On the Performance of Low-complexity Decoders of LDPC and Polar Codes

Qingqing Peng, Dawei Yin, Dongxu Chang, Yuan Li,
Huazi Zhang, Guiying Yan, Guanghui Wang

Abstract

Efficient decoding is crucial to high-throughput and low-power wireless communication scenarios. A theoretical analysis of the performance-complexity tradeoff toward low-complexity decoding is required for a better understanding of the fundamental limits in the above-mentioned scenarios. This study aims to explore the performance of decoders with complexity constraints. Specifically, we investigate the performance of LDPC codes with different numbers of belief-propagation iterations and the performance of polar codes with an SSC decoder. We found that the asymptotic error rates of both polar codes and LDPC codes are functions of complexity T and code length N, in the form of $2^{-a2^{b\frac{T}{N}}}$, where a and b are constants that depend on channel and coding schemes. Our analysis reveals the different performance-complexity tradeoffs for LDPC and polar codes. The results indicate that if one aims to further enhance the decoding efficiency for LDPC codes, the key lies in how to efficiently pass messages on the factor graph. In terms of decoding efficiency, polar codes asymptotically outperform (J, K)-regular LDPC codes with a code rate $R \leq 1 - \frac{J(J-1)}{2^J+(J-1)}$ in the low-complexity regime ($T \leq O(NlogN)$).

Index Terms

low-density parity-check codes, polar codes, successive-cancellation decoding, and belief propagation decoding.

I. Introduction

Through decades of efforts in pursuit of the Shannon limit, channel coding has seen major breakthroughs in both theory and practice. The channel coding schemes employed in 5G NR, including LDPC codes and polar codes have theoretically approached or achieved Shannon limit [1], [2]. As a result, communication systems now has high spectral efficiency. However, the vision for 6G application scenarios sets higher requirements for coding theory, aiming at throughput exceeding Tb/s and energy efficiency below pJ/bit, which underscores the need to address decoding efficiency in addition to coding gain [3]–[5].

This work is partially supported by the National Key R&D Program of China, (2023YFA1009603) (Corresponding author: Dawei Yin)

Qingqing Peng, Dawei Yin, Dongxu Chang, and Guanghui Wang are with the School of Mathematics, Shandong University, Jinan, 250100 China (e-mail:{pqing, daweiyin, dongxuchang} @mail.sdu.edu.cn, ghwang@sdu.edu.cn).

Yuan Li and Huazi Zhang are with Huawei Technologies Co. Ltd., Hangzhou, 310051 China (e-mail: {liyuan299, zhanghuazi}@huawei.com).

Guiying Yan is with the Academy of Mathematics and Systems Science, CAS, University of Chinese Academy of Sciences, Beijing, 100190 China (e-mail: yangy@amss.ac.cn).

Decoding efficiency, characterized by a complexity-performance tradeoff, varies across different decoding algorithms. Research on LDPC code performance-complexity tradeoffs stems from the McEliece conjecture [6]. It suggests that when the designed code rate is $1-\epsilon$ of the channel capacity C, the complexity should be approximately $\frac{1}{\epsilon}\ln\frac{1}{\epsilon}$ for a given block error rate (BLER) P_B. On the other hand, for a given code rate $R<C$, Lentmaier et al. [7] establish the BLER upper bound for LDPC codes with $O(\log N)$ belief propagation iterations, where N is the block length. The decoding efficiency of polar codes has also been extensively researched. The simplified successive-cancellation (SSC) decoder can significantly reduce the complexity of successive-cancellation (SC) decoder [8] without performance loss. Seyyed et al. show that in the BMS channel, for a fixed $P_B \in (0,1)$ and a sequence of polar codes $\mathcal{C}_{polar}(P_B, W, N)$ of increasing block lengths with rates approaching the channel capacity, there exists $\bar{N}(\epsilon)$ for any $\epsilon > 0$, such that for any $N \geq \bar{N}(\epsilon)$, the latency of the SSC decoder with P processing elements is upper bounded by $O(N^{1-1/\mu} + (2+\epsilon)\frac{N}{P}\log_2\log_2\frac{N}{P})$ [9]. Specifically, a fully serial SSC decoder ($P=1$) has a complexity upper bound of $(2+\epsilon)N\log_2\log_2 N$. The tradeoff between the decoding circuit complexity and performance in VLSI models has also attracted research interest [10], [11]. In [11], Frank demonstrates that either the energy consumption $E(N) \geq \Omega(N\sqrt{-\ln f(N)})$ or the error rate $P_B > f(N)$. The results mentioned above inspire our investigation into the decoding efficiency of polar codes and LDPC codes.

Our previous work has compared the decoding efficiency of LDPC and polar codes, using the number of messages passed (NMP) to measure complexity and a statistical distance to the Maximum A Posteriori (MAP) estimate to measure performance [12]. However, the results are mainly numerical and lack asymptotic analysis. Retaining NMP as the complexity measure in this study, we aim to establish a direct relationship between complexity T and error rate P_B, that is, $P_B = f(T)$.

For LDPC codes, Grover et al. predict a double exponential reduction of BER with the number of iterations [13]. In this study, we provide an explicit expression for this double-exponential relationship. To be specific, we establish a lower bound of the error rate by observing the number of messages passed from channel output to a variable node. This can be formulated as a graph theory problem, that is, how many non-repetitive neighbors of a given vertex are present in the Tanner graph during the decoding process. For polar codes, we also characterize the trade-off between the complexity of the polar code SSC decoder and its BLER.

Our primary contribution lies in providing the asymptotic error rates of LDPC and polar codes, formulated as $2^{-a2^{b\frac{T}{N}}}$, where a and b are constants that depend on channel and coding schemes. Note that b is the main term determining the order of the error rate. We demonstrate that for LDPC codes, $b \in (\frac{\log_2(J-1)}{2J}, \frac{\log_2(J-1)(K-1)}{2J})$, while for polar codes, $b=0.5$. This implies that polar codes asymptotically outperform (J,K)-regular LDPC codes with a code rate $R \leq 1 - \frac{J(J-1)}{2^J+(J-1)}$ in the low-complexity regime ($T \leq O(NlogN)$). This conclusion aligns with the findings in [12] and corroborates results on terabits-per-second SC decoders [14], [15].

II. PRELIMINARIES

In this section, we will review the definitions of LDPC codes and polar codes, as well as their decoding algorithms, namely belief propagation (BP) decoding for LDPC codes and simplified successive-cancellation (SSC) decoding for polar codes.

A. Bipartite Graphs and LDPC Codes

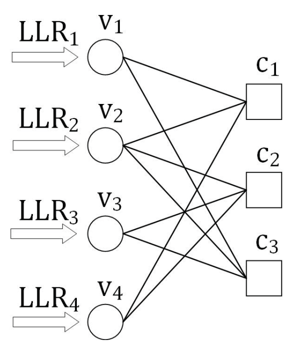
An undirected bipartite graph $G = (V_G \cup C_G, E_G)$ is defined as two disjoint sets of vertices V_G and C_G, and a set of edges E_G, where E_G is a subset of the pairs $\{\{v, c\} : v \in V_G, c \in C_G\}$, and $|V_G|$ and $|C_G|$ denote the number of vertices in V_G and C_G, respectively. The degree of a vertice v refers to the number of edges connected to v, and it can be denoted as $\deg(v)$. A bipartite graph G is said to be bi-regular if all the vertices in V_G have the same degree $\deg(v)$ and all the vertices in C_G have the same degree $\deg(c)$.

In a bipartite graph, a path of length $2k$ is a sequence of vertices $\{v_1, c_1, v_2, \ldots, v_k, c_k, v_{k+1}\}$ in $V_G \cup C_G$ such that $\{v_i, c_i\}, \{c_i, v_{i+1}\} \in E$ for all $i \in \{1, \ldots, k\}$, and all the vertices $v_1, c_1, v_2, \ldots, v_k, c_k, v_{k+1}$ are distinct. The distance, denoted as d_{v_i,v_j}, in graph G between two vertices v_i and v_j is defined as the length of the shortest path connecting v_i and v_j in G.

An LDPC code with a parity check matrix H is a linear block code that can be represented by an undirected bipartite graph $G = (V_G \cup C_G, E_G)$, where V_G and C_G refer to the sets of variable nodes (VNs) and check nodes (CNs) respectively. This bipartite graph is also known as a Tanner graph, and an edge exists between VN v_i and CN c_j if and only if $H(c_j, v_i) = 1$. The parity-check matrix and Tanner graph for an LDPC code $\mathcal{C}$ with a length of 4 are given in Fig. 1. Note that $|V_G|$ denotes the code's block length N, and the code rate is generally defined as $1 - |C_G|/|V_G|$.

$$
H = \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array}
$$

(a) parity check matrix H



(b) Tanner graph G

Fig. 1. Parity check matrix and Tanner graph for an LDPC code with a length of 4. In the Tanner graph, circles represent VNs, and squares represent CNs. The i-th VN receives channel message LLR_i.

B. The Ensemble of LDPC Codes

Consider the normalized degree distribution from a node perspective:

$$L_{deg}(x) = \sum_i L_i x^i, \tag{1}$$

$$R_{deg}(x) = \sum_j R_j x^j, \tag{2}$$

where L_i denotes the proportion of VNs with a degree of i, and R_j denotes the proportion of CNs with a degree of j.

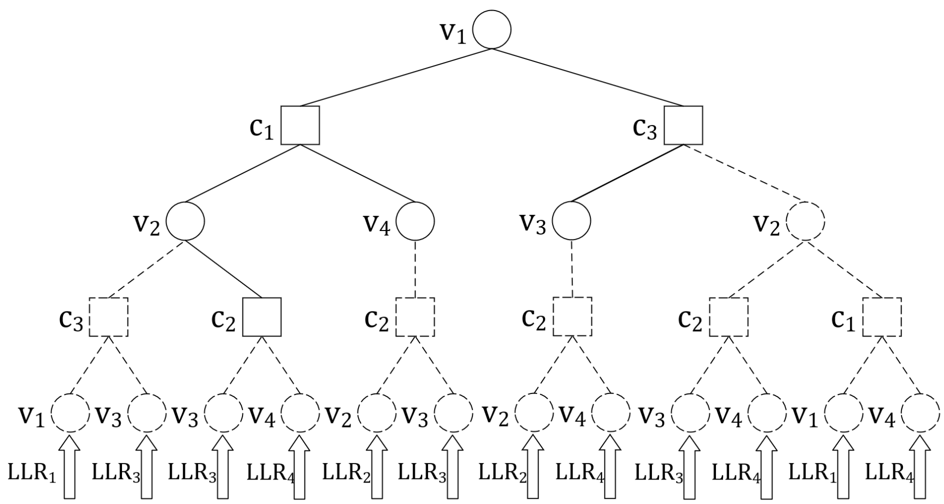


Fig. 2. A computation graph of height 4 for v_1 in Fig. 1. Solid lines represent the first appearances of nodes, whereas dashed lines indicate their subsequent occurrences.

Let $J = d_1 \geq d_2 \geq \cdots \geq d_N$ and $K = d'_1 \geq d'_2 \geq \cdots \geq d'_M$, where $M = \frac{L'_{deg}(1)}{R'_{deg}(1)}N$. Suppose the degree sets $\{d_i\}$ and $\{d'_j\}$ are selected according to the distributions $L_{deg}(x)$ and $R_{deg}(x)$, respectively. The Ensemble LDPC(N, L_{deg}, R_{deg}) is the probability space of all bipartite graphs with node set (V_G, C_G), where $V_G = \{v_1, v_2, \cdots, v_N\}$, $C_G = \{c_1, c_2, \cdots, c_M\}$, and $deg(v_i) = d_i, deg(c_j) = d'_j$. Within this space, the distribution of bipartite graphs is uniform.

It is necessary to describe the construction of LDPC(N, L_{deg}, R_{deg}) [16]. For each node z, we consider a bin that contains $deg(z)$ cells. We then consider random perfect matchings to pair the cells on the V_G side of the graph with the cells on the C_G side. Corresponding to each matching, there is a so-called configuration, in which the matched cells on the two sides of the graph are connected by an edge. We assume that configurations are selected uniformly at random. Corresponding to each matching (configuration), we construct a bipartite graph such that if there is an edge between two cells, then we place an edge between the corresponding nodes (bins) in the bipartite graph. We denote the ensemble of bipartite graphs so constructed by $\mathcal{G}$. We note that $\mathcal{G}$ contains bipartite graphs with parallel edges. The ensemble LDPC(N, L_{deg}, R_{deg}) is obtained by removing all bipartite graphs with parallel edges from $\mathcal{G}$. With the condition that the bipartite graphs constructed from random configurations have no parallel edges, the distribution of bipartite graphs (those in LDPC(N, L_{deg}, R_{deg})) is uniform.

C. BP Decoding and Computation Graph

One effective decoder for LDPC codes is BP decoding. In a Tanner graph, each VN receives a message from the channel, and BP decoding is achieved by passing messages along the edges. Assume that $\boldsymbol{x} = \{x_1, \ldots, x_N\} \in \{0,1\}^N$ are transmitted through binary-input memoryless channels, and $\boldsymbol{y} = \{y_1, ..., y_N\} \in \{0,1\}^N$ are received signals. Let

$$LLR_i = \ln \frac{p(y_i|x_i = 0)}{p(y_i|x_i = 1)} \tag{3}$$

denote the initial log-likelihood ratio (LLR) for the i-th VN, and LLR_i can also be regarded as the channel message of the i-th VN, as illustrated in Fig. 1.

The message-passing process of BP decoding consists of variable-to-check (V2C) message updates and check-to-variable (C2V) message updates. The message update rule of V2C is

$$LLR^{(l)}_{v_i \to c_j} = LLR_i + \sum_{c \in \mathcal{N}(v_i) \setminus c_j} LLR^{(l-1)}_{c \to v_i}, \tag{4}$$

and the message update rule of C2V is

$$LLR^{(l)}_{c_j \to v_i} = 2 \tanh^{-1} \left(\prod_{v \in \mathcal{N}(c_j) \setminus v_i} \tanh(LLR^{(l-1)}_{v \to c_j}/2) \right), \tag{5}$$

where l is the number of iterations, $\mathcal{N}(v)$ denotes the nodes connected directly to node v, $v_i \to c_j$ means from VN v_i to CN c_j and $c_j \to v_i$ means from CN c_j to VN v_i. The initial message $LLR^{(0)}_{v_i \to c_j}$ and $LLR^{(0)}_{c_j \to v_i}$ is 0.

The decoding process for a VN can be depicted as a computation graph [17]. For example, the decoding of v_1 in Fig. 1 depends on the messages from its neighboring nodes, c_1 and c_2. It processes the message received from c_1 and c_2 along with its own message to complete the decoding. As an example, the outgoing message from c_1 is a function of the messages it receives from v_2 and v_4. When we unroll this dependency structure for VN v_1, we arrive at the computation graph of height 4 shown in Fig. 2, which corresponds to 2 iterations of BP decoding. This computation graph is depicted as a tree, but in fact, it is not: several of the VNs and CNs appear repeatedly. For example, v_2 appears as a child of both c_1, c_2, and c_3. It is worth noting that the number of distinct VNs in Fig.2 is equivalent to the number of channel messages collected by v_1 in the decoding process.

D. Polar Codes and SSC Decoding

Polar codes [1] are parameterized by $(N, k, \mathcal{A})$, where N is the code length and $\mathcal{A}$ is a set of information indices that carry k information bits $\mathbf{u}_{\mathcal{A}}$. The complement of $\mathcal{A}$ is the frozen indices $\mathcal{A}^c$ that carry frozen bits $\mathbf{u}_{\mathcal{A}^c}$. The polar generator matrix is $\mathbf{G}_N = \mathbf{B}_N \mathbf{F}^{\otimes n}$ for any $N = 2^n$, where $\mathbf{B}_N$ is a bit-reversal permutation matrix, $\mathbf{F}^{\otimes n}$ denotes the $n-$th Kronecker power of $\mathbf{F} \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. A codeword is generated by $\mathbf{c} = \mathbf{u} G_N$, where $\mathbf{u} = (\mathbf{u}_{\mathcal{A}}, \mathbf{u}_{\mathcal{A}^c})$.

SC decoding is a message-passing algorithm on the factor graph of polar codes, as shown in Fig. 3 for a polar code of length $N = 8$ [1]. At level n of the factor graph, the LLR values $\boldsymbol{\alpha}_n^{0:N-1} = \{\alpha_n^0, \alpha_n^1, \cdots, \alpha_n^{N-1}\}$, that are calculated from the received channel-output vector, are fed to the decoder. At each level s, we have:

$$\alpha_s^i = \begin{cases} f(\alpha_{s+1}^i, \alpha_{s+1}^{i+2^s}) & \text{if } \lfloor \frac{i}{2^s} \rfloor \mod 2 = 0, \\ g(\alpha_{s+1}^i, \alpha_{s+1}^{i-2^s}, \beta_s^{i-2^s}) & \text{if } \lfloor \frac{i}{2^s} \rfloor \mod 2 = 1 \end{cases} \tag{6}$$

where $f(a, b) = 2 \operatorname{arctanh}(\tanh(\frac{a}{2}) \tanh(\frac{b}{2}))$, $g(a, b, c) = a + (1 - 2c)b$, and β_s^i is the i-th bit estimate at level s of the factor graph. As shown in Fig.4, the decoding process also can be represented as a binary tree. The bit estimates $\boldsymbol{\beta}_s = \{\beta_s^0, \beta_s^1, \cdots, \beta_s^{N-1}\}$ are calculated as

$$\beta_s^i = \begin{cases} \beta_{s-1}^i \oplus \beta_{s-1}^{i+2^s} & \text{if } \lfloor \frac{i}{2^s} \rfloor \mod 2 = 0, \\ \beta_{s-1}^i & \text{if } \lfloor \frac{i}{2^s} \rfloor \mod 2 = 1, \end{cases} \tag{7}$$

where $\oplus$ is the bit-wise XOR operation. All frozen bits are assumed to be zero. Hence at level $s = 0$, the i-th bit u^i is estimated as

$$\hat{u}_i = \beta_0^i = \begin{cases} 0 & \text{if } u_i \text{ is a frozen bit or } \alpha_0^i > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

The SSC decoding algorithm [8] identifies two types of nodes in the SC decoding tree. The bits within each node can be decoded efficiently in one shot without traversing its descendent nodes. These two types of nodes are:

- Rate-0 node: A node at level s of the SC decoding tree all of whose leaf nodes at level 0 are frozen bits. For a Rate-0 node at level s, bit estimates can be directly calculated at the level where the node is located as

$$\beta_i^s = 0. \tag{9}$$

- Rate-1 node: A node at level s of the SC decoding tree whose leaf nodes at level 0 are all information bits. For a Rate-1 node at level s, the bit estimations can be directly calculated at the level where the node is located as

$$\beta_s^i = \begin{cases} 0 & \text{if } \alpha_s^i > 0, \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

SSC decoding can decode Rate-0 and Rate-1 nodes in a single time step. As shown in Fig.5, in a binary tree representation of SC decoding, this corresponds to pruning all the nodes that are the descendants of a Rate-0 node or a Rate-1 node.

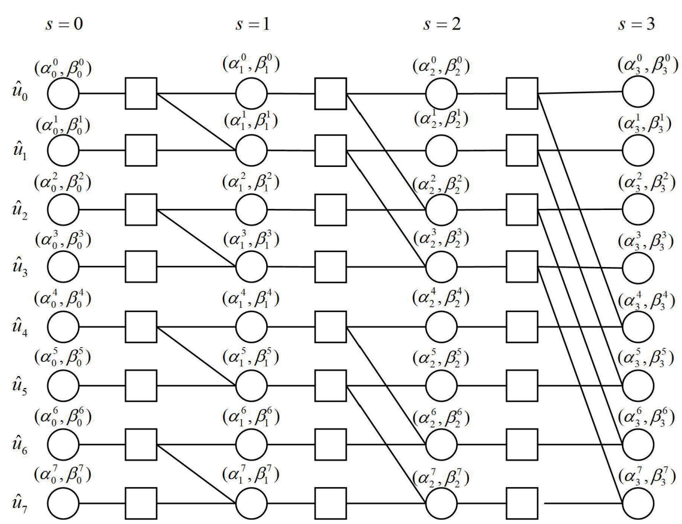


Fig. 3. SC decoding on the factor graph representation of Polar codes with $N = 8$.

E. Complexity metric and Big-O Notation

In this study, we define passing a V2C or C2V message on a directed edge as a unit of complexity [12], and the complexity T of a decoder is the number of messages passed (NMP). This notion of complexity is proportional to the complexity measured in arithmetic operations [6], and aligns with the simplification approach in [9], [18], [19].

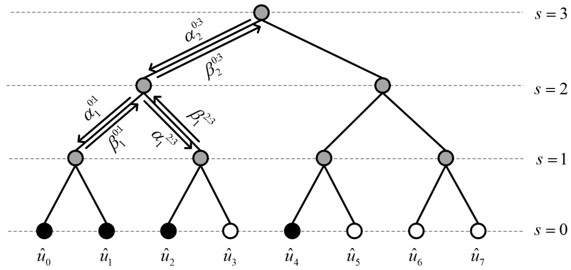


Fig. 4. The binary tree representation of SC decoding with $N = 8$ and $R = 0.5$. The white nodes represent information bits and the black nodes represent frozen bits.

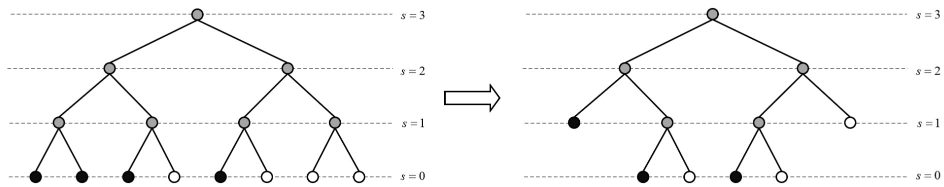
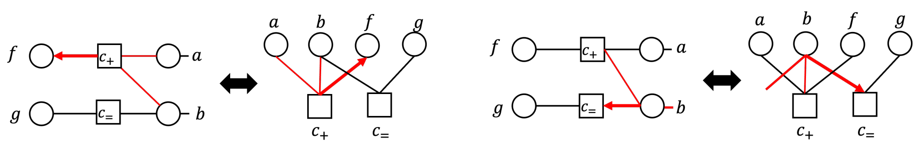


Fig. 5. The SSC decoding algorithm on decoding tree. The white nodes represent the R1 nodes and the black nodes represent the R0 nodes.

The polar factor graph is equivalent to a bipartite graph as shown in Fig. 6 [20]. Hence the same rules to measure complexity can be applied to polar codes.



(a) C2V in polar code

(b) V2C in polar code

Fig. 6. Four types of message passing in LDPC and polar codes.

We use standard Bachmann-Landau notation in this paper. For any non-negative real-valued functions $f(x)$ and $g(x)$, the notation $f(x) = O(g(x))$ (or equivalently $f(x) \leq O(g(x))$) implies that for sufficiently large x, $f(x)$ is bounded above by $cg(x)$, where c is a positive constant. Similarly, the notation $f(x) = \Omega(g(x))$ (or equivalently $f(x) \geq \Omega(g(x))$) indicates that for sufficiently large x, $f(x)$ is bounded below by $cg(x)$, where c is a constant.

III. Performance Bounds of LDPC Codes

This section aims to establish a lower bound on the BER of LDPC codes as a function of decoding complexity. We will analyze the average BER within two distinct ensembles: LDPC codes with finite degrees and regular LDPC codes. The proof techniques for these bounds are similar and can be divided into two main steps. Firstly, we calculate the number of channel messages that VN v can collect under l iterations, denoted as $n(v, l)$. This represents the number of distinct VNs in the computation graph rooted at v with a height of $2l$. Secondly, the lower bound on the error probability for v with $n(v,l)$ channel messages is established. Assume that the LDPC codes considered in this section are transmitted over a binary-input output-symmetric memoryless channel, and each codeword is chosen uniformly at random for transmission.

A. the Error Probability of VN with Given Number of Messages

This subsection will introduce a lemma that provides a lower bound on the error probability for a VN v, given a specific number of channel messages. This lemma results in a lower bound on the BER for LDPC codes.

Lemma 1. *Consider an LDPC code with Tanner graph G. Let $n(v,l)$ denote the number of channel messages that VN v can collect under l iterations of BP decoding, the error probability of v can be expressed as follows,*

$$P_v \geq 2^{-n(v,l)c_1-c_1}, \tag{11}$$

where

$$c_1 = |\log_2 \sqrt{\alpha}|,$$

and α denotes the error probability in a single-channel transmission by using the MAP decoder. Therefore, α is a constant dependent on the channel.

The proof of Lemma 1 will be provided in Appendix A. Subsequently, we demonstrate how Lemma 1 is applied to assess the performance of LDPC codes.

Consider an LDPC code with a maximum degree of J for VNs and a maximum degree of K for CNs. For any VN v_i, since the maximum degree of VNs is J, the number of first-layer nodes (neighbors of v_i) in the computation graph for v_i is at most J. Similarly, for the CNs in the first layer of the computation graph, there can be at most $K-1$ children nodes. Therefore, the number of VNs in the second layer is at most $J(K-1)$. Following this pattern, in the $2l$-th layer, there can be at most $J(J-1)^{l-1}(K-1)^l$ VNs. This means that in the computation graph with a height of $2l$, there can be at most $c_2([(K-1)(J-1)]^l-1)$ VNs, where $c_2 = \frac{J\times(K-1)}{(J-1)(K-1)-1}$. In other words, v_i can collect at most $c_2([(K-1)(J-1)]^l-1)$ channel messages under l iterations of BP decoding.

Substituting $n(v_i,l) \leq \min\{c_2([(K-1)(J-1)]^l-1), N\}$ into Lemma 1, we obtain the following Corollary.

Corollary 2. *Consider an LDPC code with a maximum degree of J for VNs and a maximum degree of K for CNs. If we perform BP decoding with $l \leq \log_{(J-1)(K-1)}(\frac{N}{c_2}+1)$ iterations, the BER P_b is given by*

$$P_b \geq 2^{-[(K-1)(J-1)]^l\times c_1c_2+c_1c_2-c_1}, \tag{12}$$

where c_1 and c_2 are expressions as defined earlier.

Here is an intuitive interpretation of Corollary 2. If we perform a finite number of iterations for BP decoding, then increasing the code length will not result in a performance gain, such as QC-LDPC codes. Furthermore, if one desires an exponential decrease in error rate with respect to code length N, the complexity should be at least $\Omega(N\log_{(J-1)(K-1)} N)$. This conclusion is also consistent with the $O(N)$-complexity codes like irregular repeat–accumulate (IRA) and accumulate-repeat-accumulate (ARA) codes [21], [22], where the maximum degree tends to infinity as the code length increases.

B. Average BER over the Ensemble of LDPC Codes

When there is a significant disparity between the maximum and minimum degrees, the bound provided by the Corollary 2 is rather loose. In this subsection, we will present a more precise estimate of the expected number of messages collected by a random VN, which provides a lower bound on the average BER $\overline{P}_b$ over the ensemble of LDPC codes under l iterations. We uniformly randomly choose a graph G from LDPC(N, L_{deg}, R_{deg}) and a pair of nodes from G. Let d be a random variable denoting the distance between this pair of VNs. Then, $P(d > 2t | d > 2t-1)$ (denoted as $P(2t)$ for brevity) denotes the conditional probability that the d is greater than $2t$ given that it is greater than $2t-1$. We can recursively determine the number of messages collected during the t-th iteration using $P(2t)$ and subsequently calculate the error rate of nodes by employing Lemma 1.

Theorem 3. *Consider the ensemble LDPC(N, L_{deg}, R_{deg}). If we perform BP decoding with l iterations, then the average BER over the ensemble $\overline{P}_b$ can be expressed as follows,*

$$\overline{P}_b \geq 2^{-N\left(1-\prod_{t=1}^{l} P(2t)\right)c_1 - c_1}, \tag{13}$$

where

$$P(2t) = \sum_{k=1}^{\infty} L_k \left(\widetilde{P}(2t-1)\right)^k, \tag{14}$$

$$\widetilde{P}(2t-1) = \sum_{k=1}^{\infty} \frac{kR_k}{\sum_{k=1}^{\infty} kR_k} \left(\widetilde{P}(2t-2)\right)^{k-1}, \tag{15}$$

$$\widetilde{P}(2t-2) = \sum_{k=1}^{\infty} \frac{kL_k}{\sum_{k=1}^{\infty} kL_k} \left(\widetilde{P}(2t-3)\right)^{k-1}, \tag{16}$$

$$\widetilde{P}(1) = \sum_{k=1}^{\infty} \frac{kR_k}{\sum_{k=1}^{\infty} kR_k} \left(1 - \frac{1}{N}\right)^{k-1}. \tag{17}$$

Equations (14), (15), and (16) are valid for $t \geq 1$, $t \geq 2$, and $t \geq 2$, respectively.

$P(2t)$ can be calculated recursively using (14) $\sim$ (17), and $\widetilde{P}(\cdot)$ can be viewed as an intermediate variable in the recursion. In Appendix B, we will demonstrate that $\widetilde{P}(\cdot)$ denotes the conditional probability in the cavity graph.

The proof of Theorem 3 is based on an analysis of the expected number of messages collected by a random VN v in the ensemble, similar to the approach in Corollary 2.

Lemma 4. *Consider the ensemble LDPC(N, L_{deg}, R_{deg}). Let $\overline{n}(l)$ be the expected number of messages collected by a random VN in the ensemble under l iterations of BP decoding, where the expectation is over all instances of the code and the choice of the VN. We have*

$$\overline{n}(l) = N(1 - P(d > 2l)) = N\left(1 - \prod_{t=1}^{l} P(2t)\right), \tag{18}$$

and $P(2t)$ can be calculated by (14) $\sim$ (17).

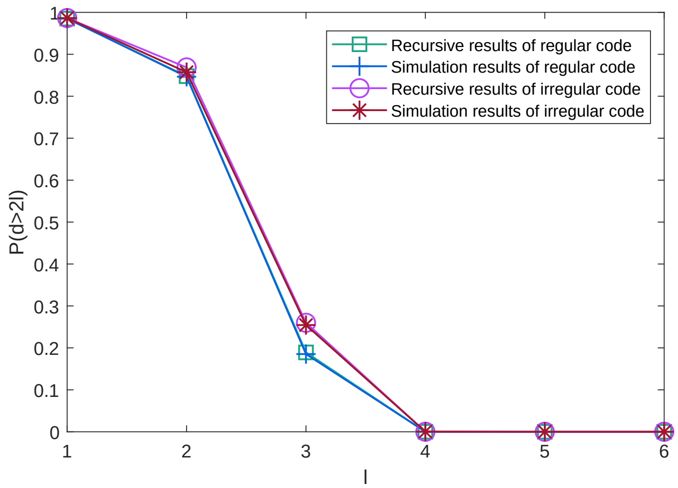


Fig. 7. The tail distribution $P(d > 2l)$ were obtained from both recursive equations and numerical simulations. These calculations were performed on two LDPC code ensembles: the ensemble of $(3, 6)$-regular LDPC codes and the ensemble of irregular LDPC codes, both with a block length of $N = 1000$. The degree distributions for the irregular LDPC code ensemble are specified as $L_{deg}(x) = 0.38354x^2 + 0.04237x^3 + 0.57409x^4$ and $R_{deg}(x) = 0.24123x^5 + 0.75877x^6$. The numerical results were averaged over 50 graph instances.

The proof of Lemma 4 will be presented in Appendix B. In Fig. 7, we depict the tail distribution $P(d > 2l)$ for two LDPC code ensembles, as obtained from (14): the ensemble of $(3, 6)$-regular LDPC codes and the ensemble of irregular LDPC codes, both with a block length of $N = 1000$. The degree distributions for the ensemble of irregular LDPC codes are specified as $L_{deg}(x) = 0.38354x^2+0.04237x^3+0.57409x^4$ and $R_{deg}(x) = 0.24123x^5+0.75877x^6$. The results are compared with computer simulations showing excellent agreement.

For a randomly selected VN v in the ensemble, Let $\overline{P}_v$ denote the expected error probability of v, where the expectation is over all instances of the code, the choice of the VNs, and the realization of the channel noise. According to Lemma 1, we have

$$\begin{aligned}\overline{P}_v &\overset{Lemma\ 1}{\geq} \mathbb{E}\left(2^{-n(v,l)c_1-c_1}\right)\\ &\overset{(a)}{\geq} 2^{-\overline{n}(l)c_1-c_1}\\ &\overset{Lemma\ 4}{=} 2^{-N\left(1-\prod_{t=1}^{l}P(2t)\right)c_1-c_1},\end{aligned} \tag{19}$$

where (a) follows Jensen's inequality and a $P(2t)$ can be calculated by (14) $\sim$ (17), which completes the proof of Theorem 3. Compared with Corollary 2, Theorem 3 provides a tighter lower bound for irregular LDPC codes.

C. Average BER over the Ensemble of Regular LDPC Codes

Assume that $L_{deg}(x) = x^J$ and $R_{deg}(x) = x^K$, then the ensemble mentioned in the previous section corresponds to the (J, K)-regular LDPC codes ensemble. By substituting $L_J = 1$ and $R_K = 1$ into Lemma 4 and Theorem 3, we can obtain Corollary 5 and Corollary 6.

Corollary 5. *Consider the ensemble of (J, K)-regular LDPC codes with a block length of N. Let $\overline{n}(l)$ denote the expected number of channel messages collected by a random VN in the ensemble under l iterations of BP decoding, where the expectation is over all instances of the code, the choice of VN. We have*

$$\overline{n}(l) = N\Big(1 - P(d > 2l)\Big), \tag{20}$$

where

$$P(d > 2l) = \exp\bigg\{ - \frac{J}{N} \frac{(K-1)^{l+1}(J-1)^l - K + 1}{(K-1)(J-1) - 1} \bigg\}. \tag{21}$$

In Corollary 5, $P(d > 2l)$ can be interpreted as the expected proportion of VNs that are not present in the computation graph of height $2l$ for a random VN v. $NP(d = 2l)$ denotes the expected number of solid-lined VNs at the $2l$-th layer in the computation graph of v, and it can be calculated as follows,

$$NP(d = 2l) = N\Big(P(d > 2l - 2) - P(d > 2l)\Big). \tag{22}$$

Remark. *If $l \leq c \log_{(J-1)(K-1)} N$, where c is a constant less than 1, then we have*

$$\lim_{N \to \infty} \frac{NP(d = 2l + 2)}{NP(d = 2l)} \approx (J-1)(K-1), \tag{23}$$

which reflects the good expansion property of random bipartite graphs. When the number of iterations l is fixed, the probability of the computation graph being a tree approaches 1 as the block length N tends to infinity [23]. Following the proof approach in [23], we can prove that when $l \leq \frac{c}{2} \log_{(J-1)(K-1)} N$ with $c < 1$, the computation graph is a tree with probability 1 in the limit of infinitely long blocklengths. By using Corollary 5, it can be demonstrated that when $l \in [\frac{1}{2} \log_{(J-1)(K-1)} N, c \log_{(J-1)(K-1)} N]$ with $\frac{1}{2} < c < 1$, although the computation graph might not be a tree, the number of nodes approximates that of a tree.

Corollary 6. *Consider the ensemble of (J, K)-regular LDPC codes with a block length of N. The decoder performs l iterations of BP decoding, the average BER over the ensemble*

$$\overline{P}_b \geq 2^{-c_1 N \left(1 - exp\{-\frac{c_2}{N} \times ([(K-1)(J-1)]^l - 1)\}\right) - c_1}, \tag{24}$$

where c_1 and c_2 are constants as defined earlier.

When $l \leq c \log_{(J-1)(K-1)} N$ with $c < 1$, the lower bounds in (12) and (24) are asymptotically equivalent. A single LDPC code iteration requires $2JN$ message-passing steps, hence, we can conclude that for (J, K)-regular LDPC codes when the decoding complexity is

$$T < \frac{2J}{\log_2 (J-1)(K-1)} N \log_2 (\frac{N}{c_2} + 1),$$

the average BER and average BLER are bounded by

$$\overline{P}_B(N, T) \geq \overline{P}_b(N, T) \geq \Omega(2^{-c_1 c_2 2^{\frac{\log_2 (J-1)(K-1)}{2J} \frac{T}{N}}}). \tag{25}$$

D. The Discussion about the Tightness of the Lower Bound

In this subsection, we will discuss the tightness of the order of the average BER over the ensemble of regular LDPC codes as presented in Section III-C. To achieve this goal, we reference the upper bound detailed in [7].

We refer to an upper bound on the LDPC code with a tree-like computation graph as presented in [7] and compare it with the lower bound we have provided. We rewrite it as follows.

Theorem 7 [7]. *Consider the iterative decoding of (N, J, K)-regular LDPC codes, where $J \geq 3$ and all computation graphs with a height of $2l$ are tree-like for any $l \leq \frac{\log_2 N}{\log_2 (J-1)(K-1)}$. Assume that l_1 is a constant determined by J, K and channel. If the number of iterations $l \in \left(l_1, \frac{\log_2 N}{\log_2 (J-1)(K-1)}\right)$, then there exists a constant c_4 such that P_b and P_B are approximately upper-bounded by the inequalities*

$$P_b < 2^{-c_4 2^{l \log_2 (J-1)}} = 2^{-c_4 2^{\frac{\log_2 (J-1)}{2J} \frac{T}{N}}} \tag{26}$$

and

$$P_B < N P_b < N 2^{-c_4 2^{l \log_2 (J-1)}}. \tag{27}$$

Note that the lower bound in (24) and the upper bound in (26) are both in the form of $2^{-a 2^{b \frac{T}{N}}}$, where a and b are constants. The values of b in the lower bounds provided by Corollary 6 is $\frac{\log_2 (J-1)(K-1)}{2J}$, where b is the main term determining the order of the lower bounds. The main term determining the order of upper bound is $\frac{\log_2 (J-1)}{2J}$, and the difference from the lower bound is only $\frac{\log_2 (K-1)}{2J}$.

IV. Polar codes

The decoding process of polar codes can also be represented in the Tanner graph [20]. Building upon this, a comparative analysis of the decoding process between LDPC and polar codes based on the number of messages passed was conducted in [12]. The SC decoding algorithm employs a unique message-passing scheduling strategy, ensuring that all information bits gather channel information and maintaining a complexity of only $N log_2 N$. Consequently, Lemma 1 can similarly be applied to polar codes, which provides a lower bound on the BER of polar codes that can be derived as

$$P_b \geq \Omega(2^{-c_1 N - c_1}).$$

This bound is relatively loose, and its BLER lower bound is roughly $\Omega(2^{-\sqrt{N}})$ [24].

Through the analysis of LDPC codes in the previous section, we have come to realize that the ability to effectively collect and utilize information in the Tanner graph is important. Given that SSC decoding inherits the scheduling strategy from SC decoding, this insight has led us to believe that polar codes may possess significant potential for low-complexity scenarios. In this section, we will enhance the results of [9] to establish the BLER-complexity tradeoff for polar codes assuming different code constructions and SSC decoding complexity. This analysis will require a different construction rule for polar codes to facilitate theoretical analysis.

Specifically, we redefine polar code construction as follows.

Definition 1. *For a given block length $N = 2^n$, binary memoryless symmetric (BMS) channel W, and probability of error $P_B = f(N)$, the polar code $\mathcal{C}_{polar}(P_B, W, N)$ is constructed by assigning the information bits to the* *positions corresponding to all the synthetic channels whose Bhattacharyya parameter is less than P_B/N and by assigning a predefined (frozen) value to the remaining positions.*

With the code construction rule of Definition 1, the error probability under SC decoding and SSC decoding is guaranteed to be at most P_B.

We can find that the construction of polar codes will vary according to the error probability P_B. Seyyed et al. show that for a fixed $P_B \in (0, 1)$ and a sequence of polar codes $\mathcal{C}_{polar}(P_B, W, N)$ of increasing block lengths with rates approaching the channel capacity, there exists $\bar{N}(\epsilon)$ for any $\epsilon > 0$, such that for any $N \geq \bar{N}(\epsilon)$, the latency of the SSC decoder with P processing elements is upper bounded by $O(N^{1-1/\mu} + (2+\epsilon)\frac{N}{P} \log_2 \log_2 \frac{N}{P})$. Specifically, the proof of [9, Theorem 1] demonstrates that when P equals 1, i.e., each unit of latency corresponds to one V2C and one C2V message passing, the complexity of the SSC decoder is upper-bounded by $(2 + \epsilon)N \log_2 \log_2 N$.

Typically, the complexity of an SSC decoder is fixed. However, by treating certain frozen bits as information bits, more aggressive simplification can be pursued, leading to even lower complexity. Of course, this comes at a cost of performance loss. The performance-complexity tradeoff of the ”modified” SSC decoder is illustrated below. Consider a polar code of length 8 and code rate 0.5, wherein the four positions corresponding to the smallest Bhattacharyya parameters are for information bits. Suppose that, during decoding, a frozen bit is decoded as an information bit. This modification incurs additional error at the frozen bit position which was error-free but, as shown in Fig. 8(b), it introduces a new rate-1 node to the SSC decoding tree, and thus reducing complexity. We can generalize this idea by treating more frozen bits as information bits, and obtain further complexity reduction in the tradeoff. Note that we do not actually increase the code rate from the encoder's perspective, but only treat the code as a higher-rate one at the decoder.

Marco et al. clarify the trade-off between the error rate and the gap to capacity in [25], indicating that the aforementioned adjustments to the SSC decoder can merely alter the scaling at how the code rate approaches the channel capacity, without affecting the asymptotic code rate. Thus, the decoder is always capable of decoding within the bounds of the channel capacity. Consequently, we will expound upon how this trade-off influences the complexity of the SSC decoder. The implications of this observation are encapsulated in the subsequent theorem.

Theorem 8. *Let W be a given BMS channel with symmetric capacity $I(W)$. There exists a sequence of polar codes $\mathcal{C}_{polar}(P_B, W, N)$ which are decoded by SSC decoders with complexity $T \in (2Nlog_2log_2N, Nlog_2N)$, such that for sufficient large N, the block error rate can be bounded by*

$$P_B \leq 2^{-2^{0.5\frac{T}{N}}}.$$

The proof of Theorem 8 continues the line of reasoning established in [9], with specific details available in Appendix C. Theorem 8 delivers an upper bound of the BLER of polar codes when $T \in (2Nlog_2log_2N, Nlog_2N)$, and we have

$$\Omega(2^{-\sqrt{N}}) \leq P_B(N, T) \leq O(2^{-2^{0.5\frac{T}{N}}}).$$

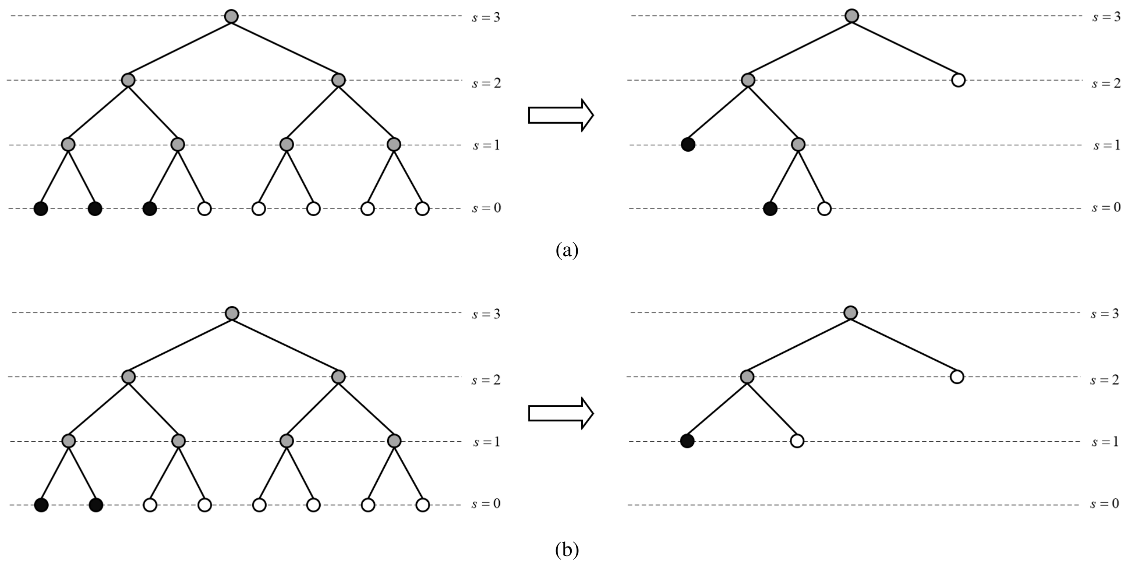


Fig. 8. The complexity of the SSC decoder varies with the code rate

Note that on the left side of the interval T, we obtain a better BLER estimate than that of [9]. With the corresponding complexity $T = (2+\epsilon)2Nlog_2 log_2 N$, the SSC decoding can achieve an error rate that is lower than any polynomial order. On the right side, our result is consistent with the conclusions in [24], where the upper and lower bounds coincide.

V. Conclusion

In this study, we provide the BER bounds for (J, K)-regular LDPC codes and the BLER bounds for polar codes which both can be unified in the form of

$$2^{-a2^{b\frac{T}{N}}},$$

where $b \in (\frac{\log_2(J-1)}{2J}, \frac{log_2(J-1)(K-1)}{2J})$ for LDPC codes and $b = 0.5$ for polar codes.

In Fig. 9, we illustrate how the performance of (J, K)-regular LDPC codes changes with different graph densities. When we fix the decoding complexity T and the code rate R, we observe that the main term b increases and then decreases as the degree J gets larger. This implies that there is an ideal graph density for each code rate that optimizes efficiency. These findings align with previous studies on the density of parity-check matrices found in [17].

These results also indicate that polar codes are more efficient than (J, K)-regular LDPC codes with code rate $R \leq 1 - \frac{J(J-1)}{2^J+(J-1)}$ in the low-complexity regime. Specifically, as shown in Fig. 10, when $T \leq N \log_2 N$ and for any $J \geq 3$, the BLER upper bound of polar codes is lower than the BER upper bound of LDPC codes. Moreover, when $R \leq 1 - \frac{J(J-1)}{2^J+(J-1)}$, the BLER upper bound of polar codes is even lower than the BER lower bound of regular LDPC codes.

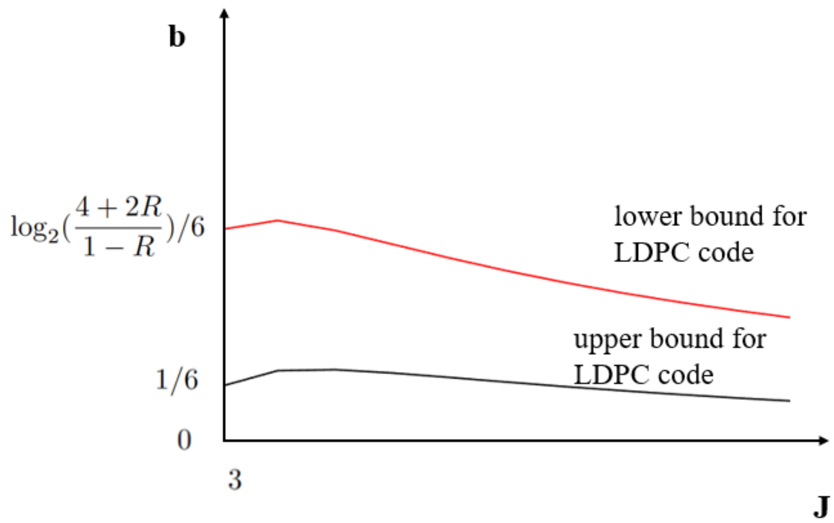


Fig. 9. b vs. J for the lower and upper bound of (J, K)-regular LDPC codes given the complexity T, code rate R.

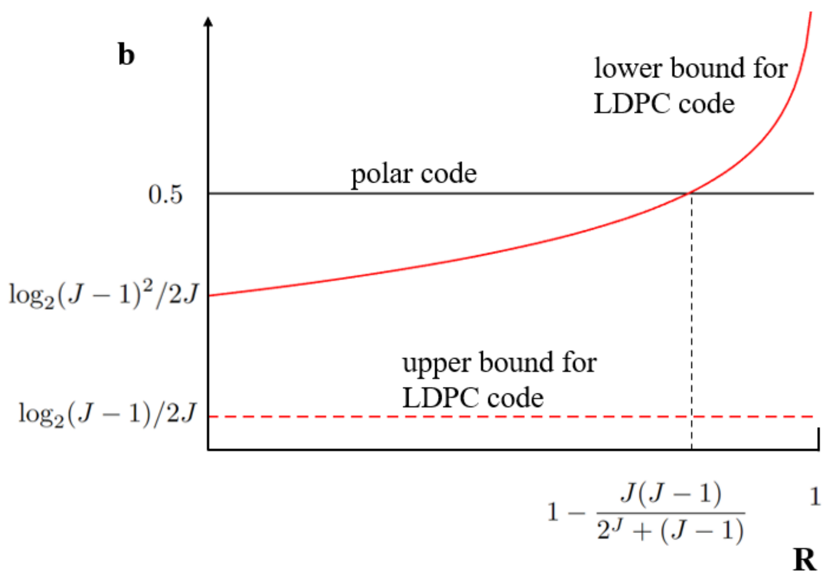


Fig. 10. b vs. R for bounds of LDPC BER and polar BLER when $T = Nlog_2N$.

The disadvantage range of LDPC codes changes with variations in graph density and degree distribution. When $Nlog_2N < T < \frac{2J}{\log_2(J-1)(K-1)} N \log_2(\frac{N}{c_2}+1)$, the performance of polar codes no longer improves with increasing complexity, whereas LDPC codes continue to exhibit performance improvements.

This study's practical contribution lies in indicating potential avenues for enhancing LDPC and polar codes. The findings related to LDPC codes underline the significance of the capacity to gather and utilize information on the Tanner graph. It is promising to note the growing interest in recent research centered on the scheduling policy of LDPC codes [26], [27]. For Polar codes, their strong capability in gathering and utilizing information suggests that efforts to enhance decoding performance should focus on other metrics, perhaps including code distance. A noteworthy accomplishment in this field is the exploration of how the polar spectrum can be improved through pre-transformation [28].

APPENDIX A
PROOF OF LEMMA 1

We start with the definition of tree code.

Definition 2. *Consider an LDPC code $\mathcal{C}$. Let $\mathcal{T}$ denote the computation tree with its root node as v and a height of $2l$. Define tree code $\mathcal{C}_v$ to be the set of valid codewords on $\mathcal{T}$ [17]. More precisely, $\mathcal{C}_v$ is the set of $0/1$ assignments on the variables contained in $\mathcal{T}$ that fulfill the constraints on the tree.*

Note that the block length of $\mathcal{C}_v$ is $n(v,l)$. Assume that the transmitter chooses the codeword uniformly at random from $\mathcal{C}_v$. Project the global codewords of $\mathcal{C}$ onto the set of variables contained in $\mathcal{T}$. This set of projections can be a strict subset of $\mathcal{C}_v$. The error probability of node v is only related to the computation graph $\mathcal{T}$, independent of the transmitted codewords. Thus, when performing l iterations of BP decoding, the error probability of node v in code $\mathcal{C}$ is equal to the error probability of node v in code $\mathcal{C}_v$. We denote the former as P_v and the latter as P_1.

Let $\mathcal{C}_v^{0/1}$ denote the codewords in $\mathcal{C}_v$ such that v is $0/1$. It is well known that $|\mathcal{C}_v^0| = |\mathcal{C}_v^1|$. Without loss of generality, we assume that the codewords in $\mathcal{C}_v^i$ are denoted as $\boldsymbol{x}_{i,1},...,\boldsymbol{x}_{i,t}$ with $i=0,1$. Denote the union of BP decoding regions of codewords $\boldsymbol{x}_{0,1},...,\boldsymbol{x}_{0,t}$ as Y_0, and the union of BP decoding regions of codewords $\boldsymbol{x}_{1,1},...,\boldsymbol{x}_{1,t}$ as Y_1. This gives us the following relationship:

$$\begin{aligned}
P_v = P_1 &= \sum_{i=1}^{t} P(\boldsymbol{x}_{0,i}) \int_{\boldsymbol{y}\in Y_1} P(\boldsymbol{y}|\boldsymbol{x}_{0,i}) d\boldsymbol{y} \\
&\quad + \sum_{i=1}^{t} P(\boldsymbol{x}_{1,i}) \int_{\boldsymbol{y}\in Y_0} P(\boldsymbol{y}|\boldsymbol{x}_{1,i}) d\boldsymbol{y} \\
&= \frac{1}{2t}\sum_{i=1}^{t} \left(\int_{\boldsymbol{y}\in Y_1} P(\boldsymbol{y}|\boldsymbol{x}_{0,i}) d\boldsymbol{y} + \int_{\boldsymbol{y}\in Y_0} P(\boldsymbol{y}|\boldsymbol{x}_{1,i}) d\boldsymbol{y} \right) \\
&\geq \frac{1}{2t}\sum_{i=1}^{t} \int_{\boldsymbol{y}\in Y_0\cup Y_1} \min\Big\{ P(\boldsymbol{y}|\boldsymbol{x}_{0,i}), P(\boldsymbol{y}|\mathbf{x}_{1,i}) \Big\} d\boldsymbol{y},
\end{aligned} \tag{28}$$

where $P(\boldsymbol{x})$ denotes the probability of codeword $\boldsymbol{x}$ being sent, and $P(\boldsymbol{y}|\boldsymbol{x})$ denotes the transition probability of a binary-input output-symmetric memoryless channel. The term $\frac{1}{2}\int_{\boldsymbol{y}\in Y_0\cup Y_1} \min\big\{P(\boldsymbol{y}|\boldsymbol{x}_{0,i}), P(\boldsymbol{y}|\boldsymbol{x}_{1,i})\big\} d\boldsymbol{y}$ can be considered as the BLER of a code composed of two codewords, $\boldsymbol{x}_{0,i}$ and $\boldsymbol{x}_{1,i}$, with each codeword being sent with a probability of $1/2$, and ML decoding is used. Clearly, this BLER is greater than or equal to P_2, the BLER of a repetition code REP_n with the same length $n(v,l)$ under ML decoding. This gives us the following result,

$$P_1 \geq \frac{1}{t}\sum_{i=1}^{t} P_2 = P_2 \tag{29}$$

In the case of the repetition code REP_n, where the an all-zero codeword is denoted as $\boldsymbol{x}_0$ and its ML decoding region is Y_0', and an all-one codeword is denoted as $\boldsymbol{x}_1$ with its ML decoding region being Y_1', we have:

$$\begin{aligned}P_2 &= P(\boldsymbol{x}_0)\int_{\boldsymbol{y}\in Y_1'} P(\boldsymbol{y}|\boldsymbol{x}_0)\mathrm{d}\boldsymbol{y}\\ &\quad + P(\boldsymbol{x}_1)\int_{\boldsymbol{y}\in Y_0'} P(\boldsymbol{y}|\boldsymbol{x}_1)\mathrm{d}\boldsymbol{y}\\ &\geq \frac{1}{2}2\alpha^{\frac{n(v,l)+1}{2}}\\ &= 2^{-n(v,l)|\log_2\sqrt{\alpha}|-|\log_2\sqrt{\alpha}|},\end{aligned} \tag{30}$$

where α denotes the error probability in a single-channel transmission. The reason for the validity of this inequality is rooted in the fact that if the first $\frac{n(v,l)+1}{2}$ positions of $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are both transmitted with errors, the ML decoder fails to decode.

By (28), (29) and (30), we conclude the proof of Lemma 1.

Appendix B
Proof of Lemma 4

In this section, we will provide a method to calculate the expected number of distinct VNs in the computation graph of height $2l$ for a random VN v in the ensemble, where the expectation is over all instances of the code and the choice of v. This problem is equivalent to finding the expected number of VNs that have a distance from a random VN v in the ensemble less than $2l$, where the expectation is over all instances of the Tanner graph and the choice of v.

It is easy to know that

$$\begin{aligned}\overline{n}(l) &= N\Big(1 - P(d > 2l)\Big)\\ &= N\Big(1 - P(d>0)\prod_{t=1}^{2l} P(d>t|d>t-1)\Big).\end{aligned} \tag{31}$$

Since we choose two different VNs as the initial and final nodes, we can establish that $P(d > 0) = 1$. Within a bipartite graph, the path length between VN pairs is always even, hence $P(d > 2t+1|d > 2t) = 1$, $t = 0, \ldots, l-1$. This observation further simplifies the (31):

$$P(d > 2l) = \prod_{t=1}^{l} P(d > 2t|d > 2t-1) = \prod_{t=1}^{l} P(2t). \tag{32}$$

Therefore, the key to proving Lemma 4 lies in the calculation of $P(2t)$. Next, we will provide a detailed calculation.

A. some recursive expressions in a given Tanner graph

We employ a recursive approach to provide the formula for computing $P(2t)$, drawing inspiration from [29], where the distribution of shortest path lengths in random graphs is discussed. Before calculating $P(2t)$ in Section B-B, it is necessary to establish some recursive expressions for the shortest path lengths in a given Tanner graph in this subsection.

In a given Tanner graph G, consider a pair of VNs (v_i, v_j) where the distance between them is denoted as d_{v_i,v_j}. To establish recursive expressions, we first introduce the definitions of conditional indicator functions and mean conditional indicator functions.

Definition 3. *The definitions of the indicator functions are as follows:*

$$\mathcal{X}(d_{v_i,v_j} > 2t) = \begin{cases} 1 & d_{v_i,v_j} > 2t, \\ 0 & d_{v_i,v_j} \leq 2t. \end{cases} \tag{33}$$

The definitions of the conditional indicator functions are as follows:

$$\begin{aligned} &\mathcal{X}(d_{v_i,v_j} > 2t | d_{v_i,v_j} > 2t-1) \\ &= \frac{\mathcal{X}(d_{v_i,v_j} > 2t \cap d_{v_i,v_j} > 2t-1)}{\mathcal{X}(d_{v_i,v_j} > 2t-1)}. \end{aligned} \tag{34}$$

Note that, $\mathcal{X}(d_{v_i,v_j} > 2t | d_{v_i,v_j} > 2t-1)$ indicates whether d_{v_i,v_j} is greater than $2t$, given that d_{v_i,v_j} is greater than $2t-1$. If this is true, $\mathcal{X}(d_{v_i,v_j} > 2t | d_{v_i,v_j} > 2t-1)$ takes a value of 1; otherwise, it assumes a value of 0. In case where the condition $d_{v_i,v_j} > 2t-1$ is not satisfied, the value of the conditional indicator function is undetermined. Though our primary focus is on the conditional indicator functions between pairs of VN, the same definitions apply equally to pairs of CNs as well as to VN and CN pairs.

Definition 4. *If we take the average of the conditional indicator function $\mathcal{X}(d_{v_i,v_j} > 2t | d_{v_i,v_j} > 2t-1)$ with respect to the final node v_j, and the averaging is done over the final nodes v_j in G where $d_{v_i,v_j} > 2t-1$, we obtain the mean conditional indicator function $m_{v_i}(t)$:*

$$m_{v_i}(2t) = \mathbb{E}_{v_j}\left[\mathcal{X}(d_{v_i,v_j} > 2t | d_{v_i,v_j} > 2t-1)\right]. \tag{35}$$

Based on the provided definition, for a given VN v_i, $m_{v_i}(2t)$ denotes the proportion of VNs that satisfy $d_{v_i,v_j} > 2t$ among the set of VNs in G where $d_{v_i,v_j} > 2t-1$.

Now, we present the recursive properties of the paths, which serve as the foundation for establishing the recursive expressions of the conditional indicator function and the mean conditional indicator function. As illustrated in Fig. 11, a path of length $2t$ from an initial node i to a final node j can be decomposed into an edge from i to $r \in \mathcal{N}(i)$, and a path of length $2t-1$ from r to j. In other words, if there is no path of length $2t$ between node i and node j, it implies that any neighbor r of node i does not have a path of length $2t-1$ to reach node j. The graph obtained by removing vertex i from graph G is referred to as the cavity graph of graph G, denoted as $\widetilde{G}$. All second-section paths decomposed from the paths i to j should be embedded in $\widetilde{G}$, implying that the paths from r to j must not traverse through node i. Using $\mathcal{X}^{(i)}(d_{rj} > 2t-1 | d_{rj} > 2t-2)$ to denote the conditional indicator function for the shortest path from r to j in $\widetilde{G}$, it is also referred to as the cavity indicator function. The superscript (i) stands for the fact that the node r is reached by a link from node i. Drawing upon the recursive nature of paths, it can be deduced that the conditional indicator function $\mathcal{X}(d_{v_i,v_j} > 2t | d_{v_i,v_j} > 2t-1)$ can be denoted as the product of conditional

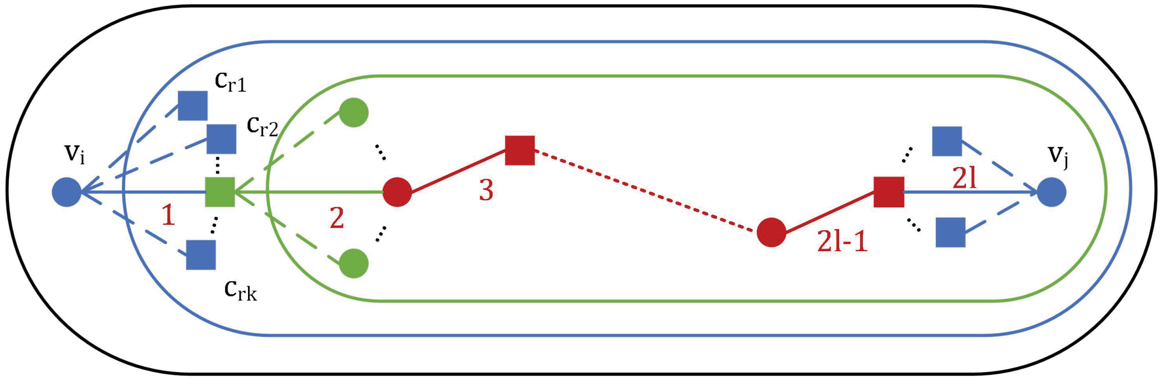


Fig. 11. Illustration of the possible paths of length $2l$ between two random VNs, v_i and v_j, in a Tanner graph G. The first edge of such a path connects node v_i to some other node, c_r, which may be any one of the k neighbors of node v_i. The rest of the path, from node c_r to node v_j is of length $2l-1$ and it resides on the cavity graph of graph G.

indicator functions for shorter paths (referred to as cavity indicator functions) between nodes $c_r \in \mathcal{N}(v_i)$ and v_j.

$$\begin{aligned} &\mathcal{X}(d_{v_i,v_j} > 2t | d_{v_i,v_j} > 2t-1) \\ &= \prod_{c_r \in \mathcal{N}(v_i)} \mathcal{X}^{(v_i)}(d_{c_r,v_j} > 2t-1 | d_{c_r,v_j} > 2t-2). \end{aligned} \tag{36}$$

Similar to $m_i(2t)$, averaging $\mathcal{X}^{(i)}(d_{c_r,v_j} > 2t-1 | d_{c_r,v_j} > 2t-2)$ with respect to the final node v_j yields the mean cavity indicator function $m_{c_r}^{(v_i)}(2t-1)$,

$$m_{c_r}^{(v_i)}(2t-1) = \mathbb{E}_j \left[\mathcal{X}^{(v_i)}(d_{c_r,v_j} > 2t-1 | d_{c_r,v_j} > 2t-2) \right]. \tag{37}$$

The averaging is done over the final nodes v_j in $\widetilde{G}$ where the length of the shortest path between c_r and v_j, embedded in $\widetilde{G}$, is greater than $2t-2$.

Under the assumption that the local structure of the network is tree-like, one can approximate the average of the product in Eq. (36) by the product of the averages. This assumption is fulfilled in the limit of large networks. In the analysis below we assume that $N \to \infty$ and thus obtain recursion equations of the form

$$m_{v_i}(2t) = \prod_{c_r \in \mathcal{N}(v_i)} m_{c_r}^{(v_i)}(2t-1). \tag{38}$$

The mean cavity indicator function $m_{c_r}^{(v_i)}(2t-1)$ obeys a similar equation of the form

$$m_{c_r}^{(v_i)}(2t-1) = \prod_{v_s \in \mathcal{N}(c_r) \setminus \{v_i\}} m_{v_s}^{(c_r)}(2t-2). \tag{39}$$

B. recursive expression for P(2t)

Pick a bipartite graph uniformly at random from the ensemble LDPC(N, L, R) and randomly select a VN v_i uniformly. With a probability of L_k, node v_i has a degree of k. If VN v_i is connected to a check node c_r by an edge, the probability that c_r has a degree of k is given by $\rho_k = \frac{kR_k}{R'(1)}$, where ρ_k denotes the check degree distributions from an edge perspective. Similarly, if the CN c_r is directly connected to a VN v_s (distinct from v_i), then the probability that v_s has a degree of k is given by $\lambda_k = \frac{kL_k}{L'(1)}$, where λ_k denotes the variable degree distributions from an edge perspective. Now, under this model, we will delve into the recursive expression for the conditional probability $P(2t)$.

Let $P\big(m_{v_i}(2t) = m\big)$ denote the probability that the mean indicator function associated with VN v_i takes on the value m, $\widetilde{P}\big(m_{c_r}^{(v_i)}(2t-1) = \widetilde{m}\big)$ denote the probability that the cavity mean indicator function associated with CN c_r takes on the value $\widetilde{m}$. Based on (38), we can derive:

$$\begin{aligned} & P\Big(m_{v_i}(2t) = m\Big) \\ & = \sum_{k=1}^{\infty} L_k \int_0^1 \int_0^1 \cdots \int_0^1 \prod_{r=1}^{k} \widetilde{P}\Big(m_{c_r}^{(v_i)}(2t-1) = m_r\Big) \cdot \\ & \delta\Big(m - \prod_{r=1}^{k} m_r\Big) dm_1 dm_2 \dots dm_k, \end{aligned} \tag{40}$$

where

$$\delta\Big(m - \prod_{r=1}^{k} m_r\Big) = \begin{cases} 1 & \text{if } m = \prod_{r=1}^{k} m_r, \\ 0 & \text{if } m \neq \prod_{r=1}^{k} m_r. \end{cases} \tag{41}$$

When the degree of node v_i is k, $m_{v_i}(2t)$ is equal to m if and only if the product of cavity mean indicator functions corresponding to the k neighbors of node v_i equals m. In other words, $P\left(m_{v_i}(2t) = m\right) = P(\prod_{r=1}^{k} m_{c_r}^{(v_i)}(2t-1) = m)$. $P(\prod_{r=1}^{k} m_{c_r}^{(v_i)}(2t-1) = m)$ can be expressed by the integral term in (40). And the δ function constrains the integration domain to the region where $m = \prod_{r=1}^{k} m_{c_r}^{(v_i)}(2t-1)$. The validity of (40) relies on the probability of a node v_i having a degree k being L_k.

Similarly, based on (39), the following expression holds:

$$\begin{aligned} & \widetilde{P}\Big(m_{c_r}^{(v_i)}(2t-1) = m\Big) \\ & = \sum_{k=1}^{\infty} \frac{kR_k}{R'(1)} \int_0^1 \cdots \int_0^1 \prod_{s=1}^{k-1} \widetilde{P}\Big(m_{v_s}^{(c_r)}(2t-2) = m_s\Big) \cdot \\ & \delta\Big(m - \prod_{s=1}^{k-1} m_s\Big) dm_1 \dots dm_{k-1}, \end{aligned} \tag{42}$$

$$\begin{aligned} & \widetilde{P}\Big(m_{v_s}^{(c_r)}(2t-2) = m\Big) \\ & = \sum_{k=1}^{\infty} \frac{kL_k}{L'(1)} \int_0^1 \cdots \int_0^1 \prod_{z=1}^{k-1} \widetilde{P}\Big(m_{c_z}^{(v_s)}(2t-3) = m_z\Big) \cdot \\ & \delta\Big(m - \prod_{z=1}^{k-1} m_z\Big) dm_1 \dots dm_{k-1}. \end{aligned} \tag{43}$$

Equation (42) calculates the probability of the cavity mean indicator function $m_{c_r}^{(v_i)(2t-1)}$ taking the value m. For node $c_r \in \mathcal{N}(v_i)$, the probability of its degree being equal to k is denoted as ρ_k, and as an intermediate node, one of its edges is consumed by the incoming link, leaving only $k-1$ links for the outgoing paths. Therefore, this equation slightly differs from (40).

The expected values of $P\Big(m_{v_i}(2t) = m\Big)$,$\widetilde{P}\Big(m_{c_r}^{(v_i)}(2t-1) = m\Big)$and $\widetilde{P}\Big(m_{v_s}^{(c_r)}(2t-2) = m\Big)$ provide the conditional probabilities

$$\begin{aligned} P(2t) &= P(d > 2t | d > 2t-1) \\ &= \int_0^1 mP\Big(m_{v_i}(2t) = m\Big)dm, \end{aligned} \tag{44}$$

$$\begin{aligned} \widetilde{P}(2t-1) &= \widetilde{P}(d > 2t-1 | d > 2t-2) \\ &= \int_0^1 m\widetilde{P}\Big(m_{c_r}^{(v_i)}(2t-1) = m\Big)dm \end{aligned} \tag{45}$$

and

$$\begin{aligned} \widetilde{P}(2t-2) &= \widetilde{P}(d > 2t-2 | d > 2t-3) \\ &= \int_0^1 m\widetilde{P}\Big(m_{v_z}^{(c_r)}(2t-2) = m\Big)dm. \end{aligned} \tag{46}$$

Plugging Eqs. (40) ,(42)and (43) into Eqs. (44), and (45), respectively, we obtain the recursion equations

$$P(2t) = \sum_{k=1}^{\infty} L_k \Big(\widetilde{P}(2t-1)\Big)^k, \tag{47}$$

$$\widetilde{P}(2t-1) = \sum_{k=1}^{\infty} \frac{kR_k}{R'(1)} \Big(\widetilde{P}(2t-2)\Big)^{k-1} \tag{48}$$

and

$$\widetilde{P}(2t-2) = \sum_{k=1}^{\infty} \frac{kL_k}{L'(1)} \Big(\widetilde{P}(2t-3)\Big)^{k-1}, \tag{49}$$

which are valid for $t \geq 1$,$t \geq 2$ and $t \geq 2$, respectively. For $t = 1$ when the number of nodes is sufficiently large, we can approximately obtain the result

$$\widetilde{P}(1) = \sum_{k=1}^{\infty} \frac{kR_k}{R'(1)} \Big(1 - \frac{1}{N}\Big)^{k-1}. \tag{50}$$

This completes the proof of Lemma 4.

Appendix C
Proof of Theorem 8

We first introduce the relevant results regarding the scaling exponent and the number of unpolarized nodes [9, Lemma 1].

Definition 5. *We say that μ is an upper bound on the scaling exponent if there exists a function $h(x) : [0,1] \to [0,1]$ such that $h(0) = h(1) = 0$, $h(x) > 0$ for any $x \in (0,1)$, and*

$$\sup_{\substack{x \in (0,1) \\ y \in [x\sqrt{2-x^2}, 2x-x^2]}} \frac{h(x^2) + h(y)}{2h(x)} < 2^{-1/\mu}. \tag{51}$$

For BI-AWGN channel [30] shows that $\mu \approx 4$, and for BSC it is conjectured that $\mu \approx 4.2$. For the BEC, the condition (51) can be relaxed to

$$\sup_{x\in(0,1)} \frac{h(x^2)+h(2x-x^2)}{2h(x)} < 2^{-1/\mu}, \tag{52}$$

which gives a numerical value $\mu \approx 3.63$.

Lemma 9 [9]. *Let W be a BMS channel and let $Z_n = Z(W_n)$ be the random process that tracks the Bhattacharyya parameter of W_n. Let μ be an upper bound on the scaling exponent according to Definition 5. Fix $\gamma \in (\frac{1}{1+\mu}, 1)$. Then, for $n \geq 1$,*

$$\mathbb{P}\left(Z_n \in \left[2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}, 1-2^{-2^{n\gamma h_2^{(-1)}\left(\frac{\gamma(\mu+1)-1}{\gamma\mu}\right)}}\right]\right) \leq c_0 2^{-n(1-\gamma)/\mu}, \tag{53}$$

where c_0 is a numerical constant that does not depend on n, W, or γ, and $h_2^{(-1)}$ is the inverse of the binary entropy function $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$ for $x \in [0, 1/2]$.

We now provide a refined version of [9, Lemma 2].

Lemma 10. *Let W be a BMS channel. For any $\epsilon > 0$, $T \in ((2+\epsilon)N\log_2\log_2 N, N\log_2 N)$, $P_B \geq 2^{-2^{\frac{T}{(2+0.1\epsilon)N}}}$, $N = 2^n$ and $M = 2^m$ with $m < n$. Consider the polar code $\mathcal{C}_{polar}(P_B/M, W, N/M)$ constructed according to Definition 1. Then, there exists an integer n_0, which depends on P_B, such that for all $n \geq n_0$, the following holds:*
1) If $Z(W) \leq 2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}$, then the polar code $\mathcal{C}_{polar}(P_B/M, W, N/M)$ has rate 1.
2) If $Z(W) \geq 1-2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}$, then the polar code $\mathcal{C}_{polar}(P_B/M, W, N/M)$ has rate 0.

Proof. Similar to the proof of Lemma 2 in [31], we start with the case that $Z(W) \leq 2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}$. Note that, for $n \geq 1$

$$Z_n \begin{cases} \in & [Z_{n-1}\sqrt{2-Z_{n-1}^2}, 2Z_{n-1}-Z_{n-1}^2] \quad \text{w.p. } 1/2, \\ = & Z_{n-1}^2 \quad \text{w.p. } 1/2. \end{cases} \tag{54}$$

Thus $Z_n \leq 2Z_{n-1}$. Thus, as $Z(W) \leq 2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}$, for any $i \in \{1, ..., N/M\}$ and sufficiently large N, we have that

$$Z(W_{n-m}^{(i)}) \leq \frac{2^{n-m}}{2^{2^{\frac{T}{(2+0.5\epsilon)N}}}} \leq \frac{N}{2^{2^{\frac{T}{(2+0.5\epsilon)N}}}} \leq \frac{P_B}{N}. \tag{55}$$

The last inequality holds since when $T > (2+\epsilon)NloglogN$, we have $2^{\frac{T}{(2+0.1\epsilon)N}} - 2^{\frac{T}{(2+0.5\epsilon)N}} > 2logN$ for sufficiently large N. Therefore, $\mathcal{C}_{polar}(P_B/M, W, N/M)$ has rate 1.

Now consider the second case where $Z(W) \geq 1-2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}$. Consider the random process $1-Z_n$ and note that (54) implies that $1-Z_n \leq 2(1-Z_{n-1})$. As $1-Z(W) \leq 2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}$, the proof is consistent with the first case. $\square$

The proof of Theorem 8 is presented as follows.

Proof. Consider pruning the decoding tree at depths k_1 and k_1+k_2, with

$$\begin{aligned} k_1 &= \frac{T}{N}, \\ k_2 &= min\{\frac{50T}{N}, \log_2 N - k_1\}. \end{aligned} \tag{56}$$

If $k_1 \leq \frac{1}{2}\log_2 N$, there are two constants γ_1, γ_2 such that for sufficiently large values of N,

$$\begin{aligned} 2^{-2^{k_1\gamma_1 h_2^{(-1)}\left(\frac{\gamma_1(\mu+1)-1}{\gamma_1\mu}\right)}} &\leq 2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}, \\ 2^{-2^{k_2\gamma_2 h_2^{(-1)}\left(\frac{\gamma_2(\mu+1)-1}{\gamma_2\mu}\right)}} &\leq 2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}. \end{aligned} \tag{57}$$

Note that, γ_1 that satisfies the first inequality certainly exists, since we have

$$\lim_{\gamma_1 \to 1} \gamma_1 h_2^{(-1)}\left(\frac{\gamma_1(\mu+1)-1}{\gamma_1\mu}\right) = \frac{1}{2}. \tag{58}$$

For the second inequality, k_2 is greater than k_1, therefore γ_2 also exists. Now, partition the decoding tree into three parts: (i) nodes that appear above depth k_1, (ii) what remains between depth k_1 and the next k_2 levels after pruning the tree at depth k_1, and (iii) what remains in the decoding tree after pruning at depth $k_1 + k_2$.

For part (i), the total decoding complexity sums up to

$$\sum_{i=1}^{k_1} 2^i \frac{N}{2^i} = k_1 N. \tag{59}$$

At depth k_1, there are a total of 2^{k_1} nodes prior to the pruning. By using Lemma 9 and the first inequality in (57), there are at most

$$a_1 \triangleq c_0 2^{k_1(1-\frac{1-\gamma_1}{\mu})} \tag{60}$$

nodes whose Bhattacharyya parameter is in the interval $[2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}, 1-2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}]$. Thus, by applying Lemma 10 with $M = 2^{k_1}$, all but those a_1 nodes can be pruned. Hence, part (ii) of the decoding tree consists of at most a_1 sub-trees with depth k_2. Consequently, the total decoding complexity for part (ii) can be upper bounded by

$$a_1 \sum_{i=1}^{k_2} 2^i \frac{N}{2^{i+k_1}} = a_1 k_2 \frac{N}{2^{k_1}} = o(k_1 N). \tag{61}$$

At depth k_2, each of the sub-trees has a total of 2^{k_2} nodes before pruning. By usng Lemma 9 and the second inequality in (57), at most $c_0 2^{k_2(1-\frac{1-\gamma_2}{\mu})}$ of these nodes have Bhattacharyya parameter in the interval $[2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}, 1-2^{-2^{\frac{T}{(2+0.5\epsilon)N}}}]$. Thus, by applying Lemma 10 with $M = 2^{k_1+k_2}$, the number of remaining nodes after pruning at depth $k_1 + k_2$ can be upper bounded by

$$a_2 \triangleq c_0^2 2^{k_1(1-\frac{1-\gamma_1}{\mu})} 2^{k_2(1-\frac{1-\gamma_2}{\mu})}. \tag{62}$$

Consequently, the total decoding complexity for part (iii) can be upper bounded by

$$\begin{aligned} &a_2 \sum_{i=1}^{\log_2 N - k_1 - k_2} 2^i \frac{N}{2^{i+k_1+k_2}} \\ &= a_2\big(\log_2 N - k_1 - k_2\big)\frac{N}{2^{k_1+k_2}} \\ &\leq \frac{N\log_2 N}{2^{k_2\frac{1-\gamma_2}{\mu}}} = o(k_1 N). \end{aligned} \tag{63}$$

By summing the complexities of the three parts, the complexity is upper bounded by T for sufficiently large N. If $k_1 > \frac{1}{2}\log_2 N$, since there is no part (iii) of the decoding tree, it is only necessary to sum the first two parts of the decoding tree. At this point, it is unnecessary to consider whether the second inequality in equation 57 holds. Combining the above two cases, and since ϵ can be arbitrarily small, the proof of Theorem 8 is thus established. □

REFERENCES

[1] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on information Theory*, 55(7):3051–3073, 2009.

[2] David JC MacKay and Radford M Neal. Near shannon limit performance of low density parity check codes. *Electronics letters*, 33(6):457–458, 1997.

[3] Marwa Chafii, Lina Bariah, Sami Muhaidat, and Merouane Debbah. Twelve scientific challenges for 6G: Rethinking the foundations of communications theory. *IEEE Communications Surveys & Tutorials*, 2023.

[4] Yang Lu and Xianrong Zheng. 6G: A survey on technologies, scenarios, challenges, and the related issues. *Journal of Industrial Information Integration*, 19:100158, 2020.

[5] Panagiotis Skrimponis, Sourjya Dutta, Marco Mezzavilla, Sundeep Rangan, Seyed Hadi Mirfarshbafan, Christoph Studer, James Buckwalter, and Mark Rodwell. Power consumption analysis for mobile mmwave and sub-THz receivers. In *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pages 1–5. IEEE, 2020.

[6] Robert J McEliece. Are turbo-like codes effective on nonstandard channels? *IEEE Information Theory Society Newsletter*, 51(4):1–8, 2001.

[7] Michael Lentmaier, Dmitri V Truhachev, Kamil Sh Zigangirov, and Daniel J Costello. An analysis of the block error probability performance of iterative decoding. *IEEE Transactions on Information Theory*, 51(11):3834–3855, 2005.

[8] Amin Alamdar-Yazdi and Frank R Kschischang. A simplified successive-cancellation decoder for polar codes. *IEEE communications letters*, 15(12):1378–1380, 2011.

[9] Seyyed Ali Hashemi, Marco Mondelli, Arman Fazeli, Alexander Vardy, John M Cioffi, and Andrea Goldsmith. Parallelism versus latency in simplified successive-cancellation decoding of polar codes. *IEEE Transactions on Wireless Communications*, 21(6):3909–3920, 2021.

[10] Clark David Thompson. *A complexity theory for VLSI*. Carnegie Mellon University, 1980.

[11] Christopher G Blake and Frank R Kschischang. Energy, latency, and reliability tradeoffs in coding circuits. *IEEE Transactions on Information Theory*, 65(2):935–946, 2018.

[12] Dawei Yin, Yuan Liy, Xianbin Wang, Jiajie Tong, Huazi Zhang, Jun Wang, Guanghui Wang, Guiying Yan, and Zhiming Ma. On the message passing efficiency of polar and low-density parity-check decoders. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 528–534. IEEE, 2022.

[13] Pulkit Grover, Kristen Woyach, and Anant Sahai. Towards a communication-theoretic understanding of system-level power consumption. *IEEE Journal on Selected Areas in Communications*, 29(8):1744–1755, 2011.

[14] Jiajie Tong, Xianbin Wang, Qifan Zhang, Huazi Zhang, Jun Wang, and Wen Tong. Fast polar codes for terabits-per-second throughput communications. In *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 1–6, 2023.

[15] Altuğ Süral, E Göksu Sezer, Yiğit Ertuğrul, Orhan Arikan, and Erdal Arikan. Terabits-per-second throughput for polar codes. In *2019 IEEE 30th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, pages 1–7. IEEE, 2019.

[16] Ali Dehghan and Amir H Banihashemi. On the tanner graph cycle distribution of random LDPC, random protograph-based LDPC, and random quasi-cyclic LDPC code ensembles. *IEEE Transactions on Information Theory*, 64(6):4438–4451, 2018.

[17] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008.

[18] Chun-Hao Hsu and Achilleas Anastasopoulos. Capacity-achieving codes for noisy channels with bounded graphical complexity and maximum likelihood decoding. *submitted to IEEE Transactions on Information Theory*, 2006.

[19] Igal Sason and Gil Wiechman. Bounds on the number of iterations for turbo-like ensembles over the binary erasure channel. *IEEE transactions on information theory*, 55(6):2602–2617, 2009.

[20] Marc Fossorier. Polar codes: Graph representation and duality. *IEEE Commun. Lett.*, 19(9):1484–1487, 2015.

[21] Henry D Pfister and Igal Sason. Accumulate–repeat–accumulate codes: Capacity-achieving ensembles of systematic codes for the erasure channel with bounded complexity. *IEEE transactions on information theory*, 53(6):2088–2115, 2007.

[22] Henry D Pfister, Igal Sason, and Rudiger Urbanke. Capacity-achieving ensembles for the binary erasure channel with bounded complexity. *IEEE Transactions on Information Theory*, 51(7):2352–2379, 2005.

[23] Thomas J Richardson and Rüdiger L Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on information theory*, 47(2):599–618, 2001.

[24] Nadine Hussami, Satish Babu Korada, and Rudiger Urbanke. Performance of polar codes for channel and source coding. In *2009 IEEE International Symposium on Information Theory*, pages 1488–1492, 2009.

[25] Marco Mondelli, S Hamed Hassani, and Rüdiger L Urbanke. Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors. *IEEE Transactions on Information Theory*, 62(12):6698–6712, 2016.

[26] Min Jang, Kyeongyeon Kim, Seho Myung, Hongsil Jeong, Kyung-Joong Kim, and Sang-Hyo Kim. A design of layered decoding for QC-LDPC codes based on reciprocal channel approximation. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 554–559. IEEE, 2022.

[27] Dongxu Chang, Guanghui Wang, Guiying Yan, and Dawei Yin. An optimization model for offline scheduling policy of low-density parity-check codes. *arXiv:2303.13762*, 2023.

[28] Yuan Li, Zicheng Ye, Huazi Zhang, Jun Wang, Guiying Yan, and Zhiming Ma. On the weight spectrum improvement of pre-transformed reed-muller codes and polar codes. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 2153–2158, 2023.

[29] Mor Nitzan, Eytan Katzav, Reimer Kühn, and Ofer Biham. Distance distribution in configuration-model networks. *Physical Review E*, 93(6):062309, 2016.

[30] Satish Babu Korada, Andrea Montanari, Emre Telatar, and Rüdiger Urbanke. An empirical scaling law for polar codes. In *2010 IEEE International Symposium on Information Theory*, pages 884–888. IEEE, 2010.

[31] Marco Mondelli, Seyyed Ali Hashemi, John M Cioffi, and Andrea Goldsmith. Sublinear latency for simplified successive cancellation decoding of polar codes. *IEEE Transactions on Wireless Communications*, 20(1):18–27, 2020.