Computing the Low-Weight codewords of Punctured and Shortened Pre-Transformed polar Codes

Malek Ellouze, Romain Tajan, Camille Leroux, Christophe Jégo, Charly Poulliat

Computing the Low-Weight codewords of Punctured and Shortened Pre-Transformed polar Codes

Malek Ellouze, Romain Tajan, Camille Leroux and Christophe Jégo
University of Bordeaux
Bordeaux INP - ENSEIRB-MATMECA
IMS Lab, UMR CNRS 5218, France
Email: {surname.name}@ims-bordeaux.fr

Charly Poulliat
University of Toulouse
INPT-ENSEEIHT
IRIT Lab, UMR CNRS 5505, France
Email: charly.poulliat@enseeiht.fr

***Abstract*—In this paper, we present a deterministic algorithm to count the low-weight codewords of punctured and shortened pure and pre-transformed polar codes. The method first evaluates the weight properties of punctured/shortened polar cosets. Then, a method that discards the cosets that have no impact on the computation of the low-weight codewords is introduced. A key advantage of this method is its applicability, regardless of the frozen bit set, puncturing/shortening pattern, or pre-transformation. Results confirm the method's efficiency while showing reduced computational complexity compared to state-of-the-art algorithms.**

I. Introduction

The growing interest regarding polar codes [1] is due to their ability to achieve the channel capacity asymptotically. Polar codes, nonetheless, possess two primary constraints. The first limitation arises from the fact that, for moderate code lengths, polar codes have poor distance properties. The enhancement of the distance properties of polar codes saw a notable stride with the introduction of precoding. Among the precoding techniques, the concatenation of cyclic redundancy check (CRC) with polar codes [2] combined with a successive cancellation list (SCL) decoder [3] helped improve the distance properties and the decoding performance of polar codes. A more recent alternative is to apply precoding before the polar transformation, as seen in *Dynamic Frozen Bits (DFB)* polar codes [4] and *Polarization-Adjusted Convolutional* (PAC) codes [5].

The second limitation is that from the pure polar code construction perspective, polar codes can only have sizes expressed as powers of 2. Methods such as puncturing and shortening have been introduced to manage the constraint on code sizes, enabling the construction of rate-compatible polar codes and leading to the development of various puncturing and shortening strategies [6]–[8].

Shortening is, however, more challenging in the case of PAC and DFB codes, as the values fixed by the shortening patterns can be modified due to the precoding process. To solve this issue, [9] suggested a constraint on the convolutional precoding for PAC codes that enables preserving the shortening pattern initial values.

All of the aforementioned strides were coupled with the introduction of methods to evaluate the distance properties of pure, precoded, rate-compatible polar codes as it enables estimating the performance of codes under Maximum-Likelihood (ML) decoding. The minimum distance computation for polar codes originated in [10]. In [11], the computation of the number of minimum weight codewords assumes frozen sets of decreasing monomial codes. A more general yet complex and non-deterministic approach given in [2] estimates partial distances using Monte Carlo simulations with large list sizes. In [12], a deterministic algorithm for calculating polar code weight distribution was proposed, but its high computational complexity restricted its use to codes up to length 128 under certain conditions. [13] introduced a method for computing the average partial weight spectrum of pre-transformed polar codes. [14] and [15] focused on partial weight spectrum calculations but were limited to decreasing monomial constructions and are not applicable to pre-transformed polar codes. [16] and [17] developed low-complexity methods for computing the minimum distance properties of pre-transformed polar codes, but only when the minimum distance matches that of pure polar codes. In [18], low-weight codewords are enumerated for polar, rate-compatible, and precoded polar codes through a recursive decomposition. However, it can yield high complexity in the case of precoded and rate-compatible polar codes.

This paper introduces a new low-complexity algorithm for the computation of the partial spectrum for rate-compatible pure and pre-transformed polar codes. Unlike [13], this method is deterministic, i.e it allows the determination of the exact reduced spectrum. In contrast to [14]–[17], it can be adapted to punctured/shortened polar codes and compute the partial spectrum beyond just the minimum distance properties. This approach is an extension to the one presented in [19] and offers the advantage of not assuming any specific structure (a) for the frozen bit set, (b) for the pre-transformation or (c) for the shortening and puncturing patterns. It, therefore, aids in the design of puncturing shortening patterns, pre-transformation parameters, and frozen bit sets for punctured/shortened polar codes, thereby enhancing their decoding performance.

II. Preliminaries

A. Polar codes and polar cosets

The polar code $(N = 2^n, K)$ transformation matrix is given by the n-fold polar Kronecker matrix $\mathbf{G} = {G_2}^{\otimes n}$

where $(\cdot)^{\otimes n}$ denotes the n^{th} Kronecker product power and $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ The codewords of a polar code are obtained such that $\mathbf{x} = \mathbf{u}\mathbf{G}$, where $\mathbf{u} \in \mathbb{F}_2^N$ is an information vector for which K positions are assigned to the information bits, whereas the remaining ones are *frozen*, i.e set to some known values. This operation is called *rate-profiling*. We note $\mathcal{F}$ the set of indices of the components of $\mathbf{u}$ corresponding to the frozen bits. We refer to polar codes with all frozen bits set to zero as *pure polar codes*.

B. Pre-Transformed polar codes

A pre-transformation of polar codes consists in applying a linear mapping before the multiplication with the transformation matrix. This can be conveyed as a vector-matrix multiplication with an upper triangular matrix T. The overall encoding process can be described as $\mathbf{x} = \underbrace{\mathbf{v}T}_{\mathbf{u}}\mathbf{G}$. Various polar code variants, such as DFB and PAC codes, can be unified under the concept of pre-transformed polar codes.
In the case of PAC codes [5], the pre-transformation consists of a convolutional encoding using the generator function $\mathbf{g}$ of degree m with coefficients $[g_0, g_1, ..., g_{m-1}]$, i.e, given a vector v_i, the associated u_i is obtained as: $u_i = \mathbf{g}(\mathbf{v}_0^i) = \sum_{j=0}^{m-1} g_j v_{i-j}$.

C. Rate-compatible polar codes

Let $\mathcal{S}$ and $\mathcal{P}$ denote the shortening and puncturing patterns, respectively. We also note $S = |\mathcal{S}|$ and $P = |\mathcal{P}|$. The lengths of a shortened and punctured polar code, respectively given by $N_s = N - S$ and $N_p = N - P$, are derived from a parent polar code of length N. When shortening a polar code, a designated subset of the parent code is selected. Within this subset, a total of S codeword bits are fixed to a predetermined value, e.g 0. Since the shortened codeword bits are perfectly known to the decoder, they lead to very reliable elements $\mathcal{U}_\mathcal{S}$ of $\mathbf{u}$ qualified as *overcapable* [7] and need to be frozen. We denote by $\mathcal{F}_\mathcal{S} = \mathcal{F} \bigcup \mathcal{U}_\mathcal{S}$ the set of frozen bits in the case of shortened polar codes.
In the puncturing process of polar codes, a total of P codeword bits are treated as erased and, consequently not transmitted. The unreliability of codeword bits $\mathbf{x}_\mathcal{P}$, due to being unknown to the decoder, affects the initially transmitted vector $\mathbf{u}$. As a result, a set $\mathcal{U}_\mathcal{P}$ of bits of $\mathbf{u}$ are deemed *incapable* [7], and must be frozen. The set of frozen bits for punctured polar codes is denoted as $\mathcal{F}_\mathcal{P} = \mathcal{F} \bigcup \mathcal{U}_\mathcal{P}$.
In the context of pre-transformed polar codes, for punctured polar codes, the pre-transformation remains unaffected as the punctured codewords bits are not transmitted. It, therefore, does not alter the pre-transformation rules used for pure polar codes. However, in the case of shortening pre-transformed polar codes, [9] introduced a constraint on the pre-transformation. This constraint is designed to ensure that the shortening condition is met, i.e. $\mathbf{x}_\mathcal{S} = \mathbf{0}$. In the case of PAC codes, the constrained pre-transformation consists in:

$$u_i = \begin{cases} 0 \quad \text{if} \quad i \in \mathcal{S} \\ \sum_{j=0}^{m-1} g_j v_{i-j} \quad \text{otherwise} \end{cases} \tag{1}$$

In this paper, in the case of shortened pre-transformed polar codes, we consider the constraint on the pre-transformation in Equation (1).

III. Minimum Weight Enumeration Function (MWEF) and Reduced Weight Enumeration Function (RWEF) of rate-compatible polar cosets

A. Computation of the MWEF and RWEF of polar cosets

As in [12], given $\mathbf{u}_0^{i-1} \triangleq (u_0, u_1, \ldots, u_{i-1}) \in \mathbb{F}_2^{i-1}$ and $u_i \in \mathbb{F}_2$, a polar coset $\mathcal{C}_N$ can be defined as:

$$\mathcal{C}_N(\mathbf{u}_0^i) = \{[\mathbf{u}_0^i, \mathbf{u}_{i+1}^{N-1}]\mathbf{G} | \mathbf{u}_{i+1}^{N-1} \in \mathbb{F}_2^{N-i-1}\} \tag{2}$$

A polar coset $\mathcal{C}_N(\mathbf{u}_0^i)$ thus describes the codewords' affine space generated by the prefix $\mathbf{u}_0^i$. and can also be expressed as:

$$\mathcal{C}_N(\mathbf{u}_0^i) = \mathbf{p} \oplus \left\{ \mathbf{s} \in \mathbb{F}_2^n \middle| \mathbf{s}H^{(i)T} = 0 \right\} \tag{3}$$

where $\mathbf{p} = \mathbf{u}_0^{i-1}\mathbf{G}_0^{i-1}$ and $H^{(i)}$ denotes a parity check matrix associated to $\mathbf{G}_{i+1}^{N-1}$ the $(N - i - 1)$ last rows of $\mathbf{G}$, i.e. $\mathbf{G}_{i+1}^{N-1}H^{(i)T} = 0$.
As shown in [20], in the specific case of polar codes with Arikan's kernel, there exists an extended code $[\mathbf{s}, \mathbf{t}, u_i]$ associated to an extended parity check matrix whose Tanner graph is a tree. Therefore, the message passing formalism is used to compute the distance properties of polar cosets. In [12], the Minimum Weight Enumeration (WEF) function $A_N(\mathcal{C}_N(\mathbf{u}_0^i))(X)$ is defined as:

$$A_N(\mathcal{C}_N(\mathbf{u}_0^i))(X) \triangleq \sum_{w=0}^{N} A_w X^w \tag{4}$$

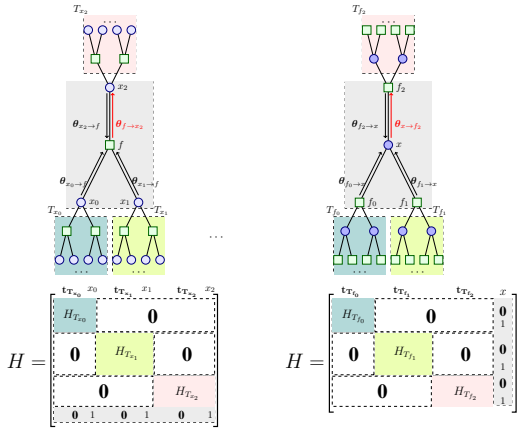
where A_w is the number of words of $\mathcal{C}_N(\mathbf{u}_0^i)$ with weight w. In this work, we define the MWEF $A_N^*(\mathcal{C}_N(\mathbf{u}_0^i))(X)$ and RWEF $A_N^{w_{end}}(\mathcal{C}_N(\mathbf{u}_0^i))(X)$ respectively as:

$$\begin{cases} A_N^*(\mathcal{C}_N(\mathbf{u}_0^i))(X) \triangleq A^* X^{w^*} \\ A_N^{w_{end}}(\mathcal{C}_N(\mathbf{u}_0^i))(X) \triangleq \sum\limits_{w=0}^{w_{end}} A_w X^w \end{cases} \tag{5}$$

In short, the RWEF considers the monomials associated to a weight up to w_{end}. MWEF is a specific case of RWEF where $w_{end} = w^*$ where w^* is the minimum weight of the considered coset. In [12], message passing rules are developed in order to compute the WEF of a polar coset. Those rules are adapted in the following in order to compute the RWEF or more specifically the MWEF of a polar coset. We present in the following the message passing rules for the computation of the RWEF.

During message passing formalism, two configurations can be encountered. The configuration depicted in Figure 1a shows two variable nodes, x_0 and x_1, connected to a third variable node x_2 via a parity function f. The corresponding parity

matrix H for the factor graph is also illustrated in the same figure.



(a) Parity node case (b) Variable node case

Fig. 1. Graph factor configurations

T_{x_i} is defined as $T_{x_i} = \left\{\mathbf{x}| \quad \mathbf{x}H^T_{T_{x_i}} = 0\right\}$ and to each message coming from a variable node, we associate:

$$\boldsymbol{\theta}_{x_i \to f} = \begin{pmatrix} \theta^{(0)}_{x_i \to f} \\ \theta^{(1)}_{x_i \to f} \end{pmatrix} = \begin{pmatrix} A_N^{w_{end}}(T_{x_i}|x_i = 0)(X) \\ A_N^{w_{end}}(T_{x_i}|x_i = 1)(X) \end{pmatrix} \quad (6)$$

In this case, $\boldsymbol{\theta}_{f \to x_2}$ can be computed from $\boldsymbol{\theta}_{x_0 \to f}$ and $\boldsymbol{\theta}_{x_1 \to f}$ as:

$$\boldsymbol{\theta}_{f \to x_2} = \begin{pmatrix} LT_{w_{end}}(\theta^{(0)}_{x_0 \to f}\theta^{(0)}_{x_1 \to f} + \theta^{(1)}_{x_0 \to f}\theta^{(1)}_{x_1 \to f}) \\ LT_{w_{end}}(\theta^{(0)}_{x_0 \to f}\theta^{(1)}_{x_1 \to f} + \theta^{(1)}_{x_0 \to f}\theta^{(0)}_{x_1 \to f}) \end{pmatrix} \quad (7)$$

where $LT_{w_{\text{end}}}(.)$ denotes the operator that only selects the monomials of a degree lower or equal to w_{end}.

Similarly, the second configuration represented in Fig. 1b is where two parity nodes with parity functions f_0 and f_1 are connected to a variable node x. T_{f_i} is defined as $T_{f_i} = \left\{\mathbf{x}| \quad \mathbf{x}H^T_{T_{f_i}} = 0\right\}$ and to each message coming from a parity node, we associate:

$$\boldsymbol{\theta}_{f_i \to x} = \begin{pmatrix} \theta^{(0)}_{f_i \to x} \\ \theta^{(1)}_{f_i \to x} \end{pmatrix} = \begin{pmatrix} A_N^{w_{end}}(T_{f_i}|x = 0)(X) \\ A_N^{w_{end}}(T_{f_i}|x = 1)(X) \end{pmatrix} \quad (8)$$

Given the two incoming messages $\boldsymbol{\theta}_{f_0 \to x}$ and $\boldsymbol{\theta}_{f_1 \to x}$ from the parity nodes to the variable node x, $\boldsymbol{\theta}_{x \to f_2}$ can be expressed as follows:

$$\boldsymbol{\theta}_{x \to f_2} = \begin{pmatrix} LT_{w_{end}}(\theta^{(0)}_{f_0 \to x}\theta^{(0)}_{f_1 \to x}) \\ LT_{w_{end}}(\theta^{(1)}_{f_0 \to x}\theta^{(1)}_{f_1 \to x}) \end{pmatrix} \quad (9)$$

The MWEF can be computed similarly to the RWEF by replacing the $LT_{w_{end}}(.)$ operator in Equations (7) and (9) with the $LP(.)$ operator, which only selects the monomial of lowest degree. Finally, the initial message that is sent from a leaf node x_i is:

$$\boldsymbol{\theta}_{x_i} = \begin{pmatrix} X^{p_i \oplus 0} \\ X^{p_i \oplus 1} \end{pmatrix} \quad (10)$$

B. RWEF of punctured/shortened polar cosets

In this section, we adapt the computation of the RWEF of polar cosets in order to take the puncturing or/and the shortening effect into account. We define rate-compatible punctured and shortened polar cosets respectively as:

$$\begin{cases} \mathcal{CP}_N(\mathbf{u}_0^i) = \{\mathbf{x}_{\bar{\mathcal{P}}}|\mathbf{x} \in \mathcal{C}_N(\mathbf{u}_0^i)\} \\ \mathcal{CS}_N(\mathbf{u}_0^i) = \{\mathbf{x}_{\bar{\mathcal{S}}}|\mathbf{x} \in \mathcal{C}_N(\mathbf{u}_0^i), \mathbf{x}_{\mathcal{S}} = \mathbf{0}\} \end{cases} \quad (11)$$

Where $\bar{\mathcal{V}}$ denotes the complement of the set $\mathcal{V}$. This representation is different from $\mathcal{C}_N(\mathbf{u}_0^i)$ in the way that it takes into account the effect of puncturing or shortening. It is possible thus to compute $A_N^{w_{end}}(\mathcal{CP}_N(\mathbf{u}_0^i))(X)$ and $A_N^{w_{end}}(\mathcal{CS}_N(\mathbf{u}_0^i))(X)$ using an approach that is similar to the one used to define the LLR values of rate-compatible polar codes.

1) Case of puncturing: In the case of punctured polar codes, given x_i such that $i \in \mathcal{P}$, the value of x_p is erased. Therefore, it does not play any role into the determination of the different codewords weights and therefore adds no weight to the final words of the rate-compatible coset. When taking this into consideration, each leaf node $x_i, i \in \mathcal{P}$ is initialised as follows:

$$\boldsymbol{\theta}_{x_i} = \begin{pmatrix} 1X^0 \\ 1X^0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (12)$$

However, there is a modification that needs to be taken into consideration in the case of punctured polar codes. Actually, as a punctured polar coset $\mathcal{CP}_N(\mathbf{u}_0^i)$ describes the affine space generated by the punctured last $N-i-1$ rows of the generator matrix, the resulting punctured matrix $\mathbf{GP}_{i+1}^{N-1}$ may not be full rank due to puncturing. This results into taking into account a word from the coset more than once. Therefore, in the case of punctured polar codes, the number of occurrences of words with weight w has to be divided by $2^{N-i-1-rk(\mathbf{GP}_{i+1}^{N-1})}$, where $rk(.)$ computes the rank of a matrix.

Example III.1. *An illustration of a weight factor graph of bit u_3 is provided by Fig. 2. We consider the punctured polar codes with parameters $N = 8$, $K = 2$ and $P = 4$, $\mathcal{P} = \{0, 2, 4, 6\}$, the frozen bit set $\mathcal{F} = \{0, 1, 2, 3, 4, 6\}$ and $\mathbf{p} = [0, 0, 0]G_0^2 = [0, 0, 0, 0, 0, 0, 0, 0]$. The aim is to compute the RWEFs $A_8^2(\mathcal{CP}_8([0, 0, 0], u_3 = 0))$ and $A_8^2(\mathcal{CP}_8([0, 0, 0], u_3 = 1))$ for $w_{end} = 2$. The RWEFs on the different nodes are computed using Equations (7) and (9). The monomials with a power greater than 2 are discarded.*

It has also to be noted that the cosets $\mathcal{CP}_8([0, 0, 0], 0)$ and $\mathcal{CP}_8([0, 0, 0], 1)$ describe the space generated by the punctured rows of the matrix highlighted in green in Fig. 3. We can see from Fig. 3 that due to the presence of two zero rows (the fifth and the seventh rows), the rank of the matrix $\mathbf{GP}_4^7$ is equal to 2 instead of 4 when taking the puncturing into account. This means that $A_8^2(\mathcal{CP}_8([0, 0, 0], 0)) = \frac{1}{2^{8-3-1-2}}(4+8X^2) = 1 + 2X^2$ and $\mathcal{CP}_8([0, 0, 0], 1) = \frac{1}{2^{8-3-1-2}}.16X^2 = 4X^2$.

2) Case of shortening: In the case of shortened polar codes, the shortening pattern is defined to guarantee that $x_i = 0, i \in$

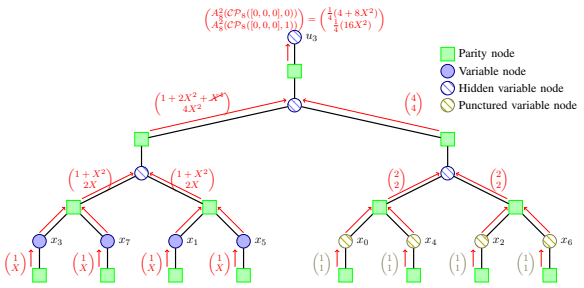


Fig. 2. Tanner Graph of u_3 decoding for a punctured polar code with $N = 8$ and $P = 4$

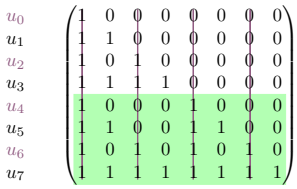


Fig. 3. Transformation matrix for $N = 8$ and $P = 4$

$\mathcal{S}$. This means that there are no codewords configurations for which $x_i = 1$. This leads to the following initialization for every leaf node x_i such that $i \in \mathcal{S}$:

$$\boldsymbol{\theta}_{x_i} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{13}$$

The message passing rules remain unchanged.

IV. Enumeration of low-weight codewords for rate-compatible pre-transformed polar codes

In this section, we leverage the findings from Section III to enumerate all the codewords of weight less or equal to a fixed threshold of punctured and shortened pure and pre-transformed polar codes.

A. Rate-compatible polar codes as union of rate-compatible cosets

Example IV.1. *Let us consider the polar code shortened from the parent code (16,7) with $\mathcal{S} = \{7, 15\}$. Fig. 4 shows the transformation matrix $\mathbf{G}_{16}$. In this figure, the frozen bits of indexes $\mathcal{F} = \{0, 1, 2, 3, 4, 5, 8\}$, are represented in black, the shortened bits in green and the remaining information bits are represented in red for the ones before the last frozen bit and in blue for the ones after the last frozen bit.*

Fig. 4. Polar transformation $\mathbf{G}_{16}$

Let us consider the shortened polar coset $\mathcal{CS}_{16}(\mathbf{u}_0^8)$. For any $\mathbf{u}_0^8$ such that $u_i = 0 \quad \forall i \in \mathcal{F}$, and $u_i = \{0, 1\} \quad \forall i \in \mathcal{I}$. The shortened polar coset $\mathcal{CS}_{16}(\mathbf{u}_0^8)$ forms a subset of the shortened polar code $\mathcal{CS}$ as by definition of $\mathcal{CS}_{16}(\mathbf{u}_0^8)$, the shortened bits u_7 and u_{15} are constrained to be zero. Therefore, the total shortened polar code $\mathcal{C}_\mathcal{S}$ can be described as:

$$\mathcal{C}_\mathcal{S} = \bigcup_{u_0^8 \in L_8} \mathcal{CS}_{16}(\mathbf{u}_0^8) \tag{14}$$

Where $L_8 = \left\{ \mathbf{u}_0^8 \; in \; \{0,1\}^8 \, | u_i = 0 \quad \forall i \in \mathcal{F} \right\}$.

More generally, any shortened polar code $\mathcal{C}_\mathcal{S}$ can be expressed as:

$$\mathcal{C}_\mathcal{S} = \bigcup_{\mathbf{u}_0^s \in L_s} \mathcal{CS}_N(\mathbf{u}_0^s) \tag{15}$$

where $s = \max(F)$ and $L_s = \{\mathbf{u}_0^s \; in \; \{0,1\}^s \, | u_i = 0 \quad \forall i \in \mathcal{F}\}$. The RWEF $A_N^{w_{end}}(\mathcal{C}_\mathcal{S})(X)$ of the shortened polar code is therefore obtained as:

$$A_N^{w_{end}}(\mathcal{C}_\mathcal{S})(X) = \sum_{\mathbf{u}_0^s \in L_s} A_N^{w_{end}}(\mathcal{CS}_N(\mathbf{u}_0^s)) \tag{16}$$

The same reasoning remains valid in the case of a punctured polar code $\mathcal{C}_\mathcal{P}$.

Exploring all the possible cosets may become prohibitive even for moderate code sizes. We propose in the next section an algorithm that only explores relevant cosets.

B. Reduced spectrum of rate-compatible polar codes

We propose an algorithm that deterministically computes the partial spectrum up to a specified weight w_{end} while restricting the exploration to only the relevant cosets during the process.

We are able to compute the minimum weight w^* of any rate-compatible polar coset $\mathcal{CV}_N(\mathbf{u}_0^i)$, where $\mathcal{V}$ represents $\mathcal{S}$ when dealing with shortening and $\mathcal{P}$ when handling puncturing. This is achieved by calculating the MWEF of the coset. Therefore, it is possible to propose an enumeration structure that has the advantage of pruning cosets with a constraint of their minimal weight. The fixed value of the last explored weight w_{end} is used as a threshold to eliminate irrelevant prefixes. This operates as follows:

- For each $i \in [0, N-1]$, all the prefixes $\mathbf{u}_0^i$ where $u_i = \mathbf{g}(\mathbf{v}_0^i)$ that remained in the list at an exploration stage i are listed.
- For each of the aforementioned prefixes, $w^*(\mathcal{CV}_N(\mathbf{u}_0^i))$ is computed.
- The cosets with $w^*(\mathcal{CV}_N(\mathbf{u}_0^i)) > w_{end}$ are discarded. Those cosets are irrelevant to the computation the partial weight spectrum with the threshold on weight w_{end} as:

$$w^*(\mathcal{CV}_N(\mathbf{u}_0^i, u_{i+1})) \geqslant w^*(\mathcal{CV}_N(\mathbf{u}_0^{i-1}, u_i)) \tag{17}$$

  In other words, if a coset has a minimum weight $w > w_{end}$, then no codeword within that coset can have

a weight lower than or equal to w_{end}.

- When $i = s$, the partial weight spectrum is obtained as the sum of the RWEFs of the cosets remaining in the list.

Algorithm 1 gives the details of the proposed algorithm for punctured or shortened PAC codes. It is important to note that it can also be applied to polar codes with DFB as this only affects the way the pre-transformation is realised. Algorithm 1 consists of $s-1$ loop iterations where the minimum weight of C_i cosets is evaluated at each enumeration stage and one iteration where the RWEF of C_s cosets is evaluated. The computational complexity of the proposed method is driven by the total number of evaluated cosets $n_c = \sum_{i=0}^{s} C_i$.

Algorithm 1: Reduced spectrum of punctured/shortened PAC codes

Input: $N, K, \mathcal{F}, \mathbf{g}, \mathcal{V}, w_{end}$
Output: Reduced spectrum up to w_{end}

$L \leftarrow 1$
$\mathcal{L} \leftarrow \{0\}$ /* List to store prefixes */

for $i \in [\![0; s]\!]$ **do**
  if $i \in \mathcal{F}$ **then**
    for $l \in [\![1; L]\!]$ **do**
      $v_i[l] \leftarrow 0$
      if $i \in \mathcal{S}$ **then**
        $u_i[l] \leftarrow 0$ /* Equation (1) */
      else
        $u_i[l] \leftarrow \mathbf{g}(\mathbf{v}_0^i[l])$
      end
      Compute w^* of $\mathcal{CV}_N(\mathbf{u}_0^{i-1}[l], u_i[l])$
      Discard the prefixes for which $w^* > w_{end}$
      $L \leftarrow |\mathcal{L}|$
    end
  else
    $\mathcal{L} \leftarrow \mathcal{L} \bigcup \mathcal{L}'$ /* $\mathcal{L}'$ is a copy of $\mathcal{L}'$ */
    for $l \in [\![1; L]\!]$ **do**
      $[v_i[l], v_i[l']] \leftarrow [0, 1]$
      $u_i[l] \leftarrow \mathbf{g}(\mathbf{v}_0^i[l])$ $u_i[l'] \leftarrow \mathbf{g}(\mathbf{v}_0^i[l'])$
      Compute w^* of $\mathcal{CV}_N(\mathbf{u}_0^{i-1}[l], u_i[l])$ and $\mathcal{CV}_N(\mathbf{u}_0^{i-1}[l], u_i[l'])$
      Discard the cosets for which $w^* > w_{end}$
      $L \leftarrow |\mathcal{L}|$
    end
  end
  if $i = s$ **then**
    Compute the RWEF of the remaining paths in the list
    Compute the RWEF of the overall code $A_N^{w_{end}}(\mathcal{C}_\mathcal{V})$
end
Return $A_N^{w_{end}}(\mathcal{C}_\mathcal{V})$

TABLE I
PARTIAL WEIGHT DISTRIBUTION OF PUNCTURED AND SHORTENING POLAR AND PAC CODES

(N, K)	Type	(w, A_w)
$(80, 20)$	Polar	$(8, 30), (16, 173), (20, 256)(24, 8040)$ $(28, 7424)$
	PAC	$(8, 2), (12, 8), (14, 8), (16, 109), (18, 56)$ $(20, 920), (22, 504), (24, 2456)$ $(26, 5352), (28, 11528), (30, 11304)$ $(32, 34194)$
$(160, 40)$	Polar	$(8, 12), (16, 382), (24, 2220), (32, 55533)$ $(40, 663536)$
	PAC	$(8, 8), (16, 230), (18, 64), (20, 64)$ $(22, 64), (24, 2120), (26, 960)$ $(28, 1216), (30, 1472), (32, 16557)$ $(34, 16448), (36, 19264)$ $(38, 22080), (40, 286240)$
$(320, 80)$	Polar	$(16, 476), (24, 11584), (28, 12288)$ $(32, 117598), (36, 12288)$ $(40, 678208), (44, 589824)$ $(48, 15764476)$
	PAC	$(16, 76), (18, 16), (20, 32), (22, 16)$ $(24, 208), (26, 112), (28, 352), (30, 112)$ $(32, 9694), (34, 2928), (36, 12512)$ $(38, 8176), (40, 160848), (42, 52496)$ $(44, 224544), (46, 192784), (48, 3669484)$
$(640, 160)$	Polar	$(16, 344), (24, 2688), (32, 117004)$ $(40, 3741824)$
	PAC	$(16, 20), (24, 24), (32, 11652), (34, 64)$ $(36, 704), (38, 64), (40, 47032)$
$(80, 40)$	Polar	$(8, 1078), (12, 32128), (14, 45056)$ $(16, 971821), (18, 2191360)$ $(20, 35615872)$
	PAC	$(8, 582), (10, 608), (12, 14848)$ $(14, 46624), (16, 446125), (18, 1810400)$ $(20, 11718144)$
$(160, 80)$	Polar	$(8, 508), (12, 2496), (16, 320030)$ $(20, 9821632)$
	PAC	$(8, 300), (12, 2112), (16, 92862)$ $(18, 15616), (20, 2453568)$
$(320, 160)$	Polar	$(8, 120), (16, 183116), (20, 731136)$
	PAC	$(8, 120), (16, 74540), (18, 6080)$ $(20, 568832)$
$(640, 320)$	Polar	$(16, 69496), (24, 27166592)$
	PAC	$(16, 43544), (18, 736), (20, 25408)$ $(22, 6208), (24, 16013568)$

V. EXPERIMENTAL RESULTS

This section summarizes the experimental results obtained on the partial weight distribution for a wide range of pure and pre-transformed rate-compatible polar codes. For each code, we compute the exact number A_w of codewords of weight w for all $w \leq w_{end}$.

Table I summarizes the partial weight spectrum of punctured shortened polar and PAC codes for $N = \{128, 256, 512, 1024\}$, $P = \{48, 96, 192, 384\}$ and $S = \{48, 96, 192, 384\}$ respectively. We apply puncturing for the codes with rate $R = 0.25$ and shortening for codes with rate $R = 0.5$. This choice aligns with the 5G standardization, where shortening is used for high rates and puncturing for low rates [21]. In the case of PAC codes, the polynomial

TABLE II
PARTIAL WEIGHT DISTRIBUTION OF $(200, 100)$ RANDOMLY PUNCTURED / SHORTENED POLAR CODES

(N, K)	$(200, 100)$ Random Shortening	$(200, 100)$ Random puncturing
(w, A_w)	$(8, 194)$	$(7, 12)$
	$(12, 456)$	$(8, 35)$
	$(16, 67867)$	$(9, 115)$
	$(20, 1319413)$	$(10, 332)$
	$(22, 696208)$	$(11, 710)$
		$(12, 1349)$
		$(13, 2934)$
		$(14, 6737)$
		$(15, 16490)$
		$(16, 41033)$
		$(17, 98835)$
		$(18, 235252)$
		$(19, 561588)$
TC [18]	56257×10^9	128664×10^9
Our TC	78×10^9	48×10^9

$\mathbf{g} = [1, 0, 1, 1, 0, 1, 1]$ is chosen. The frozen bit sets are the ones specified in the 5G standard [21]. The puncturing and shortening patterns are the ones defined with the bit-reversal permutation [7].

The results for the number of codewords with minimum weight of shortened polar and PAC codes (results highlighted in red) were corroborated with results in [9]. To the best of the authors' knowledge, the full results for the partial weight spectrum of PAC codes have not been reported in the literature.

Table II provides a computational complexity comparison of the proposed algorithm to the one introduced in [18]. To this end, the punctured and shortened patterns were defined randomly to accommodate the results of [18]. The results are shown for a $(200, 100)$ polar code with the same rate-profiling. We compare for both method the Time Complexity (TC) defined as the number of arithmetic operations for both methods.

Table II shows that the number of codewords with a specific weight computed via Algorithm 1 is in the same range of the results computed in [18]. Note that since the puncturing and shortening are done randomly, we cannot reproduce the exact same results. The computational complexity of Algorithm 1 is lower by several orders of magnitude. For instance , it is indicated in [18] that the overall running time for a C++ implementation on a computer with 6 cores i7 and a 3.2GHz processor in the case of a randomly shortened $(200, 100)$ polar code is approximately 28 hours. In contrast, our MATLAB implementation on a computer with 2 cores i5 and a 3.1GHz processor achieves a running time of less than 5 minutes.

VI. CONCLUSION

A low-complexity algorithm is detailed to compute the partial weight spectrum of punctured and shortened pure and pre-transformed polar codes. The proposed approach takes advantage of the computation of cosets minimum weights to explore only the relevant cosets defining a polar code. It has been shown to have a significantly lower complexity when compared to state-of-the-art algorithms. Besides, the computation is feasible regardless of the frozen bit set or the punctured/shortened pattern.

REFERENCES

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

[2] B. Li, H. Shen, and D. Tse, "An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check," *IEEE Communications Letters*, vol. 16, no. 12, pp. 2044–2047, 2012.

[3] I. Tal and A. Vardy, "List decoding of polar codes," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2011, pp. 1–5.

[4] P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *2013 IEEE Information Theory Workshop (ITW)*, 2013, pp. 1–5.

[5] E. Arıkan, "From sequential decoding to channel polarization and back again," *arXiv preprint arXiv:1908.09594*, 2019.

[6] K. Niu, K. Chen, and J.-R. Lin, "Beyond turbo codes: Rate-compatible punctured polar codes," in *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 3423–3427.

[7] V. Bioglio, F. Gabry, and I. Land, "Low-complexity puncturing and shortening of polar codes," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2017, pp. 1–6.

[8] R. Wang and R. Liu, "A novel puncturing scheme for polar codes," *IEEE Communications Letters*, vol. 18, no. 12, pp. 2081–2084, 2014.

[9] X. Gu, M. Rowshan, and J. Yuan, "Rate-compatible shortened pac codes," in *2023 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, 2023, pp. 1–6.

[10] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *2009 IEEE International Symposium on Information Theory*, 2009, pp. 1488–1492.

[11] M. Bardet, V. Dragoi, A. Otmani, and J.-P. Tillich, "Algebraic properties of polar codes from a new polynomial formalism," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 230–234.

[12] H. Yao, A. Fazeli, and A. Vardy, "A deterministic algorithm for computing the weight distribution of polar codes," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1218–1223.

[13] Y. Li, H. Zhang, R. Li, J. Wang, G. Yan, and Z. Ma, "On the weight spectrum of pre-transformed polar codes," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 1224–1229.

[14] M. Rowshan, S. H. Dau, and E. Viterbo, "On the formation of min-weight codewords of polar/pac codes and its applications," *IEEE Transactions on Information Theory*, 2023.

[15] M. Rowshan, V.-F. Drăgoi, and J. Yuan, "On the closed-form weight enumeration of polar codes: $1.5d$-weight codewords," *arXiv preprint arXiv:2305.02921*, 2023.

[16] M. Rowshan and J. Yuan, "Fast enumeration of minimum weight codewords of PAC codes," in *2022 IEEE Information Theory Workshop (ITW)*, 2022, pp. 255–260.

[17] A. Zunker, M. Geiselhart, and S. Ten Brink, "Enumeration of minimum weight codewords of pre-transformed polar codes by tree intersection," in *2024 58th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2024, pp. 1–6.

[18] V. Miloslavskaya, B. Vucetic, and Y. Li, "Computing the partial weight distribution of punctured, shortened, precoded polar codes," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7146–7159, 2022.

[19] M. Ellouze, R. Tajan, C. Leroux, C. Jégo, and C. Poulliat, "Low-complexity algorithm for the minimum distance properties of pac codes," in *2023 12th International Symposium on Topics in Coding (ISTC)*, 2023, pp. 1–5.

[20] R. Mori and T. Tanaka, "Performance and construction of polar codes on symmetric binary-input memoryless channels," in *2009 IEEE International Symposium on Information Theory*, 2009, pp. 1496–1500.

[21] 3GPP TS 38.212 V17.4.0, "5G; NR; multiplexing and channel coding," 2023.