# Capacity-Achieving Polar-Based Codes With Sparsity Constraints on the Generator Matrices

James Chin-Jen Pang, Hessam Mahdavifar, *Member, IEEE*, and S. Sandeep Pradhan, *Fellow, IEEE*

***Abstract*— In general, the generator matrix sparsity is a critical factor in determining the encoding complexity of a linear code. Further, certain applications, e.g., distributed crowdsourcing schemes utilizing linear codes, require most or even all the columns of the generator matrix to have some degree of sparsity. In this paper, we leverage polar codes and the well-established channel polarization to design capacity-achieving codes with a certain constraint on the weights of all the columns in the generator matrix (GM) while having a low-complexity decoding algorithm. We first show that given a binary-input memoryless symmetric (BMS) channel $W$ and a constant $s \in (0, 1]$, there exists a polarization kernel such that the corresponding polar code is capacity-achieving with the *rate of polarization* $s/2$, and the GM column weights being bounded from above by $N^s$. To improve the sparsity versus error rate trade-off, we devise a column-splitting algorithm and two coding schemes for BEC and then for general BMS channels. The *polar-based* codes generated by the two schemes inherit several fundamental properties of polar codes with the original $2 \times 2$ kernel including the decay in error probability, decoding complexity, and the capacity-achieving property. Furthermore, they demonstrate the additional property that their GM column weights are bounded from above sublinearly in $N$, while the original polar codes have some column weights that are linear in $N$. In particular, for any BEC and $\beta < 0.5$, the existence of a sequence of capacity-achieving polar-based codes where all the GM column weights are bounded from above by $N^\lambda$ with $\lambda \approx 0.585$, and with the error probability bounded by $\mathcal{O}(2^{-N^\beta})$ under a decoder with complexity $\mathcal{O}(N \log N)$, is shown. The existence of similar capacity-achieving polar-based codes with the same decoding complexity is shown for any BMS channel and $\beta < 0.5$ with $\lambda \approx 0.631$.**

***Index Terms*— Linear codes, polar codes, channel coding, Hamming weight, sparse matrices, error probability.**

## I. Introduction

CAPACITY-APPROACHING error-correcting codes such as low-density parity-check (LDPC) codes [1] and polar codes [2] have been extensively studied for applications in wireless and storage systems. Besides conventional applications of codes for error correction, a surge of new applications has also emerged in the past decade including crowdsourcing [3], [4], distributed storage [5], and speeding up distributed machine learning [6], [7]. To this end, new motivations have arisen to study codes with sparsity constraints on their generator and/or parity-check matrices. For instance, the stored data in a failed server needs to be recovered by downloading data from a few servers only, due to bandwidth constraints, imposing sparsity constraints in the decoding process in a distributed storage system. In crowdsourcing applications, e.g., when workers are asked to label items in a dataset, each worker can be assigned only a few items due to capability limitations, imposing sparsity constraints in the encoding process. More specifically, codes defined by sparse generator matrices become relevant for such applications [8], [9].

In this paper, we focus on polar codes in order to construct a sequence of codes defined by sparse GMs with practical utility, such as low decoding complexity, explicit construction, sufficiently fast decay in the error probability, and the potential to approach capacity at large block-length.

### A. Polar Codes

Channel polarization, introduced by Arıkan [2], [10], is one of the most recent breakthroughs in coding theory. Polar codes are a class of provably capacity-achieving channel codes with explicit construction for general BMS channels, and have attracted significant attention due to their error correction performance, as well as their low-complexity decoding algorithms. Within the ongoing fifth generation wireless systems (5G) standardization process, polar codes have been adopted for uplink and downlink control information for the enhanced mobile broadband (eMBB) communication service. Furthermore, polar codes and polarization phenomenon have been successfully applied to a wide range of problems including data compression [11], [12], broadcast channels [13], [14], multiple access channels [15], [16], physical layer security [17], [18], and coded modulations [19].

### B. LDGM and Related Works

A related line of work on studying linear codes with sparsity constraints on their generator matrices is by associating them with sparse graph representations [20]. In this context, they are referred to as low-density generator matrix (LDGM) codes, also regarded as the counterpart of LDPC codes. The sparsity

Manuscript received 18 August 2022; revised 16 February 2023 and 15 May 2023; accepted 25 May 2023. Date of publication 5 June 2023; date of current version 18 September 2023. This work was supported by the National Science Foundation under grants CCF–1763348, CCF–1909771, CCF–2007878, and CCF–2132815. An earlier version of this paper was presented in part at the 2020 IEEE International Symposium on Information Theory [DOI: 10.1109/ISIT44484.2020.9174358]. The associate editor coordinating the review of this article and approving it for publication was P. Trifonov. *(Corresponding author: James Chin-Jen Pang.)*

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: cjpang@umich.edu; hessam@umich.edu; pradhanv@umich.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2023.3282603.

Digital Object Identifier 10.1109/TCOMM.2023.3282603

of the generator matrices of LDGM codes leads to a low encoding complexity, and has been adopted in applications such as lossy source compression [21] and multiple description coding [22]. In [23] and [24] it was pointed out that certain constructions of LDGM codes are not asymptotically *good*, a behavior which is also studied using an error floor analysis in [25] and [26].

In terms of the sparsity of the GM, the authors of [27] showed the existence of capacity-achieving codes over binary symmetric channels (BSC) using random linear coding arguments when the column weights of the GM are upper bounded by $\epsilon N$, for any $\epsilon > 0$, where $N$ is the code block length. Also, it is conjectured in [27] that column weight upper bounds that scale sublinearly in $N$ suffice to achieve the capacity. For binary erasure channels (BEC), bounds that scale as $\mathcal{O}(\log N)$ suffice for achieving the capacity, again using random linear coding arguments [27]. Furthermore, the scaling exponent of such random linear codes are studied in [28]. Later, in [29], the existence of capacity-achieving systematic LDGM ensembles over any BMS channel with the expected value of the weight of the entire GM bounded by $\epsilon N^2$, for any $\epsilon > 0$, is shown. While the (ensemble-averaged) block-error probability for the codes goes to zero as the block-length grows large, the speed of decay in the error probability is not provided in [27] and [29].

In [8], the problem of label learning through queries from a crowd of workers was formulated as a coding theory problem. Due to practical constraints in such crowdsourcing scenarios, each query can only contain a small number of items. In [9], we considered the same setting as in [8] with the additional consideration that some workers may not respond to queries, a scenario that resembles a binary erasure channel. Then we showed that a combination of LDPC codes and LDGM codes gives a query scheme where the number of queries approaches the information-theoretic lower bound [9].

In the realm of quantum error correction, quantum low-density-generator-matrix (QLDGM) codes, quantum low-density-parity-check (QLDPC) codes, and other sparse-graph-based schemes have been extensively studied due to the small numbers of quantum interactions per qubit during the encoding and/or error correction procedure, avoiding additional quantum gate errors and facilitating fault-tolerant decoding. Amongst these schemes, the error correction performance of the LDGM-based codes proposed in [30] was shown to outperform all other Calderbank-Steane-Shor (CSS) and non-CSS codes of similar complexity.

In both applications highlighted above, the benefit of the LDGM codes follows from a certain upper bound on the column weights of the GM, ensuring the columns are relatively low weight. Motivated by these applications, the main goal of this work is to construct sequences of codes where all of the columns of the GM are *low weight*, where certain upper bounds on the weight will be specified later.

### C. Our Contributions

In this paper, we study capacity-achieving polar and polar-based codes over BMS channels with sparsity constraints on generator matrix column weights. Leveraging polar codes based on general kernels, with rates of polarization studied in [31], we show that capacity-achieving polar codes with column weights bounded from above by $N^s$ exist for any given $s > 0$, where $N$ is the code block length. This verifies the conjecture given in [27]. There is, however, a trade-off between the sparsity parameter $s$ and the rate of polarization, given by $\frac{s}{2}$.

For the case when the speed of decay for block-error probability and the GM sparsity are both constrained, we propose two new code constructions with sparse GM columns, which provide a better trade-off for $s > 0.585$. We first consider BEC, and propose a splitting algorithm termed *decoder-respecting splitting (DRS)* algorithm, which, roughly speaking, splits *heavy* columns in the GM into several *light* columns. Note that if one splits the heavy columns in an arbitrary manner to form a new GM, the code defined by the new GM may be substantially different from the original one in terms of the error probability and/or having a low-complexity decoder. Leveraging the fact that the polarization transform of a BEC leads to BECs, the DRS algorithm converts the encoder of a standard polar code into an encoder defined by a sparse GM without hurting the reliability of the bit-channels observed by the source bits. Furthermore, the specific structure of DRS enables a low-complexity successive cancellation decoder in a recursive fashion inheriting that of original polar codes. In particular, we show a sequence of codes defined by GMs with column weights upper bounded by $N^\lambda$, for any $\lambda > \lambda^* \approx 0.585$ and the existence of a decoder with computation complexity $\mathcal{O}(N \log N)$ under which the block-error probability is bounded by $2^{-N^\beta}$ for any $\beta < 0.5$.

Next, for general BMS channels, we propose an enhancement of the DRS-based encoding scheme, referred to as *augmented-DRS (ADRS)* scheme, which requires additional channel uses and decoding complexity. In spite of these limitations, we show that there exists a sequence of capacity-achieving codes, referred to as the *polar-ADRS* codes. The sequence of codes is defined by GMs with column weights upper bounded by $N^\lambda$, for any $\lambda > \lambda^\dagger \approx 0.631$, and can be decoded with complexity $\mathcal{O}(N \log N)$.

The rest of the paper is organized as follows. In Section II, we introduce basic notations and definitions for channel polarization and polar codes. Section III provides a sparsity result for polar codes with general kernels. In Section IV, we introduce the DRS algorithm and the ADRS scheme, and the corresponding code constructions over the BEC and BMS channels, respectively. The successive cancellation decoders are also described and shown to be of low computation complexity. Finally, Section V concludes the paper. The proofs for the results in Sections III and IV are included in the Appendix.

## II. Preliminaries

Let $h_b(\cdot)$ denote the binary entropy function, $\exp_2(x)$ denotes the function $2^x$, $\ln(\cdot)$ be the logarithmic function with base $e$, and $\log(\cdot)$ be the logarithmic function with base 2. $Z(W)$ denote the Bhattacharyya parameter of a channel $W$.

We give formal definitions for the BMS channel and capacity-achieving codes for readers' reference.

*Definition 1:* A binary memoryless symmetric channel (BMS) $W : \mathcal{X} \to \mathcal{Y}$ is a noisy memoryless channel with binary input alphabet $\mathcal{X}$, and channel output alphabet $\mathcal{Y}$, (we use $\mathcal{X} = \{0, 1\}$, and assume $\mathcal{Y}$ is finite, in this paper.) such that $\Pr[Y = y|X = 0] = \Pr[Y = \phi(y)|X = 1]$ for all $y \in \mathcal{Y}$ for some involution $\phi$ on $\mathcal{Y}$.

*Definition 2:* A type of code is said to be capacity achieving over a BMS channel $W$ with capacity $C = I(W) > 0$ if, for any given constant $R < C$, there exists a sequence of codes with rate $R$ and the block-error probability vanishes as the block length $N$ grows large. The block-error probability is evaluated under the maximum likelihood (ML) decoder, unless a different decoding scheme is specified.

### A. Channel Polarization and Polar Codes

The *channel polarization* phenomenon was discovered by Arıkan [2] and is based on the polarization transform as the building block. Let $\mathcal{W}$ denote the class of all BMS channels. The channel transform $W \mapsto (W^-, W^+)$ that maps $\mathcal{W}$ to $\mathcal{W}^2$, where $W^- : \mathcal{X} \to \mathcal{Y}^2$ and $W^+ : \mathcal{X} \to \mathcal{Y}^2 \times \mathcal{X}$, is defined in [2] and is often referred to as a polarization recursion. Then a channel $W^{s_1,s_2,\ldots,s_n}$ with $s_i \in \{-,+\}, i = 1, 2, \ldots, n$, can be defined by applying the channel transform $n$ times recursively, as in [2].

For $N = 2^n$, the polarization transform is obtained from the $N \times N$ matrix $G_2^{\otimes n}$, where $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ [2], and $A^{\otimes n}$ denote the $n$-fold Kronecker product of $A$. A polar code of length $N$ is constructed by selecting certain rows of $G_2^{\otimes n}$ as its generator matrix. More specifically, let $K$ denote the code dimension. Then all the $N$ bit-channels in the set $\{W^{s_1,s_2,\ldots,s_n} : s_i \in \{-,+\} \text{ for } i = 1, 2, \ldots, n\}$, resulting from the polarization transform, are sorted with respect to an associated parameter, e.g., their probability of error (or Bhattacharyya parameter), the best $K$ of them with the lowest probability of error are selected, and then the corresponding rows from $G_2^{\otimes n}$ are selected to form the GM. Hence, the GM of an $(N, K)$ polar code is a $K \times N$ sub-matrix of $G_2^{\otimes n}$. Then the probability of error of this code, under successive cancellation (SC) decoding, is upper bounded by the sum of probabilities of error of the selected $K$ best bit-channels [2].

### B. General Kernels and Error Exponent

It is shown in [31] that if $G_2$ is replaced by an $l \times l$ matrix $G_l$, then polarization still occurs if and only if $G_l$ is an invertible matrix in $\mathbb{F}_2$ and it is not upper triangular under any column permutation, in which case the matrix $G_l$ is called a *polarization kernel*. Furthermore, the authors of [31] provided a general formula for the speed of the error rate decay of polar codes constructed based on an arbitrary $l \times l$ polarization kernel $G_l$. More specifically, let $N = l^n$ denote the block length and $C$ denote the capacity of the channel. For any fixed $\beta < E(G_l)$ and fixed code rate $R < C$, where $E(G_l)$ denotes the rate of polarization (see [31, Definition 7]), there is a sequence of polar codes based on $G_l$ with probability of error $P_e$ under SC decoding bounded by $P_e(n) \leqslant 2^{-N^\beta}$, for all sufficiently large $n$. The rate of polarization $E(G_l)$ is given by $E(G_l) = \frac{1}{l}\sum_{i=1}^{l} \log_l D_i$, where $\{D_i\}_{i=1}^{l}$ are the *partial distances* of $G_l$. More specifically, for $G_l = [g_1^T, g_2^T, \ldots, g_l^T]^T$, the partial distances $D_i$ are given by $D_i \triangleq d_H(g_i, \text{span}(g_{i+1}, \ldots, g_l))$ for $i = 1, 2, \ldots, l$, where $d_H(a, b)$ is the Hamming distance between two vectors $a$ and $b$, and $d_H(a, U)$ is the minimum distance between a vector $a$ and a subspace $U$, i.e., $d_H(a, U) = \min_{u \in U} d_H(a, u)$.

## III. Sparse Polar Code Constructions Based on Large Kernels

In this section we first show the existence of capacity-achieving polar codes with generator matrices for which all column weights scale at most polynomially with arbitrarily small degree in the block length $N$, hence validating the conjecture in [27]. Second, we show that, for any polar code of rate 1, *almost* all of the column weights of the GM are polynomial in $N$.

*Theorem 1:* For any fixed $s \in (0, 1)$ and any BMS channel, there are capacity-achieving polar codes under SC decoding, with generator matrices having column weights bounded by $N^s$, where $N$ denotes the block length of the code.

*Proof:* Consider an $l \times l$ polarizing matrix $G_l = \begin{bmatrix} I_{\frac{l}{2}} & 0_{\frac{l}{2}} \\ I_{\frac{l}{2}} & I_{\frac{l}{2}} \end{bmatrix}$, where $l$ is an even integer such that $l \geqslant 2^{\frac{1}{s}}$. The partial distances are $D_i = 1$ for $1 \leqslant i \leqslant \frac{l}{2}$ and $D_i = 2$ for $\frac{l}{2} + 1 \leqslant i \leqslant l$. Hence, the rate of polarization $E(G_l) = \frac{1}{2}\log_l 2 > 0$, and there is a sequence of capacity-achieving polar codes constructed using $G_l$ as the polarizing kernel. Note that in $G_l$, each column has weight at most 2 and, hence, the column weights of $G_l^{\otimes n}$ are upper bounded by $2^n$. By the specific choice of $l$, we have $2^n \leqslant (l^s)^n = (l^n)^s = N^s$, where $N = l^s$ is the block length of the code. This completes the proof. ∎

*Remark 1:* While Theorem 1 provides a theoretical guarantee on the existence of capacity-achieving polar codes with sparse generator matrices, the sparsity comes at a cost. Specifically, the rate of polarization $E(G_l) = \frac{1}{2}\log_l 2 \leqslant \frac{s}{2}$ is smaller than that associated with the kernel $G_2$, given by $E(G_2) = 0.5$. On the other hand, while the SC decoding complexity for polar codes defined by general $l \times l$ kernels behaves as $\mathcal{O}(\frac{2^l}{l} N \log_l N)$ [31], in this case, the complexity scales as $\mathcal{O}(N \log_l N)$ by considering the following viewpoint. Interleave $(l/2)^n$ copies of the polar code with block length $2^n$ based the standard $G_2$ kernel, to form a code with block length $N = l^n$ with an $n$-stage recursive encoder structure. By decoding each copy with complexity $\mathcal{O}(2^n \log 2^n) = \mathcal{O}(n2^n)$ under the SC decoder, the entire code can be decoded with complexity $\mathcal{O}((l/2)^n \cdot n2^n) = \mathcal{O}(N \log_l N)$.

Since we can construct capacity-achieving codes with column weights upper bounded by $N^s$ with any fixed $s > 0$, by using polar codes, the question now is whether it is possible to further improve the sparsity of polar code GMs. For instance, we know it is possible to have an upper bound of $\mathcal{O}(\log N)$ on all the GM column weights of capacity-achieving codes, over the BEC, by utilizing random linear

ensembles [27]. For rate-1 polar codes, the proposition below answers the inquiry in the negative, by showing that almost all the GM columns have weights lower bounded by a polynomial in $N$.

*Proposition 1:* Given any $l \geqslant 2$, $l \times l$ polarizing kernel $G_l$, and $\frac{1}{l} > r > 0$, the fraction of columns in $G_l^{\otimes n}$ with $\mathcal{O}(N^{r \log_l 2})$ Hamming weight vanishes as $n$ grows large, where $N = l^n$.

*Proof:* The proof is given in Appendix Section A. ∎

The trade-off highlighted in Remark 1 suggests that off-the-shelf polar code constructions with large kernels may not be the ideal option when the speed of decay of the error probability is a concern. However, the heaviest column in the polar code with kernel $G_2$ scales as $\Theta(N)$ for any code rate. To construct codes with sparse GM and suitable decay of the error probability, in the next section, we propose a *splitting* algorithm for the generator matrix and investigate the resulting codes in terms of the error probability, GM column sparsity, and the decoding complexity.

## IV. Sparse Polar-Based Codes With Low-Complexity Decoding

When all columns of a matrix $G$ are required to be sparse, that is, have low Hamming weights, a *splitting algorithm* is applied. Given a column weight threshold $w_{u.b.}$, a splitting algorithm splits any column in $G$ with weight exceeding $w_{u.b.}$ into columns that sum to the original column both in $\mathbb{F}_2$ and in $\mathbb{R}$, and that have weights no larger than $w_{u.b.}$.

Note that a column of $G$ is left intact by the splitting algorithm as long as its Hamming weight does not only exceed $w_{u.b.}$. Thus a splitting algorithm would be described as an algorithm which takes as input a column vector $\mathbf{v}$ and a weight threshold $w_{u.b.}$, and returns a set of column vectors whose lengths are equal to the length of $\mathbf{v}$. Given a matrix $A$ with $m$ columns and a threshold $w_{u.b.}$, with a slight abuse of notation, the matrix *generated* by a splitting algorithm is defined as the matrix whose column vectors are those from the $m$ sets, which are respectively the outputs of the algorithm for each column of $A$. For example, consider a $4 \times 2$ matrix $A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}^T = [\mathbf{a_1}, \mathbf{a_2}]$, a threshold $w_{u.b.} = 2$, and a splitting algorithm $\mathcal{S}$. Let $\mathcal{S}(\mathbf{a_i}, w_{u.b.})$, $i = 1, 2$, be the sets of vectors returned by $\mathcal{S}$, given by $\mathcal{S}(\mathbf{a_1}, w_{u.b.}) = \left\{[1,0,1,0]^T, [0,0,0,1]^T\right\}, \mathcal{S}(\mathbf{a_2}, w_{u.b.}) = \left\{[1,0,1,0]^T, [0,1,0,0]^T\right\}$. The matrix generated by $\mathcal{S}$ for $A$ is then a $4 \times 4$ matrix of the form $[[1,0,1,0]^T, [0,0,0,1]^T, [1,0,1,0]^T, [0,1,0,0]^T]$, or a column permutation of it.

Let an $(N, K)$ polar code $\mathcal{C}$ have a $K \times N$ submatrix of $G = G_2^{\otimes n}$ as the generator matrix, and $G'$ denote the $N \times N(1+\gamma)$ matrix generated by the splitting algorithm, where $N = 2^n$. A new code *based on* $G'$ selects the same $K$ rows of $G'$ as the polar code $\mathcal{C}$ to form the generator matrix, where all the column weights are bounded by $w_{u.b.}$. Such a code is referred to as a *polar-based code corresponding to* $G'$, or a PB$(G')$ code, in this paper.

Note that more detailed description is needed to uniquely specify a splitting algorithm, which then determines the term $\gamma$ and the performance of the PB$(G')$ code. Specifically, the channel polarization phenomenon and the recursive encoding and decoding structure may be invalid when the GM is modified by the splitting algorithm. These changes also imply that the codes with the split GM may suffer from drawbacks such as weaker bounds on error probability and larger decoding complexity, as well as the rate loss with a multiplicative factor of $1 + \gamma$, when compared to the polar codes. In this section, we introduce a *splitting* algorithms, referred to as the decoder-respecting splitting (DRS) algorithm, and two encoding schemes that are effective in avoiding the drawbacks. These schemes enable low-complexity SC decoders based on likelihood ratios that can be calculated with a recursive algorithm. Specifically, when the threshold $w_{u.b.}$ is chosen appropriately, we show in Section IV-A that the term $\gamma$ goes to 0 exponentially fast in $n$, when the DRS algorithm is applied to columns of the matrix $G = G_2^{\otimes n}$. The encoding of the resulting PB$(G')$ codes can be realized by a encoding scheme which inherits the recursive structure of the original polar codes, except only at locations that corresponds to a split of a column of $G$, as dictated by the DRS algorithm. At these locations, the exclusive-OR operations are removed and additional copies of the underlying channel are used. The PB$(G')$ codes suffer only a negligible $1 + \gamma$ multiplicative factor of rate loss compared to the original polar codes for large $n$. For BEC, this sequence of codes is capacity-achieving with an error exponent of $\frac{1}{2}$, under a new SC decoding scheme (see Theorem 2 in Section IV-B). For general BMS channels, another encoding scheme, referred to as the *ADRS* scheme, is proposed in Section IV-C. This scheme introduces additional 'noise' nodes and requires even more copies of the underlying channel when the DRS algorithm requires a split. For codes generated by this scheme, results similar to that in Theorem 2 are available with a slightly stricter condition on the choice of $w_{u.b.}$.

### A. Decoder-Respecting Splitting Algorithm

The main idea of the DRS algorithm is to construct a generator matrix that can be realized with an encoding pattern similar to conventional polar codes such that the column weights of the matrix associated with the diagram are at most $w_{u.b.}$. The pseudo code for the algorithm is provided in Algorithm 1.

The core of the algorithm is the DRS-Split function. When the weight of the input the vector $\mathbf{x}$ is larger than the threshold, it splits the vector in half into vectors $\mathbf{x}_h$ and $\mathbf{x}_t$, and recursively finds two sets, $Y_h$ and $Y_t$, composed of vectors with the length halved compared to the length of $\mathbf{x}$. The vectors are then appended to the length of $\mathbf{x}$, which collectively form the output of the function. For a vector $\mathbf{u} \in \{0,1\}^{m \times 1}$, let $|\mathbf{u}| = m$ denote its length, and $w_H(\mathbf{u})$ its Hamming weight. We note that the weights of vectors in $Y_h$ and $Y_t$ are respectively upper bounded by the weights of $\mathbf{x}_h$ and $\mathbf{x}_t$, both of which are bounded by $k = |\mathbf{x}_h| = |\mathbf{x}_t|$, and that the value of $k$ is halved each iteration. Hence, the function is guaranteed to terminate as long as the threshold is a positive integer.

We use a simple example to illustrate the algorithm. Let $n = 3$, $v = [0,0,0,0,1,1,1,1]^T$ and $w_{u.b.} = 2$. Since

**Algorithm 1** DRS Algorithm

**Input:** weight threshold $w_{u.b.} \in \mathbb{N}$, a column vector $\mathbf{v} \in \{0,1\}^{2^n \times 1}$

**Output:** the set of vectors with length $2^n$ returned by DRS-SPLIT$(w_{u.b.}, \mathbf{v})$

1: **function** DRS-SPLIT$(w_{u.b.}, \mathbf{x})$
2: **if** $w_H(\mathbf{x}) > w_{u.b.}$ **then**
3: $k \leftarrow \text{length}(\mathbf{x})/2$
4: $\mathbf{x}_h \leftarrow (x_1, \ldots, x_k)^T$, $\mathbf{x}_t \leftarrow (x_{k+1}, \ldots, x_{2k})^T$
5: $Y_h \leftarrow$ DRS-SPLIT$(w_{u.b.}, \mathbf{x}_h)$
6: $Y_t \leftarrow$ DRS-SPLIT$(w_{u.b.}, \mathbf{x}_t)$
7: **if** $\mathbf{x}_h = \mathbf{0}_{k\times 1}$ **then**
8: **return** $\bigcup_{y\in Y_t} \{(\mathbf{0}_{1\times k}, y^T)^T\}$
9: **else if** $\mathbf{x}_t = \mathbf{0}_{k\times 1}$ **then**
10: **return** $\bigcup_{y\in Y_h} \{(y^T, \mathbf{0}_{1\times k})^T\}$
11: **else**
12: **return** $\bigcup_{y\in Y_t} \{(\mathbf{0}_{1\times k}, y^T)^T\} \cup \bigcup_{y\in Y_h} \{(y^T, \mathbf{0}_{1\times k})^T\}$
13: **end if**
14: **else if** $w_H(\mathbf{x}) = 0$ **then**
15: **return** $\{\}$
16: **else**
17: **return** $\{\mathbf{x}\}$
18: **end if**
19: **end function**

the weight of $v$ exceeds the threshold, it is first split into $\mathbf{x}_h = [0,0,0,0]^T$ and $\mathbf{x}_t = [1,1,1,1]^T$. Since $\mathbf{x}_h$ is an all-zero vector, $Y_h$ is an empty set according to line 14 to 15. To compute $Y_t$ =DRS-SPLIT$(2, [1,1,1,1]^T)$, the function splits the input into half again, thereby obtaining $\mathbf{x}'_h = [1,1]^T$ and $\mathbf{x}'_t = [1,1]^T$. The corresponding $Y'_h$ and $Y'_t$ are then both $\{[1,1]^T\}$ and, hence, we have $Y_t = \{[0,0,1,1]^T\} \cup \{[1,1,0,0]^T\} = \{[0,0,1,1]^T, [1,1,0,0]^T\}$. Since $\mathbf{x}_h = \mathbf{0}_{4\times 1}$, the function proceeds to lines 7 and 8, and returns $\{[0,0,0,0,0,0,1,1]^T, [0,0,0,0,1,1,0,0]^T\}$.

In order to analyze the effect of the DRS algorithm on the matrix $G_2^{\otimes n}$, we show that the size of the algorithm output does not depend on the order of a sequence of Kronecker product operations, where the *size* of a set of vectors stands for the number of vectors in the set. Suppose that the Kronecker product operations with the vector $[1,\ 1]^T$ for $n_1$ times and with the vector $[0,\ 1]^T$ for $n_2$ times are applied on a vector $v$, where $n = n_1 + n_2$ and the order of the operations is specified by a sequence $(s_1, s_2, \ldots, s_n) \in \{-,+\}^n$ with $|\{i : s_i = -\}| = n_1$ and $|\{i : s_i = +\}| = n_2$. Also, let $v^{(i)}$ denote the output of applying the first $i$ Kronecker product operations on $v$. It is defined by the following recursive relation:

$$v^{(i)} = \begin{cases} v^{(i-1)} \otimes [1,1]^T, & \text{if } s_i = -, \\ v^{(i-1)} \otimes [0,1]^T, & \text{if } s_i = +, \end{cases} \tag{1}$$

for $i \geqslant 1$ and the initial condition $v^{(0)} = v$. We use $v^{(s_1,s_2,\ldots,s_i)}$ instead of $v^{(i)}$ when the sequence is needed for clarity. The following lemma shows that any two vectors of the form $v^{(s_1,s_2,\ldots s_n)}$ will be split into the same number of columns under the DRS algorithm as long as the sequences associated with them contain the same number of $-$ and $+$ signs.

*Lemma 1:* Let $n = n_1 + n_2$ and $(s_1, \ldots, s_n) \in \{-,+\}^n$ be a sequence with $n_1$ minus signs and $n_2$ plus signs. Let $v^{(n)}$ be the vector defined by a vector $v$ and the sequence $(s_1, \ldots, s_n)$ through equation (1). Then the size of the DRS algorithm output for $v^{(n)}$ depends only on the values $n_1$ and $n_2$.

*Proof:* The proof is given in Appendix Section B.1. ■

Let a $K \times N$ matrix $M = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]$ and a threshold $w_{u.b.}$ be given. Suppose that the DRS algorithm is applied to each column in $M$ and the sum of the sizes of the output sets is $N(1+\gamma)$. Then DRS$(M)$ is defined as the $K \times N(1+\gamma)$ matrix consisting of all the vectors in the output sets (with repetition).

We study the effect of the DRS algorithm in terms of the multiplicative rate loss, i.e., $1+\gamma$. Since all the columns of $G_2^{\otimes n}$ are in the form of $v^{(s_1,s_2,\ldots,s_n)}$ with $v = [0,1]^T$ or $v = [1,1]^T$, Lemma 1 substantially simplifies the analysis for $\gamma$. In particular, the following proposition shows an appropriate choice of $w_{u.b.}$ guarantees the existence of a sparse polar-based GM with vanishing $\gamma$.

*Proposition 2:* Let the columns of $G_2^{\otimes n}$ be the inputs for the DRS algorithm and DRS$(G_2^{\otimes n})$ be the $N \times N(1+\gamma)$ matrix generated by the DRS algorithm for $G_2^{\otimes n}$. The term $\gamma$ vanishes exponentially fast as $n$ goes to infinity for any $w_{u.b.} = 2^{n\lambda}$ with $\lambda > \lambda^* \triangleq h_b(\frac{2}{3}) - \frac{1}{3} \approx 0.585$.

*Proof:* The proof is given in Appendix Section B.2. ■

For the effect of the DRS algorithm on $G_2^{\otimes n}$ with finite $n$, we compute values of $\gamma$ for various combinations of $n$ and $\lambda$, as shown in Figure 1. The numerical results with $6 \leqslant n \leqslant 26$ indicate that, for $0.5 \leqslant \lambda < 0.6$, the multiplicative rate loss $\gamma$ is larger with larger $n$, and for $\lambda \geqslant 0.65$, $\gamma$ is smaller with larger $n$. The fact that the $n = 26$ does not provide the smallest $\gamma$ for $\lambda$ close to $\lambda^*$ should not be considered a contradiction to Proposition 2. Instead, the closer $\lambda > \lambda^*$ is, the larger $n$ it takes for the exponential decay of $\gamma$ to dominate.

### *B. Low-Complexity Decoder for Polar-Based Codes: BEC*

In this section, we show two results for the polar-based code corresponding to DRS$(G_2^{\otimes n})$ over the BEC. Such codes are referred to as the *polar-DRS* codes in this paper. First, we propose a low-complexity suboptimal decoder for the polar-DRS codes. Second, with the low-complexity suboptimal decoder, the polar-DRS codes are capacity-achieving for suitable column weight threshold.

It is known that when the channel transformation with kernel $G_2$ is applied to two BECs, the two new bit-channels are also BECs. Specifically, for two binary erasure channels $W_1$ and $W_2$ with erasure probabilities $\epsilon_1$ and $\epsilon_2$, respectively, the polarized bit-channels $W^-(W_1, W_2)$ and $W^+(W_1, W_2)$ are binary erasure channels with erasure probabilities $\epsilon_1+\epsilon_2-\epsilon_1\epsilon_2$ and $\epsilon_1\epsilon_2$, respectively.

The mutual information $I(\cdot)$ and Bhattacharyya parameter $Z(\cdot)$ of a BEC $W$ with erasure probability $\epsilon$ are given by:

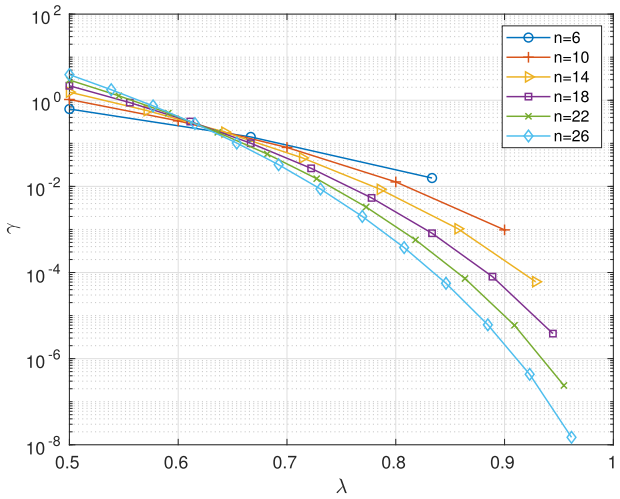



Fig. 1. Multiplicative rate loss factor $\gamma$ versus $\lambda$, where $w_{u.b.} = N^{\lambda}$.

$I(W) = 1 - \epsilon, Z(W) = \epsilon$. For a sequence $(s_1, s_2, \ldots, s_n) \in \{-,+\}^n$, the function $\mathsf{Bi2De}(s_1, s_2, \ldots, s_n)$ returns the decimal value of the binary string in which a minus sign for $s_i$ is regarded as a 0 and a plus sign as a 1, e.g., $\mathsf{Bi2De}(-,+,+) = (011)_2 = 3$. Let $G$ denote $G_2^{\otimes n}$ and $G'$ denote $\mathrm{DRS}(G_2^{\otimes n})$, and let $Z_G^{(s_1 s_2 \ldots s_n)}$ denote the Bhattacharyya parameter of the bit-channel $W^{s_1 s_2 \ldots s_n}$, which is equal to $W_N^{(\mathsf{Bi2De}(s_1, s_2, \ldots, s_n)+1)}$ in [2, page 3]. The term $Z_{G'}^{(s_1 s_2 \ldots s_n)}$ denotes the Bhattacharyya parameter of the bit-channel observed by the source bit of the same index corresponding to $G'$.

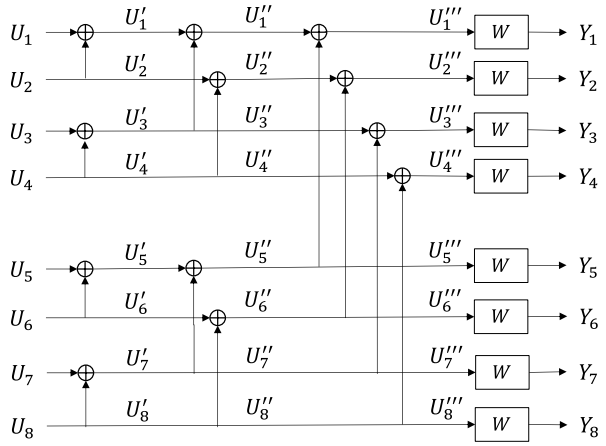
The following lemma shows that the bit-channel observed by each source bit is better in terms of the Bhattacharyya parameter when $G'$ is the generator matrix instead of $G$.

*Lemma 2:* Let $w_{u.b.}$ and $n$ be given, and let $G$ denote $G_2^{\otimes n}$ and $G'$ denote $\mathrm{DRS}(G_2^{\otimes n})$. The following is true for any $(s_1, s_2, \ldots, s_n) \in \{-,+\}^n$:

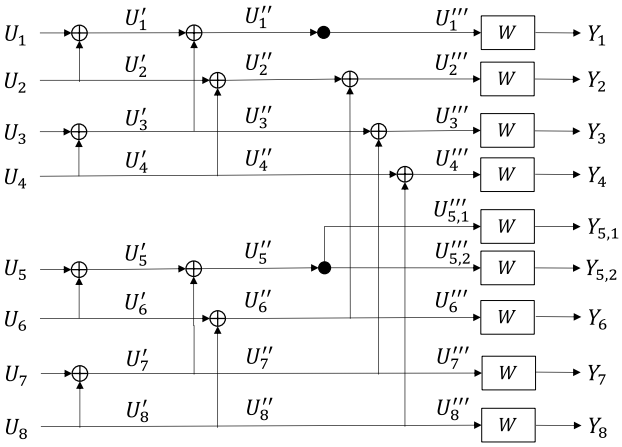
$$Z_{G'}^{(s_1 s_2 \ldots s_n)} \leqslant Z_G^{(s_1 s_2 \ldots s_n)}.$$

*Proof:* The proof is given in Appendix C. ∎

*Remark 2:* A key to the proof of Lemma 2 is a recursive encoding scheme for the relationship $\mathbf{x} = \mathbf{u}G'$, where $\mathbf{u}$ and $\mathbf{x}$ are row vectors of lengths $N = 2^n$ and $N(1+\gamma)$, respectively. The encoding scheme is most easily understood by considering the low-complexity encoding structure for the standard polar code, as seen in [32], and replacing the exclusive-OR (XOR) operations at locations that correspond to the splitting operations dictated by the DRS algorithm. Specifically, when a split is required on a column of $G$, the corresponding XOR node is removed, the input bit for the 'worse' channel remains untouched, and two copies of the input bit for the 'good' channel are transmited through the underlying channel. For example, consider $G = G_2^{\otimes 3}$. The encoding diagram for codes




(a) Encoding block for generator matrix $G = G_2^{\otimes 3}$




(b) Encoding block for generator matrix $G'$

Fig. 2. Encoding structure change due to the DRS algorithm with $N = 8, w_{u.b.} = 4$.

defined by $G$ is shown in Figure 2a. We have

$$G = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 1&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0 \\ 1&0&0&0&1&0&0&0 \\ 1&1&0&0&1&1&0&0 \\ 1&0&1&0&1&0&1&0 \\ 1&1&1&1&1&1&1&1 \end{bmatrix},$$

$$G' = \begin{bmatrix} 1&0&0&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0&0 \\ 1&0&1&1&1&0&0&0&0 \\ 0&1&0&0&0&1&0&0&0 \\ 0&1&1&0&0&1&1&0&0 \\ 0&1&0&1&0&1&0&1&0 \\ 0&1&1&1&1&1&1&1&1 \end{bmatrix},$$

where $G'$ is the matrix $\mathrm{DRS}(G_2^{\otimes 3})$ when $w_{u.b.} = 4$. The encoding structure for $G'$ is shown in Figure 2b. Since the first column is the only column of $G$ split by the DRS algorithm

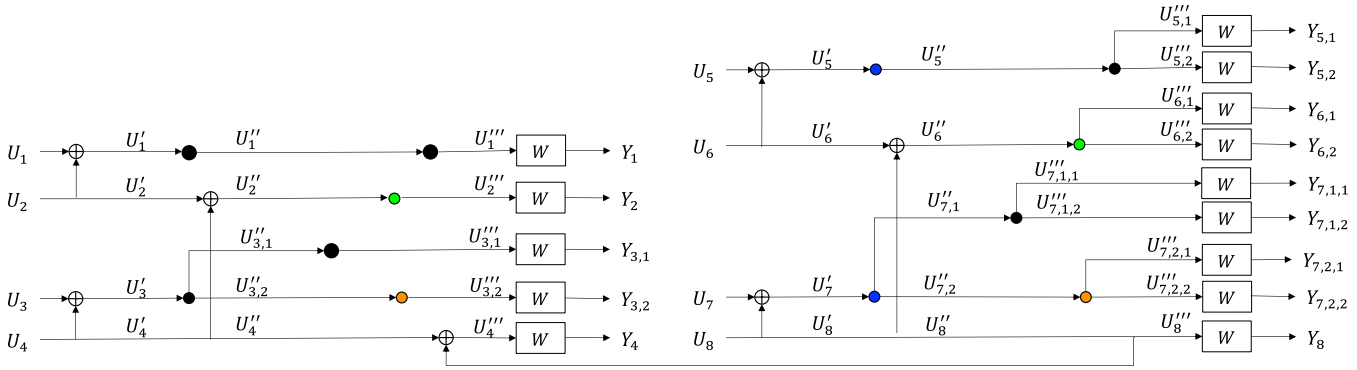



Fig. 3. Encoding block for generator matrix $\mathrm{DRS}(G_2^{\otimes 3})$ when $w_{u.b.} = 2$.

when $w_{u.b.} = 4$, we remove the XOR node that performs $U'''_1 = U''_1 + U''_5$, and assigns $U'''_1 = U'_1$, $U'''_{5,1} = U''_5$ and $U'''_{5,2} = U''_5$. Two solid circles, representing transparent nodes where the output variable(s) are identical to the input variable, are used to indicate the location of the removed XOR node. For the case when $w_{u.b.} = 2$, the DRS algorithm would split the first column of $G$ into three vectors, and the second, third, and fifth column once each. The corresponding encoding diagram is shown in Figure 3, where we color the solid circles associated with splits on the first, second, third, and fifth columns by black, green, orange, and blue, respectively.

We are ready to show the existence of a sequence of capacity-achieving codes over the BEC with GMs where the column weights are bounded by a polynomial in the blocklength, and that the block error probability under a low complexity decoder vanishes as $n$ grows large.

*Theorem 2:* Let $\beta < E(G_2) = 0.5$, $\lambda > \lambda^* = h_b(\frac{2}{3}) - \frac{1}{3} \approx 0.585$, and a BEC $W$ with capacity $C$ be given. There exists a sequence of polar-based codes corresponding to $\mathrm{DRS}(G_2^{\otimes n})$ with the following properties for all sufficiently large $n$: (1) The error probability under a SC decoder is upper bounded by $2^{-N^\beta}$, where $N = 2^n$, (2) The Hamming weight of each column of the GM is upper bounded by $N^\lambda$, (3) The rate approaches $C$ as $n$ grows large, and (4) The codes can be decoded by a SC decoding scheme with complexity $\mathcal{O}(N \log N)$.

*Proof:* Let the threshold for DRS algorithm be $w_{u.b.} = 2^{n\lambda}$, $G$ denote $G_2^{\otimes n}$, and $G'$ denote $\mathrm{DRS}(G_2^{\otimes n})$. We prove the four claims in order. First, Lemma 2 shows that for a given $n$ and any $t > 0$, the following is true:

$$\{\mathbf{s} \in \{-,+\}^n : Z_G^{\mathbf{s}} \leqslant t\} \subseteq \{\mathbf{s} \in \{-,+\}^n : Z_{G'}^{\mathbf{s}} \leqslant t\}. \quad (2)$$

Using [2, Theorem 2], for any $\beta < \frac{1}{2}$, we have

$$\liminf_{n\to\infty} \frac{1}{N} \left| \left\{ \mathbf{s} \in \{-,+\}^n : Z_G^{\mathbf{s}} \leqslant 2^{-N^\beta} \right\} \right| = I(W) = C \quad (3)$$

Let $\boldsymbol{S}_G$ and $\boldsymbol{S}_{G'}$ denote the sets of the sequences $\mathbf{s} \in \{-,+\}^n$ that satisfy $Z_G^{\mathbf{s}} \leqslant 2^{-N^\beta}$ and $Z_{G'}^{\mathbf{s}} \leqslant 2^{-N^\beta}$, respectively. Equation (2) guarantees that $\boldsymbol{S}_G$ is a subset of $\boldsymbol{S}_{G'}$. Assume the code corresponding to $G$ freezes the input bits observing bit-channels $W^{s_1 s_2 \cdots s_n}$ for all $(s_1, s_2, \ldots, s_n) \notin \boldsymbol{S}_G$. For the code corresponding to $G'$, we use the bit-channels with the same index as the code corresponding to $G$, for transmission of information bits, and leave the rest as frozen. The probability of block error under SC decoding, which is described in the last part of this proof, for the code corresponding to $G'$, $P_{e,G'}$, can be bounded above, as in [2], by the sum of the Bhattacharyya parameters of the bit-channels for the source bits (that are not frozen), that is,

$$P_{e,G'} \leqslant \sum_{\mathbf{s}\in\boldsymbol{S}_G} Z_{G'}^{\mathbf{s}} \leqslant \sum_{\mathbf{s}\in\boldsymbol{S}_G} 2^{-N^\beta} = |\boldsymbol{S}_G|\, 2^{-N^\beta},$$

where the second inequality follows because, for $\mathbf{s} \in \boldsymbol{S}_G$, we must have $\mathbf{s} \in \boldsymbol{S}_{G'}$ and thus $Z_{G'}^{\mathbf{s}} \leqslant 2^{-N^\beta}$. From (3), for all sufficiently large $n$, we have $P_{e,G'} \leqslant NC2^{-N^\beta}$. With some calculus, one may show that, for any $\beta' < \frac{1}{2}$, $P_{e,G'} \leqslant 2^{-N^{\beta'}}$ for all sufficiently large $n$.

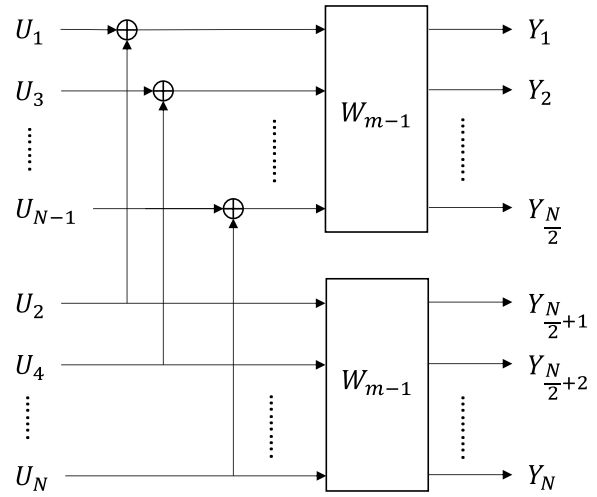
The second claim follows from the fact that the GM for the code corresponding to $G'$ is a submatrix of $G'$, and the Hamming weight of each column of $G'$ is upper bounded by $w_{u.b.} = 2^{n\lambda} = N^\lambda$.

The third claim is a consequence of Proposition 2 and Lemma 2. The number of information bits of the code corresponding to $G'$ is given by $|\boldsymbol{S}_G|$, and the length of the code is $N(1+\gamma)$. Hence the code rate is $\frac{|\boldsymbol{S}_G|}{N(1+\gamma)}$. Since the term $\gamma$ vanishes as $n$ grows large, we have

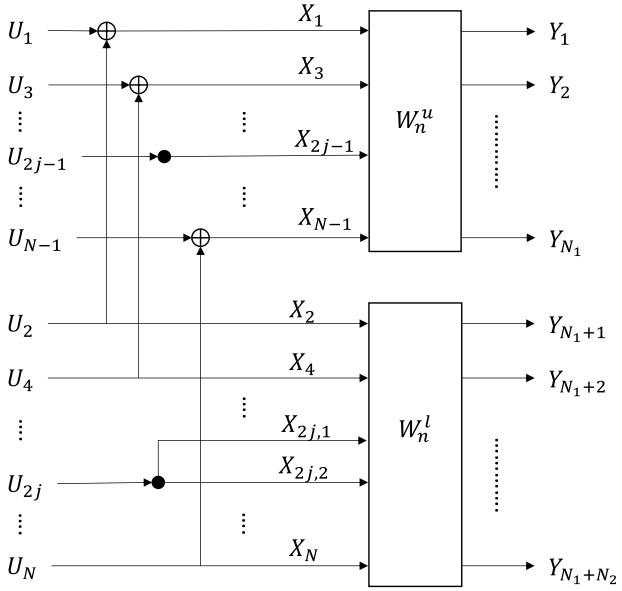
$$\liminf_{n\to\infty} \frac{|\boldsymbol{S}_G|}{N(1+\gamma)} = \liminf_{n\to\infty} \frac{|\boldsymbol{S}_G|}{N} = I(W) = C. \quad (4)$$

Finally, we prove the claim for the existence of a low-complexity decoder. Just like that of the SC decoder for conventional polar codes with kernel $G_2$, the decoding algorithm proceeds in a recursive manner. Let $U_1, \ldots, U_N$ be the inputs, and $Y_1, \ldots, Y_{N_1}, Y_{N_1+1}, \ldots, Y_{N_1+N_2}$ the outputs, where $N_1 + N_2 = N(1+\gamma)$, as shown in Figure 4b. However, while the polar code based on $G_2^{\otimes n}$, as shown in Figure 4a, is recursive in the encoder structure, i.e., the two encoding sub-blocks corresponding to $G_2^{\otimes n-1}$ are identical, the code based on $(G_2^{\otimes n})'$ is not, as the blocks $W_n^u$ and $W_n^l$ are not necessarily equal. In fact, when there is a split in the GM due to the DRS algorithm, i.e., when one or more of the XOR operations shown in Figure 4b is replaced by two solid black circle, the number of inputs of the block $W_n^l$ will be larger than that of $W_n^u$.

Let $\mathcal{F} \subseteq \{1, \ldots, N\}$ be the set of the indices of the frozen bits. The decoder declares estimates $\hat{U}_i$ of the inputs, for




(a) Encoding block for generator matrix $G_2^{\otimes n}$




(b) Encoding block for generator matrix $(G_2^{\otimes n})'$

Fig. 4. Encoding/decoding diagrams for standard and DRS-modified polar codes.

$1 \leqslant i \leqslant N$, sequentially by:

$$\hat{U}_i = \begin{cases} u_i, & \text{if } i \in \mathcal{F}, \\ \psi_i(Y_1^{N_1+N_2}, \hat{U}_1^{i-1}, W_n) & \text{if } i \notin \mathcal{F}, \end{cases} \tag{5}$$

where $\psi_i(Y_1^{N_1+N_2}, \hat{U}_1^{i-1}, W_n)$ can be found in following four cases, and $W_n$ denotes the encoding block shown in Figure 4b. Let the symbol $e$ denotes an erasure, and assume $e \oplus b = e$ for $b \in \{0, 1, e\}$. We write $\psi_i$ in place of $\psi_i(Y_1^{N_1+N_2}, \hat{U}_1^{i-1}, W_n)$ for the sake of space in the following.

- If $i$ is odd and $X_i = U_i \oplus U_{i+1}$, which corresponds to an unsplit XOR operation observed by $U_i$,

$$\psi_i \triangleq \begin{cases} \hat{X}_i \oplus \hat{X}_{i+1} & \text{if } \hat{X}_i \neq e, \hat{X}_{i+1} \neq e, \\ e, & \text{otherwise.} \end{cases}$$

- If $i$ is odd and $X_i = U_i$, which corresponds to a split XOR operation, $\psi_i \triangleq \hat{X}_i$.

- If $i$ is even and $X_{i-1} = U_i \oplus U_{i-1}$, which corresponds to an unsplit XOR operation,

$$\psi_i \triangleq \begin{cases} \hat{X}_i, & \text{if } \hat{X}_i \neq e, \hat{X}_{i-1} \neq e, \\ & \hat{X}_i = \hat{X}_{i-1} \oplus \hat{U}_{i-1} \\ & \text{or } \hat{X}_i \neq e, \hat{X}_{i-1} = e, \\ \hat{X}_{i-1} \oplus \hat{U}_{i-1}, & \text{if } \hat{X}_i = e, \hat{X}_{i-1} \neq e, \\ & \hat{U}_{i-1} \neq e \\ e, & \text{otherwise.} \end{cases}$$

- If $i$ is even and $X_{i,1} = X_{i,2} = U_i$, which corresponds to a split XOR operation,

$$\psi_i \triangleq \begin{cases} \hat{X}_{i,1}, & \text{if } \hat{X}_{i,1} \neq e, \hat{X}_{i,2} = e, \\ & \text{or } \hat{X}_{i,1} = \hat{X}_{i,2} \neq e, \\ \hat{X}_{i,2}, & \text{if } \hat{X}_{i,1} = e, \hat{X}_{i,2} \neq e, \\ e, & \text{otherwise.} \end{cases}$$

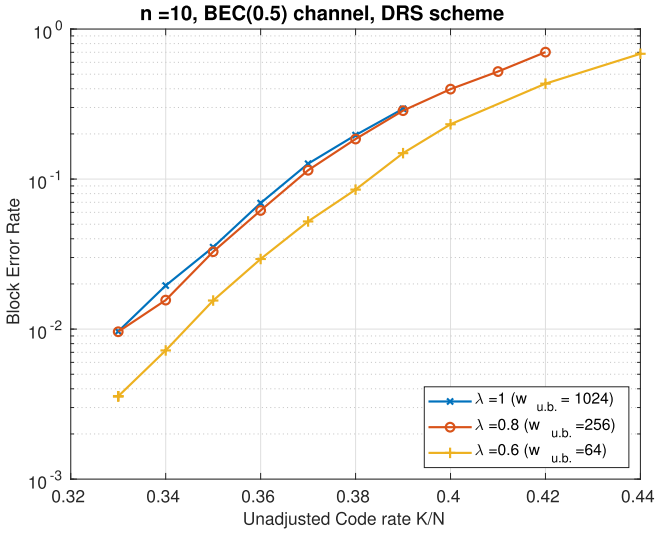
The estimates $\hat{X}_1, \hat{X}_3, \ldots, \hat{X}_{N-1}$ and $\hat{X}_2, \hat{X}_4, \ldots, \hat{X}_{2j,1}, \hat{X}_{2j,2}, \ldots, \hat{X}_N$ are found in a similar approach using the blocks $W_n^u$ and $W_n^l$ along with the outputs $Y_1, \ldots, Y_{N_1}$ and $Y_{N_1+1}, \ldots, Y_{N_1+N_2}$, respectively.

For the right-most variables, the blocks they observe are identical copies of the BEC $W$. Hence the estimates of the variables, denoted as $\hat{X}_1^{(n)}, \hat{X}_2^{(n)}, \ldots, \hat{X}_{N_1+N_2}^{(n)}$ are naturally defined by the outputs of the channels, i.e., $\hat{X}_i^{(n)} = Y_i$ for $i = 1, 2, \ldots, N_1 + N_2$.

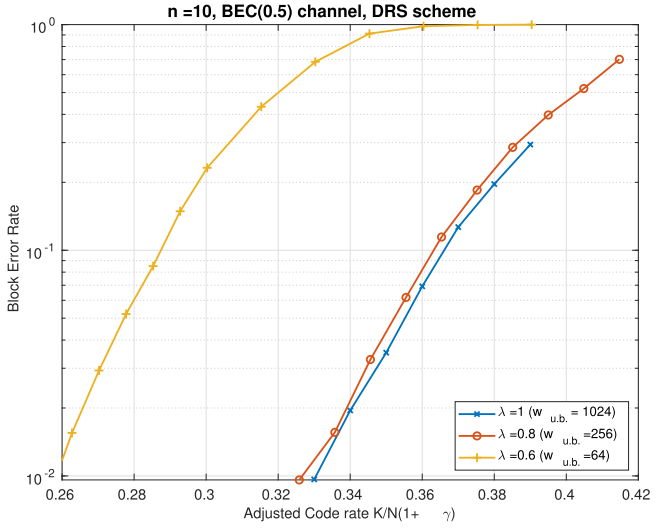
At each stage there are at most $N_1 + N_2 = N(1 + \gamma) = \mathcal{O}(N)$ estimates to make, and the recursion ends in $\log(N)$ steps. Since each estimate is obtained with constant complexity, the total decoding complexity for the code based on DRS($G_2^{\otimes n}$) is bounded by $\mathcal{O}(N \log N)$. ■

We evaluate the performance of the polar-DRS codes with $n = 10$ and $\lambda = 0.6, 0.8, 1.0$, under the SC decoding scheme described in the proof of Theorem 2, over the BEC with erasure probability $\epsilon = 0.5$. In Figure 5a, the block error probabilities for the curves with smaller $\lambda$ are smaller, due to the improvement of some of the Bhattacharyya parameters observed by the information bits. That is, there are sequences $(s_1, s_2, \ldots, s_n) \in \{-, +\}^n$ for which the inequality in Lemma 2 is strict. However, after factoring in multiplicative rate loss $\gamma$, we may observe in Figure 5b that the performance of the codes with $\lambda = 0.6$ are substantially worse than the original polar code ($\lambda = 1$ curve), and those with $\lambda = 0.8$ deliver trade-off between code rate and error probability comparable to the original polar code, while guaranteeing the threshold $w_{u.b.}$ is one-fourth of the latter.

*Remark 3:* We note that for general BMS channels, Lemma 2 may fail. One key part in the proof (see Appendix C) is the fact that the Bhattacharyya parameter for the bit-channel observed by $U_i$ is a non-decreasing function of those of $W(X_1), \ldots, W(X_{f(m)})$ for $i \leqslant 2^m$, and of $W(X_{f(m)+1}), \ldots, W(X_{2f(m)})$ for $i > 2^m$, when all the channels are BECs. We now provide an example where we see the argument for Lemma 2 fail for BMS channels. Let $a, a', b, b'$ be four distinct elements and $\mathcal{Y} = \{a, a', b, b'\}$. Let two BMS channels $W_1, W_2 : \{0, 1\} \to \mathcal{Y}$ be given, and that $W_1(y|0) = W_1(\phi(y)|1)$ and $W_2(y|0) = W_2(\phi(y)|1)$ for all $y \in \mathcal{Y}$ where




(a) Unadjusted code rate

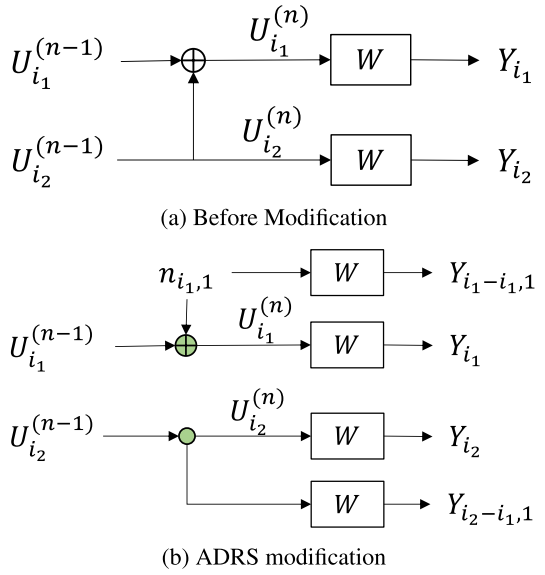



(b) Adjusted code rate

Fig. 5. Error probability for polar-DRS codes with $n = 10$ with $w_{u.b.} \in \{64, 256, 1024\}$.

the involution $\phi$ maps $a \mapsto a', b \mapsto b'$. Assume the channel transition probabilities are $W_1(a|0) = 6/9, W_1(b|0) = 1/9$, $W_1(b'|0) = 1/9, W_1(a'|0) = 1/9$ and $W_2(a|0) = 5/11$, $W_2(b|0) = 4/11$, $W_2(b'|0) = 1/11$, $W_2(a'|0) = 1/11$. The Bhattacharyya parameters for $W_1, W_2$ are respectively 0.7666 and 0.7702. If $m = 1$ and $B_m$ is simply the kernel $G_2$, the symbols $X_1, X_2$ are functions of $U_1, U_2$ given by $X_1 = U_1 + U_2$ and $X_2 = U_2$.

We now consider two possible cases for the pair $(W(X_1), W(X_2))$. If $(W(X_1), W(X_2)) = (W_1, W_2)$, the Bhattacharyya parameters for the bit-channels observed by $U_1, U_2$ are respectively 0.9147 and 0.5904. If $(W(X_1), W(X_2)) = (W_2, W_2)$, the Bhattacharyya parameters for the bit-channels observed by $U_1, U_2$ are respectively 0.9137 and 0.5932. We note that while the Bhattacharyya parameters for $W(X_1)$, $W(X_2)$ in the second case are no less than in the first case, the Bhattacharyya parameter $Z(U_1)$ in the second case is smaller than in the first case. With the above observation, one can not claim the validity of Theorem 2 for general BMS channels. This motivates a new code construction for general BMS channels.




(a) Before Modification

(b) ADRS modification

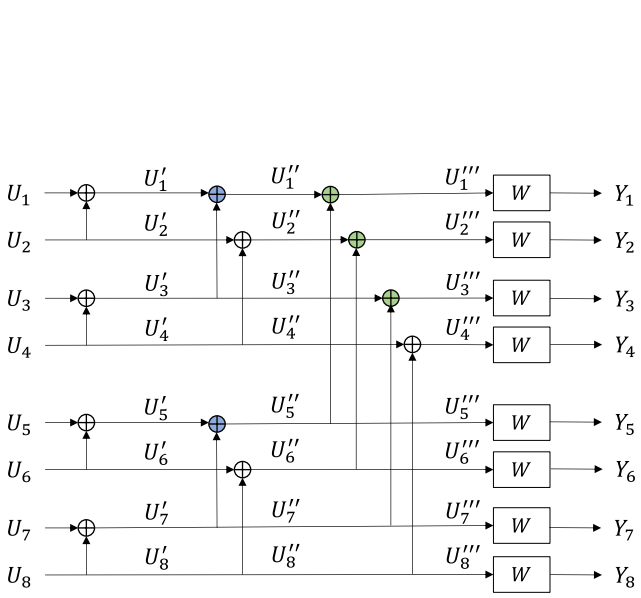
Fig. 6. ADRS scheme for a split XOR of first iteration of polarization.

### C. Low-Complexity Decoder for Polar-Based Codes: BMS

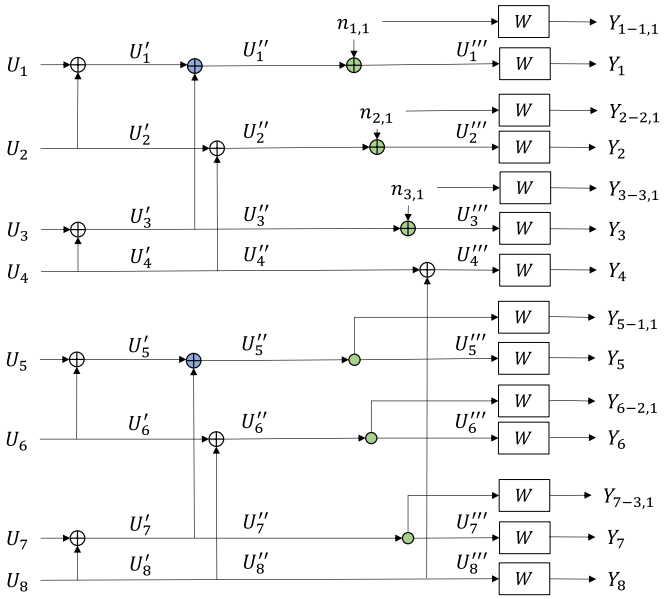
This section introduces a capacity-achieving polar-based coding scheme with low-complexity decoder for general BMS channels. For general BMS channels, the Bhattacharyya parameter of the bit-channel $W^-$ cannot be expressed only in terms of parameters of the channel $W$. This implies that Lemma 2 and Theorem 2 are not applicable for channels other than BEC, as pointed out in Remark 3. A procedure that augments the generator matrix corresponding to $G'$, the output of the DRS algorithm for the matrix $G_2^{\otimes n}$, may be used to construct a capacity-achieving linear code over any BMS channel $W$.

*1) ADRS Scheme:* The encoding scheme, termed augmented-DRS (ADRS) scheme, avoids heavy columns in the GM and, at the same time, guarantees that the bit-channels observed by the source bits $U_i$ have the same statistical characteristics as when they are encoded with the generator matrix $G_2^{\otimes n}$. Specifically, the ADRS scheme modifies the encoder for $G_2^{\otimes n}$ starting from the split XOR operations associated with the first polarization recursion, then the second recursion, and proceed all the way to the $n$-th recursion, where a XOR operation is split if and only if it is split in an encoder with generator matrix DRS($G_2^{\otimes n}$).

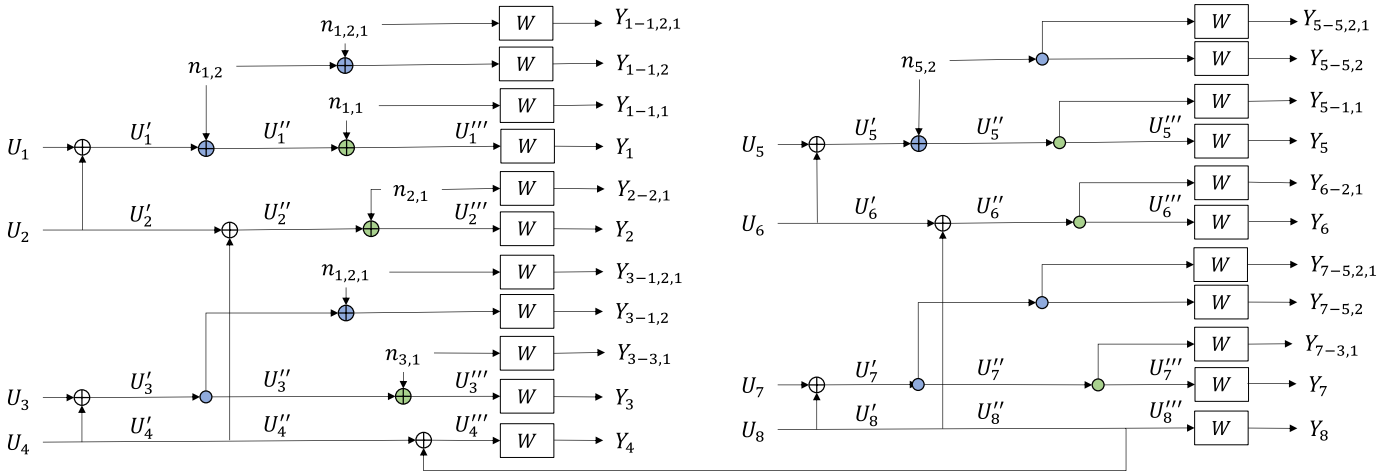
Assume an XOR operation with operands $U_{i_1}^{(n-j)}$ and $U_{i_2}^{(n-j)}$ and the output $U_{i_1}^{(n-j+1)}$, where $i_1 = \mathsf{Bi2De}(s_1, \ldots, s_{j-1}, s_j = -, s_{j+1}, \ldots, s_n) + 1$ and $i_2 = \mathsf{Bi2De}(s_1, \ldots, s_{j-1}, s_j = +, s_{j+1}, \ldots, s_n) + 1 = i_1 + 2^{n-j}$, is to be split (see Section IV-B for the function $\mathsf{Bi2De}(\cdot)$). If $j = 1$, before modification, the variables $U_{i_1}^{(n)}$ and $U_{i_2}^{(n)}$ are transmitted through two copies of $W$, and the bit-channels observed by $U_{i_1}^{(n-1)}$ and $U_{i_2}^{(n-1)}$ are $W^-$ and $W^+$, respectively, as shown



(a) Encoding diagram for $G_2^{\otimes 3}$



(b) ADRS for splits corresponding to $s_1$ in $G_2^{\otimes 3}$



(c) ADRS Encoding Diagram for $G_2^{\otimes 3}$ with $w_{u.b.} = 2$

Fig. 7. ADRS example with $N = 8$ and $w_{u.b.} = 2$.

in Figure 6a. If the XOR operation is split according to DRS($G_2^{\otimes n}$), ADRS scheme replaces the structure by that given in Figure 6b, where $n_{i_1,1}$ is a Bernoulli(0.5) random variable independent of all the other variables.

If $j \geqslant 2$, assume that the ADRS modification for the split operations for the first $(j-1)$ recursions are completed. Let $n_{i_1,j}$ be a Bernoulli(0.5) random variable independent of all the other given variables. The part of encoding diagram to the right of $U_{i_1}^{(n-j+1)}$ is replicated, where $n_{i_1,j}$ takes the place of $U_{i_1}^{(n-j+1)}$ in the replica. And then we let $U_{i_1}^{(n-j+1)} = U_{i_1}^{(n-j)} \oplus n_{i_1,j}$. In addition, the part of encoding diagram to the right of $U_{i_2}^{(n-j+1)}$ is replicated, and a copy of $U_{i_2}^{(n-j)}$ is transmitted through the replica. The variable $U_{i_2}^{(n-j+1)}$ remains $U_{i_2}^{(n-j+1)} = U_{i_2}^{(n-j)}$.

We demonstrate the procedure described above through the following example. Assume $n = 3$, $N = 8$, and $w_{u.b.} = 2$. The encoding diagram for $G_2^{\otimes 3}$ is shown in Figure 7a, and the XOR operations that are split in DRS($G_2^{\otimes 3}$) are marked in green and blue, which indicate the operations are due to the first and the second polarization recursions (i.e., $s_1$ and $s_2$), respectively. The notations $U_i', U_i'', U_i'''$ are used to represent $U_i^{(1)}, U_i^{(2)}, U_i^{(3)}$. Replacing the XOR operations marked in green as described for the case of $j = 1$, the encoding diagram is now shown in Figure 7b. For the XOR operations marked in blue, we proceed by using the step for $j \geqslant 2$ and obtain the diagram shown in Figure 7b.

It can be noted that the bit-channels observed by each of $U_i^{(j)}$, for $i = 1, 2, \ldots, N$ and $j = 0, 1, 2, \ldots, n$, in the ADRS encoder are the same as those in the standard encoder for the generator matrix $G_2^{\otimes n}$ (The variable $U_i^{(0)}$ are given by $U_i$ for $1 \leqslant i \leqslant N$). When an XOR operation associated with the $j$-th recursion, with operands $U_{i_1}^{(n-j)}$ and $U_{i_2}^{(n-j)}$ and the output $U_{i_1}^{(n-j+1)}$, is split and modified under the ADRS scheme, the complexity of computing the likelihood or log-likelihood for

$U_{i_1}^{(n-j)}$ and $U_{i_2}^{(n-j)}$ can be upper bounded by $2(2^1+2^2+\ldots+2^j)c = 2(2^{j+1}-2)c$, for some constant $c>0$.

*2) Polar-ADRS Code Performance:* We are now ready to show the performance of the polar-based code whose encoding structure is given by the ADRS scheme, referred to as the *polar-ADRS* code. First we show the existence of a low-complexity decoder.

*Proposition 3:* Let a constant $\lambda > \lambda^\dagger \triangleq (\log_2 3)^{-1} \approx 0.631$ be given. The decoding complexity for a SC decoder for the polar-ADRS code is bounded by $\mathcal{O}(N\log N)$ for all sufficiently large $n$ if the threshold for the DRS algorithm is $w_{u.b.} = 2^{n\lambda}$.

*Proof:* The proof is provided in Appendix D.1. ■

Second, it can be observed that the number of additional copies of channels due to the modification for an XOR operation at the $j$-th polarization recursion is $2^j$. We find the total number of extra channel uses and the ratio $\gamma$ of that to the number $N=2^n$ of channel uses for the code corresponding to $G_2^{\otimes n}$ in the following. Assume that the column weight threshold of the DRS algorithm is given by $w_{u.b.} = 2^{n\lambda}$.

*Proposition 4:* Let $N(1+\gamma)$ be the number of channel uses of the encoder for the ADRS scheme based on DRS$(G_2^{\otimes n})$ with $w_{u.b.} = 2^{n\lambda}$. Then the term $\gamma$ goes to 0 as $n$ grows large, if we have $\lambda > \lambda^\dagger$.

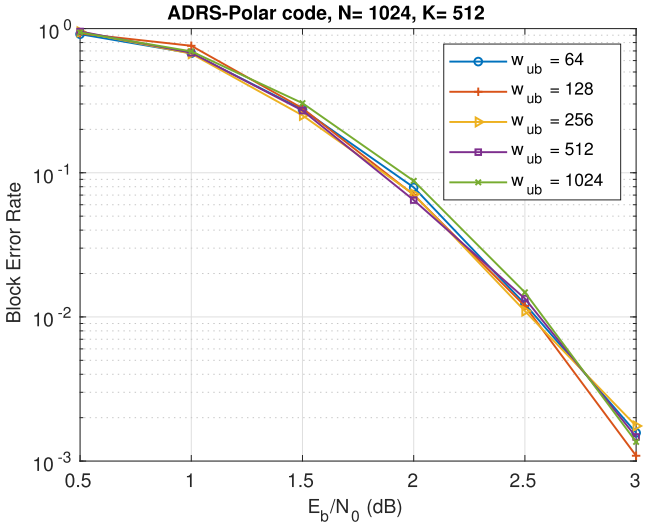
*Proof:* The proof is provided in Appendix D.2. ■

We are ready to show the existence of a sequence of capacity-achieving codes over general BMS channels with GMs where the column weights are bounded by a polynomial in the blocklength, and that the block error probability under a low complexity decoder vanishes as $n$ grows large. Note that while this result is also applicable when the underlying channel is a BEC, the constraint on $\lambda$ is stricter than that in Theorem 2, due to the difference in the encoding and decoding schemes.

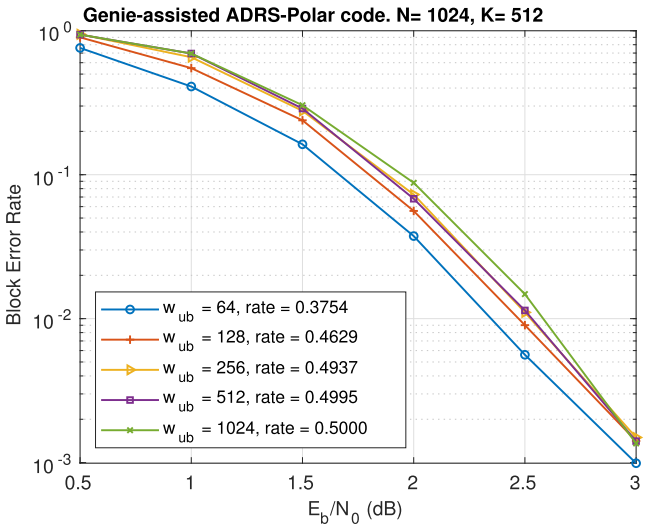
*Theorem 3:* Let $\beta < E(G_2) = 0.5$, $\lambda > \lambda^\dagger$, and a BMS channel $W$ with capacity $C$ be given. There exists a sequence of codes with the following properties for all sufficiently large $n$: (1) The error probability under SC decoding is upper bounded by $2^{-N^\beta}$, where $N=2^n$, (2) The Hamming weight of each column of the GM is upper bounded by $N^\lambda$, (3) The rate approaches $C$ as $n$ grows large, and (4) The codes can be decoded by a SC decoding scheme with complexity $\mathcal{O}(N\log N)$.

*Proof:* We prove the four properties in order as follows. First, similar to the proof of Theorem 2, for $i=1,2,\ldots,N$, the bit $U_i$ is frozen in the polar-ADRS code with rate $R<C$ if and only if it is frozen in the polar code with kernel $G_2$, blocklength $N=2^n$, and the rate $R$. Hence, the probability of error of the polar-ADRS code can be bounded in the same way as its polar-code counterpart, since the bit-channels observed by the source bits $U_i$, and the corresponding Bhattacharyya parameters, are identical to those when they are encoded with the standard polar code.

Second, when the ADRS scheme is based on DRS$(G_2^{\otimes n})$ with $w_{u.b.} = 2^{n\lambda}$, the generator matrix for the polar-ADRS code is a submatrix of DRS$(G_2^{\otimes n})$. The column weights of the GM for the polar-ADRS code are thus upper bounded by $w_{u.b.} = 2^{n\lambda} = N^\lambda$. The third claim holds by using an argument similar to the one used in the proof of Theorem 2. This is because the term $\gamma$ vanishes as $n$ grows large according to Proposition 4. Finally, note that the fourth claim is equivalent to Proposition 3. ■




(a) Random auxiliary noise nodes




(b) Frozen auxiliary noise nodes

Fig. 8. Error probability for polar-ADRS codes with $N=1024$, $K=512$ and $w_{u.b.} \in \{64,128,256,512,1024\}$.

We evaluate the performance of the polar-ADRS codes with $N=1024, K=512$ and weight thresholds $w_{u.b.} \in \{64,128,256,512,1024\}$, under the SC decoding scheme, over the BI-AWGN channels whose SNR ($E_b/N_0$) ranges from 0.5 to 3 dB. The indices of frozen bits are designed using a Monte-Carlo scheme for the BI-AWGN channel with $E_b/N_0 = 2$ dB. The auxiliary noise variables are generated with independent Bernoulli$(0.5)$ distribution. The stopping criterion for each data point is ITERATION $= 10^5$ or BLOCK ERROR COUNT $= 100$, whichever is reached first. In Figure 8a, the block error probabilities are very close for codes with different weight thresholds over the range of simulated SNR. This is because the bit-channels observed by each information bit are exactly the same with or without the

ADRS scheme, and that the block error probability of the codes are determined by the characteristics of the information bit channels.

Note that the use of the auxiliary noise variables are of theoretical reasons and is required for the proof of Theorem 3. In practice, we may freeze the auxiliary noise variables to avoid the channel uses for the transmission of them. In particular, when the auxiliary noise variables are frozen to fixed values known to both the encoder and decoder, the decoder can be considered as a genie-assisted decoder and its performance is better or equivalent to that of a decoder without the information of the auxiliary noise variables. We simulate the performance of the ADRS-polar code with $N = 1024, K = 512$ and weight thresholds $w_{u.b.} \in \{64, 128, 256, 512, 1024\}$, where the values of the auxiliary noise variables are fixed to 0's, over the BI-AWGN channels with $0.5 \leqslant E_b/N_0 \leqslant 3$ (dB), and obtain Figure 8b. The error rates of codes with small thresholds are lower than those with large or no thresholds, a trend which is similar to that observed in Figure 5a.

## V. Conclusion

This paper provided three constructions for capacity-achieving linear codes, based on polar coding, where all the GM column weights are upper bounded sublinearly in the block length. The first construction is a sequence of polar codes based on general polarization kernels where the GM column weights are upper bounded by $N^s$ for any fixed $s > 0$, and allows the codes to be decoded by a SC decoder. In order to attain a better trade-off between the GM sparsity and the fall in error probability, we then proposed a column-splitting algorithm for the GM, termed the DRS algorithm. With the DRS algorithm, we designed two encoding schemes which yield two polar-based codes, referred to as polar-DRS codes and polar-ADRS codes, that are decodable with low-complexity decoders for the BECs and general BMS channels, respectively. The polar-based codes preserve several fundamental properties of the standard polar code with $G_2$ kernel including the asymptotic error rate upper bound and decoding complexity. Further, the GM column weights of the polar-DRS and polar-ADRS codes are bounded from above by $N^\lambda$, for $\lambda \approx 0.585$ and $\lambda \approx 0.631$, respectively, while the best bound for the standard polar codes scales linearly in $N$. The proposed constructions are also distinct from known constructions for codes with constraints on the GM sparsity by having analytical error probability upper bounds scaling as $\mathcal{O}(2^{-N^t})$ under SC decoders, for any $t < 0.5$. A future direction is to design splitting algorithm and/or encoding schemes that preserve key properties of the polar codes based on general polarization kernels, and show that the corresponding polar-based codes exhibit better sparsity versus error rate trade-off.

## Appendix

### A. Proof for Section III

*Proof of Proposition 1:* Since $G_l$ is a polarization kernel, there is at least one column in $G_l$ with weight at least 2. To see this, note that $G_l$ being invertible implies that all rows and columns are nonzero vectors. If all the columns of $G_l$ have weights equal to 1, then all the rows must also have weights equal to 1, i.e., $G_l$ is a permutation matrix. Then $D_i = 1, \forall i$, and $E(G_l) = 0$, which implies that $G_l$ can not be polarization kernel. The contradiction shows that at least one column in $G_l$ have a weight at least 2.

Let $k \geqslant 1$ denote the number of columns in $G_l$ with a weight at least 2. Let $\mathbf{v}$ be a randomly uniformly chosen column of $G_l^{\otimes n}$, and $w(\mathbf{v})$ be the Hamming weight of $\mathbf{v}$. For $\frac{1}{l} > r > 0$,

$$\begin{aligned}&\Pr\left(w(\mathbf{v}) = \mathcal{O}(N^{r\log_l 2})\right)\\ &\quad\leqslant \Pr\left(2^{\sum_{i=1}^n F_i} = \mathcal{O}(N^{r\log_l 2}) = \mathcal{O}(2^{nr})\right),\end{aligned}$$

where $F_i$ is the indicator variable that one of $k$ non-unit-weight columns is used in the $i$-th Kronecker product of $G_l$ to form $\mathbf{v}$. The variables $F_1, F_2, \ldots, F_n$ are i.i.d. as $\text{Ber}(k/l)$. The law of large numbers implies that $\sum_{i=1}^n F_i \geqslant \frac{kn}{l} > nr$ with high probability. Thus, $\Pr\left(2^{\sum_{i=1}^n F_i} = \mathcal{O}(2^{nr})\right) \to 0$ as $n \to \infty$ for any $r > 0$. ∎

### B. Proofs for Subsection IV-A

*1) Proof of Lemma 1:* In order to understand the DRS algorithm's effect on $G = G_2^{\otimes n}$, we first study how the order of two special Kronecker product operations affects the number of output vectors. We present the following Lemmas 3 and 4 toward the proof of Lemma 1.

*Lemma 3:* Let a column vector $v$ and a column weight threshold $w_{u.b.}$ be given. Then the outputs of the DRS algorithm for $(v \otimes [1,1]^T) \otimes [0,1]^T$ and $(v \otimes [0,1]^T) \otimes [1,1]^T$ contain the same number of vectors.

*Proof:* The input vectors can be denoted by:

$$\begin{aligned}(v \otimes [1,1]^T) \otimes [0,1]^T &\equiv v_{LR}\\ (v \otimes [0,1]^T) \otimes [1,1]^T &\equiv v_{RL}.\end{aligned}$$

We note that $2\, w_H(v) = w_H(v_{LR}) = w_H(v_{RL})$, and prove the lemma in two cases:

1) $2w_H(v) \leqslant w_{u.b.}$: In this case, the algorithm will not split either $v_{LR}$ or $v_{RL}$. Both outputs contain exactly one vector.
2) $2w_H(v) > w_{u.b.}$: Let $n_{DRS}(v)$ denote the size, or more precisely, the number of column vectors the DRS algorithm returns when it is applied to $v$.
   For $v_{LR}$, the DRS algorithm observes $\mathbf{x}_h = \mathbf{0}$, hence the number of output vectors is the same as the size of DRS-Split$(w_{u.b.}, (v^T, v^T)^T)$ (see Section IV-A). With $2w_H(v) > w_{u.b.}$, the size of DRS-Split$((v^T, v^T)^T)$ is the sum of the sizes of $Y_h =$ DRS-Split$(w_{u.b.}, \mathbf{x}_h = v)$ and $Y_t =$ DRS-Split$(w_{u.b.}, \mathbf{x}_t = v)$. By assumption, $|Y_h| = |Y_t| = n_{DRS}(v)$, giving $n_{DRS}(v_{LR}) = 2\, n_{DRS}(v)$.
   For $v_{RL}$, the number of vectors in the DRS algorithm output is the sum of the sizes of two sets $Y_h =$ DRS-Split$(w_{u.b.}, \mathbf{x}_h = (\mathbf{0}^T, v^T)^T)$ and $Y_t =$ DRS-Split$(w_{u.b.}, \mathbf{x}_t = (\mathbf{0}^T, v^T)^T)$. It is easy to see that $|Y_h| = |Y_t| = |$DRS-Split$(w_{u.b.}, v)|$, which equals $n_{DRS}(v)$. Thus, $n_{DRS}(v_{RL}) = 2\, n_{DRS}(v)$.

∎

The next lemma shows the effect of the DRS algorithm from a different perspective. If there are two vectors with the same column weights, and numbers of vectors of the DRS algorithm outputs are identical when they are the inputs, the properties will be preserved when they undergo some basic Kronecker product operations.

*Lemma 4:* Let $u_1$ and $u_2$ be two vectors with equal Hamming weights. Assume, for a given $w_{u.b.}$, the DRS algorithm splits $u_1$ and $u_2$ into the same number of vectors. Then the DRS algorithm also returns the same number of vectors for $u_1 \otimes [1,1]^T$ and $u_2 \otimes [1,1]^T$, as well as for $u_1 \otimes [0,1]^T$ and $u_2 \otimes [0,1]^T$.

*Proof:* We first discuss the case when $u_1 \otimes [1,1]^T$ and $u_2 \otimes [1,1]^T$ are processed by the DRS algorithm. If $2w_H(u_1) = 2w_H(u_2) \leqslant w_{u.b.}$, no splitting is done. If $2w_H(u_1) = 2w_H(u_2) > w_{u.b.}$, the size of the DRS algorithm output for the input $u_1 \otimes [1,1]^T$ is the sum of the sizes of $Y_h = \text{DRS-SPLIT}(w_{u.b.}, \mathbf{x}_h = u_1)$ and $Y_t = \text{DRS-SPLIT}(w_{u.b.}, \mathbf{x}_t = u_1)$, both of which are $n_{DRS}(u_1)$. The size of the output for the input $u_2 \otimes [1,1]^T$ can be found in a similar way to be $2n_{DRS}(u_2)$. Note that $n_{DRS}(u_1) = n_{DRS}(u_2)$ by assumption. Therefore, the sizes of the outputs of the DRS algorithm, when $u_1 \otimes [1,1]^T$ and $u_2 \otimes [1,1]^T$ are the inputs, are equal. Similarly, one can easily show that when $u_1 \otimes [0,1]^T$ and $u_2 \otimes [0,1]^T$ are processed by the DRS algorithm, the number of output columns are equal. ∎

*Proof of Lemma 1:* Suppose that there is an index $i$ such that $(s_i, s_{i+1}) = (+,-)$. Let $v^{(i+1)}$ and $(v^{(i+1)})'$ be defined by (1) with sequences $(s_1, \ldots, s_{i-1}, s_i = +, s_{i+1} = -)$ and $(s_1, \ldots, s_{i-1}, s'_i = -, s'_{i+1} = +)$, respectively. We note that

$$v^{(i+1)} = \left(v^{(i-1)} \otimes [0,1]^T\right) \otimes [1,1]^T$$
$$(v^{(i+1)})' = \left(v^{(i-1)} \otimes [1,1]^T\right) \otimes [0,1]^T.$$

Lemma 3 shows that the DRS algorithm splits $v^{(i+1)}$ and $(v^{(i+1)})'$ into the same number of columns. Furthermore, Lemma 4 shows that the number of output vectors of the DRS algorithm for $v^{(n)} = [v^{(i+1)}]^{(s_{i+2},\ldots,s_n)}$ and $(v^{(n)})' = [(v^{(i+1)})']^{(s_{i+2},\ldots,s_n)}$ are equal.

Therefore, an occurrence of $(s_i, s_{i+1}) = (+,-)$ in a sequence can be replaced by $(s_i, s_{i+1}) = (-,+)$ without changing the number of output vectors of the DRS algorithm. Since any sequence $(s_1, s_2, \ldots, s_n)$ with $n_1$ minus signs and $n_2$ plus signs can be permuted into $(s'_1, s'_2, \ldots s'_n)$, where $s'_i = -$ for $i \leqslant n_1$ and $s'_i = +$ for $i > n_1$, by repeatedly replacing any occurrence of $(+,-)$ by $(-,+)$, the above arguments show $n_{DRS}(v^{(n)}) = n_{DRS}(v^{(s'_1,s'_2,\ldots s'_n)})$ always holds. Hence, the size of DRS algorithm output for $v^{(n)}$ depends only on the values $n_1$ and $n_2$. ∎

*2) Proof for Proposition 2:* First note that there is a bijection between $\{-,+\}^n$ and the columns of $G_2^{\otimes n}$ as follows. For each $\mathbf{s} = (s_1, \ldots, s_n) \in \{-,+\}^n$, there is exactly one column of $G_2^{\otimes n}$ in the form $1^{(\mathbf{s})}$ (see equation (1) and the paragraph following it, where we use $v = v^{(0)} = [1] \in \mathbb{F}_2$ to be the length-1 vector). The term $\gamma$ can be characterized as follows:

$$\gamma = \left[\frac{1}{N} \sum_{\mathbf{s} \in \{-,+\}^n} n_{DRS}(1^{(\mathbf{s})})\right] - 1.$$

By Lemma 1, the terms in the summation can be grouped according to the number of minus and plus signs in the sequence. Hence,

$$\gamma = \left[\frac{1}{N} \sum_{i=0}^{n} \binom{n}{i} n_{DRS}(1^{(s_1=-,\ldots,s_i=-,s_{i+1}=+,\ldots,s_n=+)})\right] - 1.$$

Let $u_i$ denote the vector $1^{(s_1,\ldots,s_n)}$ with $s_l = -$ for $l \leqslant i$ and $s_l = +$ for $l > i$. Without loss of generality, let $n\lambda \in \mathbb{N}$. For $i \leqslant n\lambda$, the Hamming weight of $u_i$ is $2^i \leqslant 2^{n\lambda} = w_{u.b.}$. Hence, $n_{DRS}(u_i) = 1$. For $i > n\lambda$, $u_i$ is split into $2^{i-n\lambda}$ vectors, each of which having weight equal to $2^{n\lambda}$. Therefore, $n_{DRS}(u_i) = 2^{i-n\lambda}$.

The term $\gamma$ can be written as follows:

$$\gamma = \sum_{i=0}^{n\lambda} \frac{1}{N}\binom{n}{i} + \sum_{i=n\lambda+1}^{n} \frac{1}{N}\binom{n}{i} 2^{i-n\lambda} - 1 = \sum_{i=n\lambda+1}^{n} a_i, \tag{6}$$

where $a_i \triangleq \frac{1}{N}\binom{n}{i}(2^{i-n\lambda}-1)$. Now, let $\alpha = i/n$. Since $i > n\lambda$ for each summand $a_i$, we consider $\alpha > \lambda > \frac{1}{2}$ in the following calculations. The term $a_i$ can be written as

$$\begin{aligned} a_i = a_{n\alpha} &= 2^{-n}\binom{n}{n\alpha} 2^{n\alpha - n\lambda + o(1)} \\ &= 2^{-n} \cdot 2^{nh_b(\alpha)+o(1)} \cdot 2^{n\alpha-n\lambda+o(1)} \\ &= 2^{n\cdot f(\alpha,\lambda)+o(1)}, \end{aligned} \tag{7}$$

where the third equality is due to an asymptotic approximation of the binomial coefficient, and $f(\alpha,\lambda) \triangleq h_b(\alpha) + \alpha - \lambda - 1$.

Consider $f(\alpha,\lambda)$ as a function of $\alpha$ over the interval $[0,1]$. We find its first and second derivatives with respect to $\alpha$ as follows:

$$\frac{\partial f(\alpha,\lambda)}{\partial \alpha} = 1 - \log\frac{\alpha}{1-\alpha}, \tag{8}$$

$$\frac{\partial^2 f(\alpha,\lambda)}{\partial^2 \alpha} = -\frac{1}{\ln 2}\left(\frac{1}{\alpha} + \frac{1}{1-\alpha}\right) < 0, \text{ for } 0 < \alpha < 1. \tag{9}$$

Thus, for any fixed $\lambda$, $f(\alpha,\lambda)$ is a concave function of $\alpha$ and has local maximum when $\partial f(\alpha,\lambda)/\partial\alpha = 0$. From (8), the equality holds if and only if $\alpha = \frac{2}{3}$, and the maximum is

$$\sup_{\alpha\in(0,1)} f(\alpha,\lambda) = h_b\left(\frac{2}{3}\right) + \frac{2}{3} - \lambda - 1 = \lambda^* - \lambda. \tag{10}$$

Also, when $\alpha = 0$, $f(0,\lambda) = -\lambda - 1$, and when $\alpha = 1$, $f(1,\lambda) = -\lambda$. Hence, $\sup_{\alpha\in[0,1]} f(\alpha,\lambda) = \lambda^* - \lambda$. When $\lambda > \lambda^*$, we know $f(\frac{i}{n},\lambda) \leqslant \sup_\alpha f(\alpha,\lambda) < 0$ for all integers $0 \leqslant i \leqslant n$, and (7) implies that $a_i \to 0$ exponentially fast for each $i$. Equation (6) then shows that $\gamma$ vanishes exponentially fast in $n$. ∎

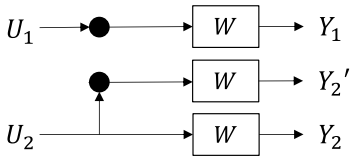



Fig. 9. Encoding Block for $G_2'$.

### C. Proof for Subsection IV-B

*Proof for Lemma 2:* We show the claim by proving the following: when we encode the source bits according to $G'$, the bit-channels observed by the source bits are BECs and that the erasure probabilities are less than or equal to those when $G$ is used. We use proof by induction on $n$. For ease of notation, we use $M'$ to denote $\mathrm{DRS}(M)$ for a given matrix $M$ in this proof.

For $n = 1$, if $G_2 = G_2'$, we naturally have $Z_{G_2'}^{(s_1)} = Z_{G_2}^{(s_1)}$ for $s_1 \in \{-, +\}$. If $G_2 \neq G_2'$, the latter must be $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$, corresponding to the encoding block diagram in Figure 9, where the solid black circles indicate a *split* of the XOR operation, i.e., the two operands of the original XOR operation are transmitted through two copies of channel $W$. The bit-channels observed by $U_1$ and $U_2$, denoted by $W^{\boxminus}$ and $W^{\boxplus}$, are BECs with erasure probability $\epsilon$ and $\epsilon^2$, respectively. Note that the Bhattacharyya parameters satisfy the following:

$$Z_{G_2'}^{(-)} = Z(W^{\boxminus}) = \epsilon \leqslant Z_{G_2}^{(-)} = Z(W^{-}) = 2\epsilon - \epsilon^2,$$
$$Z_{G_2'}^{(+)} = Z(W^{\boxplus}) = \epsilon^2 = Z_{G_2}^{(+)} = Z(W^{+}).$$

Suppose that, for a fixed $w_{u.b.}$, the claim holds for all $n \leqslant m$ for some integer $m \geqslant 1$. Let $B_m$ denote the encoding block corresponding to the generator matrix $(G_2^{\otimes m})'$, with inputs $U_1, \ldots, U_{2^m}$ and encoded bits $X_1, \ldots, X_{f(m)}$, where $f(m)$ is the number of columns in $(G_2^{\otimes m})'$. Using the matrix notation, the relation between the input bits and encoded bits is:

$$(U_1, \ldots, U_{2^m})(G_2^{\otimes m})' = (X_1, \ldots, X_{f(m)}).$$

For $n = m + 1$, the matrix $(G_2^{\otimes m+1})'$ is associated with $(G_2^{\otimes m})'$ as follows:

$$(G_2^{\otimes m+1})' = \mathrm{DRS}\left(\begin{bmatrix} (G_2^{\otimes m})' & \mathbf{0} \\ (G_2^{\otimes m})' & (G_2^{\otimes m})' \end{bmatrix}\right). \tag{11}$$

Since $(G_2^{\otimes m})'$ consists of the outputs of the DRS algorithm, the columns in the right half of the input matrix in equation (11) remain unaltered in the output. For the columns in the left half, they are of the form $[v^T, v^T]^T$ for some column $v$ of $(G_2^{\otimes m})'$. If $2w_H(v) > w_{u.b.}$, the outputs of the DRS algorithm are $[\mathbf{0}^T, v^T]^T$ and $[v^T, \mathbf{0}^T]^T$ because the vector $v$ must have weight no larger than the threshold. If $2w_H(v) \leqslant w_{u.b.}$, the algorithm leaves the vector unchanged. We may represent the encoding block $B_{m+1}$ as in Figure 10, where it is assumed that the $j$-th column of the input matrix in (11) is halved by the DRS algorithm.

The erasure probabilities for the bit-channels observed by $X_i$, denoted here as $W(X_i)$, are less than or equal to $2\epsilon - \epsilon^2$ for $1 \leqslant i \leqslant f(m)$, and are equal to $\epsilon^2$ for $f(m) + 1 \leqslant i \leqslant 2\ f(m)$, respectively. Hence we may replace the XOR operations to the right of the $X_i$'s as well as the transmission over $W$'s by BECs $W(X_1), \ldots, W(X_{f(m)})$, $W(X_{f(m)+1}), \ldots, W(X_{2f(m)})$, as in Figure 11.

One may observe that the erasure probability for the bit-channel observed by $U_i$ is a non-decreasing function of those of $W(X_1), \ldots, W(X_{f(m)})$ for $i \leqslant 2^m$, and of $W(X_{f(m)+1}), \ldots, W(X_{2f(m)})$ for $i > 2^m$. Thus, for $i \leqslant 2^m$, we have

$$\begin{aligned} &Z_{(G_2^{\otimes m+1})'}(U_i \mid Z(W) = \epsilon) \\ &\leqslant Z\left[W(U_i) \mid Z(W(X_j)) = 2\epsilon - \epsilon^2, \text{ for } 1 \leqslant j \leqslant f(m)\right] \\ &= Z_{(G_2^{\otimes m})'}\left(U_i \mid Z(W) = 2\epsilon - \epsilon^2\right) \\ &\leqslant Z_{G_2^{\otimes m}}\left(U_i \mid Z(W) = 2\epsilon - \epsilon^2\right) \\ &= Z_{G_2^{\otimes m+1}}\left(U_i \mid Z(W) = \epsilon\right), \end{aligned}$$

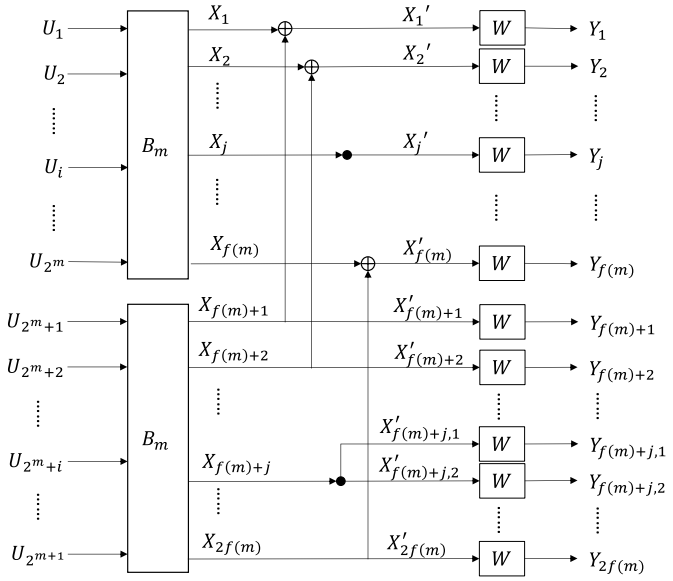



Fig. 10. Encoding Block $B_{m+1}$.

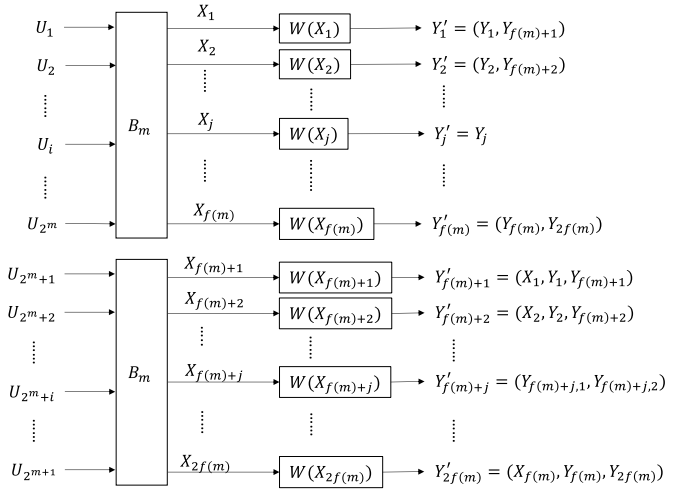



Fig. 11. Equivalent Encoding Block $B_{m+1}$.

where the first inequality is due to $Z(W(X_j)) \leqslant 2\epsilon - \epsilon^2$ for $1 \leqslant j \leqslant f(m)$ and the second inequality follows from the hypothesis of the induction.

Similarly, for $i > 2^m$, we have

$$\begin{aligned}
&Z_{(G_2^{\otimes m+1})'}\left(U_i \mid Z(W) = \epsilon\right) \\
&= Z\left[W(U_i) \mid Z(W(X_j)) = \epsilon^2, \text{ for } f(m)+1 \leqslant j \leqslant 2f(m)\right] \\
&= Z_{(G_2^{\otimes m})'}\left(U_{i-f(m)} \mid Z(W) = \epsilon^2\right) \\
&\leqslant Z_{G_2^{\otimes m}}\left(U_{i-f(m)} \mid Z(W) = \epsilon^2\right) \\
&= Z_{G_2^{\otimes m+1}}\left(U_i \mid Z(W) = \epsilon\right).
\end{aligned}$$

Hence the inequality holds when $n = m + 1$ as well. ∎

### D. Proofs for Subsection IV-C

*1) Proof of Proposition 3:* First note that each XOR operation at the $j$-th recursion can be associated with exactly one vector $\mathbf{s} = (s_1, s_2, \ldots, s_n) \in \{-,+\}^n$ and $s_j = -$. For example, assume that the number of minus signs in $\mathbf{s}$, denoted as $m(\mathbf{s})$, is larger than $n_{lub} = \log w_{u.b.} = n\lambda$, and let $\tau = \tau(\mathbf{s})$ be the index such that $m(s_\tau, s_{\tau+1}, \ldots, s_n) = n_{lub}$ and $s_\tau = -$. Then for each index $i$ in the set $\{k : 1 \leqslant k < \tau, s_k = -\}$, there is a bijection between the pair $(\mathbf{s}, i)$ and an XOR operation at the $i$-recursion which is split and modified in the ADRS scheme. Hence, the extra complexity of the SC decoder for the ADRS scheme, compared to that of the SC decoder for the code based on $G_2^{\otimes n}$, is given by

$$\begin{aligned}
&\sum_{l=1}^{n-n_{lub}+1} \left|\{\mathbf{s} \in \{-,+\}^n : \tau(\mathbf{s}) > l, s_l = -\}\right| 2(2^{l+1}-2)c \\
&= \sum_{l=1}^{n-n_{lub}+1} \sum_{k=l+1}^{n-n_{lub}+1} \left|\{\mathbf{s} \in \{-,+\}^n : \tau(\mathbf{s}) = k, s_l = -\}\right| \cdot \\
&\qquad 2(2^{l+1}-2)c \\
&= \sum_{k=1}^{n-n_{lub}+1} \left[\binom{n-k+1}{n_{lub}} 2^{k-2} \sum_{l=1}^{k-1} 2(2^{l+1}-2)c\right] \\
&\leqslant 4c \sum_{k=1}^{n-n_{lub}+1} \binom{n-k+1}{n_{lub}} 2^{k-2} \sum_{l=1}^{k-1} 2^l \\
&= 4c \sum_{k=1}^{n-n_{lub}+1} \binom{n-k+1}{n_{lub}} 2^{k-2}(2^k-2) \\
&\leqslant 4c \sum_{k=0}^{n-n_{lub}} \binom{n-k}{n_{lub}} 2^{2k}. \qquad (12)
\end{aligned}$$

Now, let $\alpha = \frac{k}{n} \in [0, 1-\lambda]$. Using Stirling's approximation we have

$$\binom{n-k}{n_{lub}} 2^{2k} = \binom{n(1-\alpha)}{n\lambda} 2^{2\alpha n} \approx 2^{n(1-\alpha)h_b(\frac{\lambda}{1-\alpha})+2\alpha n},$$

for all sufficiently large $n$. It suffices to assume that $\lambda < \frac{3}{4}$. For $\lambda \geqslant \frac{3}{4}$, note that fewer XOR operations are split and modified, and that the resulting additional decoding complexity is not larger than when $\lambda < \frac{3}{4}$ is used. Let $f(\alpha, \lambda) = (1-\alpha)h_b(\frac{\lambda}{1-\alpha}) + 2\alpha$. We find its maximum, for a given $\lambda$, by solving

$$\begin{aligned}
0 = \frac{\partial}{\partial\alpha} f(\alpha,\lambda) &= \frac{\partial}{\partial\alpha}\Big[-(1-\alpha)\left(\frac{\lambda}{1-\alpha}\log\frac{\lambda}{1-\alpha}\right) \\
&\quad -(1-\alpha)\left(1-\frac{\lambda}{1-\alpha}\right)\log\left(1-\frac{\lambda}{1-\alpha}\right) + 2\alpha\Big] \\
&= \frac{1}{\ln 2}\left[-\ln(1-\alpha) + \ln(1-\alpha-\lambda)\right] + 2,
\end{aligned}$$

which is true if and only if $\alpha = 1 - \frac{4}{3}\lambda$. And note that $\frac{\partial^2 f(\alpha,\lambda)}{\partial\alpha^2} = \frac{1}{\ln 2}\left[\frac{1}{1-\alpha} - \frac{1}{1-\alpha-\lambda}\right] < 0$ for all $\alpha \in [0, 1-\lambda]$. The maximum of the function is then given by $f(1-\frac{4}{3}\lambda, \lambda) = 2 - \lambda\log 3$. Using the union bound, the sum in (12) can be bounded by $4cn2^{n(2-\lambda\log 3)}$, and the ratio of the sum to $N = 2^n$, denoted by $\gamma_C$, is bounded from above as $\gamma_C \leqslant 4cn2^{n(1-\lambda\log 3)}$. Since the exponent $n(1-\lambda\log 3)$ goes to negative infinity as $n$ grows when $\lambda > \lambda^\dagger = 1/\log 3 \approx 0.631$, the ratio $\gamma_C$ vanishes exponentially in $n$ when $\lambda > \lambda^\dagger$. The proposition follows by noting that the SC decoding complexity for the code based on $G_2^{\otimes n}$ is $N \log N$. ∎

*2) Proof of Proposition 4:* Similar to the proof of Proposition 3, the number of additional channels due to the ADRS scheme modification is given by

$$\begin{aligned}
&\sum_{l=1}^{n-n_{lub}+1} \left|\{\mathbf{s} \in \{-,+\}^n : \tau(\mathbf{s}) > l, s_l = -\}\right| 2^l \\
&= \sum_{l=1}^{n-n_{lub}+1} \sum_{k=l+1}^{n-n_{lub}+1} \left|\{\mathbf{s} \in \{-,+\}^n : \tau(\mathbf{s}) = k, s_l = -\}\right| 2^l \\
&\leqslant \sum_{k=1}^{n-n_{lub}+1} \binom{n-k+1}{n_{lub}} 2^{k-2} \sum_{l=1}^{k-1} 2^l \\
&\leqslant \sum_{k=0}^{n-n_{lub}} \binom{n-k}{n_{lub}} 2^{2k}. \qquad (13)
\end{aligned}$$

By the argument in the proof of Proposition 3, the sum in (13) can be upper bounded by $n2^{n(2-\lambda\log 3)}$, and the ratio of the sum to $N = 2^n$, denoted by $\gamma$, is bounded from above as $\gamma \leqslant n2^{n(1-\lambda\log 3)}$. Since the exponent $n(1-\lambda\log 3)$ goes to negative infinity as $n$ grows when $\lambda > \lambda^\dagger = 1/\log_2 3 \approx 0.631$, the ratio $\gamma$ vanishes exponentially in $n$ when $\lambda > \lambda^\dagger$. ∎

## REFERENCES


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

[2] E. Arikan, "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] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1953–1961.

[4] A. Vempaty, L. R. Varshney, and P. K. Varshney, "Reliable crowdsourcing for multi-class labeling using coding theory," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 667–679, Aug. 2014.

[5] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[6] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Mar. 2018.

[7] S. Horii, T. Yoshida, M. Kobayashi, and T. Matsushima, "Distributed stochastic gradient descent using LDGM codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1417–1421.
[8] A. Mazumdar and S. Pal, "Semisupervised clustering, AND-queries and locally encodable source coding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6489–6499.
[9] J. C. Pang, H. Mahdavifar, and S. S. Pradhan, "Coding for crowdsourced classification with XOR queries," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Aug. 2019, pp. 1–5.
[10] E. Arikan and E. Telatar, "On the rate of channel polarization," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2009, pp. 1493–1495.
[11] E. Arikan, "Source polarization," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 899–903.
[12] E. Abbe, "Polarization and randomness extraction," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2011, pp. 184–188.
[13] M. Mondelli, S. H. Hassani, I. Sason, and R. L. Urbanke, "Achieving Marton's region for broadcast channels using polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 2, pp. 783–800, Feb. 2015.
[14] N. Goela, E. Abbe, and M. Gastpar, "Polar codes for broadcast channels," *IEEE Trans. Inf. Theory*, vol. 61, no. 2, pp. 758–782, Feb. 2015.
[15] E. Şaşoğlu, E. Telatar, and E. Yeh, "Polar codes for the two-user binary-input multiple-access channel," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6583–6592, Jan. 2013.
[16] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Achieving the uniform rate region of general multiple access channels by polar coding," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 467–478, Feb. 2016.
[17] H. Mahdavifar and A. Vardy, "Achieving the secrecy capacity of wiretap channels using polar codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6428–6443, Oct. 2011.
[18] M. Andersson, V. Rathi, R. Thobaben, J. Kliewer, and M. Skoglund, "Nested polar codes for wiretap and relay channels," *IEEE Commun. Lett.*, vol. 14, no. 8, pp. 752–754, Aug. 2010.
[19] H. Mahdavifar, M. El-Khamy, J. Lee, and I. Kang, "Polar coding for bit-interleaved coded modulation," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3115–3127, May 2016.
[20] V. V. B. Chandar, "Sparse graph codes for compression, sensing, and secrecy," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 2010.
[21] A. Golmohammadi, D. G. M. Mitchell, J. Kliewer, and D. J. Costello, "Encoding of spatially coupled LDGM codes for lossy source compression," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5691–5703, Nov. 2018.
[22] Y. Zhang, S. Dumitrescu, J. Chen, and Z. Sun, "LDGM-based multiple description coding for finite alphabet sources," *IEEE Trans. Commun.*, vol. 60, no. 12, pp. 3671–3682, Dec. 2012.
[23] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.
[24] D. J. MacKay and R. M. Neal, "Good codes based on very sparse matrices," in *Proc. IMA Int. Conf. Cryptogr. Coding*. Cham, Switzerland: Springer, 1995, pp. 100–111.
[25] W. Zhong, H. Chai, and J. Garcia-Frias, "Approaching the Shannon limit through parallel concatenation of regular LDGM codes," in *Proc. Int. Symp. Inf. Theory*, 2005, pp. 1753–1757.
[26] J. Garcia-Frias and W. Zhong, "Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Commun. Lett.*, vol. 7, no. 6, pp. 266–268, Jun. 2003.
[27] A. M. Kakhaki, H. K. Abadi, P. Pad, H. Saeedi, F. Marvasti, and K. Alishahi, "Capacity achieving linear codes with random binary sparse generating matrices over the binary symmetric channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 621–625.
[28] H. Mahdavifar, "Scaling exponent of sparse random linear codes over binary erasure channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 689–693.
[29] W. Lin, S. Cai, B. Wei, and X. Ma, "Coding theorem for systematic LDGM codes under list decoding," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2018, pp. 1–5.
[30] P. Fuentes, J. E. Martinez, P. M. Crespo, and J. Garcia-Frias, "Approach for the construction of non-Calderbank–Steane–Shor low-density-generator-matrix–based quantum codes," *Phys. Rev. A, Gen. Phys.*, vol. 102, no. 1, Jul. 2020, Art. no. 012423.
[31] S. B. Korada, E. Sasoglu, and R. Urbanke, "Polar codes: Characterization of exponent, bounds, and constructions," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6253–6264, Dec. 2010.
[32] E. Arikan, "On the origin of polar coding," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 209–223, Feb. 2016.

**James Chin-Jen Pang** received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2016. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with the University of Michigan, Ann Arbor. His research interests lie in the areas of coding and information theory with applications to quantum communication and machine learning.

**Hessam Mahdavifar** (Member, IEEE) received the B.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2007, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of California at San Diego (UCSD), La Jolla, in 2009 and 2012, respectively. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. He was with Samsung U.S. Research and Development, San Diego, USA, as a Staff Research Engineer, between 2012 and 2016. His research interests include coding and information theory with applications to wireless communications, storage systems, security, and privacy.

He received the NSF Career Award in 2020. He also received the Best Paper Award in 2015, IEEE International Conference on RFID, and the 2013 Samsung Best Paper Award. He also received two silver medals at the International Mathematical Olympiad in 2002 and 2003 and two gold medals at Iran National Mathematical Olympiad in 2001 and 2002.

**S. Sandeep Pradhan** (Fellow, IEEE) received the M.E. degree from the Indian Institute of Science in 1996 and the Ph.D. degree from the University of California at Berkeley in 2001. From 2002 to 2008, he was an Assistant Professor and from 2008 to 2015, he was an Associate Professor with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, where he is currently a Professor. His research interests include network information theory, coding theory, and quantum information theory. He was an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY in the area of Shannon theory from 2014 to 2016.