R = \frac{K}{\sum_{\ell=1}^{L} N_\ell}$. Notwithstanding the flexible rate-compatibility, a delicate code optimization is required to attain a high HAQR performance by carefully choosing the blocklengths N_ℓ and information bits K_ℓ for each layer $\ell \in \{1, 2, \ldots, L-1\}$. Solving this optimization problem remains as a promising future topic.

IV. Examples

In this section, we provide some examples of the design of deep polar codes in a short blocklength regime to better understand the proposed encoding method. Throughout the examples, we shall focus on a short packet transmission scenario with a length of 32 over a BEC with an erasure probability of 0.5, i.e., $I(W) = 0.5$ at two different rates $R \in \{\frac{11}{32}, \frac{15}{32}\}$.

A. Channel Polarization and Row Weights of $\mathbf{G}_{32}$

For a deep polar code construction, it is important to understand the polarized bit-channel capacities and the row weight of $\mathbf{G}_{32}$. As shown in Fig. **??**, the red circles indicate the bit-channel capacities for the BEC with $I(W) = 0.5$, i.e., $I\left(W_{32}^{(i)}\right)$ for $i \in [32]$. The blue cross indicates the normalized weight of $\mathbf{G}_{32}$ according to index $i \in [32]$, i.e., $\frac{\mathsf{wt}(\mathbf{g}_{32,i})}{32}$. The ordered index sets are mismatched according to bit-channel capacities and the normalized weights. For instance, $I\left(W_{32}^{(25)}\right) > I\left(W_{32}^{(12)}\right)$ but $\frac{\mathsf{wt}(\mathbf{g}_{32,25})}{32} < \frac{\mathsf{wt}(\mathbf{g}_{32,12})}{32}$. When including the bit-index 25 in the information set to facilitate SC decoding, the minimum distance of the codewords is necessarily reduced. It becomes crucial to carefully select the bit-indices to improve the coding performance, considering both their weights and the bit-channel capacities.

B. Example 1

This example explains how to construct a two-layered deep polar code with a code rate of $R = \frac{11}{32}$. The encoding process involves utilizing two transformation matrices: $\mathbf{G}_8^\top$ for layer 1 and $\mathbf{G}_{32}$ for layer 2. Assuming that a target minimum distance of the second layer is $\mathsf{d}_2^{\min} = 8$, we select the indices of rows in the matrix $\mathbf{G}_{32}$ whose weights are greater than or equal to $\mathsf{d}_2^{\min} = 8$, i.e.,

$$\mathcal{R}_2 = \{8, 12, 14, 15, 16, 20, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32\}.$$

Then, from Fig. **??**, the information set for the second layer, a subset of $\mathcal{R}_2$, is chosen as

$$\begin{aligned}\mathcal{I}_2 &= \left\{i \in \mathcal{R}_2 : I\left(W_{32}^{(i)}\right) \geq 0.98\right\} \\ &= \{32, 31, 30, 28, 24, 16, 29\},\end{aligned}$$

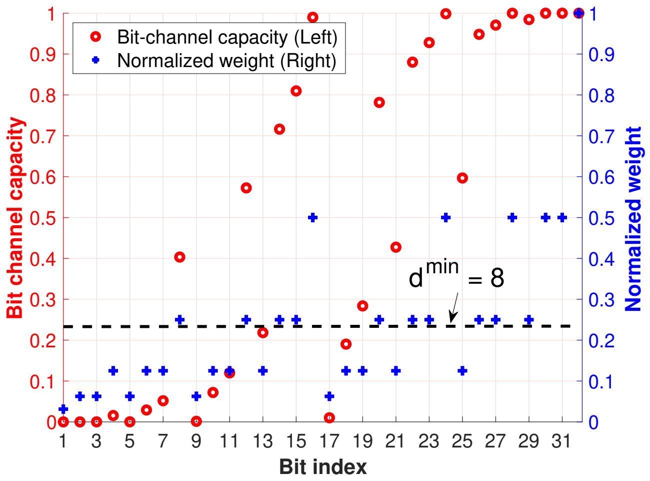


Fig. 2. Polarized bit-channel capacity (left) and normalized weight (right) for the BEC with $I(W) = 0.5$.

TABLE I
COMPARISON OF THE WEIGHT DISTRIBUTIONS

	[32, 11]			[32, 15]		
Weight	Polar	RM-type	Proposed	Polar	RM-type	Proposed
0	1	1	1	1	1	1
4	0	0	0	8	0	0
8	76	40	20	444	364	300
12	192	336	416	6328	6720	6976
16	1510	1294	1174	19206	18598	18214
20	192	336	416	6328	6720	6976
24	76	40	20	444	364	300
28	0	0	0	8	0	0
32	1	1	1	1	1	1

where $|\mathcal{I}_2| = K_2 = 7$. Considering that $N_1 = 8$, we identify the eight bit-channel indices that yield the highest capacity in $\mathcal{R}_2/\mathcal{I}_2$ to form the connection set for the second layer as

$$\mathcal{A}_2 = \{27, 26, 23, 22, 15, 20, 14, 12\}.$$

It is worth mentioning that the bit-channel index of 25 was excluded in the connection set although its capacity is larger than that of the index of 12, i.e., $I\left(W_{32}^{(25)}\right) \geq I\left(W_{32}^{(12)}\right)$. This is because $\mathsf{wt}\left(\mathbf{g}_{32,25}\right) = 4 < \mathsf{wt}\left(\mathbf{g}_{32,12}\right) = 8$. The frozen set of the second layer becomes

$$\mathcal{F}_2 = \{1, 2, \ldots, 32\}/\{\mathcal{I}_2 \cup \mathcal{A}_2\}.$$

In the first layer, we identify the four indices that result in the highest bit-channel capacity using the polar transform matrix $\mathbf{G}_8^{\top}$, while ensuring that the minimum distance $\mathsf{d}_1^{\min}$ is greater than or equal to 4. The corresponding information and frozen sets are chosen as $\mathcal{I}_1 = \{1, 2, 3, 5\}$ and $\mathcal{F}_1 = \{8, 7, 6, 4\}$, respectively. The output vector of the first layer, denoted as $\mathbf{v}_1 = \mathbf{u}_1\mathbf{G}_8^{\top}$, is then connected to the input of connection set $\mathbf{u}_{\mathcal{A}_2} = \mathbf{v}_1$.

C. Example 2

We present an additional example to construct an $\left(32, 15, \{\mathcal{I}_1, \mathcal{I}_2\}, \{\phi, \mathcal{A}_2\}, \{\mathbf{G}_4^{\top}, \mathbf{G}_{32}\}\right)$ deep polar code. Applying the same approach as in Example 1, we first select the indices corresponding to the row vectors of $\mathbf{G}_{32}$ with a weight greater than 8 in an identical manner as

$$\mathcal{R}_2 = \{8, 12, 14, 15, 16, 20, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32\}.$$

We adopt the deep polar rate-profiling method by selecting the information and connection sets as

$$\mathcal{I}_2 = \{15, 16, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32\}$$

with $K_2 = |\mathcal{I}_2| = 12$ and

$$\mathcal{A}_2 = \{8, 12, 14, 20\}.$$

Then, we choose the information set for the first layer as

$$\mathcal{I}_1 = \{1, 2, 3\}$$

with $K_1 = |\mathcal{I}_1| = 3$.

D. Comparisons with RM-type and Polar Codes

We compare our deep polar codes in Examples 1 and 2 with existing RM and polar codes. We generate the polar codes for fair comparison by selecting the top $K \in \{11, 15\}$ bit-channel indices that provide the highest capacities. For the RM code construction with information bit size $K \in \{11, 15\}$, we consider a subcodeword set of a $[32, 16]$ RM code. Since the information set of the $[32, 16]$ RM code is

$$\mathcal{I}_{\mathsf{RM}}^{16} = \{i : \mathsf{wt}(\mathbf{g}_{32,i}) \geq 8\}, \tag{30}$$

we choose the information set for $K = 15$ as a subset of $\mathcal{I}_{\mathsf{RM}}^{16}$, i.e., $\mathcal{I}_{\mathsf{RM}}^{15} \subseteq \mathcal{I}_{\mathsf{RM}}^{16}$. In particular, to optimize the code performance, we evaluate the weight distributions of 16 possible sub-codebooks of the $[32, 16]$-RM code and select the best one with the smallest number of the codewords with the minimum weights.

Weight distribution: As shown in Table **??**, the proposed deep polar codes provide superior weight distributions than those of the RM and polar codes in both code rates. Specifically, deep polar codes have fewer codewords having the minimum weight than the RM codes while keeping the identical minimum distance of 8 in both code rates. In addition, the deep polar code with a rate of $R = \frac{15}{32}$ can improve both the minimum distance and the number of codewords having the smallest weights.

BLER performance: To demonstrate the effect of the weight distribution improvement in the code design, we plot BLERs for the three codes under ML decoding as increasing the erasure probabilities of the BEC. As illustrated in Fig. **??**, the proposed deep polar provides noticeable BLER improvement compared to the RM and polar codes when $K = 11$. This BLER gain stems from the considerable reduction of the codewords with the minimum weight. When $K = 15$, the minimum distance of deep polar and RM codes is larger than the polar code, which

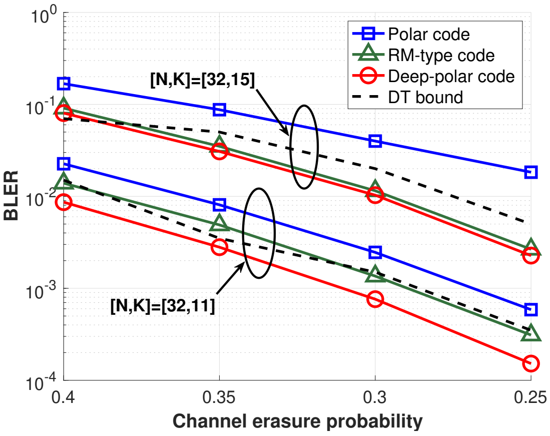


Fig. 3. BLER performance comparison for the BEC channel with $N = 32$ and $K \in \{11, 15\}$.

leads to improved BELR performance. Although the deep polar and RM codes have identical minimum distances, the deep polar code slightly outperforms the RM code by reducing the codewords with the minimum weight, as shown in Table **??**. One remarkable result is that the deep polar codes can achieve better BLER performances than the dependence-testing (DT) bound, one of the strongest achievability bounds for the BEC in the finite blocklength regime.

Pre-transform using a sub-upper triangular matrix: In contrast to conventional pre-transformed polar codes that utilize an upper triangular matrix for pre-transformation, our resulting pre-transformation matrix across multiple layers can be seen as a sub-matrix of an upper triangular matrix. For instance, we express our two-layered encoding structure with a unified pre-transform matrix $\mathbf{T} \in \mathbb{F}_2^{32\times 32}$ as follows:

$$[\mathbf{u}_{\mathcal{F}_2}, \mathbf{u}_1, \mathbf{u}_{\mathcal{I}_2}] \underbrace{\begin{bmatrix} \mathbf{0}_{17} & \mathbf{0}_{17} & \mathbf{0}_{17} \\ \mathbf{0}_{8} & \mathbf{G}_8^{\top} & \mathbf{0}_{8} \\ \mathbf{0}_{7} & \mathbf{0}_{7} & \mathbf{I}_{7} \end{bmatrix}}_{\mathbf{T}} \mathbf{G}_{32} = \mathbf{x}. \tag{31}$$

It is evident that the resulting pre-transformed matrix $\mathbf{T}$ does not exhibit an upper diagonal structure; instead, a sub-matrix of $\mathbf{T}$ takes the form of an upper triangular matrix. This condition is less strict than the PAC and the existing pre-transformed polar codes, where the entire $\mathbf{T}$ matrix must be Toeplitz or upper triangular.

V. deep polar Decoders

In this section, we present two decoding algorithms for deep polar codes. The first decoding method is SCL with backpropagation parity check (SCL-BPC), which provides flexibility in balancing complexity and performance through list size control. Motivated by the superposition code interpretation, the second algorithm is the parallel-SCL, which aims to reduce the decoding latency at the cost of increased hardware complexity.

A. SCL-BPC Decoder

The central concept behind SCL-BPC is to efficiently prune decoding paths that fail to satisfy the successive parity check equations within the deep polar encoder during the SCL decoding process. As a stepping stone toward understanding the overall SCL-BPC decoder, it is instructive to explain the BPC mechanism used in decoding, which is the key ingredient of the proposed decoder. The key concept behind the BPC mechanism is to leverage the reverse process of the deep polar encoder. From (**??**), the deep polar encoder generates the connection bits of layer $\ell \in [L]$, $\mathbf{u}_{\mathcal{A}_\ell}$, using the input of encoder at layer $\ell - 1$ and $\mathbf{G}_{N_{\ell-1}}^{\top}$ as

$$\mathbf{u}_{\mathcal{A}_\ell} = [\mathbf{u}_{\mathcal{I}_{\ell-1}},\ \mathbf{u}_{\mathcal{A}_{\ell-1}},\ \mathbf{u}_{\mathcal{F}_{\ell-1}}]\mathbf{G}_{N_{\ell-1}}^{\top}. \tag{32}$$

From this successive encoder structure, we know that if $\mathbf{u}_{\mathcal{A}_\ell}$ is successfully decoded, the reverse encoding using $\mathbf{T}_{\ell-1}^{-1} = \mathbf{G}_{N_{\ell-1}}^{\top}$ ensures to produce the frozen bits of the previous layer:

$$\mathbf{u}_{\mathcal{F}_{\ell-1}} = \left(\mathbf{u}_{\mathcal{A}_\ell}\mathbf{G}_{N_{\ell-1}}^{\top}\right)_{\mathcal{F}_{\ell-1}}, \tag{33}$$

for $\ell \in \{2, 3, \ldots, L\}$. Precisely estimating $\hat{\mathbf{u}}_{\mathcal{A}_L}$ is crucial for effective decoding using the BPC mechanism. However, accurately estimating the connection bits in layer L presents a significant challenge due to their transmission over less reliable bit-channels than the information bits during SCL decoding. Consequently, incorrect estimation of $\hat{\mathbf{u}}_{\mathcal{A}_L}$ can lead to a degradation in the performance of SCL decoding, mainly when the list size is insufficiently large.

We propose an efficient SCL-BPC decoder by leveraging a bit-wise BPC mechanism to improve the decoding performance. The major idea is to verify the backpropagation syndrome check condition at the bit level within each layer, specifically for the elements denoted as $u_{L,i}$, where i belongs to the set $\mathcal{A}_L$. To simplify our notation, we define $\mathbf{G}_{N_\ell,1:k}^{\top} \in \mathbb{F}_2^{k\times k}$ as the upper-left submatrix of $\mathbf{G}_{N_\ell}^{\top}$. It is important to note that our pre-transform technique using the polar transform matrix possesses a unique property: its inverse is equal to itself. As a result, the inverse pre-transformation matrix is simply calculated itself as

$$\left(\mathbf{G}_{N_\ell,1:k}^{\top}\right)^{-1} = \mathbf{G}_{N_\ell,1:k}^{\top},\ \forall \ell \in [L-1], \forall k \in [K]. \tag{34}$$

If we denote the connection bits of layer ℓ as $\hat{\mathbf{u}}_{\mathcal{A}_\ell} = [\hat{u}_{\ell,i_1}, \hat{u}_{\ell,i_2}, \ldots, \hat{u}_{\ell,i_{N_{\ell-1}}}]$ such that $i_1 < i_2 < \cdots < i_{N_{\ell-1}}$, we express

$$\hat{\mathbf{u}}_{\ell-1,1:k} = \hat{\mathbf{u}}_{\mathcal{A}_\ell,1:k}\mathbf{G}_{N_{\ell-1},1:k}^{\top}, \tag{35}$$

where $\hat{\mathbf{u}}_{\ell-1,1:k}$ is a subsequence that comprises the first k elements of $\hat{\mathbf{u}}_{\ell-1}$. Next, we extract two portions from the estimated bits $\hat{\mathbf{u}}_{\ell-1}$ of layer $\ell - 1$: one portion is used for parity checking, and the other portion consists of connection bits that are recursively applied using (**??**). Finally, we gather the estimated frozen bits from each

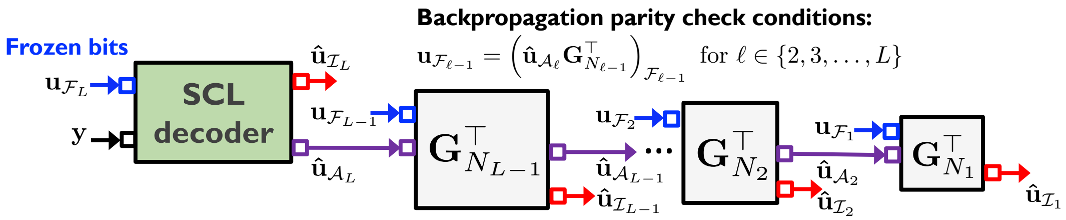

Backpropagation parity check conditions:
$\mathbf{u}_{\mathcal{F}_{\ell-1}} = \left(\hat{\mathbf{u}}_{\mathcal{A}_{\ell}}\mathbf{G}_{N_{\ell-1}}^{\top}\right)_{\mathcal{F}_{\ell-1}}$ for $\ell \in \{2,3,\ldots,L\}$
Frozen bits
$\mathbf{u}_{\mathcal{F}_L}$
$\mathbf{y}$
SCL decoder
$\hat{\mathbf{u}}_{\mathcal{I}_L}$
$\mathbf{u}_{\mathcal{F}_{L-1}}$
$\hat{\mathbf{u}}_{\mathcal{A}_L}$
$\mathbf{G}_{N_{L-1}}^{\top}$
$\mathbf{u}_{\mathcal{F}_2}$
$\hat{\mathbf{u}}_{\mathcal{A}_{L-1}}$
$\hat{\mathbf{u}}_{\mathcal{I}_{L-1}}$
$\mathbf{G}_{N_2}^{\top}$
$\mathbf{u}_{\mathcal{F}_1}$
$\hat{\mathbf{u}}_{\mathcal{A}_2}$
$\hat{\mathbf{u}}_{\mathcal{I}_2}$
$\mathbf{G}_{N_1}^{\top}$
$\hat{\mathbf{u}}_{\mathcal{I}_1}$

Fig. 4. An illustration of the SCL decoder using the backpropagation parity check principle.

layer $\ell \in [L]$ and verify their syndromes as depicted in Fig. **??**.

Leveraging this bit-wise BPC mechanism, the proposed SCL decoder employs a level-by-level search strategy on a binary tree. At each bit $u_{L,i}$, where $i \in \mathcal{A}_L \cup \mathcal{I}_L$, the decoder extends the list of candidate paths by exploring both paths of the binary tree and appending either $u_{L,i} = 0$ or $u_{L,i} = 1$ to each candidate path. Consequently, the number of paths is doubled but limited to a predetermined maximum value S. In contrast to the standard SCL decoder, it is included in the list whenever a new path satisfies the BPC and ranks among the S most reliable paths. Conversely, it is discarded if a new path fails the BPC or exhibits lower reliability than the existing S paths in the list. This iterative process continues until $i \in [N_L]$. Ultimately, the decoder selects the path with the highest reliability metric as the output. When $S = 1$, our decoder simplifies to the SC decoder utilizing the backpropagation parity check. The deep polar SCL decoding procedure is summarized in Algorithm **??**.

The proposed decoder introduces an extra decoding complexity due to the BPC operation compared to the standard SCL decoding complexity with a list size of S, which is $\mathcal{O}(SN_L \log N_L)$. For each layer $\ell \in \{1, 2, \ldots, N_{L-1}\}$, the parity check can be performed with a complexity of $\mathcal{O}(N_\ell \log N_\ell)$. Consequently, the overall complexity is the sum of the complexity required for SCL decoding and computing the inverse of the pre-transforms, i.e.,

$$\mathcal{O}(SN_L \log N_L) + \mathcal{O}\left(\sum_{\ell=1}^{L-1} N_\ell \log N_\ell\right). \tag{36}$$

It is worth noting that the additional complexity introduced by the inverse of the pre-transform can be ignored when N_L is significantly larger than N_ℓ for $\ell \in [L-1]$.

B. A Low-Latency Decoder

We also present a method for achieving low-latency decoding of deep polar codes. Our approach involves leveraging standard SCL decoders in parallel to estimate the information vector $\mathbf{u}_{\mathcal{I}_L}$, assuming the connection bits in layer L, $\mathbf{u}_{\mathcal{A}_L}$, as an additional frozen set alongside with frozen bits $\mathbf{u}_{\mathcal{F}_L}$. There are 2^{K-K_L} potential connection bit patterns for $\mathbf{u}_{\mathcal{A}_L}$. Let $\mathbf{u}_{\mathcal{A}_L}^j \in \mathbb{F}_2^{N_{L-1}}$ represent the jth connection bit pattern, where $j \in \left\{1, 2, \ldots, 2^{K-K_L}\right\}$. Given the jth connection bit pattern $\mathbf{u}_{\mathcal{A}_L}^j$ and the frozen bits of the last layer $\mathbf{u}_{\mathcal{F}_L} = \mathbf{0}$, the SCL decoder identifies the most reliable information vector from a list of size S. Denoting the result of SCL decoding for the jth possible connection bit pattern and frozen bits as $\hat{\mathbf{u}}_{\mathcal{I}_L}^j$, the decoder selects the most reliable estimate $\hat{\mathbf{u}}_{\mathcal{I}_L}^{j^\star}$ for the information vector from the set $\{1, 2, \ldots, 2^{K-K_L}\}$ as

$$j^\star = \underset{j\in\{1,2,\ldots,2^{K-K_L}\}}{\arg\max} \log p\left(\mathbf{y}_L \mid \mathbf{u}_{\mathcal{F}_L}, \mathbf{u}_{\mathcal{A}_L}^j, \hat{\mathbf{u}}_{\mathcal{I}_L}^j\right). \tag{37}$$

The parallel-SCL decoding method diminishes the decoding latency by harnessing the power of parallel processing. However, the hardware complexity of this method increases exponentially with the number of information bits encoded in ℓ layers, where $\ell \in \{1, 2, \ldots, L-1\}$ and $K - K_L$. Consequently, practical usage of this method is limited to scenarios where the number of $K - K_L$ is small enough. In addition, the overall computational complexity of the parallel-SCL decoding becomes

$$\mathcal{O}\left(2^{K-K_L} SN_L \log N_L\right). \tag{38}$$

The list size S of this parallel-SCL decoding can be chosen very small because the information bits are sent through the bit-channels with the highest capacities under the premise that the actual connection bits are utilized as the frozen bits. Therefore, the implementation needs to carefully optimize the list size and the number of parallel-SCL decoders so that $2^{K-K_L}S$ is comparable to the other SCL-type decoders.

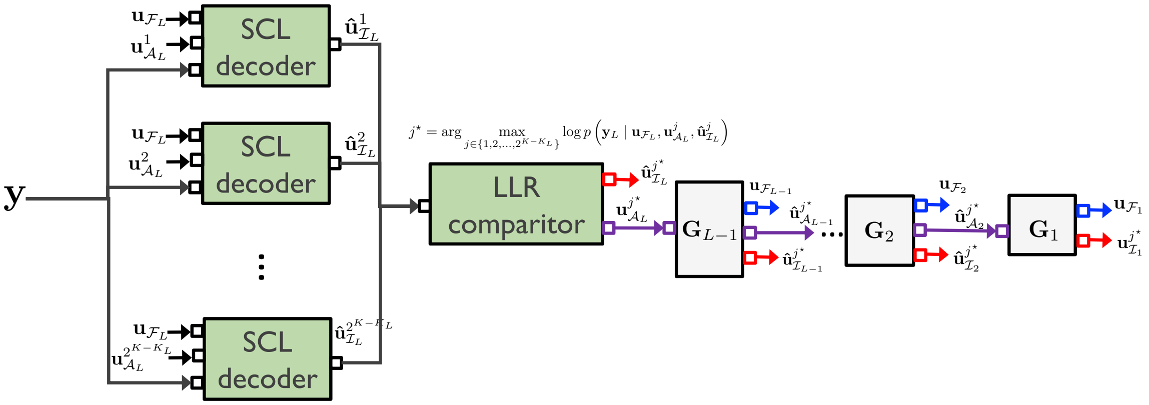
VI. Simulation Results

This section compares the BLERs of different encoding and decoding methods under BI-AWGN channel.

A. Proposed Code Construction

We implement two types of deep polar codes with/without CRC precoding.

- **Deep polar codes:** We have developed deep polar codes with varying numbers of layers, $L \in \{2, 3, 4\}$, based on the size of the information bits K. For instance, Table **??** shows encoding network configurations for deep polar codes. A significant finding from our investigations is that the minimum distance of the deep polar code, d^{min}, can exceed the minimum weight of the selected rows from $\mathbf{G}_{128}$ in $\mathcal{A}_L$, i.e., d_L^{min}. This



$\mathbf{u}_{\mathcal{F}_L}$
$\mathbf{u}^1_{\mathcal{A}_L}$
SCL decoder
$\hat{\mathbf{u}}^1_{\mathcal{I}_L}$
$\mathbf{u}_{\mathcal{F}_L}$
$\mathbf{u}^2_{\mathcal{A}_L}$
SCL decoder
$\hat{\mathbf{u}}^2_{\mathcal{I}_L}$
$\mathbf{y}$
$\mathbf{u}_{\mathcal{F}_L}$
$\mathbf{u}^{2^{K-K_L}}_{\mathcal{A}_L}$
SCL decoder
$\hat{\mathbf{u}}^{2^{K-K_L}}_{\mathcal{I}_L}$
$j^\star = \arg \max_{j\in\{1,2,\ldots,2^{K-K_L}\}} \log p\left(\mathbf{y}_L \mid \mathbf{u}_{\mathcal{F}_L}, \mathbf{u}^j_{\mathcal{A}_L}, \hat{\mathbf{u}}^j_{\mathcal{I}_L}\right)$
LLR comparitor
$\hat{\mathbf{u}}^{j^\star}_{\mathcal{I}_L}$
$\mathbf{u}^{j^\star}_{\mathcal{A}_L}$
$\mathbf{G}_{L-1}$
$\mathbf{u}_{\mathcal{F}_{L-1}}$
$\hat{\mathbf{u}}^{j^\star}_{\mathcal{A}_{L-1}}$
$\hat{\mathbf{u}}^{j^\star}_{\mathcal{I}_{L-1}}$
$\mathbf{G}_2$
$\mathbf{u}_{\mathcal{F}_2}$
$\hat{\mathbf{u}}^{j^\star}_{\mathcal{A}_2}$
$\hat{\mathbf{u}}^{j^\star}_{\mathcal{I}_2}$
$\mathbf{G}_1$
$\mathbf{u}_{\mathcal{F}_1}$
$\mathbf{u}^{j^\star}_{\mathcal{I}_1}$

Fig. 5. The proposed backpropagation decoding method using parallel-SCL decoders.

observation indicates that using multi-layered pre-transformations can increase the minimum distance of the code.

- **CA-deep polar codes:** We employ CRC precoding as an extra pre-transformation before applying the successive deep polar encoding. We utilize the specific 6-bit CRC polynomial $1 + D^5 + D^6$ for the CRC precoding. The effectiveness of CRC precoding in enhancing the codes is demonstrated in Table **??**, where it is observed that the minimum distance is increased with only a small number of layers for encoding. For example, the $(128, 64)$ CA-deep polar code with two layers exhibits an increased minimum distance compared to the two-layered deep polar code without CRC precoding. This result suggests that CRC precoding serves as an additional pre-transformation that introduces more layers into the encoding process. Consequently, it contributes to the overall improvement in the performance of the deep polar codes.
- **Decoders:** We employ three decoding algorithms for the deep polar codes. These algorithms include the proposed SCL-BPC decoding and parallel-SCL decoding techniques with a list size of S, which are primarily utilized to reduce decoding complexity and delays during simulations. The ML decoding algorithm is also employed to gauge the gap-to-capacity performance specifically for low code rates, such as $K = 16$ and $N \in \{64, 128, 256\}$.

B. Benchmarks

We explain benchmark encoding and decoding schemes to compare the BLER performance in a short blocklength regime. The benchmarks are listed as follows:

- **Theoretical bounds [?]:** We utilize theoretical benchmarks to assess the performance of our proposed encoding and decoding methods. Specifically, we plot the dispersion normal approximation and the meta-converse bounds for a given code length and rates, as described in [?]. The dispersion and meta-converse bounds were calculated using the *spectre* package, which can be accessed online at https://github.com/yp-mit/spectre. These bounds allow us to demonstrate the gaps between the achievable performance of our methods and the upper bounds of the finite-blocklength capacity.
- **CA-polar codes [?]:** The CA-polar codes serve as the baseline methods. The information sets are chosen according to the 5G standard specification in [?] when constructing the polar codes. In addition, we take into account the CRC precoding with the polynomial of $1 + D^5 + D^6$, which has shown the lowest BLER among CRC polynomials in 3GPP standards [?]. For decoding the CA-polar codes, we use the standard SCL decoder with a list size of S.
- **PAC codes [?]:** The PAC codes heavily depend on the rate-profile algorithm and generator polynomials of convolutional code. The optimized rate-profiling method is known for information bit sizes of $K = 29$ and $K = 64$ under blocklength of 128 [?]. Specifically, the PAC codes adopt the generator polynomials $\mathbf{c} = 3211$ and $\mathbf{c} = 133$ in the octal notation for $K = 29$ and $K = 64$, respectively.
- **CA-BOSS codes [?]:** We also consider comparing a block orthogonal sparse superposition (BOSS) code. Concatenating with CRC codes, the CA-BOSS code was shown to achieve the meta converse bounds within 1 dB for a short blocklength regime. For comparison, we implement a CA-BOSS code with the CRC polynomial of $1 + D + D^3$. We refer the reader to [?] for the construction details.

C. BLER Comparisons

Comparison with PAC codes: Fig. **??** presents the comparisons of BLER performances among three coding schemes: deep polar codes, the PAC codes with optimized rate-profiling, and 5G standard CA-polar codes, each evaluated at two distinct code rates, $R \in \{\frac{29}{128}, \frac{64}{128}\}$. The

```
Algorithm 1: SCL-BPC Decoder
Data: List size $S$, channel output $\mathbf{y}$, transform matrices $\mathbf{T}_\ell$, information sets $\mathcal{I}_\ell$, connection sets $\mathcal{A}_\ell$, frozen sets $\mathcal{F}_\ell$, $\forall \ell \in [L]$, and frozen bits $\mathbf{u}_{\mathcal{F}_L}$.
Result: Estimated information bits $\widehat{\mathbf{u}}_{\mathcal{I}_\ell}[s^\star]$, $\forall \ell \in [L]$;
*** Initialization ***;
$\mathcal{P}_{\mathsf{alive}} \leftarrow \{1\}$; $\mathcal{P}_{\mathsf{pool}} \leftarrow [2S]/\mathcal{L}_{\mathsf{alive}}$;
$\mathsf{PM}[s] \leftarrow 0,\ \forall s \in [S]$;
*** Decoding ***;
for $i = 1, 2, \dots, N_L$ do
    for $s \in \mathcal{P}_{\mathsf{alive}}$ do // path cloning
        $\eta_i \leftarrow \log \frac{p(\mathbf{y}, \widehat{\mathbf{u}}_{L,1:i-1}[s] | u_{L,i}=0)}{p(\mathbf{y}, \widehat{\mathbf{u}}_{L,1:i-1}[s] | u_{L,i}=1)}$;
        if $i \in \mathcal{F}_L$ then // frozen bits
            $\hat{u}_{L,i}[s] \leftarrow u_{L,i}$;
            $\mathsf{PM}[s] \leftarrow \mathsf{PM}[s] + \log\left(1 + e^{-(1-2\hat{u}_{L,i}[s])\eta_i}\right)$;
        else // information bits
            Copy the path $s$ into a new path $s' \in \mathcal{P}_{\mathsf{pool}}$;
            $\mathcal{P}_{\mathsf{alive}} \leftarrow \mathcal{P}_{\mathsf{alive}} \cup \{s'\}$; $\mathcal{P}_{\mathsf{pool}} \leftarrow \mathcal{P}_{\mathsf{pool}}/\{s'\}$;
            $\hat{u}_{L,i}[s] \leftarrow 0$; $\hat{u}_{L,i}[s'] \leftarrow 1$;
            $\mathsf{PM}[s] \leftarrow \mathsf{PM}[s] + \log\left(1 + e^{-(1-2\hat{u}_{L,i}[s])\eta_i}\right)$;
            $\mathsf{PM}[s'] \leftarrow \mathsf{PM}[s'] + \log\left(1 + e^{-(1-2\hat{u}_{L,i}[s'])\eta_i}\right)$;
        end
    end
    if $i \in \mathcal{A}_L$ then // parity check
        for $s \in \mathcal{P}_{\mathsf{alive}}$ do
            Apply inverse transform according to (??);
            $\widehat{\mathbf{u}}_{\mathcal{F}_\ell,1:k_\ell} \leftarrow$ currently available frozen bits of layer $\ell \in [L]$;
            if $\widehat{\mathbf{u}}_{\mathcal{F}_\ell,1:k_\ell} \neq \mathbf{0}$ for some $\ell$ then
                Kill the path $s$ ;
                $\mathcal{P}_{\mathsf{alive}} \leftarrow \mathcal{P}_{\mathsf{alive}}/\{s\}$; $\mathcal{P}_{\mathsf{pool}} \leftarrow \mathcal{P}_{\mathsf{pool}} \cup \{s\}$ ;
            end
        end
    end
    if $|\mathcal{P}_{\mathsf{alive}}| > S$ then // list pruning
        $\tau_{\mathsf{threshold}} \leftarrow$ the $S$th smallest $\mathsf{PM}[s]$ for $s \in \mathcal{P}_{\mathsf{alive}}$;
        for $s \in \mathcal{P}_{\mathsf{alive}}$ do
            if $\mathsf{PM}[s] > \tau_{\mathsf{threshold}}$ then
                Kill the path $s$;
                $\mathcal{P}_{\mathsf{alive}} \leftarrow \mathcal{P}_{\mathsf{alive}}/\{s\}$; $\mathcal{P}_{\mathsf{pool}} \leftarrow \mathcal{P}_{\mathsf{pool}} \cup \{s\}$ ;
            end
        end
    end
end
Find $s^\star \leftarrow \arg\min_{s \in \mathcal{P}_{\mathsf{alive}}} \mathsf{PM}[s]$;
*** Information bits extraction ***;
for $\ell = L, \dots, 2$ do
    $\widehat{\mathbf{u}}_{\mathcal{I}_{\ell-1}}[s^\star] \leftarrow \left(\widehat{\mathbf{u}}_{\mathcal{A}_\ell}[s^\star]\mathbf{G}_{N_{\ell-1}}^\top\right)_{\mathcal{I}_{\ell-1}}$;
    $\widehat{\mathbf{u}}_{\mathcal{A}_{\ell-1}}[s^\star] \leftarrow \left(\widehat{\mathbf{u}}_{\mathcal{A}_\ell}[s^\star]\mathbf{G}_{N_{\ell-1}}^\top\right)_{\mathcal{A}_{\ell-1}}$;
end
Return $\widehat{\mathbf{u}}_{\mathcal{I}_\ell}[s^\star],\ \forall \ell \in [L]$;
```

meta-converse and the normal approximation bounds are plotted as theoretical benchmarks for the corresponding code rates. For decoding, SCL-BPC decoders are utilized for deep polar codes, while SCL decoders are employed for both PAC and CA-polar codes with varying list sizes, $S \in \{8, 32, 256\}$. The solid lines on the graph correspond to the BLER performances of SCL-BPC and SCL decoders when using a short list size of $S = 8$. The results demonstrate that deep polar codes outperform PAC and CA-polar codes when employing a short list size, particularly where $R = \frac{64}{128}$, PAC codes exhibit a worse BLER performance than CA-polar codes. Additionally, we observe the dotted lines on the plot, representing the BLER performances with larger list sizes, i.e., $S = 32$ for $K = 29$ and $S = 256$ for $K = 64$. Notably, with sufficiently large list sizes, both deep polar and PAC codes closely achieve the normal approximation bound, while CA-polar codes still exhibit a gap from the bound.

Comparison with CA-polar codes: Fig. **??** illustrates the comparison of BLER performance between the proposed deep polar and 5G CA-polar codes using two decoding methods: SCL-BPC and SCL with a list size of $S = 8$. The simulation result reveals that our deep polar codes consistently outperform the CA-polar counterparts at three specific code rates, namely $R \in \{\frac{32}{128}, \frac{56}{128}, \frac{96}{128}\}$. This result suggests that the proposed deep polar code exhibits superior performance compared to the 5G CA-polar codes across various code rates, while the increase in decoding complexity remains marginal.

Performance of parallel-SCL decoder: Fig. **??** shows the BLER performance for the deep polar, PAC, and CA-polar codes under parallel-SCL decoding. To implement the parallel-SCL decoder for the PAC and CA-polar codes, we select the $K - K_L$ smallest information indices and generate 2^{K-K_L} combinations of information bit patterns as possible frozen bit patterns. Then, we apply the parallel-SCL decoder by adding the generated information bit patterns as additional frozen bits. For each possible frozen bit pattern, the decoder applies the SCL decoding with a list size of $S = 2$. The simulation result shows that deep polar codes outperform CA-polar and PAC codes under parallel-SCL decoding by attesting that local pre-transform gives robustness to parallel-SCL decoding.

Fig. **??** shows the BLER performance of deep polar codes decoded by parallel-SCL and CA-polar codes by SCL decoder. We use a parallel-SCL decoder with list size $S = 2$ to decode deep polar codes with message length $K = 29$ and list size $S = 4$ for $K = 64$. The BLER curve of deep polar codes under parallel-SCL decoder lies between that of CA-polar codes under SCL decoder with list size $S = 16$ and $S = 32$. Even if the total number of computations of parallel-SCL is larger than SCL counterparts, we can parallelize the decoding operation to achieve low decoding latency, thereby attest-

TABLE II
Simulation parameters

	Fig.	(N,K)	Rate-profile	Pre-transform	Decoder	$\mathsf{d}_L^{\min}$		(N_ℓ, K_ℓ)			
						Design	Estimate	$\ell=1$	$\ell=2$	$\ell=3$	$\ell=4$
DP	??	$(128,29)$	5G [?]	Polar	SCL-BPC	16	32	$(128,19)$	$(16,8)$	$(4,2)$	
DP	?? (—)	$(128,64)$	5G [?]	Polar	SCL-BPC	8	8	$(128,51)$	$(16,13)$		
CA-DP	?? (- -)	$(128,64)$	5G [?]	Polar + CRC6	SCL-BPC	8	12	$(128,67)$	$(16,3)$		
DP	??	$(128,32)$	5G [?]	Polar	SCL-BPC	16	24	$(128,21)$	$(16,6)$	$(8,3)$	$(4,2)$
DP	??	$(128,56)$	5G [?]	Polar	SCL-BPC	16	16	$(128,52)$	$(8,4)$		
DP	??	$(128,96)$	5G [?]	Polar	SCL-BPC	8	8	$(128,94)$	$(4,2)$		
DP	??-??	$(128,29)$	DEGA 1.5 dB	Polar	parallel-SCL	16	32	$(128,24)$	$(32,3)$	$(8,1)$	$(2,1)$
DP	??-??	$(128,64)$	DEGA 6 dB	Polar	parallel-SCL	8	16	$(128,59)$	$(32,3)$	$(8,1)$	$(2,1)$
DP	??	$(64,16)$	5G [?]	Polar	ML	16	16	$(64,6)$	$(16,8)$	$(4,2)$	
DP	??	$(128,16)$	5G [?]	Polar	ML	32	32	$(128,6)$	$(16,8)$	$(4,2)$	
DP	??	$(256,16)$	5G [?]	Polar	ML	64	64	$(256,6)$	$(16,8)$	$(4,2)$	
CA-DP	??	$(64,16)$	5G [?]	Polar + CRC6	ML	8	16	$(64,11)$	$(16,11)$		
CA-DP	??	$(128,16)$	5G [?]	Polar + CRC6	ML	32	32	$(128,12)$	$(16,10)$		
CA-DP	??	$(256,16)$	5G [?]	Polar + CRC6	ML	64	96	$(256,13)$	$(16,9)$		
CA-polar	??-??	$\forall(N,K)$	5G [?]	CRC6	SCL						
PAC	??,??	$(128,29)$	RM [?]	CC ($\mathbf{c}=3211$)	SCL		32				
PAC	??,??	$(128,64)$	RM [?]	CC ($\mathbf{c}=133$)	SCL		16				

** DP: deep polar, DEGA: Density Evolution with Gaussian Approximation [?], CC: Convolutional Coding
** Estimate $\mathsf{d}_L^{\min}$ is computed using SCL(-BPC) decoder with list size $S=10^4$.

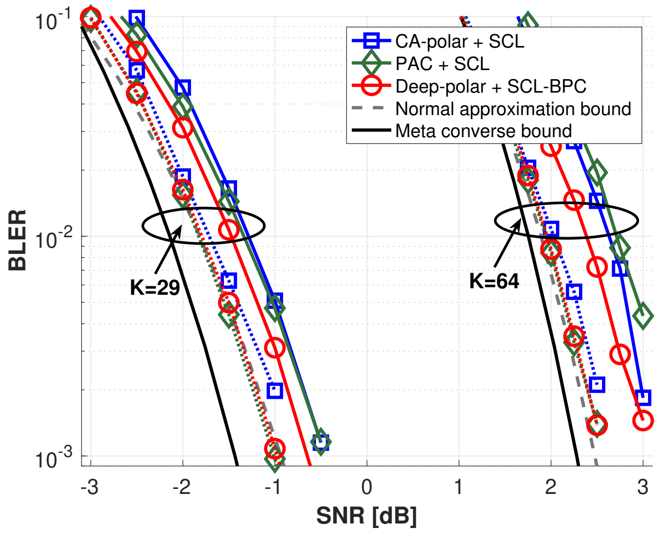


Fig. 6. BLER performance of PAC, CA-polar, and deep polar codes. Parameters for solid line are $(K,S) \in \{(29,8),(64,8)\}$, and for dotted line are $(K,S) \in \{(29,32),(64,256)\}$.

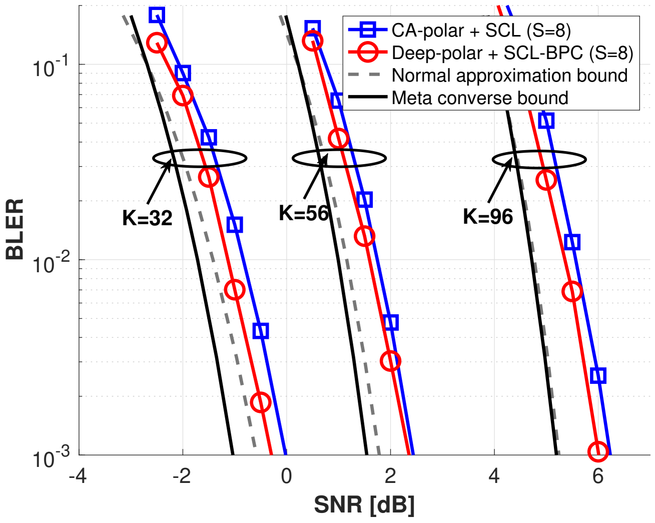


Fig. 7. BLER performance comparison bewteen deep polar and CA-polar codes. The code parameters are $N=128$ and $K \in \{32,56,96\}$.

ing to the suitability of deep polar codes for low-latency communication applications.

Comparison with CA-BOSS codes: Fig. **??** presents the ML decoding performance of four coding schemes: deep polar codes, CA-deep polar codes, CA-BOSS codes, and CA-polar codes, at different code rates $R \in \{\frac{16}{256}, \frac{16}{128}, \frac{16}{64}\}$. The results demonstrate the effectiveness of the proposed codes in improving the weight spectrum. By utilizing ML decoding, we assess the performance improvement achieved by the proposed codes. For example, when $R=\frac{16}{128}$, the CA-BOSS code marginally outperforms the deep polar code. However, with the integration of CRC precoding, the CA-deep polar code exhibits superior decoding performance compared to all other considered coding schemes, even achieving the meta converse bound within 0.4 dB. This remarkable outcome highlights the significant benefits of harmonizing local pre-transform (e.g., deep polar code) with global pre-transform (e.g., CRC) for boosting decoding performance at low rates.

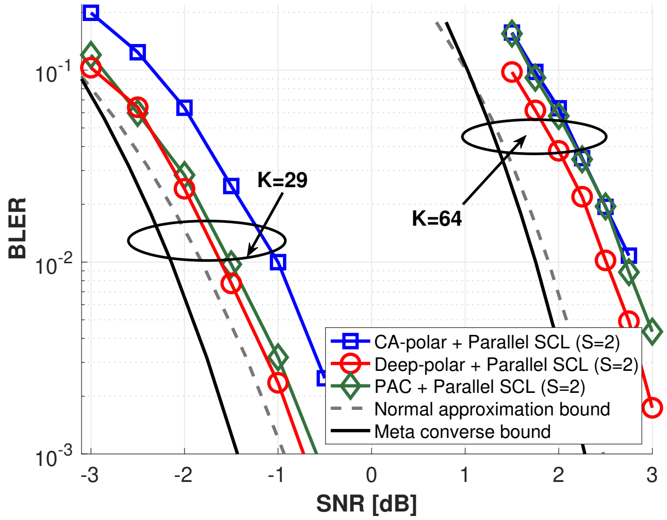


Fig. 8. BLER performance under parallel-SCL decoder.

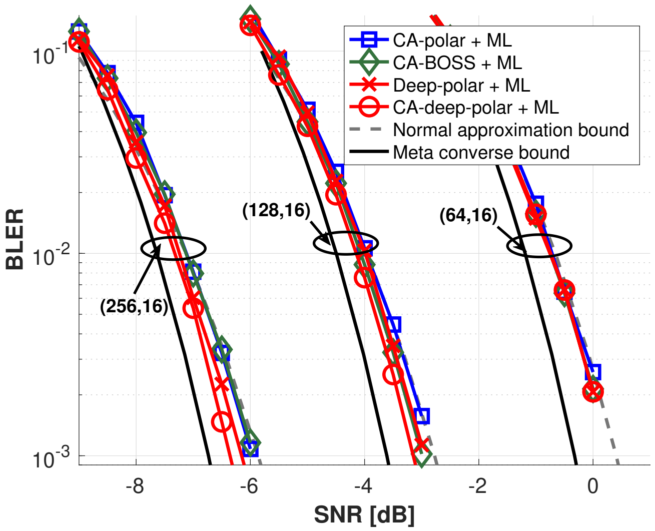


Fig. 10. BLER performance of deep polar, CA-deep polar, CA-polar, and CA-BOSS codes. The CRC polynomial is $1 + D^5 + D^6$ for CA-deep polar and CA-polar codes and $1 + D + D^3$ for CA-BOSS codes.

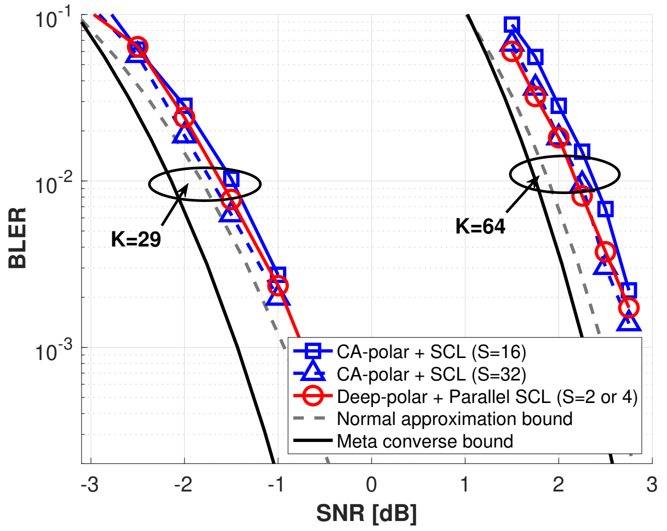


Fig. 9. BLER performance of deep polar codes under parallel-SCL decoder and CA-polar codes under SCL decoder. We use list size $S = 2$ for decoding deep polar codes for $K = 29$, and $S = 4$ for $K = 64$.

VII. Conclusion

We have introduced a novel type of pre-transformed polar code called the deep polar code. The deep polar codes are generated through successive multi-layered encoding with a flexible rate-profiling method. The main technical innovation involves creating a unique yet universal pre-transformed architectures of polar codes, which is inspired by deep layered transform coding. These innovative architectures open up new possibilities for exploring a far more diverse range of pre-transformed polar code structures that were not explored before. Our deep polar code is a superposition code of pre-transformed and regular polar subcodes. Controlling the rates of these two subcodes enables the design of low-complexity decoding while significantly improving the weight distribution of the codes. Leveraging this superposition property, we have also introduced two decoding methods: the SCL-BPC decoder, which effectively reduces decoding complexity, and the parallel-SCL decoder, which minimizes decoding latency. Through simulations, we have demonstrated that our codes achieve close to finite-blocklength capacity and consistently outperform all existing state-of-the-art pre-transformed polar codes at various rates and short blocklengths, all while maintaining a reasonable decoding complexity.

A promising further research direction would be to extend the deep polar code design to moderate blocklengths, such as up to 2048, in order to bridge the gap towards the finite-blocklength capacity in moderate blocklength regimes. Additionally, exploring the development of a more efficient decoding algorithm for larger code blocklengths holds great promise, as the complexity of the SCL-BPC decoder may become computationally prohibitive in moderate blocklengths. Lastly, it is also interesting to design an efficient rate-compatible deep polar code to optimally support HARQ protocols.