Sparsely Pre-transformed Polar Codes for Low-Latency SCL Decoding

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

***Abstract*—Deep polar codes, which employ multi-layered polar kernel pre-transforms in series, are recently introduced variants of pre-transformed polar codes. These codes can reduce the number of minimum-weight codewords, thus closely achieving finite-block length capacity with successive cancellation list (SCL) decoders in certain scenarios. However, for low-latency applications requiring small SCL decoder list sizes, reducing minimum-weight codewords doesn't necessarily improve decoding performance. To address this limitation, we propose an alternative pre-transform technique to enhance the suitability of polar codes for SCL decoders with practical list sizes. Leveraging the fact that the SCL decoding error event set can be decomposed into two exclusive error event sets, our approach applies two different types of pre-transformations, each targeting the reduction of one of the two error event sets. Extensive simulation results across various block lengths and code rates demonstrate that our codes consistently outperform existing state-of-the-art pre-transformed polar codes, including CRC-aided polar codes and polarization-adjusted convolutional codes, when decoded using SCL decoders with small list sizes.**

I. Introduction

The demand for ultra-reliable low-latency communication (URLLC) continues to grow in next-generation wireless communication systems. URLLC aims to deliver extremely high-speed data packets with a very low packet error rate within a very short timeframe [2]–[4]. Achieving these goals requires cutting-edge channel coding technology. This technology must be capable of providing excellent error-correcting performance and fast decodability within a finite blocklength regime [5]–[10].

Pre-transformed polar codes are a variant of polar codes that involve applying an upper-triangular pre-transform before the polar coding process. This pre-transformation has been shown to enhance the weight spectrum of the codes by either reducing the number of minimum-weight (min-weight) codewords or increasing the minimum distance. Numerous pre-transformation techniques have been proposed in the literature [11]–[19]. Most of these methods aim to improve the code weight spectrum and are particularly effective when maximum likelihood (ML) decoders or successive cancellation list (SCL) decoders with large list sizes are used [20]–[24]. However, utilizing an SCL decoder with a small list size is essential to minimize decoding latency to enable URLLC in practice. Unfortunately, when using SCL decoders with small list sizes (e.g., fewer than 8), improving the code weight spectrum through pre-transformation does not necessarily lead to better decoding performance.

The SCL decoder can fail to decode the transmitted codeword under two exclusive error conditions. The first occurs when the transmitted codeword is not in the final list of the codewords identified by the SCL decoder. The second error happens when the transmitted codeword is present in the final list, but another codeword in the list is closer to the received signal. As a result, under a small list size of the SCL decoder, it is essential to design a pre-transform method that can reduce the two exclusive error events simultaneously.

In this paper, we advance the design of pre-transformed polar codes to enhance decoding performance with SCL decoders using limited list sizes. Our key innovation is to apply two different types of polar pre-transformation methods in parallel, with each method targeting the reduction of one of two distinct error event sets. This improvement is crucial for enabling the next generation of URLLC.

A. Related Works

Since the introduction of polar codes [25], proven to achieve asymptotic capacity over binary input discrete memoryless channels (B-DMC) with non-random construction and low-complexity successive cancellation (SC) decoding algorithms, many studies have focused on selecting information index sets for various channels and SC decoding methods [25]–[33]. In the seminal paper [25], Bhattacharyya parameters were used as reliability metrics, with a simple recursive relation for binary erasure channels (BEC) and Monte-Carlo simulations required for other channels. For efficient construction, a method to approximate bit-channels with large output alphabets by those with smaller alphabets was presented in [26]. For additive white Gaussian noise (AWGN) channels, density evolution (DE) [27] tracks log-likelihood ratios (LLR) through the SC decoding process. To reduce DE complexity, Gaussian approximations (GA) were proposed in [28], [29], approximating LLRs as Gaussian random variables to estimate their means. The information index sets from these methods depend on channel types and qualities. Additionally, channel-independent methods, based on partial order, have been developed in [30]–[32].

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00398449), in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2022-NR070834), and in part by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT2302-04. An earlier version of this paper was presented in part at the 2024 IEEE International Symposium on Information Theory (ISIT) [1].

Geon Choi is with the School of Electrical Engineering, Korea University, Seoul 02841, South Korea (e-mail: simon03062@korea.ac.kr).

Namyoon Lee is with the Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea (e-mail: nylee@postech.ac.kr).

In the finite blocklength regime, the channel polarization effect is incomplete, leading to performance degradation of SC decoders due to the dominance of the worst bit-channel error probability. To overcome this problem, a list decoding method was considered as an efficient decoder for the polar codes [34]. Unlike the SC decoder, however, finding the optimal information index set is highly challenging when the SCL decoder is used. The main difficulty arises from the fact that the DE methods developed under SC decoder [25]–[32] do not guarantee the optimality of the performance for the SCL decoder. One approach to resolve this issue was to harness the path metric range of the SCL decoder to replace some frozen bits with information bits, maintaining a large path metric range [35]. Another relevant prior work was in [36], in which the required list size of the SCL decoder was identified to achieve ML decoding performance. As subsequent work, a genetic type algorithm was proposed in [23] to generate the optimized information set for SCL decoding, which showed better decoding performance than that in [36]. Despite these advancements, there is still no optimal design method to determine the information set for a given code rate, blocklength, and list size.

Identifying min-weight codewords in polar codes is crucial for designing pre-transform techniques to reduce their number [22], [37]–[40]. For example, the number of min-weight codewords was characterized by examining the automorphism group in decreasing monomial codes [37]. Further, a study demonstrates how combining rows of the polar transform matrix generates min-weight codewords and shows that carefully replacing information indices with frozen indices can reduce their number [38]. This insight is used to explore the impact of pre-transforms, confirming their ability to effectively eliminate min-weight codewords [41].

Pre-transformed polar codes encompass various instances, including cyclic-redundancy-check-aided polar (CA-polar) [11], polar codes with dynamic frozen bits [12], parity-check polar (PC-polar) [13], [14], PAC codes [15], row-merged polar codes [16], [17], and deep polar codes [18], [19]. Numerous studies focus on optimizing the pre-transform matrix to reduce the number of low-weight codewords [20]–[22]. Numerical results show that pre-transformation significantly enhances the weight spectrum of codes. For example, PAC codes with convolutional precoding achieve the normal approximation bound with Fano decoding for short block lengths. Additionally, deep polar codes with a multi-layered polar pre-transform have been shown to reach the normal approximation bound, especially with the SCL decoder featuring a large list size and backpropagation parity check (SCL-BPC). Despite these improvements, pre-transformed polar codes still require an ML-like decoder to approach the theoretical bound.

Finding an efficient pre-transform matrix and rate-profile method for a low-complexity SCL decoder remains a significant challenge. Several studies focus on optimizing the pre-transform matrix to reduce low-weight codewords [16], [20]–[22], [39]. Rate-profile is typically determined by i) the Reed-Muller rule for SCL decoding with large list sizes (i.e., ML-like decoding) or ii) a polarization-based rule for SC decoding error probability. However, the path metric of the decoding candidate must be jointly taken into account for SCL decoding with smaller or moderate list sizes. Most prior studies in [23], [24] have addressed SCL decoding with relatively large list sizes, ranging from 32 to 1024. Unfortunately, optimizing the pre-transform with rate profiling for small list sizes in SCL decoding remains an unresolved issue. In this paper, we address this problem by introducing a new pre-transform technique for polar codes.

B. Contributions

Our contributions are summarized as follows:

- We present a novel pre-transform polar code called sparsely pre-transformed polar (SPP) codes. The SCL decoder may fail to decode the transmitted codeword due to two distinct error events. To address this issue, we propose employing two different types of pre-transformations based on polar kernel matrices in parallel. Each pre-transform is specialized to mitigate one of the two error events. Specifically:
  - One type of pre-transform aims to prevent the use of consecutive unreliable information bits, thereby reducing the error event where the transmitted codeword is not in the final list of the SCL decoder.
  - The other type of pre-transform targets reducing the number of low-weight codewords, decreasing the error event where another codeword is closer to the received signal than the transmitted codeword when both are in the final list.
- Our parallel pre-transform structure enables us to simultaneously reduce the two distinct error events. This is a key distinction from our prior work on deep polar codes [18], where the polar-based pre-transform were applied serially, focusing only on reducing the number of low-weight codewords. Despite this change, the proposed parallel pre-transform structure maintains the advantage of low-complexity encoding by leveraging the small sizes of polar kernel-based pre-transform, resulting in a sparse pre-transform matrix structure.
- We provide theoretical justification for our proposed code construction. We use path metric range and the entropy of the SCL decoder's decoding path to argue that consecutive semi-polarized bits increase the decoding error probability by increasing the likelihood of the correct decoding path being deleted. Additionally, by examining the formation of min-weight codewords, we demonstrate that our rate-profiling and row-merging pair selection algorithm can efficiently eliminate such codewords.
- Our simulations demonstrate that our proposed method achieves state-of-the-art block error rates (BLERs) compared to existing finite block length codes, including 5G CRC-aided polar codes, PAC codes, and deep polar codes, across various code rates and short block lengths with SCL decoding. The performance gains are especially pronounced when employing SCL decoding with a small list size.

II. Preliminaries

We now introduce the key concepts and notations relevant to our work.

A. Notations

The binary field is denoted as $\mathbb{F}_2 = \{0, 1\}$. Scalar quantities are denoted by lowercase letters, such as x, and their corresponding random variables by uppercase letters, such as X. Vectors are denoted by bold lowercase letters, such as $\mathbf{x}$, while their corresponding random vectors are denoted by bold uppercase letters, such as $\mathbf{X}$. The same notation $\mathbf{X}$ is also used for matrices, but the context will clarify the meaning. For a vector $\mathbf{x}$, we generally use one-based indexing, i.e., $\mathbf{x} = [x_1, \ldots, x_N]$. Exclusively, zero-based indexing, $\mathbf{x} = [x_0, \ldots, x_{N-1}]$, is used in Section III for a more concise description. Given a set $\mathcal{A}$ and a vector $\mathbf{x}$, for instance, if $\mathcal{A} = \{1, 3\}$, we use the sub-vector notation $\mathbf{x}_\mathcal{A} = [x_1, x_3]$. For $a, b \in [N]$ with $a < b$, we use the shorthand notation $\mathbf{u}_{a:b} = [u_a, u_{a+1}, \ldots, u_b]$. Finally, the interval notations $[K] = \{1, \ldots, K\}$ and $[a, b] = \{a, a + 1, \ldots, b\}$ are used.

B. Channel Coding System

A binary linear block code $\mathcal{C}(N, K)$ with a codeword length N and code dimension K is defined by the row space of the generator matrix $\mathbf{G} \in \mathbb{F}_2^{K\times N}$. The code rate R is defined as $R = \frac{K}{N}$. A codeword $\mathbf{x} \in \mathbb{F}_2^N$ is generated from the message vector $\mathbf{m} \in \mathbb{F}_2^K$ using the relation $\mathbf{x} = \mathbf{mG}$.

The minimum distance $d_{\min}$ of linear block code $\mathcal{C}(N, K)$ is defined as

$$d_{\min} \triangleq \min_{\mathbf{x}\in\mathcal{C}\setminus\{\mathbf{0}\}} \mathsf{wt}(\mathbf{x}), \tag{1}$$

where $\mathsf{wt}(\mathbf{x})$ represents the number of ones in $\mathbf{x}$, referred to as Hamming weight. The number of codewords whose Hamming weight is $d_{\min}$ is denoted as $A_{d_{\min}}$.

The two parameters, $d_{\min}$ and $A_{d_{\min}}$, are directly related to ML decoding performance. Under ML decoding, the BLER P_{ML} at binary input AWGN (BI-AWGN) channel is upper bounded by the union bound as

$$P_{\mathsf{ML}} \leq \sum_d A_d Q\left(\sqrt{2dR\frac{E_\mathrm{b}}{N_0}}\right), \tag{2}$$

where $Q(u) = \int_u^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \mathrm{d}x$ and $\frac{E_\mathrm{b}}{N_0}$ denotes the energy per bit to noise power spectral density ratio. By increasing the minimum distance and simultaneously reducing the number of codewords with min-weight, it is possible to significantly improve the code performance.

Consider a binary input discrete memoryless channel (B-DMC) $W : \mathcal{X} \to \mathcal{Y}$, where the channel input $\mathcal{X}$ will always be the binary field $\mathbb{F}_2$ and $\mathcal{Y}$ is the arbitrary output alphabet. The channel translates a binary codeword $\mathbf{x}$ into an output sequence $\mathbf{y} \in \mathcal{Y}^N$. Given a B-DMC W, the symmetric channel capacity is defined as

$$I(W) \triangleq \sum_{x\in\mathbb{F}_2} \sum_{y\in\mathcal{Y}} \frac{1}{2} W(y|x) \log_2 \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}. \tag{3}$$

C. Polar Codes

A polar code with parameters $(N, K, \mathcal{I})$ is characterized by a polar transform matrix of size $N = 2^n (n \in \mathbb{N})$ and an index set $\mathcal{I} \subseteq [N]$ suth that $|\mathcal{I}| = K$. 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

$$\mathbf{G}_N = \mathbf{G}_2^{\otimes n}. \tag{4}$$

The input vector of the encoder, denoted as $\mathbf{u} = [u_1, u_2, \ldots, u_N] \in \mathbb{F}_2^N$, is generated based on the given information set $\mathcal{I}$. In this process, the data vector carrying K information bits, denoted as $\mathbf{d}$, is allocated to $\mathbf{u}_\mathcal{I}$. The remaining elements of $\mathbf{u}$, denoted by $\mathbf{u}_{\mathcal{I}^c}$, are assigned zeros. Here, $\mathcal{I}^c = [N]\backslash\mathcal{I}$ is referred to as the frozen index 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

$$\mathbf{x} = \mathbf{u}\mathbf{G}_N = \sum_{i\in\mathcal{I}} u_i \mathbf{g}_{N,i}, \tag{5}$$

where $\mathbf{g}_{N,i}$ is the ith row vector of $\mathbf{G}_N$.

After channel combining and splitting [25], 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:

$$W_N^{(i)}(\mathbf{y}, \mathbf{u}_{1:i-1}|u_i) = \sum_{\mathbf{u}_{i+1:N}\in\mathbb{F}_2^{N-i}} \frac{1}{2^{N-1}} W^N(\mathbf{y}|\mathbf{x}), \tag{6}$$

where $W^N(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^N W(y_i|x_i)$ is the N copies of B-DMCs. In the case of an infinite blocklength, the bit-channels are perfectly polarized into two states, i.e., $I\left(W_N^{(i)}\right) \to 0$ or $I\left(W_N^{(i)}\right) \to 1$ as $N \to \infty$. As a result, when N is sufficiently large, the proper rate profiling is to select the indices having the capacity of one, i.e.,

$$\mathcal{I} = \left\{i \in [N] : I\left(W_N^{(i)}\right) \geq 1 - \epsilon\right\}, \tag{7}$$

for small $\epsilon > 0$. This rate profiling is sufficient to achieve the capacity under simple SC decoding [25]. For a short blocklength regime under SCL decoding, however, it remains open how to optimally choose the information set for polar codes.

D. Pre-Transformed Polar Codes

A $(N, K, \mathcal{I}, \mathbf{T})$ pre-transformed polar code comprises a binary upper-triangular invertible pre-transformation matrix $\mathbf{T} \in \mathbb{F}_2^{N\times N}$ and an information index set $\mathcal{I}$ used for rate profiling. During encoding, the information vector $\mathbf{d} \in \mathbb{F}^K$, containing K information bits, is incorporated into the input vector $\mathbf{v} \in \mathbb{F}^N$ of the pre-transformation. After setting $\mathbf{v}_\mathcal{I} = \mathbf{d}$ and $\mathbf{v}_{\mathcal{I}^c} = \mathbf{0}$, the codeword $\mathbf{x}$ is generated as

$$\mathbf{x} = \mathbf{v}\mathbf{T}\mathbf{G}_N. \tag{8}$$

Selecting an appropriate pre-transform matrix $\mathbf{T}$ along with information index set $\mathcal{I}$ gives rise to a significant challenge when constructing the pre-transformed polar codes. The joint

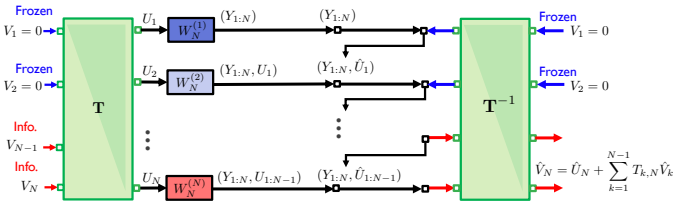

Fig. 1. Decoding of pre-transformed polar codes.

optimization of $\mathbf{T}$ and $\mathcal{I}$ to maximize the weight spectrum of a code necessitates a highly complex optimization process, even for short blocklength scenarios. Optimizing becomes more complex when factoring in the restricted decoding capabilities, specifically, a SCL decoder with a small list size.

E. SCL Decoding for Pre-Transformed Polar Codes

We explain the SCL decoding of general pre-transformed polar codes shown in Fig. 1. Recall that input vector $\mathbf{v}$ is pre-transformed into $\mathbf{u}$ by upper-triangular matrix $\mathbf{T}$ (i.e., $\mathbf{u} = \mathbf{v}\mathbf{T}$), followed by polar transform $\mathbf{G}_N$ to generate codeword $\mathbf{x} = \mathbf{u}\mathbf{G}_N$. The codeword $\mathbf{x}$ is mapped into corrupted codeword $\mathbf{y}$ by channel $W^N(\mathbf{y}|\mathbf{x})$. The decoding proceeds in reverse order. Given noisy codeword $\mathbf{y}$, the decoder sequentially estimate $\hat{u}_i$ from $i = 1$ to N. Then, the inverse pre-transform is applied to retrieve $\hat{\mathbf{v}}_{1:N} = \hat{\mathbf{u}}_{1:N}\mathbf{T}^{-1}$.

When decoding $\hat{u}_i$, the decoder utilizes the polarized bit channel $W_N^{(i)}(\mathbf{y}, \mathbf{u}_{1:i-1}|u_i)$. As to previous bits $\mathbf{u}_{1:i-1}$, the decoder presumes the estimated bits $\hat{\mathbf{u}}_{1:i-1}$ are true. To increase the probability of correctness of this assumption, the SCL decoder keeps S hypotheses of previous estimate bits. The ith bit of the sth hypothesis is denoted as $\hat{u}_i[s]$. For each decoding path s and bit index i, the decoder duplicates the current hypothesis $\hat{\mathbf{u}}_{1:i-1}[s]$ into $(\hat{\mathbf{u}}_{1:i-1}[s], 0)$ and $(\hat{\mathbf{u}}_{1:i-1}[s], 1)$. After that, the decoder prunes the decoding paths, which violate the frozen bit condition, i.e., $\hat{\mathbf{v}}_{i\in\mathcal{F}}[s] = 0$. In addition, if the number of decoding paths exceeds the predetermined threshold S, the most S probable paths in terms of $W_N^{(i)}(\mathbf{y}, \hat{\mathbf{u}}_{1:i-1}[s] \mid u_i = \hat{u}_i[s])$ are maintained and remaining paths are eliminated. For the recursive computation of $W_N^{(i)}(\mathbf{y}, \hat{\mathbf{u}}_{1:i-1}[s] \mid u_i = \hat{u}_i[s])$, we refer the reader to [42, Theorem 1].

In the decoding process, we need to check frozen bit condition in v-domain, while the decoding proceeds in u-domain. Thanks to upper-triangular structure of pre-transform matrix, we can retrieve $\mathbf{v}_{1:i}$ from $\mathbf{u}_{1:i}$ by

$$\mathbf{v}_{1:i} = \mathbf{u}_{1:i}\left(\mathbf{T}_{1:i}\right)^{-1}, \tag{9}$$

where $\mathbf{T}_{1:i}$ is upper-left sub-matrix with size $i \times i$. The same process can be performed in recursively:

$$v_i = u_i + \sum_{k=1}^{i-1} T_{k,i} v_k. \tag{10}$$

III. SCL Decoding Error Analysis

In this section, we provide an error analysis for the SCL decoder. Through this analysis, we demonstrate how two different types of error events determine the SCL decoding performance depending on the list size. We then present methods to reduce each of these error events separately, using rate profiling and pre-transform techniques. These techniques are jointly used to construct our SPP codes.

A. Two Error Events in SCL Decoding

The decoding error event $\mathcal{E}$ can be decomposed into two mutually exclusive events: $\mathcal{E}_1$ and $\mathcal{E}_2$. Here, $\mathcal{E}_1$ represents the error event where the transmitted codeword does not appear in the final list of the SCL decoder. On the other hand, $\mathcal{E}_2$ denotes the case where another codeword in the list is closer to the received signal than the transmitted codeword. Therefore, the decoding error probability under SCL decoding with a list size of S can be expressed as:

$$P_{\mathsf{SCL}}(\mathcal{E}; S) = P_{\mathsf{SCL}}(\mathcal{E}_1; S) + P_{\mathsf{SCL}}(\mathcal{E}_2; S). \tag{11}$$

The error event $\mathcal{E}_1$ mainly depends on the selection of information index set $\mathcal{I}$, i.e., the rate-profile. The event $\mathcal{E}_2$ depends on minimum distance and the number of min-weight codewords, which is a function of both rate-profile $\mathcal{I}$ and pre-transform matrix $\mathbf{T}$.

If $S = 1$, then SCL decoder reduces to SC decoder and decoding error probability of SC decoder $P_{\mathsf{SC}}(\mathcal{E})$ is given by

$$\begin{aligned} P_{\mathsf{SC}}(\mathcal{E}) &= P_{\mathsf{SCL}}(\mathcal{E}; S = 1) && (12)\\ &= P_{\mathsf{SCL}}(\mathcal{E}_1; S = 1) && (13)\\ &= 1 - \prod_{i=0}^{N-1}\left(1 - P\left(\mathcal{E}; W_N^{(i)}\right)\right), && (14)\end{aligned}$$

where $P\left(\mathcal{E}; W_N^{(i)}\right)$ is error probability of channel $W_N^{(i)}$, given by

$$\begin{aligned} &P\left(\mathcal{E}; W_N^{(i)}\right)\\ &= \mathbb{P}\left(W_N^{(i)}(\mathbf{y}, \mathbf{u}_{0:i-1}|0) < W_N^{(i)}(\mathbf{y}, \mathbf{u}_{0:i-1}|1) \mid \mathbf{u} = \mathbf{0}\right)\\ &\quad + \frac{1}{2}\mathbb{P}\left(W_N^{(i)}(\mathbf{y}, \mathbf{u}_{0:i-1}|0) = W_N^{(i)}(\mathbf{y}, \mathbf{u}_{0:i-1}|1) \mid \mathbf{u} = \mathbf{0}\right). \end{aligned} \tag{15}$$

In this case, decoding error event is dominated by information index set. For example, if a channel W is BEC, then $P\left(\mathcal{E}; W_N^{(i)}\right)$ is directly a function of bit-channel capacity $I\left(W_N^{(i)}\right)$. Therefore, the design of pre-transformed polar codes becomes finding the K most reliable bit-channels, which is well studied in [25]–[32].

On the contrary, if $S = \infty$, then SCL decoder becomes ML decoder and decoding error probability is given by

$$\begin{aligned} P_{\mathsf{SCL}}(\mathcal{E}; S = \infty) &= P_{\mathsf{SCL}}(\mathcal{E}_2; S = \infty) && (16)\\ &\lessapprox A_{d_{\min}} Q\left(\sqrt{2 d_{\min} R \frac{E_{\mathrm{b}}}{N_0}}\right). && (17)\end{aligned}$$

In this scenario, the decoding error event is primarily influenced by the minimum distance $d_{\min}$ and the number of min-weight codewords $A_{d_{\min}}$. Since $d_{\min}$ and $A_{d_{\min}}$ depend on both the rate-profile $\mathcal{I}$ and the pre-transform matrix $\mathbf{T}$, jointly

optimizing $\mathcal{I}$ and $\mathbf{T}$ is a challenging task for constructing pre-transformed polar codes that achieve high BLER performance. Most prior work has focused on finding pre-transform matrices that minimize $A_{d_{\min}}$ with a fixed $\mathcal{I}$ (e.g., Reed-Muller profiling) [20]–[22].

B. Rate Profiling Method to Reduce $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$

We explain how to design information index sets to diminish $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. Our principle for reducing the error event, where the correct codeword does not exist in the list, is to avoid using less reliable information bits consecutively. To clarify this principle, we adopt the decoder entropy analysis tool introduced in [36, Theorem 1], which was originally proposed to determine the required list size for the SCL decoder to achieve ML decoding performance.

Let $\mathcal{I}^{(m)} \triangleq \mathcal{I} \cap [0, m]$ and $\mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [0, m]$ be the sets containing information and frozen indices within the first $m+1$ input bits, respectively. The uncertainty of decoding paths when decoding u_m is characterized by the entropy of $\mathbf{U}_{\mathcal{I}^{(m)}}$ given $\mathbf{Y} = \mathbf{y}$ and $\mathbf{U}_{\mathcal{F}^{(m)}}$ as

$$d_m(\mathbf{y}) \triangleq H\left(\mathbf{U}_{\mathcal{I}^{(m)}} | \mathbf{Y} = \mathbf{y}, \mathbf{U}_{\mathcal{F}^{(m)}}\right). \tag{18}$$

From this definition, we can define the corresponding random variable $d_m(\mathbf{Y})$, which takes the value $d_m(\mathbf{y})$ when $\mathbf{Y} = \mathbf{y}$ according to the conditional distribution. To measure the uncertainty on $d_m(\mathbf{Y})$, we define the corresponding conditional entropy as

$$D_m \triangleq \mathbb{E}[d_m(\mathbf{Y})] = H\left(\mathbf{U}_{\mathcal{I}^{(m)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m)}}\right). \tag{19}$$

This conditional entropy, D_m, can be interpreted as decoding uncertainty. Consequently, it plays an important role in determining the required list size to achieve ML decoding performance under SCL decoding [36, Theorem 1].

To quantify the sole impact of decoding uncertainty introduced by the mth bit u_m, we define the incremental entropy of decoding u_m as the difference between D_m and D_{m-1} as follows:

$$\Delta_m \triangleq D_m - D_{m-1}. \tag{20}$$

The following lemma demonstrates that decoding uncertainty can increase or decrease depending on whether u_m is an information bit or a frozen bit.

Lemma 1: The incremental entropy by decoding u_m is

$$\begin{cases} \Delta_m \geq 0, & \text{if } m \in \mathcal{I} \\ \Delta_m \leq 0, & \text{otherwise.} \end{cases} \tag{21}$$

Proof: Although the proof is available in [36], we include it here for completeness. Suppose $m \in \mathcal{I}$. Then,

$$\begin{aligned} \Delta_m &= H\left(\mathbf{U}_{\mathcal{I}^{(m)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m)}}\right) - H\left(\mathbf{U}_{\mathcal{I}^{(m-1)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m-1)}}\right) \\ &= H\left(\mathbf{U}_{\mathcal{I}^{(m)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m-1)}}\right) - H\left(\mathbf{U}_{\mathcal{I}^{(m-1)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m-1)}}\right) \\ &= H\left(\mathbf{U}_{\mathcal{I}^{(m)}}, \mathbf{U}_{\mathcal{F}^{(m-1)}} | \mathbf{Y}\right) - H\left(\mathbf{U}_{\mathcal{I}^{(m-1)}}, \mathbf{U}_{\mathcal{F}^{(m-1)}} | \mathbf{Y}\right) \\ &= H(\mathbf{U}_{0:m-1} | \mathbf{Y}) + H(U_m | \mathbf{Y}, \mathbf{U}_{0:m-1}) - H(\mathbf{U}_{0:m-1} | \mathbf{Y}) \\ &= H(U_m | \mathbf{Y}, \mathbf{U}_{0:m-1}) \geq 0, \end{aligned} \tag{22}$$

where the last inequality follows from the fact that entropy is always greater than or equal to zero. Conversely, if $m \in \mathcal{F}$, we can compute the incremental entropy as

$$\begin{aligned} \Delta_m &= H(\mathbf{U}_{\mathcal{I}^{(m)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m)}}) - H(\mathbf{U}_{\mathcal{I}^{(m-1)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m-1)}}) \\ &= H(\mathbf{U}_{\mathcal{I}^{(m-1)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m-1)}}, U_m) - H(\mathbf{U}_{\mathcal{I}^{(m-1)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m-1)}}) \\ &= -I(U_m; \mathbf{U}_{\mathcal{I}^{(m-1)}} | \mathbf{Y}, \mathbf{U}_{\mathcal{F}^{(m-1)}}) \leq 0. \end{aligned} \tag{23}$$

■

From Lemma 1, if the information bits are chosen consecutively (e.g., $m, m+1 \in \mathcal{I}$), the decoder entropy continuously increases. The amount of this increment depends on the bit-channel capacities of channels $W_N^{(m)}(\mathbf{y}, \mathbf{u}_{0:m-1} | u_m)$ and $W_N^{(m+1)}(\mathbf{y}, \mathbf{u}_{0:m} | u_{m+1})$. When u_m and u_{m+1} are sent over sufficiently polarized bit-channels with $I\left(W_N^{(m)}\right) = I\left(W_N^{(m+1)}\right) \approx 1$, the decoder entropy does not increase, i.e., $H(U_m | \mathbf{Y}, \mathbf{U}_{0:m-1}) = H(U_{m+1} | \mathbf{Y}, \mathbf{U}_{0:m}) \approx 0$. As a result, the SC decoder is sufficient, provided that all information bits are transmitted over sufficiently polarized bit-channels. However, in the case of finite blocklengths, such channel polarization does not occur, and some information bits must be sent over less reliable bit-channels. When these bits are sent consecutively over less reliable bit-channels, the decoder entropy keeps increasing. In other words, the decoder requires a larger list size to maintain the transmitted codeword in the list.

When the list size is small, carefully designing information sets is crucial to keeping the transmitted codeword in the list. As a result, to reduce $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$, inserting frozen bits between information bits is an effective method since it prevents consecutive information bits. By inserting frozen bits between information bits, it is possible to reduce decoder entropy since $\Delta_m < 0$ for $m \in \mathcal{F}$, as shown in (23).

C. Bit-Swapping Method to Reduce $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$

We explain how exchanging information bits with frozen bits can reduce $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$. To begin, we introduce the relevant definitions. Given index $i \in [0, 2^n - 1]$, the binary representation of $i = \sum_{k=0}^{n-1} i_k 2^k$ is defined as $\mathsf{bin}(i) \triangleq [i_{n-1}, i_{n-2}, \ldots, i_1, i_0]$ and support of $\mathsf{bin}(i)$ as $\mathcal{S}_i = \mathsf{supp}(\mathsf{bin}(i)) \triangleq \{k \in [0, n-1] : i_k = 1\}$. Let $\mathcal{I} \subseteq [0, N-1]$ be the set of information indices satisfying the partial order property [30]–[32], [37] and $\mathcal{F} = \mathcal{I}^c = [0, N-1] \backslash \mathcal{I}$. Let $w_{\min} = \min_{i \in \mathcal{I}} \mathsf{wt}(\mathbf{g}_{N,i})$, i.e., the Hamming weight of the min-weight row. For $i \in \mathcal{I}$ such that $\mathsf{wt}(\mathbf{g}_{N,i}) = w_{\min}$, define the set $\mathcal{K}_i \triangleq \{j \in \mathcal{I} \backslash [0, i] : |\mathcal{S}_j \backslash \mathcal{S}_i| = 1\}$. It is equivalent to $\mathcal{K}_i = \{j \in [i+1, N-1] : \mathsf{wt}(\mathbf{g}_{N,j}) \geq \mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) = \mathsf{wt}(\mathbf{g}_{N,i})\}$ according to [38, Lemma 2].

Given a $(N, K, \mathcal{I})$ polar code, define a coset $\mathcal{C}_i(\mathcal{I})$ of codewords with the coset leader $\mathbf{g}_{N,i}$ as

$$\mathcal{C}_i(\mathcal{I}) \triangleq \left\{ \mathbf{g}_{N,i} \oplus \bigoplus_{h \in \mathcal{H}} \mathbf{g}_{N,h} : \mathcal{H} \subseteq \mathcal{I} \backslash [0, i] \right\}. \tag{24}$$

The coset $\mathcal{C}_i(\mathcal{I})$ generates min-weight codewords as follows. Every row $\mathbf{g}_{N,i} (i \in \mathcal{I})$ of the polar transform matrix $\mathbf{G}_N$, where $\mathsf{wt}(\mathbf{g}_{N,i}) = w_{\min}$, can form a min-weight codeword in

combination with the rows in every subset $\mathcal{J} \subseteq \mathcal{K}_i$ and the corresponding set $\mathcal{M}(\mathcal{J}) \subseteq (\mathcal{I} \cap [i+1, N-1]) \backslash \mathcal{K}_i$ as

$$\mathsf{wt}(\mathbf{g}_{N,i} \oplus \underbrace{\bigoplus_{j \in \mathcal{J}} \mathbf{g}_{N,j}}_{\text{core rows}} \oplus \underbrace{\bigoplus_{m \in \mathcal{M}(\mathcal{J})} \mathbf{g}_{N,m}}_{\text{balancing rows}}) = w_{\min}. \tag{25}$$

The rows in the set $\mathcal{J}$ are called *core* rows. The rows in the set $\mathcal{M}(\mathcal{J})$ are called *balancing* rows as their inclusion brings the weight of the sum down to $w_{\min}$ if needed. The set $\mathcal{M}(\mathcal{J})$ can be constructed by the $\mathcal{M}$-Construction described in [38, Section III-A].

Equation (25) shows one way to generate min-weight codewords, which gives a lower bound of the number of min-weight codewords of $(N, K, \mathcal{I})$ polar codes, denoted by $A_{d_{\min}}(\mathcal{I})$. Since every subset $\mathcal{J}$ of $\mathcal{K}_i$ corresponds to a different min-weight codeword, the total number of such codewords in every coset $\mathcal{C}_i(\mathcal{I})$ is lower bounded by the total number of subsets of $\mathcal{K}_i$, that is $2^{|\mathcal{K}_i|}$. This lower bound matches the result in [37, Propositions 6 and 7], which counts the number of min-weight codewords by harnessing the fact that lower-triangular affine (LTA) transformation group forms automorphism of decreasing monomial codes under the properly defined group action.

The formation of a min-weight codeword (25) is only complete when the information index set $\mathcal{I}$ follows a universal partial order [30]–[32], [37]. Otherwise, given a coset leader $\mathbf{g}_{N,i}$ and some core rows $\mathcal{J}$, it is possible that no index $i \in \mathcal{M}(\mathcal{J}) \cap \mathcal{I}$ exists, preventing the formation of a min-weight codeword. This implies that properly swapping an information bit with a frozen bit reduces the number of min-weight codewords, thereby reducing $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$.

To further illustrate, define the set $\mathcal{B}_{w_{\min}}(\mathcal{I})$ and $\mathcal{B}_{w_{\min}}(\mathcal{F})$

$$\mathcal{B}_{w_{\min}}(\mathcal{I}) \triangleq \{i \in \mathcal{I} : \mathsf{wt}(\mathbf{g}_{N,i}) = w_{\min}\}, \tag{26}$$

$$\mathcal{B}_{w_{\min}}(\mathcal{F}) \triangleq \{i \in \mathcal{F} : \mathsf{wt}(\mathbf{g}_{N,i}) = w_{\min}\}. \tag{27}$$

For each $j \in \mathcal{B}_{w_{\min}}(\mathcal{I})$, let us also define the set $\mathcal{D}_j$ and $\mathcal{G}_j$ as follows.

$$\mathcal{D}_j \triangleq \{i \in \mathcal{B}_{w_{\min}}(\mathcal{I}) : j \in \mathcal{K}_i\}, \tag{28}$$

$$\mathcal{G}_j \triangleq \{i \in \mathcal{B}_{w_{\min}}(\mathcal{F}) : j \in \mathcal{K}_i\}. \tag{29}$$

Given $j \in \mathcal{B}_{w_{\min}}(\mathcal{I})$ and $i \in \mathcal{G}_j$ satisfying,

$$\left(\sum_{x \in \mathcal{D}_j} 2^{|\mathcal{K}_x|-1} \right) + 2^{|\mathcal{K}_j|} > 2^{|\mathcal{K}_i|-1}, \tag{30}$$

according to [38], we have

$$A_{w_{\min}}(\mathcal{I}') \leq A_{w_{\min}}(\mathcal{I}) - \eta, \tag{31}$$

$$\eta = \left(\left(\sum_{x \in \mathcal{D}_j} 2^{|\mathcal{K}_x|-1} \right) + 2^{|\mathcal{K}_j|} - 2^{|\mathcal{K}_i|-1} \right) > 0, \tag{32}$$

where $\mathcal{I}' = \{i\} \cup (\mathcal{I} \backslash \{j\})$. This result implies that introducing less reliable min-weight rows as information indices while freezing more reliable min-weight rows, called *bit-swapping*, can reduce the number of min-weight codewords, thereby reducing $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$.

D. Local Pre-transform with Bit-Swapping

A local pre-transform with bit-swapping significantly reduces $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ while causing an increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. To explain this effect, we first define the local pre-transform. Let $\mathcal{A} = \{a_1, a_2, \ldots, a_{n_t}\} \subsetneq [N]$ be an index set and let $\mathbf{v} \in \mathbb{F}_2^{n_t}$ denote the input vector to the local pre-transform. To maintain the code rate, constraints must be applied to the input vector $\mathbf{v}$. Denote $\mathcal{I}$ and $\mathcal{F}$ as the information and frozen index sets, respectively, before applying the local pre-transform. Specifically, let $k_t = |\mathcal{A} \cap \mathcal{I}|$ and $\mathcal{I}_t = \{i_1, i_2, \ldots, i_{k_t}\} \subseteq [n_t]$, where $|\mathcal{I}_t| = k_t$. We allocate the information bits to $\mathbf{v}_{\mathcal{I}_t}$ and set $\mathbf{v}_{[n_t] \backslash \mathcal{I}_t} = \mathbf{0}$. The local pre-transform $\mathbf{T} \in \mathbb{F}_2^{n_t \times n_t}$ is an invertible upper-triangular matrix that transforms the input vector $\mathbf{v}$ into $\mathbf{vT}$, which is then assigned to $\mathbf{u}_{\mathcal{A}}$, where $\mathbf{u}$ is the input to the polar transform. The pre-transform introduces bit-swapping if $a_{i_j} \in \mathcal{F}$, for some index $i_j \in \mathcal{I}_t$. Here, we assume $i_1 < i_2 < \cdots < i_{k_t}$ and $a_1 < a_2 < \cdots < a_{n_t}$.

The local pre-transform with bit-swapping increases $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ as bit-swapping converts some frozen bits into information bits. These converted bits are less reliable and can impede the correct decision-making of the SCL decoder. At the same time, the local pre-transform with bit-swapping can decrease $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$. This is because the operation transforms certain information bits into dynamic frozen bits, defined as $u_{a_j} = \sum_{k=1}^{j-1} v_k T_{k,j}$. This transformation reduces the number of min-weight codewords [41], [43]. Since the error probability $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ strongly depends on the distance properties of codewords within the decoding list, a smaller number of min-weight codeword leads to a reduction in $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$.

E. Local Pre-transform without Bit-Swapping

The local pre-transform does not incur bit-swapping if $a_{i_j} \in \mathcal{I}$, for every index $i_j \in \mathcal{I}_t$. Based on experimental observations, such pre-transforms do not increase $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. Furthermore, by carefully selecting $\mathcal{A}$, we can definitely eliminate the min-weight codewords in $\mathcal{C}_i(\mathcal{I})$.

Theorem 1: Given $i, j \in [0, 2^n - 1]$ such that $\mathsf{wt}(\mathbf{g}_{N,i}) = w_{\min}$ and $\mathsf{wt}(\mathbf{g}_{N,j}) \geq w_{\min}$, suppose $j \preceq i$ with respect to universal partial order, i.e., $W_N^{(i)}$ is more reliable than $W_N^{(j)}$ [30], [31], [37]. For $\mathbf{c} \in \mathcal{C}_i(\mathcal{I})$, the following holds: $\mathsf{wt}(\mathbf{c} + \mathbf{g}_{N,j}) > w_{\min}$.

Proof: For an arbitrary set $\mathcal{I} \subseteq [0, N-1]$, we consider the min-weight codewords in $\mathcal{C}_i(\mathcal{I})$. Define the index set $\mathcal{X}_i = \{k \in [i, N-1] : \mathsf{wt}(\mathbf{g}_{N,k}) \geq w_{\min}\}$. Partition $\mathcal{X}_i$ into $\mathcal{X}_i^{(1)} = \{k \in \mathcal{X}_i : k \succeq i\}$ and its complement $\mathcal{X}_i \backslash \mathcal{X}_i^{(1)}$. Note that every min-weight codeword in $\mathcal{C}_i(\mathcal{I})$ must be a combination of rows in $\mathcal{X}_i$. Specifically, every min-weight codeword in $\mathcal{C}_i(\mathcal{I})$ must have the form described in (25) according to [44, Lemma 1], [37, Proposition 6] and [38, Proposition 2]. From [38, Lemma 6], it follows that the every element k in the set of core rows $\mathcal{J}$ and its corresponding balancing rows $\mathcal{M}(\mathcal{J})$ satisfies $k \succeq i$, meaning that k is more reliable than the coset leader i. That is, $(\mathcal{J} \cup \mathcal{M}(\mathcal{J})) \subseteq \mathcal{X}_i^{(1)}$. Therefore, if any rows $j \in \mathcal{X}_i \backslash \mathcal{X}_i^{(1)}$ are combined, a min-weight codeword cannot be generated. By the assumption $j \preceq i$, we have $j \in \mathcal{X}_i \backslash \mathcal{X}_i^{(1)}$. Hence, row j does not contribute to the formation of min-weight codewords, implying $\mathsf{wt}(\mathbf{c} + \mathbf{g}_{N,j}) > w_{\min}$. ■

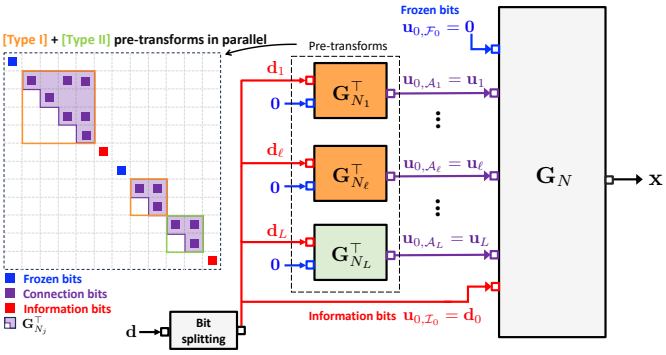

Fig. 2. Illustration of encoding structure of SPP codes. *Type-I* pre-transforms are marked in orange, and *Type-II* pre-transforms are marked in green.

Theorem 1 provides guidance on the design of pre-transform matrix toward reducing $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$. For each min-weight information row $i \in \mathcal{B}_{w_{\min}}(\mathcal{I})$ and subsequent frozen row $j \in \mathcal{F} \cap [i+1, N-1]$ such that $\mathsf{wt}(\mathbf{g}_{N,j}) \geq w_{\min}$, applying local pre-transform $\mathbf{G}_2^{\top}$ to indices $\{i, j\}$ prevent the formation of min-weight codewords in the coset $\mathcal{C}_i(\mathcal{I})$.

IV. SPP Codes

In this section, we introduce SPP codes, which improve upon deep polar codes [18], [19] under SCL decoder with small list sizes. The deep polar codes employ a serial multi-layered polar local pre-transform with bit-swapping. Although deep polar codes substantially decrease $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$, they suffer a considerable increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$, leading to performance degradation when S is small.

The SPP codes use multiple local polar pre-transform matrices in parallel, each of which belongs to either a *Type-I* or *Type-II* pre-transform. The *Type-I* pre-transform uses the transpose matrix of the polar transform kernel and aims to improve the overall error probability $P_{\mathsf{SCL}}(\mathcal{E}_1; S) + P_{\mathsf{SCL}}(\mathcal{E}_2; S)$. It accepts a slight increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ by adopting local pre-transform with bit-swapping. Despite this trade-off, it has the potential to significantly improve $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ compared to scenarios where no increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ is permitted. Unlike deep polar codes, the *Type-I* pre-transform follows the principle of inserting frozen bits between semi-polarized bits, limiting the number of consecutive semi-polarized bits. This approach reduces the potential increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. The *Type-II* pre-transform employs local pre-transform without bit-swapping, adopts the row-merging operation, and aims to improve $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ while maintaining $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$.

We first present the encoding process of SPP codes. Next, we present rate profile algorithm to determine connection indices for which the *Type-I* pre-transforms are applied. Lastly, we present a greedy algorithm to find row-merging pair to apply the *Type-II* pre-transforms.

A. Encoding

A $(N, K, \{\mathcal{I}_\ell\}_{\ell=0}^{L}, \{\mathcal{A}_\ell\}_{\ell=1}^{L}, \{\mathbf{T}_\ell\}_{\ell=1}^{L})$ SPP code is defined with the following parameters:

- i) L pre-transform matrices $\mathbf{T}_\ell \in \mathbb{F}_2^{N_\ell \times N_\ell}$, each belonging to either *Type-I* or *Type-II*;
- ii) $L+1$ information sets $\{\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_L\}$, and
- iii) L connection sets $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_L\}$.

The encoding process of the SPP code involves three steps: i) splitting message bits, ii) applying multiple polar pre-transforms, and iii) applying a polar transform. Fig. 2 illustrates the encoding process of the SPP code.

Information bit splitting and mapping: The information vector $\mathbf{d} \in \mathbb{F}_2^K$ is divided into $L+1$ sub-vectors $\mathbf{d}_\ell \in \mathbb{F}_2^{K_\ell}$, where $K_\ell = |\mathcal{I}_\ell|$ is the number of bits allocated to $\mathbf{d}_\ell$ and $\sum_{\ell=0}^{L} K_\ell = K$. We denote the input vector of the ℓth pre-transform by $\mathbf{v}_\ell = [v_{\ell,1}, v_{\ell,2}, \ldots, v_{\ell,N_\ell}] \in \mathbb{F}_2^{N_\ell}$ for $\ell \in [L]$. The index set $[N_\ell]$ for $\ell \in \{1, 2, \ldots, L\}$ is partitioned into two non-overlapping index sets as

$$[N_\ell] = \mathcal{F}_\ell \cup \mathcal{I}_\ell. \tag{33}$$

For the zeroth layer, the index set for the polar transform is divided into three non-overlapping sets as

$$[N] = \mathcal{F}_0 \cup \mathcal{I}_0 \cup \mathcal{A}_0, \tag{34}$$

where $\mathcal{A}_0 = \bigcup_{\ell=1}^{L} \mathcal{A}_\ell$ and $\mathcal{A}_i \cap \mathcal{A}_j = \phi$ for $i \neq j$. Each information sub-vector $\mathbf{d}_\ell$ for $\ell \in \{1, \ldots, L\}$ is assigned to the elements $v_{\ell,i}$ for $i \in \mathcal{I}_\ell$, i.e.,

$$\mathbf{v}_{\ell,\mathcal{I}_\ell} = \mathbf{d}_\ell, \tag{35}$$

and the zero bits are allocated to the frozen bits in $\mathcal{F}_\ell$ as

$$\mathbf{v}_{\ell,\mathcal{F}_\ell} = \mathbf{0}. \tag{36}$$

Parallel and local polar pre-transform: From the information and frozen bits assignment in (35) and (36), the input vector of the ℓ-layer pre-transform is $\mathbf{v}_\ell = [\mathbf{v}_{\ell,\mathcal{I}_\ell}, \mathbf{v}_{\ell,\mathcal{F}_\ell}]$. Note that we omit the index permutation for notational ease. The output vector of the ℓth pre-transform $\mathbf{u}_\ell \in \mathbb{F}_2^{N_\ell}$ is given by

$$\mathbf{u}_\ell = \mathbf{v}_\ell \mathbf{G}_{N_\ell}^{\top}, \tag{37}$$

where $\mathbf{G}_{N_\ell}$ is polar transform matrix with size N_ℓ and $\ell \in \{1, 2, \ldots, L\}$. In this work, we adopt $\mathbf{G}_{N_\ell}^{\top}$ for both *Type-I* and *Type-II* pre-transforms.

Global polar transform: Using the pre-transformed output vectors $\mathbf{u}_{0,\mathcal{A}_\ell} = \mathbf{u}_\ell$ for $\ell \in [L]$ in (37) and the information sub-vector $\mathbf{d}_0$, the encoder generates the input vector for the polar transform

$$\mathbf{u}_0 = [\mathbf{u}_{0,\mathcal{F}_0}, \mathbf{u}_{0,\mathcal{A}_1}, \ldots, \mathbf{u}_{0,\mathcal{A}_L}, \mathbf{u}_{0,\mathcal{I}_0}] \tag{38}$$

$$= [\mathbf{0}, \mathbf{u}_1, \ldots, \mathbf{u}_L, \mathbf{d}_0]. \tag{39}$$

Finally, applying the polar transform for the base layer, a sparesly pre-transformed codeword is constructed as

$$\mathbf{x} = \mathbf{u}_0 \mathbf{G}_N. \tag{40}$$

Our encoding method can be represented using a sparse pre-transform matrix $\mathbf{T} \in \mathbb{F}_2^{N\times N}$ with a block diagonal structure as

$$[\mathbf{0}, \mathbf{v}_1, \ldots, \mathbf{v}_L, \mathbf{d}_0] \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_1^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{G}_L^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\mathbf{T}} = [\mathbf{u}_{0,\mathcal{F}_0}, \mathbf{u}_{0,\mathcal{A}_1}, \ldots, \mathbf{u}_{0,\mathcal{A}_L}, \mathbf{u}_{0,\mathcal{I}_0}]. \quad (41)$$

The sub-block pre-transform matrices $\mathbf{G}_\ell^\top$ exhibit an upper triangular structure, as does $\mathbf{T}$. From [41], our pre-transform guarantees that the minimum distance of the code does not decrease after the transformation.

B. Type-I Pre-Transform and Rate Profile

The *Type-I* pre-transform implements three operations simultaneously: i) bit-swapping, ii) pre-transforming, and iii) limiting consecutive semi-polarized bits, by adopting multiple parallel local pre-transforms with carefully selected connection indices. The objective of the *Type-I* pre-transform is to reduce the number of min-weight codewords while accepting a small loss of polarization. To efficiently reduce $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ with minimal backoff of $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$, it is important to select connection indices where the outputs of the *Type-I* pre-transforms are applied. The consecutive semi-polarized bits should be limited, thereby reducing decoding entropy, as demonstrated in Section III-B.

Suppose an SPP code with L pre-transform matrices, comprised of L_1 *Type-I* pre-transform matrices and $L_2 = L - L_1$ *Type-II* pre-transform matrices. We explain how to select information set for the base layer $\mathcal{I}_0$ and connection sets $\mathcal{A}_\ell$ for $\ell \in [L_1]$, summarized in Algorithm 1. Define the synthesized channel

$$W_N^{(i)}(\mathbf{y}, \mathbf{u}_{1:i-1}|u_i) = \sum_{\mathbf{u}_{i+1:N}\in\mathbb{F}_2^{N-i}} \frac{1}{2^{N-1}} W^N(\mathbf{y}|\mathbf{x}), \quad (42)$$

where $W^N(\mathbf{y}|\mathbf{x})$ represents the N copies of B-DMCs $W(y|x)$. Let $n_c = \sum_{\ell=1}^{L_1} N_\ell$ denote the number of bits required for connection indices and introduce the ordered index set for $i_j \in [N]$ as

$$\mathcal{R} = \{i_j \in [N] : i_1, \ldots, i_{K_0+n_c}\} \quad (43)$$

where $I\left(W_N^{(i_1)}\right) \geq I\left(W_N^{(i_2)}\right) \geq \cdots \geq I\left(W_N^{(i_{K_0+n_c})}\right)$ and $I\left(W_N^{(i)}\right) > I\left(W_N^{(j)}\right)$ for any $i \in \mathcal{R}$ and $j \in [N]\backslash\mathcal{R}$.

We begin by partitioning the ordered index set $\mathcal{R}$ according to row weight of polar transform matrix, denoted by $\mathcal{B}_w(\mathcal{R})$, as

$$\mathcal{B}_w(\mathcal{R}) = \{i \in \mathcal{R} : \mathsf{wt}(\mathbf{g}_{N,i}) = w\} \quad (44)$$

$$= \{i_1^w, i_2^w, \ldots i_{|\mathcal{B}_w(\mathcal{R})|}^w\}, \quad (45)$$

where i_1^w is the least reliable and $i_{|\mathcal{B}_w(\mathcal{R})|}^w$ is the most reliable index in $\mathcal{B}_w(\mathcal{R})$. Let $w_{\min} = \min_{i\in\mathcal{R}} \mathsf{wt}(\mathbf{g}_{N,i})$. Using $\mathcal{B}_{w_{\min}}(\mathcal{R})$, we generate the auxiliary set $\tilde{\mathcal{R}}$ with the size of n_c, which is filled up with indices belonging to $\mathcal{B}_{w_{\min}}(\mathcal{R})$. In the process, the least reliable bit comes first. If the number of elements of $\tilde{\mathcal{R}}$ is less than n_c, we repeat the process using the next larger row weight index set $\mathcal{B}_{2w_{\min}}(\mathcal{R})$ and so on. With the auxiliary set $\tilde{\mathcal{R}}$, we generate information set $\mathcal{I}_0 = \mathcal{R}\backslash\tilde{\mathcal{R}}$. Subsequently, we reorder the elements within the set $\tilde{\mathcal{R}} = \left\{\tilde{i}_1, \ldots, \tilde{i}_{n_c}\right\}$ in a naturally ascending sequence with $\tilde{i}_j < \tilde{i}_u$ for $j < u$. Using this rearranged index set, the ℓth layer connection set is chosen as $\mathcal{A}_\ell = \left\{\tilde{i}_{\sum_{j=1}^{\ell-1} N_j+1}, \ldots, \tilde{i}_{\sum_{j=1}^{\ell} N_j}\right\}$ for $\ell \in [L_1]$. In addition, the information set $\mathcal{I}_\ell$ is constructed using RM-profiling such that

$$\mathcal{I}_\ell = \{i \in [N_\ell] : i_1, \ldots, i_{K_\ell}\} \quad (46)$$

where

$$\mathsf{wt}\left(\mathbf{g}_{N_\ell,i_1}^\top\right) \geq \mathsf{wt}\left(\mathbf{g}_{N_\ell,i_2}^\top\right) \geq \cdots \geq \mathsf{wt}\left(\mathbf{g}_{N_\ell,i_{K_\ell}}^\top\right). \quad (47)$$

Here, $\mathbf{g}_{N_\ell,i}^\top$ is the ith row of the matrix $\mathbf{G}_{N_\ell}^\top$.

The resulting connection set consists of min-weight rows and semi-polarized rows. Because each *Type-I* pre-transform includes frozen bits (since $K_\ell/N_\ell < 1$) and is applied to each connection index set $\mathcal{A}_\ell$, the number of consecutive semi-polarized bits is limited by $|\mathcal{A}_\ell|$. Depending on the selection of $\mathcal{I}_\ell$, the pair of swapped bits varies. For a better understanding of *Type-I* pre-transform including bit-swapping, consecutive semi-polarized bits, and their effects on SCL decoding, we present Example 1 to 3 in Section V.

The time complexity of Algorithm 1 is analyzed as follows:

- Line 1 to 4: The dominant factor in complexity is the computation and identification of the min-weight row. Note that $\mathcal{S}_i = \mathsf{supp}(\mathsf{bin}(i))$ and $\mathsf{wt}(\mathbf{g}_{N,i}) = 2^{|\mathcal{S}_{i-1}|}$, which requires $\mathcal{O}(\log N)$ operations. Iterating over all $i \in \mathcal{R}$, the complexity becomes $\mathcal{O}(|\mathcal{R}|N) \in \mathcal{O}(N \log N)$.
- Union of connection set: In the worst case, all elements in $\mathcal{R}$ are assigned to $\mathcal{A}$, which takes $\mathcal{O}(K_0 + n_c) \in \mathcal{O}(N)$.
- Rate-profile: Using bit-masking when constructing $\mathcal{A}$, the set difference operation takes $\mathcal{O}(|\mathcal{R}|) \in \mathcal{O}(N)$. Therefore, the sorting dominates the complexity, requiring $\mathcal{O}(N \log N)$.

Overall, Algorithm 1 has a time complexity of $\mathcal{O}(N \log N)$.

C. Type-II Pre-Transform

The *Type-II* pre-transform can reduce $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ without increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. To that end, we group some index $i \in \mathcal{I}$ with subsequent $j \in \mathcal{F}$, and apply pre-transform. As *Type-II* pre-transform matrix, we consider $\mathbf{G}_2^\top$, equivalent to row-merging operation.

To select the pair (i, j), we use Theorem 1. The objective is to merge some information bits with subsequent frozen bits to reduce the number of min-weight codewords. Let $\mathcal{M}_\ell = (\mathcal{M}_{\ell,1}, \mathcal{M}_{\ell,2})$ be the ℓth merged-pair, where $\mathcal{M}_{\ell,1}$ is information index and $\mathcal{M}_{\ell,2}$ is merged frozen index. Given information index $i \in \mathcal{B}_{w_{\min}}(\mathcal{I}_0)$ and all other previously determined merged-pair $\mathcal{M}_\ell$, the candidate of next merged index $\mathcal{P}_i$ is given as

$$\mathcal{P}_i \triangleq \{j \in [N]\backslash[i] : j \notin (\mathcal{A}_0 \cup \mathcal{I}_0), j \notin \bigcup_\ell \mathcal{M}_\ell\}, \quad (48)$$

Algorithm 1: Rate-profile

Data: L_1, $(K_\ell)_{\ell=0}^{L_1}$, $(N_\ell)_{\ell=1}^{L_1}$.
Result: $\{\mathcal{I}_\ell\}_{\ell=0}^{L_1}$,$\{\mathcal{A}_\ell\}_{\ell=1}^{L_1}$.

$n_c \leftarrow \sum_{\ell=1}^{L_1} N_\ell$ // required bits for *Type-I* pre-transform;
$\mathcal{R} \triangleq \{i_1, \ldots, i_{K_0+n_c}\}$ // ordered index set in terms of reliability;
$w \leftarrow \min_{i\in\mathcal{R}} \mathsf{wt}(\mathbf{g}_{N,i})$;
$\mathcal{A} \leftarrow \phi$;
/* *Union of connection set* */;
while $n_c > 0$ **do**
  $\mathcal{B}_w(\mathcal{R}) \leftarrow \{i \in \mathcal{R} : \mathsf{wt}(\mathbf{g}_{N,i}) = w\}$ // ordered set;
  if $|\mathcal{B}_w(\mathcal{R})| < n_c$ **then**
    $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{B}_w(\mathcal{R})$;
  else
    $\mathcal{A} \leftarrow$ the first n_c elements of $\mathcal{B}_w(\mathcal{R})$;
  end
  $n_c \leftarrow n_c - |\mathcal{B}_w(\mathcal{R})|$;
  $w \leftarrow 2w$;
end
/* *Rate-profile* */;
$\mathcal{I}_0 \leftarrow \mathcal{R}\backslash\mathcal{A}$;
$\mathcal{A} \leftarrow \mathsf{sort}(\mathcal{A}; \text{natural ascending order})$;
for $\ell = 1$ **to** L_1 **do**
  $\mathcal{A}_\ell \leftarrow$ the first N_ℓ elements of $\mathcal{A}$;
  $\mathcal{A} \leftarrow \mathcal{A}\backslash\mathcal{A}_\ell$;
  $\mathcal{I}_\ell \leftarrow$ top-N_ℓ indices with largest $\mathsf{wt}\left(\mathbf{g}_{N_\ell,i}^\top\right)$;
end
Return $\mathcal{I}_\ell$ and $\mathcal{A}_\ell$;

which collects subsequent frozen indices not belonging to any pre-transform.

According to Section III-E, the min-weight codewords are generated by the combination of min-weight row, core rows, and balancing rows. To disturb the formation of min-weight codeword, we select merged index $j \in \mathcal{P}_i$ if

$$\mathsf{wt}(\mathbf{g}_{N,j}) \geq w_{\min}. \tag{49}$$

Theorem 1 ensures that (49) eliminates the min-weight codewords in $\mathcal{C}_i(\mathcal{I})$. For each index in $i \in \mathcal{B}_{w_{\min}}(\mathcal{I}_0)$, we take a greedy approach by assigning the smallest index j satisfying (49) to $\mathcal{M}_{\ell,2}$.

Next, we repeat the same process using index $i \in (\mathcal{B}_{w_{\min}}(\mathcal{I}_0)\backslash(\bigcup_\ell \mathcal{M}_{\ell,1}))$ and takes index $j \in \mathcal{P}_i$ if

$$\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) > w_{\min}. \tag{50}$$

Again, we select the smallest such index j. Finally, we repeat the same process and takes any index $j \in \mathcal{P}_i$, which is equivalent to taking index such that

$$\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) = w_{\min}. \tag{51}$$

We summarize row-merging pair selection process in Algorithm 2. To further optimize the row-merged pair, the algorithms enumerating the number of min-weight codewords can be used such as [22] instead of greedy approach.

Algorithm 2: Design of *Type-II* pre-transform

Result: The merged pair $(\mathcal{M}_\ell)$
1 $w_{\min} \leftarrow \min_{i\in\mathcal{I}_0} \mathsf{wt}(\mathbf{g}_{N,i})$;
2 $\mathcal{B}_{w_{\min}}(\mathcal{I}_0) \leftarrow \{i \in \mathcal{I}_0 : \mathsf{wt}(\mathbf{g}_{N,i}) = w_{\min}\}$;
3 $\mathcal{I}_{w_{\min}} \leftarrow \mathcal{B}_{w_{\min}}(\mathcal{I}_0)$;
4 $\ell \leftarrow 0$;
5 state $\leftarrow 0$;
6 **while** state ≤ 2 **do**
7 **for** $i \in \mathcal{I}_{w_{\min}}$ **do** `// natural ascend. order`
8 $\mathcal{P} \leftarrow \{j \in [N]\backslash[i] : j \notin \mathcal{A}_0 \cup \mathcal{I}_0,\ j \notin \bigcup_\ell \mathcal{M}_\ell\}$;
9 **for** $j \in \mathcal{P}$ **do** `// natural ascend. order`
10 **if** state $= 0$ & $\mathsf{wt}(\mathbf{g}_{N,j}) \geq w_{\min}$ **then**
11 `addPair` (i, j);
12 **break** `// break inner for loop` ;
13 **end**
14 **if** state $= 1$ & $\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) > w_{\min}$ **then**
15 `addPair` (i, j);
16 **break**;
17 **end**
18 **if** state $= 2$ & $\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) = w_{\min}$ **then**
19 `addPair` (i, j);
20 **break**;
21 **end**
22 **end**
23 **end**
24 state $\leftarrow$ state $+ 1$;
25 $\mathcal{I}_{w_{\min}} \leftarrow \{i \in \mathcal{I}_{w_{\min}} : i \notin \mathcal{M}_{\ell,1}\}$;
26 **end**
27 **Return** $(\mathcal{M}_\ell)$;
28 **Function** `addPair` (i, j)**:**
29 $\mathcal{M}_\ell \leftarrow (i, j)$;
30 $\mathcal{I}_0 \leftarrow \mathcal{I}_0\backslash\{i\}$;
31 $\ell \leftarrow \ell + 1$;
32 **return**

The time complexity of Algorithm 2 is analyzed as follows:

- Line 1 to 4: Already performed in Algorithm 1.
- addPair operation: Using bit masking, the set difference operation has a complexity of $\mathcal{O}(1)$, and the saving pair $\mathcal{M}_\ell$ also takes $\mathcal{O}(1)$.
- While loop: The loop iterates up to 3 times since state $\in \{0, 1, 2\}$. For each state, the maximum number of addPair operations is quadratic. For condition checking, computing $\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) = 2^{|\mathcal{S}_{i-1}|} + 2^{|\mathcal{S}_{j-1}|} - 2^{1+|\mathcal{S}_{i-1}\cap\mathcal{S}_{j-1}|}$ requires $\mathcal{O}(\log N)$.

Overall, Algorithm 2 has a time complexity of $\mathcal{O}(N^2 \log N)$.

D. Discussion on Type-II Pre-Transform

To further understand how min-weight codewords are eliminated, we consider the merged pair (i, j). The ith row $\mathbf{g}_{N,i}$ contributes to the formation of min-weight codewords in two ways: i) as a coset leader of $\mathcal{C}_i(\mathcal{I} \cup \mathcal{A})$ and ii) as a core or balancing row of $\mathcal{C}_\ell(\mathcal{I}\cup\mathcal{A})$ with $\mathsf{wt}(\mathbf{g}_{N,\ell}) = w_{\min}$, $\ell \in \mathcal{I}\cup\mathcal{A}$.

Firstly, suppose the row $\mathbf{g}_{N,i}$ is a coset leader. We divide the merged pair (i, j) into three partitions:

- $\mathsf{wt}(\mathbf{g}_{N,j}) \geq w_{\min}$,
- $\mathsf{wt}(\mathbf{g}_{N,j}) < w_{\min}$ and $\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) > \mathsf{wt}(\mathbf{g}_{N,i})$,
- $\mathsf{wt}(\mathbf{g}_{N,j}) < w_{\min}$ and $\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) = \mathsf{wt}(\mathbf{g}_{N,i})$.

If $\mathsf{wt}(\mathbf{g}_{N,j}) \geq w_{\min}$, Theorem 1 explains the deletion of min-weight codewords in coset $\mathcal{C}_i(\mathcal{I} \cup \mathcal{A})$. Example 4 in Section V provides further illustration of Theorem 1. However, when $\mathsf{wt}(\mathbf{g}_{N,j}) < w_{\min}$, we can no longer rely on (25). Instead, we utilize the following results in [45, Theorem 5]. It states that given $i, j \in [0, N-1]$ such that $i < j$, for any $\mathbf{x} \in (\mathbf{g}_{N,i} + \mathcal{C}_j(\mathcal{I} \cup \mathcal{A}))$, the Hamming weight of generated codewords satisfy the following:

$$\mathsf{wt}(\mathbf{x}) \geq \mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}). \tag{52}$$

Because we select the closest subsequent merged index j, if $\mathsf{wt}(\mathbf{g}_{N,i} + \mathbf{g}_{N,j}) > \mathsf{wt}(\mathbf{g}_{N,i})$, a large portion of min-weight codewords in $\mathcal{C}_i(\mathcal{I} \cup \mathcal{A})$ is no longer min-weight codewords.

Secondly, suppose the row $\mathbf{g}_{N,i}$ is core or balancing row in coset $\mathcal{C}_\ell(\mathcal{I} \cup \mathcal{A})$ with $\mathsf{wt}(\mathbf{g}_{N,\ell}) = w_{\min}$. Depending on j, the inserted row acts as either core row or balancing row. We give an explanation based on the conjecture presented in [38]: i) if row $\mathbf{g}_{N,j}$ acts as the core row, it might require additional balancing row having lower reliability, which would be frozen row, and ii) if row $\mathbf{g}_{N,j}$ acts as the balancing row, it would become additional unnecessary balancing row since j is unreliable than ℓ. Although certain aspects of the *Type-II* pre-transform are conjectural, extensive simulation results indicate that the *Type-II* pre-transform effectively eliminates the min-weight codewords.

E. Design of SPP Codes

We propose a heuristic algorithm to determine the design parameters of the *Type-I* pre-transform (N_ℓ, K_ℓ). For simplicity, we assume in both our algorithm and simulations that that all ℓ share the same $N_\ell = N_1$ and $K_\ell = K_1$.

Algorithm description: The algorithm evaluates various combinations of (N_ℓ, K_ℓ), by computing the minimum distance $d_{\min}$ and the number of min-weight codewords $A_{d_{\min}}$, and selects the parameters with the largest $d_{\min}$. If there are ties in $d_{\min}$, the combination with the smallest $A_{d_{\min}}$ is chosen. The ML decoding performance bound can be further utilized to select the best parameter.

Additional constraints are introduced to accommodate the scenario of SCL decoding with small list sizes, which is main focus of our paper. These include the number of consecutive bits and the total number of additionally introduced bits, defined as $n_p = \sum_\ell (N_\ell - K_\ell) = L_1(N_1 - K_1)$. A larger n_p makes it easier to find parameters with a larger $d_{\min}$. However, for SCL decoder with small list sizes, an increase in $P(\mathcal{E}_1; S)$ due to a larger n_p may degrade performance. Furthermore, reducing the number of consecutive bits, limited by K_1, decreases $P(\mathcal{E}_1; S)$. Thus, we limit n_p and K_1 and evaluate $d_{\min}$ and $A_{d_{\min}}$ for feasible combinations of (N_1, K_1, L_1) to select the optimal parameters.

The feasible combinations of (N_1, K_1, L_1) must satisfy the following conditions:

$$\text{(i) } K \geq L_1 K_1, \qquad \text{(ii) } N \geq K + L_1(N_1 - K_1),$$
$$\text{(iii) } L_1 \leq \frac{C_1}{N_1 - K_1}, \qquad \text{(iv) } K_1 \leq C_2. \tag{53}$$

The condition (i) ensures that the number of message bits input into the pre-transform is less than or equal to the total number of message bits. The condition (ii) ensures that the sum of the size of the connection vector and the message vector $\mathbf{m}_0$ (without pre-transform) does not exceed the input size of the polar transform. The last two conditions restrict the number of consecutive bits and the number of introduced bits, where C_1 and C_2 are constants.

Introduce a notation $[\![N_1, K_1]\!] \triangleq \{(N_1, K_1, \ell)\}_{\ell=1}^{\ell=C_1/(N_1-K_1)}$, a collection of feasible combinations with $N_\ell = N_1$ and $K_\ell = K_1$. We generate the combinations of (N_1, K_1, ℓ) in the following order:

$$[\![2,1]\!] \mapsto [\![4,3]\!] \mapsto [\![4,2]\!] \mapsto [\![4,1]\!] \mapsto [\![8,7]\!] \mapsto [\![8,6]\!] \mapsto \cdots. \tag{54}$$

Computational complexity: The number of combinations in $[\![N_1, K_1]\!]$ is given as follows: C_1 from $[\![N_1, N_1 - 1]\!]$, $C_1/2$ from $[\![N_1, N_1 - 2]\!]$, $C_1/3$ from $[\![N_1, N_1 - 3]\!]$, and so on. Assuming $N_\ell \leq 2^m$, the total number of combinations is upper-bounded by

$$mC_1 \left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{2^m - 1}\right) = mC_1 H_{2^m - 1}, \tag{55}$$

where H_n is the nth harmonic number. The total complexity for evaluating all combinations is

$$mC_1 H_{2^m-1} \times T_{\min-\text{weight}}, \tag{56}$$

where $T_{\min-\text{weight}}$ is time complexity for evaluating $(d_{\min}, A_{d_{\min}})$.

For instance, let $m = 5$ and $C_1 = 10$. Then, (55) is approximated by $5 \times 10 \times 4.02725 \approx 202$. In the case of PAC codes, if we assume the maximum polynomial degree for convolution to be p, the number of combinations is 2^p. Letting $p = 10$ yields the following: $2^p > mC_1 H_{2^m-1}$.

V. Examples

We present some examples illustrating the effect of *Type-I* and *Type-II* pre-transform.

Example 1 (*Type-I* pre-transform and bit-swapping): Consider a short packet transmission scenario, in which a transmitter sends a codeword with a blocklength of $N = 16$ over the BEC with an erasure probability of $1/2$, denoted as $I(W_{\mathsf{BEC}}) = 0.5$. The corresponding bit channel capacity and normalized row weight are illustrated in Fig. 3. In this example, we present the effect of *Type-I* pre-transform. Consider a polar code with a code rate of $R = 8/16$. The information index set is given by $\mathcal{I}_{\mathsf{polar}} = \{8, 10, 11, 12, 13, 14, 15, 16\}$, which offers the highest bit-channel capacity. If bit-swapping is not allowed, only the 9th bit can be converted into dynamic frozen bit. Since the 8th bit is the only preceding information bit, the only feasible dynamic frozen bit pattern is $u_9 := u_8$,

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2025.3555900

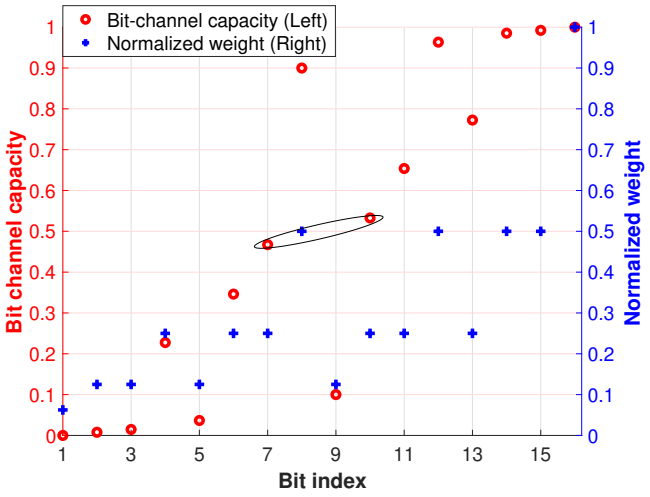

Fig. 3. (Example 1) Bit channel capacity (left) and normalized row weight (right) on binary erasure channel (BEC) with erasure probability of $1/2$.

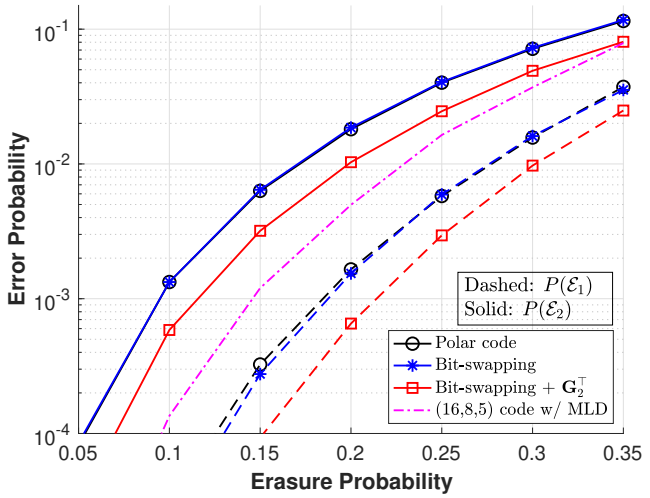

Fig. 4. (Example 2) $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ and $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ with list size of $S = 2$. The $(16, 8, 5)$ code is a binary linear code with $d_{\min} = 5$, which is the largest $d_{\min}$ for $(N, K) = (16, 8)$ that satisfies the Hamming bound. The generator matrix is constructed such that the ith row is a right circular shift by $(i-1)$ position of the vector $[1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$.

TABLE I
COMPARISON OF THE WEIGHT DISTRIBUTIONS

Weight	0	4	6	8	10	12	16
Polar code	1	28	-	198	-	28	1
$\mathbf{G}_2^\top$ $(u_9 := u_8)$	1	28	-	198	-	28	1
Bit-swapping $(7 \leftrightarrow 10)$	1	28	-	198	-	28	1
Bit-swapping & $\mathbf{G}_2^\top$	1	12	64	102	64	12	1

which preserves the weight spectrum and does not eliminate min-weight codewords (see 1st and 2nd rows of Table I). Now, introduce the one additional bit u_7, which is the most reliable bit among $\mathcal{I}^c_{\mathsf{polar}}$. Following our rate-profile, we construct connection index set as $\mathcal{A}_1 = \{7, 10\}$, which consists of two most unreliable bits with minimum row weight. Now, if we use pre-transform $\mathbf{G}_2^\top$ with $\mathcal{I}_1 = \{1\}$ and $\mathcal{F}_1 = \{2\}$ to maintain a code rate, the number of min-weight codewords $A_{\mathrm{d_{min}}}$ reduces from 28 to 12 as shown in Table I.

Example 2 (*Type-I* pre-transform and decoding error): Through this example, we explain that our *Type-I* pre-transform decreases $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ while nearly maintaining $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. Consider the same scenario with Example 1. Because $I\left(W_{16}^{(7)}\right) < I\left(W_{16}^{(10)}\right)$, the SC decoding performance becomes worse due to bit-swapping. If we use SCL decoder, however, the pre-transformed polar code can achieve better performance compared to pure polar code due to improved weight spectrum as shown in Table I. For an intuitive explanation, let us assume SCL decoding with a list size of 2 and perfect decoding of $\hat{u}_8$. Considering the decoding of u_{11}, the decoding path of pure polar code is $(\hat{u}_8, \hat{u}_{10}) = \{(\hat{u}_8, 0), (\hat{u}_8, 1)\}$ and the decoding path of bit-swapped polar code is $(\hat{u}_7, \hat{u}_8) = \{(0, \hat{u}_8), (1, \hat{u}_8)\}$. Due to the assumption of perfect estimation of $\hat{u}_8$, the above decoding path includes every possible combination of previous bits. From the perspective of the decoding subsequent bits, the behavior of the decoder is nearly identical for both codes, even if an incorrect candidate in the bit-swapped polar code passes through one additional frozen bit, u_{10}, thereby reducing decoding entropy. That is, the bit-swapped polar code exhibits nearly identical $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ when $S \geq 2$. Meanwhile, applying the pre-transform enhances the weight spectrum and reduces $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$. Fig. 4 presents the corresponding $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ and $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$. Notably, the reduction in the number of min-weight codewords achieved through the pre-transform enhances $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ as well as $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$.

Example 3 (*Type-I* pre-transform and consecutive bits): We use path metric range, introduced in [35], [46], to observe that non-consecutive information bits produced by *Type-I* pre-transform reduces $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. If we denote the path metric of the sth decoding path for the ith decoding step as $\mathsf{PM}_i[s]$, the path metric range for the ith decoding step is defined by their maximum difference,

$$\mathsf{PMR}_i = \max_s \mathsf{PM}_i[s] - \min_s \mathsf{PM}_i[s]. \tag{57}$$

A large path metric range indicates a significant difference between the most convincing decoding path and the uncertain paths in the list, signifying a low likelihood of the correct decoding path being discarded.

We present two different *Type-I* pre-transform at two different code rates $(N, K) = (128, 32)$ and $(N, K) = (128, 96)$. First, consider $(N, K) = (128, 32)$. The first pre-transform configuration is obtained by parameters $(N_\ell)_{\ell=1}^3 = (2, 2, 16)$ and $(K_\ell)_{\ell=1}^3 = (1, 1, 10)$, leading to $\mathcal{A}_1 = \{32, 48\}$, $\mathcal{A}_2 = \{56, 60\}$, and $\mathcal{A}_3 = \{62, 63, \ldots, 121\}$. It uses 8 additional bits (i.e., 8 bit-swapping), resulting in $(d_{\min}, A_{d_{\min}}) = (24, 416)$. The second pre-transform configuration is obtained by $(N_\ell)_{\ell=1}^2 = (4, 16)$ and $(K_\ell)_{\ell=1}^2 = (3, 10)$, resulting in $\mathcal{A}_1 = \{32, 48, 56, 60\}$ and $\mathcal{A}_2 = \{62, 63, \ldots, 121\}$. It uses 7 additional bits (i.e., 7 bit-swapping), and $(d_{\min}, A_{d_{\min}}) = (24, 224)$. For the first configuration, when decoding u_{56}, since u_{48} is a dynamic frozen bit, the gap of the path metric between decoding paths is large, as shown in Fig. 5 (top). However, for the second configuration, the consecutive semi-polarized bits

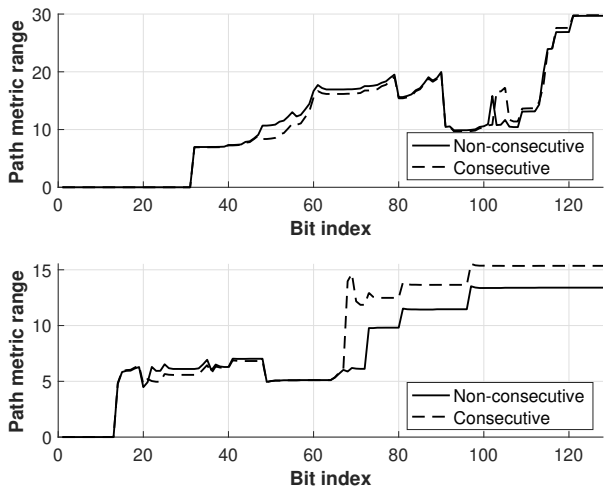

Fig. 5. (Example 3) Path metric range evaluated with list size of $S = 2$ averaged over 10^3 channel noise realizations generated at $E_b/N_0 = 3$ dB. Top: $(N, K) = (128, 32)$, Bottom: $(N, K) = (128, 96)$.

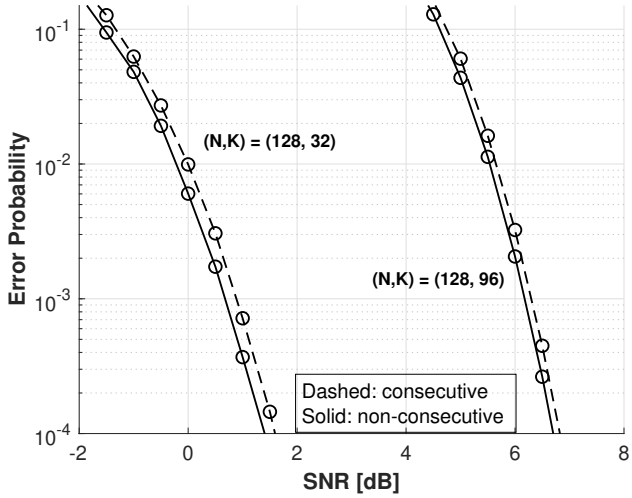

Fig. 6. (Example 3) $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ with list size of $S = 2$.

(u_{32}, u_{48}) make the correct decoding path be discarded due to a relatively small path metric range, leading to decoding performance loss.

Consider $(N, K) = (128, 96)$. The first configuration is obtained by $(N_\ell)_{\ell=1}^3 = (2, 2, 16)$ and $(K_\ell)_{\ell=1}^3 = (1, 1, 13)$, leading to $\mathcal{A}_1 = \{14, 15\}, \mathcal{A}_2 = \{20, 22\}$, and the second configuration is obtained by $(N_\ell)_{\ell=1}^2 = (2, 16)$ and $(K_\ell)_{\ell=1}^2 = (1, 12)$, leading to $\mathcal{A}_1 = \{14, 15\}, \mathcal{A}_2 = \{20, 22, 23, 26, \ldots\}$. Both configurations introduce 5 additional bits. For the first configuration, when decoding u_{24}, since u_{22} is frozen bit, the gap of path metric between decoding paths is large as shown in Fig. 5 (bottom). However, for the second configuration, the small path metric range worsens the decoding performance when decoded with the small list size. Fig. 6 presents the corresponding $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. It can be observed that a larger path metric range corresponds to smaller $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ when the list size is small.

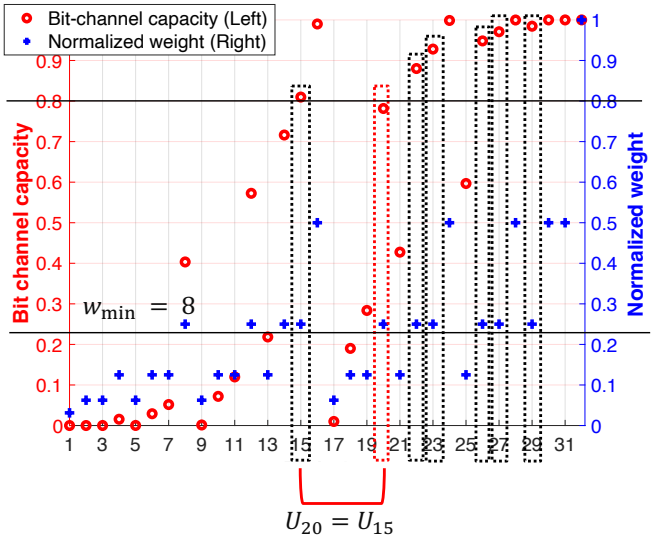

Fig. 7. (Example 4) Bit channel capacity (left) and normalized row weight (right) on binary erasure channel (BEC) with erasure probability of 1/2.

TABLE II
THE NUMBER OF MIN-WEIGHT CODEWORDS IN $\mathcal{C}_i(\mathcal{I})$ BEFORE AND AFTER THE TYPE-II PRE-TRANSFORM

Bit index	15	22	23	26	27	29
Before	32	32	16	16	8	4
After	0	32	16	16	8	4

Example 4 (*Type-II* pre-transform and Theorem 1): We present an example illustrating *Type-II* pre-transform and Theorem 1. Consider a polar code with blocklength $N = 32$, for which the bit channel capacity and normalized row weight are depicted in Fig. 7. Define the information set as $\mathcal{I} = \{i \in [N] : I\left(W_N^{(i)}\right) > 0.8\}$. Observe that a pair of indices $(15, 20)$ satisfies the condition of Theorem 1. Accordingly, all the min-weight codewords in $\mathcal{C}_{15}(\mathcal{I})$ are deleted, as shown in Table II.

VI. Simulation Results

We present the BLER performance of the proposed codes under SCL decoding on the BI-AWGN channel to evaluate the decoding performance.

A. Construction of SPP Codes and Benchmarks

We consider the following existing codes and benchmarks.

- **CA polar:** The 5G-NR channel-independent reliability sequence [47] and CRC polynomial $g_{\mathsf{CRC11}}(x) = x^{11} + x^{10} + x^9 + x^5 + 1$ is used.
- **PAC:** We use different rate-profile in each figure. Fig. 8 is simulated using RM information set, where equality is broken by 5G information set. Fig. 9 is simulated using 5G information set. The convolution polynomial is optimized to reduce the number of min-weight codewords over degree less than 10.
- **DeepPolar:** A deep polar code is a novel variant of pre-transformed polar codes where the pre-transform

TABLE III
CONSTRUCTION OF SPP CODES

Fig.	N	R	CRC	*Type-I* (N_ℓ, K_ℓ) $\ell=1$	$\ell=2$	$\ell=3$	$\ell=4$	$\ell=5$	Weight $d_{\min}$	$A_{d^{\min}}$
8	128	1/4	ϕ	$(8,3)$	$(8,3)$	-	-	-	24	224
		2/4	ϕ	$(8,2)$	-	-	-	-	12	128
		3/4	ϕ	$(8,3)$	-	-	-	-	8	16560
9	256	1/4	g_{CRC3}	$(4,1)$	$(4,1)$	-	-	-	32	420
		2/4	g_{CRC3}	$(4,2)$	-	-	-	-	16	1934
		3/4	g_{CRC3}	$(8,7)$	$(8,7)$	$(8,7)$	$(8,7)$	$(8,7)$	8	2370
10	512	1/4	ϕ	$(32,24)$	-	-	-	-	32	64
		2/4	ϕ	$(32,24)$	-	-	-	-	16	1344
		3/4	ϕ	$(32,24)$	-	-	-	-	8	192

** Reliability order: 5G sequence.
** CRC polynomial: $g_{\mathsf{CRC3}}(x) = 1 + x + x^3$.
** Estimated $d_{\min}$ is computed using SCL decoder with list size $S = 5 \times 10^4$.

consists of multi-layered nested polar encoding [18]. Fig. 8 is simulated using $(N_\ell)_{\ell=1}^3 = (2, 16, 128)$ and $(K_\ell)_{\ell=1}^3 = (1, 10, K-11)$, and Fig. 9 is simulated using $(N_\ell)_{\ell=1}^3 = (2, 32, 256)$ and $(K_\ell)_{\ell=1}^3 = (1, 26, K-27)$.

- **SPP:** The 5G-NR information set [47] is used. The remaining detail of construction is listed in Table III.
- **Theoretical bounds:** We use two finite blocklength information theoretical bounds: i) random coding union (RCU) bound and ii) meta-converse bound, which are implemented based on [48] and [49]. These bounds allow to access the gaps between the achievable decoding performance of our methods and mathematical bounds.

B. The Effect of Type-I and II Pre-Transform

To evaluate the two distinct benefits of the transforms, we provide simulation results that substantiate the following claims:

- ***Type-I* pre-transform gain**: The simulation results comparing $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ and $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ before and after applying the *Type-I* pre-transform show that although $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ usually increases slightly, $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ decreases significantly. As a result, the overall error probability decreases.
- ***Type-II* pre-transform gain**: The simulation results comparing $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ and $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ before and after applying the *Type-II* pre-transform (in combination with a specific *Type-I* pre-transform) indicate that $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ is typically remains unchanged or slightly improves, while $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ shows a significant reduction.
- ***Type-I* + *Type-II* pre-transform gain**: Without *Type-I* pre-transform (bit-swapping), applying only the *Type-II* pre-transform has limited effectiveness in eliminating min-weight codewords. This is also demonstrated in Example 1.

First, we analyze the role of the *Type-I* pre-transform by evaluating its performance under two different scenarios: i) $N = 128, K = 48$, and ii) $N = 128, K = 84$. In both cases, simulations are conducted over a Gaussian channel using BPSK modulation, the 5G information sequence, and an SCL decoder with a list size of $S = 2$. The *Type-I* pre-transform $\mathbf{G}_2^\top$ is applied with an information index set $\mathcal{I} = \{1\}$ and a frozen index set $\mathcal{F} = \{2\}$. The results are summarized in Table IV.

By applying $\mathbf{G}_2^\top$, one additional information bit is introduced, which leads to an increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. However, after applying the pre-transform, either the minimum distance increases or the number of min-weight codewords decreases. The resulting reduction in $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ is significantly greater than the increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. Consequently, the overall error probability $P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2; S)$ improves compared to its value before applying the pre-transform.

Second, we examine the benefits of the *Type-II* pre-transform. For this evaluation, the *Type-I* pre-transform is fixed as $\mathbf{G}_2^\top$, with the information index set $\mathcal{I} = \{1\}$ and the frozen index set $\mathcal{F} = \{2\}$. Following this, the *Type-II* pre-transform is applied. This additional pre-transform effectively eliminates certain min-weight codewords; notably, when $K = 48$, all min-weight codewords are removed. The *Type-II* pre-transform operates by converting specific subsequent static frozen bits into dynamic frozen bits. This process typically does not increase $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ but significantly reduces the number of min-weight codewords. Consequently, this reduction leads to a marked improvement in $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$.

Finally, we examine the combined effect of the *Type-I* and *Type-II* pre-transforms. For this analysis, we consider the case where $N = 128$ and $K = 48$. Without applying the *Type-I* pre-transform, it is impossible to identify any combinations that satisfy the three conditions necessary for constructing a *Type-II* pre-transform. However, when the *Type-I* pre-transform $\mathbf{G}_2^\top$ is applied, suitable combinations for applying the *Type-II* pre-transform can be identified, resulting in a notable improvement in decoding performance. A similar scenario is illustrated in Example 1 in Section V.

C. The Effect of Design Parameters

We consider the following parameters:

- The size of pre-transforms (N_ℓ, K_ℓ)
- The number of *Type-I* pre-transforms L_1
- The number of additional bits $n_p = \sum_{\ell=1}^{L_1} (N_\ell - K_\ell)$

The effect of K_ℓ: To investigate this effect, we present simulation results in Table V. Several *Type-I* pre-transforms

TABLE IV
THE ROLE OF TYPE-I AND TYPE-II PRE-TRANSFORM ($N = 128, S = 2$)

		$K = 48$, SNR $= 2$ dB				$K = 84$, SNR $= 5$ dB			
Type-I	*Type-II*	$d_{\min}$	$A_{d_{\min}}$	$P_{\mathsf{SCL}}(\mathcal{E}_1)$	$P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2)$	$d_{\min}$	$A_{d_{\min}}$	$P_{\mathsf{SCL}}(\mathcal{E}_1)$	$P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2)$
ϕ	X	8	16	7.20×10^{-4}	4.31×10^{-3}	8	9776	1.28×10^{-3}	2.80×10^{-3}
ϕ	O	8	16	7.20×10^{-4}	4.31×10^{-3}	8	6480	1.21×10^{-3}	2.27×10^{-3}
$\mathbf{G}_2^{\top}$	X	16	3864	9.32×10^{-4}	1.51×10^{-3}	8	7728	1.34×10^{-3}	2.59×10^{-3}
$\mathbf{G}_2^{\top}$	O	16	1624	7.13×10^{-4}	1.03×10^{-3}	8	4432	1.24×10^{-3}	2.09×10^{-3}

** *Type-I* pre-transform is $\mathbf{G}_2^{\downarrow}$ with $\mathcal{I} = \{1\}$, $\mathcal{F} = \{2\}$.

are applied to observe how $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ and $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$ vary depending on the size of K_ℓ. We consider two scenarios: i) $N = 128, K = 48$, and ii) $N = 128, K = 84$. The pre-transform $\mathbf{G}_{N_\ell}^{\top}$ is applied with sizes $(N_\ell, K_\ell) \in \{(4,3),(8,7),(16,15),(32,31)\}$. The information index set for pre-transform is $\mathcal{I} = [1, N_\ell - 1]$, and the frozen index set is $\mathcal{F} = \{N_\ell\}$.

Depending on the situation, K_ℓ sets an upper bound on the number of consecutive semi-polarized bits. As K_ℓ increases, this upper bound also increases. As shown in Table V, increasing K_ℓ generally leads to an increase in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. Since the list size is small ($S = 2$), the decoding error event is primarily dominated by the first type $\mathcal{E}_1$, i.e., the event where correct decoding candidate is not included in the list. This trend is further supported by the observed increase in $P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2; S)$. Therefore, when the list size is small and $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ dominates the error events, setting K_ℓ to a smaller value can effectively limit the number of consecutive information bits, leading to better decoding performance.

The effect of N_ℓ and L_1: An increase in L_1 leads to a rise in the number of additional bits, n_p. Similarly, for a given L_1 and K_ℓ in each layer, increasing N_ℓ also results in a higher value of n_p. As n_p grows, the patterns and positions of the parity bits become more diverse, offering greater flexibility to reduce $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$. However, this increase in n_p also introduces some negative effects on $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. These dual effects are further discussed in the following paragraph.

The number of additional bits $n_p = \sum_{\ell=1}^{L_1}(N_\ell - K_\ell)$: A larger value of n_p leads to an increased number of bit-swapping operations, a higher count of less reliable information bits, and a greater number of more reliable frozen bits. These effects have two sides. On one hand, the increase in less reliable bits negatively impacts the SCL decoder, which depends on channel polarization properties. The presence of these bits raises the decoding entropy, necessitating a larger list size to maintain ML decoding performance, which in turn results in a higher $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. On the other hand, the increase in subsequent frozen bits enhances flexibility in the pattern and positioning of dynamic frozen bits. This added degree of freedom enables better weight distribution, leading to a reduction in $P_{\mathsf{SCL}}(\mathcal{E}_2; S)$.

To illustrate, we present simulation results in Table VI providing values of $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ and $P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2; S)$ for different list sizes $S = 2$ and $S = 8$ and two values of n_p. First, the main factor influencing the BLER, i.e., $P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2; S)$, varies depending on the list size. For $S = 2$, we observe a clear relationship between K_ℓ, $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$, and $P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2; S)$. As K_ℓ increases, $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ also increases, which directly results in an increase in $P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2; S)$. For $S = 8$, the BLER $P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2; S)$ is primarily influenced by $d_{\min}$ and $A_{d_{\min}}$, at least within the considered scenarios. Second, for $S = 2$, it is essential to use smaller values of n_p and K_ℓ to achieve meaningful improvements in $d_{\min}$ and $A_{d_{\min}}$ without excessively increasing $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$. Third, for $S = 8$, leveraging the flexibility provided by a larger n_p is important. While a slight degradation in $P_{\mathsf{SCL}}(\mathcal{E}_1; S)$ may occur, the reduction in $A_{d_{\min}}$ is the dominant factor for minimizing the BLER.

D. BLER Comparison

BLER comparison with PAC codes: Fig. 8 and Fig. 9 depict the BLER performance of the considered codes at $N = 128$ and $N = 256$, respectively. The results demonstrate that our method outperforms PAC codes at various blocklengths and code rates. The decoding performance of PAC codes is inferior due to the RM rate profile (Fig. 8), which aims to improve minimum distance, and due to a large number of min-weight codewords (Fig. 9) when using the 5G information set. These findings indicate that sparse pre-transform is effective for SCL decoders with small list sizes by limiting the number of consecutive less reliable information bits while simultaneously decreasing the nubmer of min-weight codewords.

BLER comparison with CA polar codes: We consider the blocklength $N \in \{128, 256, 512\}$ and code rate $R \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$. Fig. 8 and Fig. 9 depict the BLER performance of considered codes at $N = 128$ and $N = 256$, respectively. We simulate our proposed and CA-polar codes using an SCL decoder with a list size of $S = 8$. Fig. 10 depict the BLER performance of considered codes at $N = 512$. We simulate our proposed and CA-polar codes using an SCL decoder with list sizes of $S = 2$ and $S = 8$. The results demonstrate that our method outperforms CA-polar codes at various blocklengths and code rates. In particular, the sparse pre-transform gain becomes more significant at short blocklength and low rates, even leading to approximately 1 dB coding gain at blocklength $N = 128$ and code rate $R = \frac{1}{4}$.

VII. CONCLUSION

We have introduced a novel type of pre-transformed polar code called SPP, which improves upon deep polar codes for low-latency SCL decoding. The main technical innovation

TABLE V
THE EFFECT OF K_ℓ $(N = 128, S = 2)$

	$K = 48$, SNR $= 2$ dB				$K = 84$, SNR $= 5$ dB			
Type-I	$d_{\min}$	$A_{d_{\min}}$	$P_{\mathsf{SCL}}(\mathcal{E}_1)$	$P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2)$	$d_{\min}$	$A_{d_{\min}}$	$P_{\mathsf{SCL}}(\mathcal{E}_1)$	$P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2)$
ϕ	8	16	7.20×10^{-4}	4.31×10^{-3}	8	9776	1.28×10^{-3}	2.80×10^{-3}
$\mathbf{G}_4^{\top}$	16	3352	9.21×10^{-4}	1.44×10^{-3}	8	5680	1.27×10^{-3}	2.24×10^{-3}
$\mathbf{G}_8^{\top}$	16	3608	9.49×10^{-4}	1.50×10^{-3}	8	5680	1.60×10^{-3}	2.49×10^{-3}
$\mathbf{G}_{16}^{\top}$	8	8	10.5×10^{-4}	2.90×10^{-3}	8	5424	1.67×10^{-3}	2.52×10^{-3}
$\mathbf{G}_{32}^{\top}$	8	8	10.9×10^{-4}	2.94×10^{-3}	8	5360	1.70×10^{-3}	2.51×10^{-3}

** No *Type-II* pre-transform.

TABLE VI
THE EFFECT OF n_p $(N = 128, K = 48)$

	Type-I (N_ℓ, K_ℓ)		Weight		$S = 2$, SNR $= 2$ dB		$S = 8$, SNR $= 1.5$ dB	
n_p	$\ell = 1$	$\ell = 2$	$d_{\min}$	$A_{d_{\min}}$	$P_{\mathsf{SCL}}(\mathcal{E}_1)$	$P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2)$	$P_{\mathsf{SCL}}(\mathcal{E}_1)$	$P_{\mathsf{SCL}}(\mathcal{E}_1 \cup \mathcal{E}_2)$
0	-	-	8	16	7.20×10^{-4}	4.31×10^{-3}	1.81×10^{-4}	10.3×10^{-3}
2	$(4, 2)$	-	16	2840	0.89×10^{-3}	1.35×10^{-3}	1.77×10^{-4}	2.91×10^{-3}
	$(8, 6)$	-	16	2328	1.21×10^{-3}	1.57×10^{-3}	2.97×10^{-4}	2.53×10^{-3}
4	$(8, 4)$	-	16	2072	0.99×10^{-3}	1.36×10^{-3}	1.73×10^{-4}	2.13×10^{-3}
	$(8, 6)$	$(8, 6)$	16	1240	1.53×10^{-3}	1.74×10^{-3}	3.03×10^{-4}	1.62×10^{-3}

** No *Type-II* pre-transform.

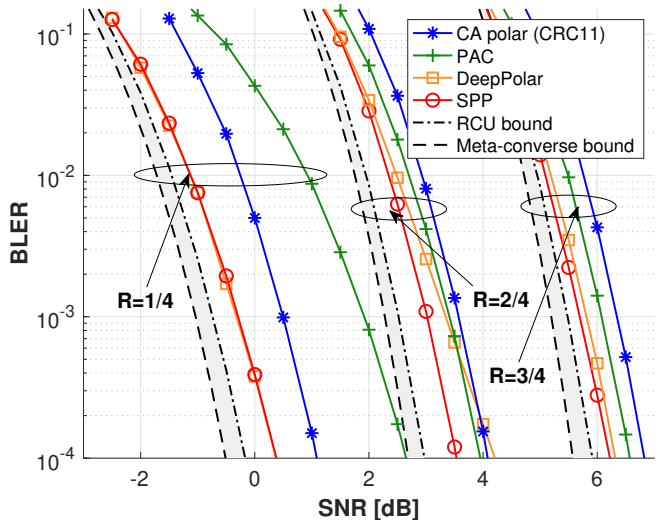

Fig. 8. BLER simulation results using SCL decoder with list size of $S = 8$ at blocklength $N = 128$.

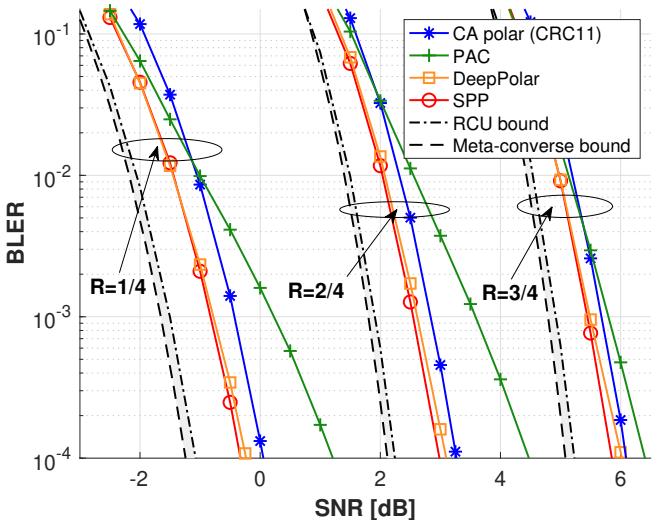

Fig. 9. BLER simulation results using SCL decoder with list size of $S = 8$ at blocklength $N = 256$.

involves limiting the number of consecutive semi-polarized information bits while attaining pre-transform gains. To achieve this, we propose applying multiple polar pre-transform matrices in parallel. This parallel pre-transform structure allows for greater flexibility in designing the pre-transform matrix, which enhances both the weight spectrum and decodability under the constraints of small SCL decoding sizes. Based on a comprehensive analysis of the SCL decoder's behavior and the formation of min-weight codewords, we designed an algorithm for selecting the connection indices to which pre-transforms are applied. These pre-transforms can be combined with global pre-transform techniques, such as CRC precoding, to further enhance the distance properties of the resulting codewords. Extensive simulation results under various blocklengths and code rates have demonstrated that our codes consistently outperform existing state-of-the-art pre-transformed polar codes, achieving superior performance at various rates and short blocklengths while maintaining low decoding complexity.

REFERENCES

[1] G. Choi and N. Lee, "Sparsely pre-transformed polar codes for low-complexity SCL decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2024.

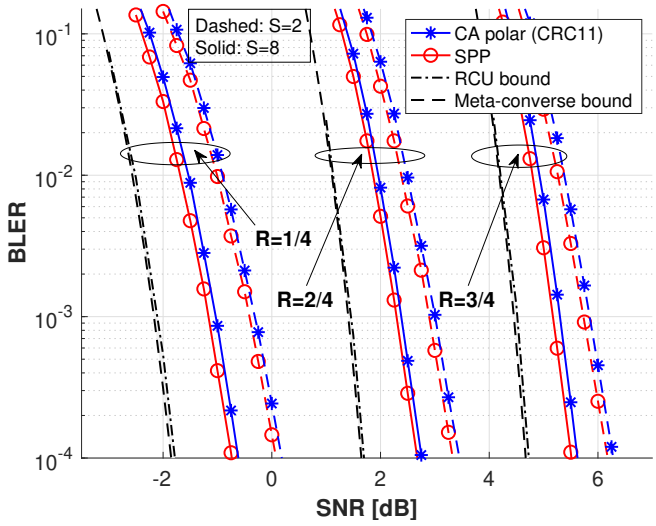

Fig. 10. BLER performance when $N = 512$. The SCL decoder is used with list size $S = 2$ and $S = 8$.

[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?" *IEEE Veh. Technol. Mag.*, vol. 13, no. 3, pp. 72–80, Sep. 2018.

[4] X. Lin and N. Lee, *5G and Beyond: Fundamentals and Standards*. Switzerland: Springer Nature, 2021.

[5] 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, Feb. 2019.

[6] C. Yue, V. Miloslavskaya, M. Shirvanimoghaddam, B. Vucetic, and Y. Li, "Efficient decoders for short block length codes in 6G URLLC," *IEEE Commun. Mag.*, vol. 61, no. 4, pp. 84–90, Apr. 2023.

[7] M. Geiselhart, F. Krieg, J. Clausius, D. Tandler, and S. ten Brink, "6G: A welcome chance to unify channel coding?" *IEEE BITS Inf. Theory Mag.*, vol. 3, no. 1, pp. 67–80, Mar. 2023.

[8] D. Han, B. Lee, S. Lee, and N. Lee, "MMSE-A-MAP decoder for block orthogonal sparse superposition codes in fading channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022.

[9] 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, 2023.

[10] 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 J. Sel. Areas Commun.*, vol. 43, no. 4, pp. 1183–1199, 2025.

[11] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, Oct. 2012.

[12] P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2013.

[13] T. Wang, D. Qu, and T. Jiang, "Parity-check-concatenated polar codes," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2342–2345, Dec. 2016.

[14] H. Zhang, R. Li, J. Wang, S. Dai, G. Zhang, Y. Chen, H. Luo, and J. Wang, "Parity-check polar coding for 5G and beyond," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018.

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

[16] S. Gelincik, P. Mary, J.-Y. Baudais, and A. Savard, "Achieving PAC code performance with SCL decoding without extra computational complexity," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2022, pp. 104–109.

[17] A. Zunker, M. Geiselhart, L. Johannsen, C. Kestel, S. ten Brink, T. Vogt, and N. Wehn, "Row-merged polar codes: Analysis, design, and decoder implementation," *IEEE Trans. Commun.*, vol. 73, no. 1, pp. 39–53, 2025.

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

[19] ——, "Deep polar codes: An efficient error correction code for short blocklength transmission," in *Proc. IEEE Global Commun. Conf. Workshops (GC Wkshps)*, 2023.

[20] M.-C. Chiu and Y.-S. Su, "Design of polar codes and PAC codes for SCL decoding," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2587–2601, May 2023.

[21] V. Miloslavskaya and B. Vucetic, "Design of short polar codes for SCL decoding," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6657–6668, Nov. 2020.

[22] A. Zunker, M. Geiselhart, and S. Ten Brink, "Enumeration of minimum weight codewords of pre-transformed polar codes by tree intersection," in *Proc. IEEE 58th Annual Conf. Information Sciences and Systems (CISS)*, 2024.

[23] V. Miloslavskaya, Y. Li, and B. Vucetic, "Frozen set design for precoded polar codes," *IEEE Trans. Commun.*, vol. 73, no. 1, pp. 77–92, 2025.

[24] V. Miloslavskaya, B. Vucetic, Y. Li, G. Park, and O.-S. Park, "Recursive design of precoded polar codes for SCL decoding," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 7945–7959, Dec. 2021.

[25] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[26] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

[27] R. Mori and T. Tanaka, "Performance and construction of polar codes on symmetric binary-input memoryless channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2009, pp. 1496–1500.

[28] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, Nov. 2012.

[29] D. Wu, Y. Li, and Y. Sun, "Construction and block error rate analysis of polar codes over AWGN channel based on gaussian approximation," *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1099–1102, 2014.

[30] C. Schürch, "A partial order for the synthesized channels of a polar code," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2016, pp. 220–224.

[31] M. Mondelli, S. H. Hassani, and R. L. Urbanke, "Construction of polar codes with sublinear complexity," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2782–2791, May 2019.

[32] 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 Global Commun. Conf.*, 2017.

[33] H. Vangala, E. Viterbo, and Y. Hong, "A comparative study of polar code constructions for the AWGN channel," *arXiv preprint arXiv:1501.02473*, 2015.

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

[35] M. Rowshan and E. Viterbo, "How to modify polar codes for list decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 1772–1776.

[36] M. C. Coşkun and H. D. Pfister, "An information-theoretic perspective on successive cancellation list decoding and polar code design," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 5779–5791, Sep 2022.

[37] M. Bardet, V. Dragoi, A. Otmani, and J.-P. Tillich, "Algebraic properties of polar codes from a new polynomial formalism," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2016, pp. 230–234.

[38] M. Rowshan, S. H. Dau, and E. Viterbo, "On the formation of min-weight codewords of polar/PAC codes and its applications," *IEEE Trans. Inf. Theory*, vol. 69, no. 12, pp. 7627–7649, Dec. 2023.

[39] M. Rowshan and J. Yuan, "On the minimum weight codewords of pac codes: The impact of pre-transformation," *IEEE J. Sel. Areas Inf. Theory*, vol. 4, pp. 487–498, 2023.

[40] V.-F. Drăgoi, M. Rowshan, and J. Yuan, "On the closed-form weight enumeration of polar codes: 1.5d-weight codewords," *IEEE Trans. Commun.*, 2024, early access.

[41] B. Li, H. Zhang, and J. Gu, "On pre-transformed polar codes," *arXiv:1912.06359*, 2019, [Online]. Available: https://arxiv.org/abs/1912.06359.

[42] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.

[43] Y. Li, H. Zhang, R. Li, J. Wang, G. Yan, and Z. Ma, "On the weight spectrum of pre-transformed polar codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2021, pp. 1224–1229.

[44] T. Kasami, N. Tokura, and S. Azumi, "On the weight enumeration of weights less than 2.5 d of Reed—Muller codes," *Information and control*, vol. 30, no. 4, pp. 380–395, 1976.
[45] R. Polyanskaya, M. Davletshin, and N. Polyanskii, "Weight distributions for successive cancellation decoding of polar codes," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7328–7336, 2020.
[46] M. Rowshan and E. Viterbo, "Stepped list decoding for polar codes," in *Proc. IEEE 10th Int. Symp. Turbo Codes Iterative Inf. Process.*, 2018.
[47] 3GPP, "NR; multiplexing and channel coding," *Tech. Rep. TS 38.212, Rel. 16*, Jul. 2020.
[48] G. Durisi and A. Lancho, "Transmitting short packets over wireless channels—an information-theoretic perspective," https://github.com/gdurisi/fbl-notes.
[49] Y. Polyanskiy, S. Chen, A. Collins, G. Durisi, T. Erseghe, G. C. Ferrante, V. Kostina, J. Östman, I. Tal, and W. Yang, "SPECTRE: short packet communication toolbox," https://github.com/yp-mit/spectre.

Geon Choi (Student Member, IEEE) received the B.S. and Ph. D. degree in electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2019 and 2024, respectively. He is currently a Postdoctoral Researcher at Korea University.

Namyoon Lee (Senior Member, IEEE) received his Ph.D. degree from The University of Texas at Austin in 2014. He was with the Communications and Network Research Group at Samsung Advanced Institute of Technology, South Korea, from 2008 to 2011, and later worked at Wireless Communications Research, Intel Labs, Santa Clara, CA, USA, from 2015 to 2016. He has been Professor at Pohang University of Science and Technology (POSTECH) since 2016. His research interests include communications theory, with a focus on advanced MIMO, channel coding technologies, and future wireless systems. Dr. Lee has received several prestigious awards, including the 2016 IEEE ComSoc AsiaPacific Outstanding Young Researcher Award, the 2020 IEEE Best Young Professional Award (Outstanding Nominee), the 2021 IEEE-IEIE Joint Award for Young Engineer and Scientist, and the 2021 KICS Haedong Young Engineering Researcher Award. He has actively contributed to IEEE journals and conferences, serving as an Associate Editor for IEEE Communications Letters from 2018 to 2020 and IEEE Transactions on Vehicular Technology from 2021 to 2023. He was also a Guest Editor for IEEE Communications Magazine in the Special Issue on Near-Field MIMO Technologies Toward 6G. Since 2021, he has been an Associate Editor for IEEE Transactions on Wireless Communications and IEEE Transactions on Communications.