Hybrid Non-Binary Repeated Polar Codes For Low-SNR Regime

Fariba Abbasi[†], Hessam Mahdavifar[‡], and Emanuele Viterbo[†]
[†] Monash University, Melbourne, VIC 3800, Australia
[‡] University of Michigan, Ann Arbor, MI 48104, USA
Email: [†]{fariba.abbasi, emanuele.viterbo}@monash.edu, [‡] hessam@umich.edu

***Abstract*—Concatenating the state-of-the-art codes at moderate rates with repetition codes have emerged as practical solutions deployed in various standards for ultra-low-power devices such as in Internet-of-Things (IoT) networks. In this paper, we propose a novel concatenation mechanism for such applications which need to operate at very low signal-to-noise ratio (SNR) regime. In the proposed scheme, the outer code is a hybrid polar code constructed in two stages, one with a binary kernel and another also with a binary kernel but applied over a binary extension field. The inner code is a non-binary multiplicative repetition code. This particular structure inherits low-complexity decoding structures of polar codes while enabling concatenation with an inner non-binary multiplicative repetition scheme. The decoding for the proposed scheme is done using cyclic redundancy check (CRC) aided successive cancellation list (SCL) decoder over AWGN channel. Simulation results show that the proposed scheme outperforms the straightforward binary polar-repetition scheme at the cost of a negligible increase in the decoding complexity.**

I. Introduction

The Third Generation Partnership Project (3GPP) has recently introduced Narrow-Band Internet-of-Things (NB-IoT) and enhanced Machine-Type Communications (eMTC) features into the cellular standard protocols. These two narrowband and complementary technologies expand the cellular networks to support low-power, wide-area (LPWA) cellular connectivity for the IoT use cases [4].

In general, IoT devices need to operate under extreme power constraints. Consequently, they often communicate at very low signal-to-noise ratio (SNR), e.g., -13 dB or 0.03 bits per transmission (translated to capacity) in NB-IoT protocols [4]. Also, they are often not equipped with advanced transceivers due to cost constraints. Therefore, the solution adopted in the standard is to use the legacy turbo codes or convolutional codes at moderate rates, e.g., the turbo code of rate $1/3$, together with many repetitions, e.g., up to 2048 repetitions in NB-IoT. This implies effective code rates as low as 1.6×10^{-4} are supported in such protocols. This repetition scheme has efficient implementations with computational complexity and latency effectively reduced to that of the outer code. However, it is expected that repeating a high-rate code to enable low-rate communication will result in rate loss and mediocre performance. As a result, studying ultra-low-rate error-correcting codes for reliable communications in such low-capacity regimes becomes necessary [1].

In [1], the authors constructed an efficient repetition scheme with outer polar codes and showed that the proposed polar-repetition scheme outperforms the Turbo-repetition code, the proposed code design in the eMTC and NB-IoT (uplink) standards, over additive white Gaussian noise (AWGN) channel. In another related work, low-rate codes for binary symmetric channels were constructed by concatenating high-rate polar codes with repetitions [5]. Also, non-binary LDPC codes concatenated with multiplicative repetition codes were introduced in [6]. By multiplicatively repeating the $(2, 3)$-regular non-binary LDPC mother code of rate $1/3$, they constructed rate-compatible codes of lower rates $1/6, 1/9, 1/12, \ldots$ which outperform the best low-rate binary LDPC codes at the cost of the increase in decoding complexity. Very recently, weakly-coded binary LDPC type code combined with polar code has been introduced in [7] which is shown to outperform uncoded modulation over high noise memoryless channels. However, the complexity of the proposed scheme in [7] is higher than that of a repetition-based scheme.

In this paper, we propose an alternative mechanism for polar-repetition schemes, referred to as hybrid non-binary multiplicative repetition. In this scheme, the outer code is a hybrid binary and non-binary polar code constructed in two stages. The first stage of the outer encoder utilizes Arıkan's binary polarization kernel applied recursively, as in original polar codes [2]. The output bits of the first stage are grouped into t-tuples and are turned into symbols over the extension binary field $GF(2^t)$. Then Arıkan's kernel is again applied recursively. Hence, the output of the outer encoder consists of symbols over $GF(2^t)$. The inner code is a non-binary multiplicative repetition code. The encoded symbols can be either turned into binary strings for transmission using a binary modulation scheme, e.g., binary phase shift keying (BPSK) or, alternatively, can be sent using a higher order modulation. The proposed structure allows for i) benefiting from the multiplicative repetition over an extension field as opposed to a simple repetition in a straightforward binary polar-repetition scheme, and ii) keeping the complexity of the encoder/decoder almost the same as those of the straightforward repetition scheme. The proposed scheme outperforms the straightforward repetition under cyclic redundancy check (CRC) aided successive cancellation list (SCL) decoder over AWGN channel at the cost of the negligible increase in encoding/decoding complexity.

II. Background

An $(n, k, \mathcal{F})$, with $n = 2^m$, polar code based on the 2×2 polarization kernel $G_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ is a linear block code generated by k rows of $G_n = G_2^{\otimes m}$, where $.^{\otimes m}$ is the m-th Kronecker power of a matrix [2]. The set of frozen bits $\mathcal{F}$, with $|\mathcal{F}| = n - k$, is the set of the indices of sub-channels with the lowest reliabilities. Arıkan's polar codes are constructed by setting entries of the input vector u_0^{n-1}[1] indexed by $\mathcal{F}$ to zeros and the remaining k bits are used to transmit the information. At the decoder side, the successive cancellation (SC) decoder, makes the decision on u_i, based on the previously decoded bits, $\hat{u}_0^{i-1}$, and channel output vector, y_0^{n-1}, according to the following log likelihood ratio (LLR) rule:

$$\hat{u}_i = \begin{cases} 1, & \text{if} \quad i \in \mathcal{F}^C \;\&\; \ln \frac{W_n^{(i)}(\hat{u}_0^{i-1}, y_0^{n-1} | u_i = 0)}{W_n^{(i)}(\hat{u}_0^{i-1}, y_0^{n-1} | u_i = 1)} < 0 \\ 0, & \text{if} \quad \text{otherwise}, \end{cases} \tag{1}$$

where $W_n^{(i)}$ is the i-th bit-channel [2]. In order to improve the error correction performance of the SC decoder, the successive cancellation list (SCL) decoding algorithm was proposed by Tal and Vardy in [3]. In SCL decoding, the L most likely paths u_0^{i-1} are tracked. When decoding u_i, for $i \in \mathcal{F}^C$, the decoder extends each path into two paths exploring both possibilities $u_i = 0$ and $u_i = 1$. If the number of obtained paths exceeds L, the decoder picks L most likely paths as the surviving ones and prunes the rest based on a certain Path Metric (PM). Let $\hat{u}_i[l]$ denote the estimate of u_i in the l-th path, for $l \in \{1, 2, \ldots, L\}$ and $S_i[l]$ denote its corresponding LLR. Then the corresponding PM is calculated as follows:

$$PM_l^{(i)} = \begin{cases} PM_l^{(i-1)} + |S_i[l]| & \text{if } \hat{u}_i[l] \neq \frac{1}{2}(1 - \text{sgn}(S_i[l])) \\ PM_l^{(i-1)} & \text{if otherwise}, \end{cases} \tag{2}$$

where $PM_l^{(-1)}$ is seto to 0. Finally, the path with the smallest PM is selected as the estimated bits $\hat{u}_0^{n-1}$. As the list size L grows large, the SCL decoder approaches the maximum-likelihood (ML) decoding performance. To further improve the performance of the SCL decoder, p-bits cyclic redundancy check (CRC) are appended to the information bits as an outer code. The SCL decoder outputs the decoding path with the smallest PM among the paths which pass the CRC, [3].

In low-capacity applications, repetition code is a simple way of designing a practical low-rate code. Let r denote the number of the repetitions and N denote the length of the code. For constructing the repetition code, first, a smaller outer code (e.g., a polar code) of length $n = N/r$ is designed and then each of its code bits is repeated r times.

III. Proposed Scheme

In this section, the proposed hybrid non-binary repeated polar code scheme is discussed. It is shown how this scheme can improve the performance of the straightforward repetition scheme in the low-SNR regime.

[1] u_0^{n-1} is a row vector $(u_0, u_1, \ldots u_{n-1})$ and u_0^i is its subvector $(u_0, u_1, \ldots u_i)$.

Owing to the recursive structure of the polar codes, one can consider the polarization transform kernel G_n as the concatenation of $G_{n/t}$ and G_t, with $G_n = G_{n/t} \otimes G_t$, where $t = 2^{m'}, m' = \{1, 2, 3, \ldots, \}$, [9]. Figure 1 (a) shows the block diagram of the encoder of the polar code with this structure, where x_0^{n-1} is the output of the transformation G_t. Figure 1 (c) shows the structure of the straightforward repetition. In this scheme, the outer code is the polar code depicted in Figure 1 (a) and the inner code is the repetition code which repeats the output of the outer polar code, z_0^{n-1}, r times and generate code C_r as follows.

$$\begin{aligned} &C_r = \{c_0^{rn-1} | c_{(r-1)n+v} = z_v, \\ &\text{for} \quad v = \{0, \ldots, n-1\}, \quad c_0^{(r-1)n-1} \in C_{r-1}\}. \end{aligned} \tag{3}$$

To improve the performance of the straightforward repetition, the scheme depicted in Figure 1 (b) and (d) is proposed. The encoder and decoder of the proposed hybrid non-binary repeated polar codes are as follows.

A. Encoding of the Proposed Hybrid Non-Binary Repeated Polar Codes

For encoding the proposed scheme, in the first stage, the binary input bits u_0^{n-1} are divided into subsets of bits of size t. Then, each of these n/t t-tuples are encoded with binary polarization kernel G_t over $GF(2)$ and the output is x_0^{n-1}. In the second stage, each of these outputs are grouped together as a 2^t-bit symbol a_i, $a_i = (x_{it}, x_{it+1}, \ldots x_{(i+1)t-1}) \in GF(2^t)$, $i = \{0, 1, \ldots, n/t-1\}$. Then, the symbols $a_0^{n/t-1}$ are encoded with the binary polarization kernel $G_{n/t}$ over $GF(2^t)$ and generate code $C_1 = \{z_0^{n/t-1} \in GF(2^t)\}$. Finally, coefficients $\rho_{n/t}^{rn/t-1}$ are chosen at random from $GF(2^t) \setminus \{0\}$ and are multiplied by $z_0^{n/t-1}$ to generate the code C_r, $r > 1$, as follows.

$$\begin{aligned} &C_r = \{c_0^{rn/t-1} | c_{(r-1)n/t+v} = \rho_{(r-1)n/t+v} z_v, \\ &\text{for} \quad v = \{0, \ldots, n/t-1\}, \quad c_0^{(r-1)n/t-1} \in C_{r-1}\}. \end{aligned} \tag{4}$$

Note that the coefficients $\rho_{(j-1)n/t}^{jn/t-1}$, $j = \{2, 3, \ldots, r\}$ are the random multiplication coefficients for the j-th repetition. Algorithm 1 shows the process of encoding the proposed scheme. The inputs to this algorithm are binary input bits u_0^{n-1}, t, n and r. The outputs is the code C_r.

The outer encoding only requires standard binary polar encoding scheme followed by binary to 2^t-ary conversion. For the purpose of decoding, the conversion from binary to 2^t-ary needs to be located to between stage 1 and stage 2.

Note that the authors in [8] constructed mixed kernels over alphabets of different sizes and improved the polarization properties of the kernel G_n. However, in this paper, by using the structure of the kernel G_n, we group the binary bits into symbols without modifying the polarization kernel G_n.

Example: Figure 2 shows an example for $n = 8$, $r = 3$, $t = 2$. The polarization kernels of the stage 1 and 2 are $G_2 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and $G_4 = G_2^{\otimes 2}$, respectively. The output codewords of this example are $c_0^{11} = (0, \alpha, 1, \alpha^2, 0, \alpha, \alpha, \alpha, 0, \alpha, \alpha^2, 1)$.

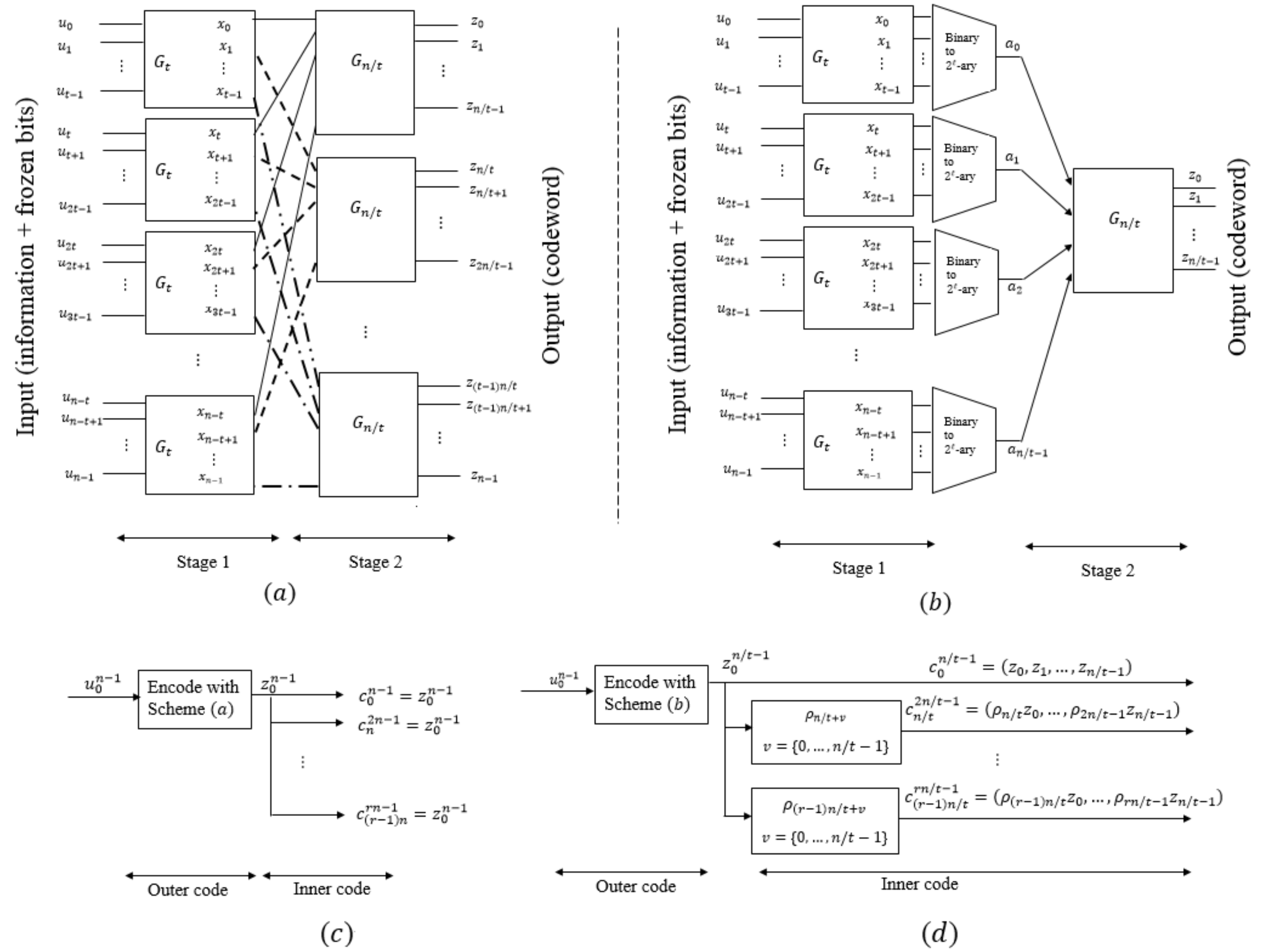

Figure 1: (a) Layered factor graph representation of a polar code (b) Layered factor graph representation of the proposed hybrid non-binary polar code (c) Repetition scheme with outer polar code (d) Hybrid non-binary repeated polar codes scheme.

Algorithm 1: Encoding Algorithm of the Proposed Hybrid Non-binary Repeated Polar Codes

```
input : u_0^{n-1}, t, n, r
output: Code C_r
Divide u_0^{n-1} into sets of t bits, u_{it}^{(i+1)t-1},
  i = {0, 1, ..., n/t - 1}.
for i ← 0 to n/t - 1 do
    x_{it}^{(i+1)t-1} ← Encode each sets of bits, u_{it}^{(i+1)t-1}, with binary
      kernel G_t over GF(2).
    Group x_{it}^{(i+1)t-1} together to make a t-bit symbol a_i.
end
C_1 = {z_0^{n/t-1}} ← Encode a_0^{n/t-1} symbols with binary kernel
  G_{n/t} over GF(2^t).
for j ← 2 to r do
    C_j ← Choose n/t coefficients ρ_{(j-1)n/t}^{jn/t-1} uniformly at random
      from GF(2^t) \ {0}, multiply them with z_0^{n/t-1}.
    Generate code C_j = (C_{j-1}, C_j).        // append C_j to
      C_{j-1}
end
return C_r
```

B. SCL Decoding of the Proposed Hybrid Non-Binary Repeated Polar Codes

For decoding the proposed scheme, we use CRC-aided LLR-based SCL decoder. Algorithm 2 shows the details of the process. The SCL decoding of the proposed scheme involves mainly 4 parts, i.e. the initial LLRs, the LLRs of the multiplicative repetition, the LLRs of the stage 1 and the LLRs of the stage 2 of the outer polar code. The details of the LLRs calculations are as follow.

Initial LLRs: The initial LLRs of the i-th symbol, $i \in \{0, 1, \ldots, rn/t-1\}$, with symbol value $s \in GF(2^t)$ are defined as

$$S_{in,i}^{(s)} = \ln \frac{W(y_i|(c_i)_M = 0_M)}{W(y_i|(c_i)_M = s_M)}, \tag{5}$$

where c_i is unmodulated codeword, $(c_i)_M$ is the modulated codewords c_i and y is the received symbol value. Let us denote $S_{in,i} = \{S_{in,i}^{(s)}\}_{s \in GF(2^t)}$, the vector of the i-th initial LLRs for all possible symbols s over $GF(2^t)$.

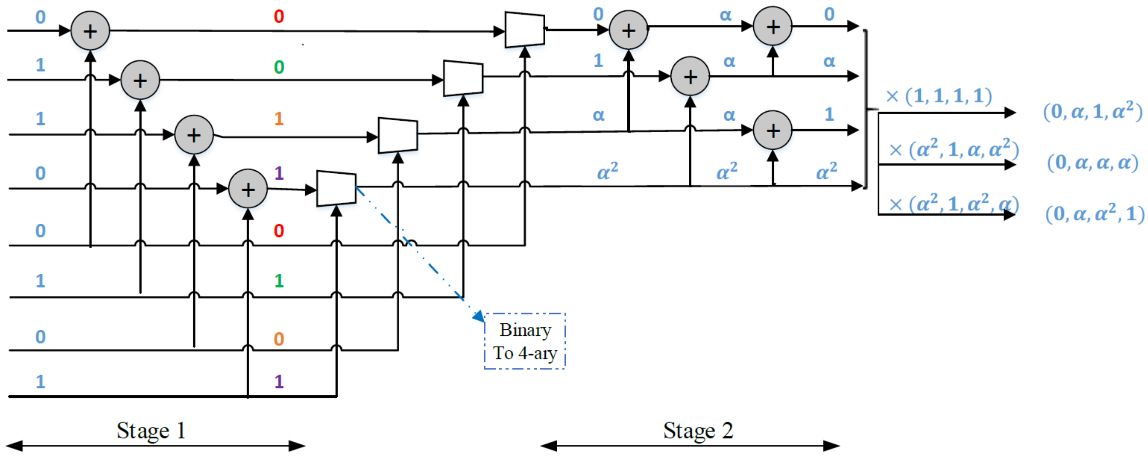

Figure 2: Example, encoder of the proposed scheme for $n=8$, $r=3$, $t=2$.

LLRs of the multiplicative repetition: Since $GF(2^t)$ is a finite field, the non-zero elements can be expressed as powers of a primitive element α in the field, i.e., $1, \alpha, \dots, \alpha^{2^t-1}$. Therefore, multiplication by an arbitrary non-zero symbols $\rho_j = \alpha^\tau$, $j = \{n/t, \dots, rn/t-1\}$, can be regarded as a cyclic shift of the field elements by τ. As a result, the decoder permutes the LLR vector $S_{in,i}$ and outputs the vector $\pi_{\rho_j}(S_{in,i})$.

Finally, the LLR of an r-tuple consisting of r independent transmissions of symbols is equal to sum of the LLRs of the individual channel outcomes after the permutations (see lines 2-8 of the Algorithm 2).

LLRs update of the stage 2**:** In general, the LLR of the i-th symbol, $i \in \{0, 1, \dots, n-1\}$, with symbol value s over kernel G_n, can be calculated according to the following formula, [2]:

$$S_i^{(s)} \triangleq \ln \frac{\sum_{u_{i+1}^{n-1}} R_{G_n}(\hat{u}_0^{i-1}, 0, u_{i+1}^{n-1})}{\sum_{u_{i+1}^{n-1}} R_{G_n}(\hat{u}_0^{i-1}, s, u_{i+1}^{n-1})}, \tag{6}$$

where $R_{G_n}(u) = \exp(-\sum_{j=0}^{n-1} S_j^{(x_j)})$ and x_j is the j-th index of the vector $x_0^{n-1} = u_0^{n-1} G_n$ and s is an element from $GF(2^t)$. One can use the following equation for simplifying eq. (6).

$$\ln(\sum_i e^{-f_i}) \approx -\min_i(f_i).$$

Now, for our binary kernel G_2, consider two input LLR vectors S_+ and S_- of size 2^t. Then, the output LLR vectors $\hat{S}_+$ and $\hat{S}_-$ can be derived from (6), as:

$$\begin{aligned} \hat{S}_+^{(s)} \approx & \min_{u_1 \in GF(2^t)} (S_+^{(s+u_1)} + S_-^{(u_1)}) \\ & - \min_{u_1 \in GF(2^t)} (S_+^{(u_1)} + S_-^{(u_1)}), \\ \hat{S}_-^{(s)} \approx & \, S_+^{(\hat{u}_0+s)} + S_-^{(s)} - S_+^{(\hat{u}_0)} - S_-^{(0)}. \end{aligned} \tag{7}$$

where $S_+^{(i)}$ and $S_-^{(j)}$ are the i-th and j-th index of the vectors S_+ and S_-, respectively and $\hat{u}_0$ is the previously decoded symbol, while u_1 is the yet to be decoded symbol.

LLRs update of the stage 1**:** For updating the LLRs of the stage 1, consider the input non-binary LLR vector $\hat{S}$. The output binary LLRs $\hat{S}_{b_i}$, $i = \{1, 2, \dots, t\}$, derived from eq. (6), are as follows.

$$\hat{S}_{b_i} \approx \min_{u_{i+1}^{t-1}} \hat{S}^{([v^{(1)}]_{2^t})} - \min_{u_{i+1}^{t-1}} \hat{S}^{([v^{(0)}]_{2^t})}, \tag{8}$$

where $[v^{(k)}]_{2^t}$ is the representation of the binary vector $v^{(k)} = (\hat{u}_0^{i-1}, k, u_{i+1}^{t-1}) G_t$ as element of $GF(2^t)$ for $k = \{0, 1\}$ and $\hat{S}^{([v_k]_{2^t})}$ is the $[v^{(k)}]_{2^t}$-th index of the vector $\hat{S}$.

The CRC-aided SCL decoder after updating the LLRs and calculating the PM for different paths based on the values of the $\hat{S}_{b_i}$ chooses the path with the smallest PM which passes the CRC.

IV. Analysis and Numerical Results

In this section, we first analyze the numerical result of the proposed scheme and compare it with the straightforward polar-repetition scheme. Then, we analyze the performance of the hybrid non-binary repeated polar code. Finally, we provide complexity analysis of the proposed scheme and the polar-repetition one.

A. Numerical Analysis

In this subsection, we provide numerical results for the proposed scheme under the SC and CRC-aided SCL decoder over AWGN channel with BPSK modulation. Figure 3 compares the performances of the proposed scheme with straight forward polar-repetition scheme for $N = 8192$, $n = 512$, $K = 40$, $t = 2$ and $r = 16$. It can seen that the performance of the proposed hybrid non-binary repeated scheme under SC decoder is almost the same as the performance of the polar-repetition scheme. However, the proposed scheme under SCL decoder with $L = 32$ and 6-bit CRC outperforms the polar-repetition scheme. Note that the construction for both schemes is based on the Monte-Carlo simulation.

Algorithm 2: List Decoding Algorithm for the Proposed Hybrid Non-binary Repeated Polar Codes

input : List size L, r, n, t, $\rho_{n/t}^{rn/t-1}$ and $y_0^{rn/t-1}$
output: The estimated bits $\hat{u}_0^{n-1}$

```
S_{in,i} ← Calculate the initial LLRs with eq. (5) for all i = {0, 1, ..., rn/t − 1}.
S_inner,i ← S_{in,i}, i = {0, ..., n/t − 1}    // Init.: Pick the first n/t elements of S_{in,i}.
for i ← n/t to rn/t − 1 do
    π_{ρ_i}(S_{in,i}) ← Permute the vector S_{in,i} based on the random coefficient ρ_i.
end
for j ← 2 to r do
    S_inner,i ← S_inner,i + π_{ρ_i}(S_{in,i})[(j − 1)n/t : jn/t − 1].
    // Output LLRs of the inner code
end
for k ← 0 to n − 1 by t do
    Using the LLRs S_inner,i, for i = {0, ..., n − 1}, update the LLRs for stage 2 with eq. (7).
    for j ← 1 to t do
        Using the updated LLRs from stage 2, update the LLRs for stage 1 with eq. (8) and obtain Ŝ_{b_j}.
        Calculate the PM for Ŝ_{b_j} with eq. (2) for each of the L paths.
        Update bits for stage 1.
    end
    Update the symbols for stage 2.
end
CRC-aided SCL decoder chooses the path with the smallest PM which passes the CRC and outputs û_0^{n−1}.
return û_0^{n−1}
```

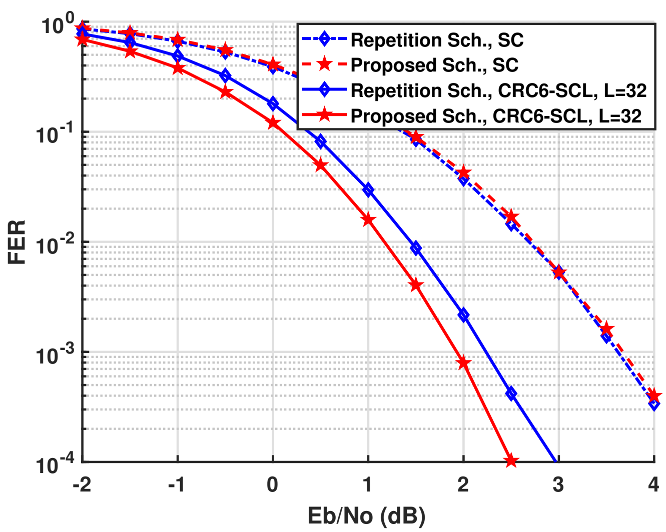

Figure 3: Performance comparison of the proposed scheme with straight forward polar-repetition for $N = 8192$, $n = 512$, $K = 40$, $r = 16$, $t = 2$.

B. Performance Analysis

In this subsection, we analyze the performance of the proposed scheme to gain insight into its better performance under the CRC-aided SCL decoder. Since the SCL decoder with large list size achieves performance very close to that of the ML decoder, we study the performance of the proposed scheme under ML decoding to have better analytical understanding.

The block error probability under ML decoding can be estimated via the truncated union bound as follows, [11].

$$P_e^{\mathrm{ML}} \leq \sum_{i=d}^{n} A_i Q(\sqrt{2iRE_b/N_o}), \tag{9}$$

where A_i is the number of the codewords of weight i and d is the minimum distance of the code. At high SNR, upper bound on P_e^{ML} depends primarily on d and A_d. Hence, to obtain a good performance under ML decoding, one needs to eliminate low-weight non-zero (LWNZ) codewords from the code.

To enumerate the LWNZ codewords, we transmit the all-zero codeword in the extremely high SNR regime under SCL decoder with very large list size, [10]. In this case, it is expected that the list most likely contains only the codewords with the least Hamming weights.

Table I compares the number of the low-weight codewords of the proposed scheme (on average) with the polar-repetition ones for $L = 2^{15}$. As it is expected the number of LWNZ codewords for the proposed scheme is less than the one of the polar-repetition scheme.

Table I: Number of Low-Weight Codewords

	$A_{64\times16}$	$A_{1329}-A_{2047}$	$A_{128\times16}$	$A_{2049}-A_{3071}$	$A_{192\times16}$	$A_{3073}-A_{4095}$	$A_{256\times16}$
Proposed Sch.	0	< 10	0	< 30	3	< 230	230
Simple Rep.	105	0	1365	0	5005	0	22819

	$A_{4097}-A_{5119}$	$A_{320\times16}$	$A_{5121}-A_{6143}$	$A_{384\times16}$	$A_{6145}-A_{7167}$	$A_{448\times16}$	$> A_{7169}$
Proposed Sch.	< 230	0	< 20	0	0	0	0
Simple Rep.	0	3003	0	455	0	15	0

C. Complexity Analysis

The complexity of the straightforward polar-repetition scheme consists of the complexity of the outer polar code of size n, $O(n \log n)$, and the repetition code of size r, nr. Therefore, the complexity of the total decoding process for the polar-repetition scheme is $O(nr + n \log n)$.

The complexity of the proposed scheme consists of the complexity of the inner and outer code. The complexity of the inner multiplicative repetition code is $2^t rn/t$. The complexity of the outer hybrid non-binary polar code consists of the i) complexity of the stage 2, $O(2^{2t} n/t \log(n/t))$ ii) complexity of the stage 1, $O(2^t n/t)$. Therefore, the complexity of the total decoding process is $O(2^t rn/t + 2^{2t}(n/t)\log(n/t) + 2^t n/t)$. If t and r be constants, the complexity of the hybrid non-binary repeated polar code will be $O(n \log n)$.

V. Conclusion

In this paper, we proposed a new concatenation scheme for improving the performance of the straightforward repetition at low-SNR regime. We concatenated the hybrid non-binary polar code with multiplicative repetition code and showed that the proposed scheme outperforms the straightforward repetition under CRC-aided SCL decoder over AWGN channel.

REFERENCES

[1] M. Fereydounian, M. V. Jamali, H. Hassani, and H. Mahdavifar, ”Channel coding at low capacity,” *Available at https://arxiv.org/abs/1811.04322*, March 2020.

[2] 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.

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

[4] R. Ratasuk, N. Mangalvedhe, Y. Zhang, M. Robert, and J.-P. Koskinen, ”Overview of narrowband IoT in LTE Rel-13,” *in Proc. IEEE Conf. Standard Commun. Netw. (CSCN)*, Berlin, Germany, Oct./Nov. 2016, pp. 1–7.

[5] I. Dumer, ”Polar codes with a stepped boundary,” *in Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2613–2617.

[6] K. Kasai, D. Declercq, C. Poulliat, and K. Sakaniwa, ”Multiplicatively repeated nonbinary ldpc codes,” *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6788–6795, 2011.

[7] I. Dumer, and N. Gharavi, ”Codes for high-noise memoryless channels”, *Available at https://www.researchgate.net/publication/340726241*, April 2020.

[8] N. Presman, O. Shapira, and S. Litsyn, ”Mixed-kernels constructions of polar codes,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 239–253, 2016.

[9] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, ”Fast multidimensional polar encoding and decoding,” *in Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, Feb. 2014, pp. 1–5.

[10] 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, December 2012.

[11] I. Sason, and S. Shamai, ”Performance Analysis of Linear Codes under Maximum-Likelihood Decoding: A Tutorial”, now, 2006.