A Probabilistic Peeling Decoder to Efficiently Analyze Generalized LDPC Codes Over the BEC

Yanfang Liu, Pablo M. Olmos, Tobias Koch

Universidad Carlos III de Madrid & Gregorio Marañón Health Research Institute

Email: {vivian,olmos,koch}@tsc.uc3m.es

Abstract

In this paper, we analyze the tradeoff between coding rate and asymptotic performance of a class of generalized low-density parity-check (GLDPC) codes constructed by including a certain fraction of generalized constraint (GC) nodes in the graph. The rate of the GLDPC ensemble is bounded using classical results on linear block codes, namely Hamming bound and Varshamov bound. We also study the impact of the decoding method used at GC nodes. To incorporate both bounded-distance (BD) and Maximum Likelihood (ML) decoding at GC nodes into our analysis without resorting on multi-edge type of degree distributions (DDs), we propose the probabilistic peeling decoding (P-PD) algorithm, which models the decoding step at every GC node as an instance of a Bernoulli random variable with a successful decoding probability that depends on both the GC block code as well as its decoding algorithm. The P-PD asymptotic performance over the BEC can be efficiently predicted using standard techniques for LDPC codes such as density evolution (DE) or the differential equation method. Furthermore, for a class of GLDPC ensembles, we demonstrate that the simulated P-PD performance accurately predicts the actual performance of the GLPDC code under ML decoding at GC nodes. We illustrate our analysis for GLDPC code ensembles with regular DDs. In all cases, we show that a large fraction of GC nodes is required to reduce the original gap to capacity, but the optimal fraction is strictly smaller than one. We then consider techniques to further reduce the gap to capacity by means of random puncturing, and the inclusion of a certain fraction of generalized variable nodes in the graph.

Index Terms

Generalized low-density parity-check codes, codes on graphs, maximum-likelihood decoding

I. INTRODUCTION

Generalized low-density parity-check (GLDPC) block codes were first proposed by Tanner [1]. In contrast to standard LDPC codes, which are represented by bipartite Tanner graphs where variable nodes and single parity-check (SPC) nodes are connected according to a given degree distribution (DD), in GLDPC codes the SPC nodes in the graph are replaced by generalized constraint (GC) nodes [1]. The sub-code associated to each GC node is referred to as the component code. Examples of component codes used in the GLDPC literature are Hamming codes [2], Hadamard codes [3] or expurgated random codes [4], [5]. For powerful component codes, GLDPC codes have many potential advantages, including improved performance in noisy channels, fast convergence speed and low error floor [4], [6].

Upon selecting a particular class of component codes, the DD of the GLDPC code ensemble can be optimized, and near-capacity iterative decoding thresholds can be achieved [2], [4], [7]. Capacity-achieving GLDPC code ensembles can also be obtained by spatially-coupling GLDPC block codes with regular DDs [8], [9]. Furthermore, the asymptotic exponents of the weight/stopping set spectrum for irregular and spatially-coupled GLDPC ensembles have been derived in [6] and [10], respectively. Based on these works, it is possible to design asymptotically good GLDPC code ensembles to achieve capacity-approaching iterative decoding thresholds and a minimum distance that grows linearly with the blocklength.

In this paper, we analyze GLDPC code ensembles using a different approach. Instead of selecting a particular class of component codes and optimizing the graph DD, we are interested in analyzing the tradeoff between coding rate and iterative decoding threshold of GLDPC code ensembles with fixed DD, referred to as the base DD, as we increase the fraction ν of GC nodes in the graph. This approach is novel in the literature and we believe it is appealing from a design perspective, since one might be interested in introducing a certain amount of GC nodes in the Tanner graph of a given LDPC code, aiming at reducing the gap to channel capacity at the resulting coding rate, and at the same time improving the minimum distance of the code and thus the error floor.

For the BEC, iterative decoding of graph-based codes, such as LDPC or GLDPC codes, can be performed by means of *peeling decoding* (PD) algorithms [11], [12], [13], which iteratively remove from the Tanner graph variable nodes whose value is known. As a result, the decoding process yields a sequence of graphs whose mean coincides with the asymptotic (in the blocklength) evolution of the ensemble. Furthermore, this evolution can be computed by solving a particular set of differential equations [11]. In the case of GLDPC codes the derivation of such differential equations requires to specify in advance the DD of the graph, and a description of what kind of erasure patterns are locally decodable at any GC node, which depends on both the component codes and the corresponding decoding algorithm. In fact, the resulting decoding threshold of GLDPC codes heavily depends on this latter point [3], [5], [13]. For instance, as we demonstrated in this paper, for a $(2,7)$ base DD in which all check nodes are $(7,4)$-Hamming GC nodes, the asymptotic threshold over the BEC is $\epsilon^* \approx 0.7025$ if maximum likelihood (ML) decoding is performed at each GC nodes. However, it drops to $\epsilon^* \approx 0.5135$ if suboptimal bounded distance (BD) decoding is used instead of ML. In both cases, the coding rate is exactly the same. The reason for this difference in performance is that BD-decoded GC nodes only resolve erasure patterns up to degree $\mathtt{d} - 1$, where $\mathtt{d}$ is the minimum distance of the component code, whereas ML-decoded GC nodes can resolve a subset of erasure patterns of degree above $\mathtt{d} - 1$.

While deriving the asymptotic differential equations to analyze PD with BD decoding at GC nodes (BD-PD for short) follows a straightforward extension of the standard PD differential equations for LDPC codes [11], the GLDPC asymptotic analysis of PD under ML-decoded component codes (ML-PD, for short) requires the use of multi-edge-type DDs [14] to track down all possible decodable erasure patterns at GC nodes [8], [13]. As a consequence, the list of code parameters to jointly optimize becomes cumbersome. Specifically, the parameters include the description of the multi-edge DD, the position of GC nodes in the graph, the edge labelling at every GC node used to determine positions in the component block code, and the list of locally ML-decodable erasure patterns. In [5], the authors were able to incorporate ML-decoded GC nodes without resorting to multi-edge type DDs by analyzing the GLDPC average performance using extrinsic information (EXIT) charts when each GC node in the graph is selected at random within the family of block component codes with fixed block length and minimum distance larger than 2. This approach has a design caveat though, as it does neither allow the use of a single type of component codes, nor to narrow down the family of component codes by fixing the minimum distance.

In this paper, we propose an analysis methodology that allows to easily incorporate into the PD algorithm ML-decoded GC nodes with specific properties, such a particular value of the minimum distance $\mathtt{d}$ or how many erasure patterns beyond minimum distance it can decode. We develop a probabilistic description of all components of the GLDPC code, namely the base DD, the presence of GC nodes

in the graph, and the decoding method implemented at GC nodes. Regarding the latter aspect, we parameterize the decoding capabilities of at every node with a blocklength-K component code by a vector $(p_1, p_2, ..., p_K)$, where $p_w \in [0, 1]$, $w \in \{1, \ldots, K\}$, is the probability that a weight-w erasure pattern chosen at random is decodable. Thus, p_w is the fraction of decodable weight-w erasure patterns. Note that if we take $p_w = 1$ for $w \leq \mathsf{d} - 1$ and $p_w = 0$ for $w = \{\mathsf{d}, \ldots, K\}$, we recover BD-PD. We show how to properly incorporate such a probabilistic description of component codes into the PD algorithm, and denote the resulting algorithm as *probabilistic PD* (P-PD). Due to its probabilistic nature, the asymptotic analysis of P-PD does not require the use of multi-edge type DDs. We show by computer simulations that the P-PD performance accurately predicts the actual GLDPC performance when ML decoding is performed at GC nodes.

The performance predicted using P-PD is valid for any linear component code of blocklength-K and decoding profile $(p_1, p_2, ..., p_K)$. To analyze a family of linear component codes of blocklength-K and minimum distance d, we employ two bounds to compute the GLDPC coding rate. The Hamming or sphere-packing bound [15] is used to determine a converse bound on the rate of the GLDPC code ensemble as a function of a triplet of (ν, d, K). The Varshamov bound is considered to determine an achievable rate of the GLDPC code ensemble [16]. In many scenarios of interest, we show that these bounds are sufficiently tight and thus relevant for the code designer.

By employing a probabilistic description of the decoding capabilities at GC nodes, we are able to analyze a large class of GLDPC code ensembles and beyond-BD decoding methods with a fairly small set of parameters. We illustrate our analysis for GLDPC code ensembles using $(2, 6)$, $(2, 7)$, $(2, 8)$ and $(2, 15)$ base DDs. To obtain realistic values for the coding capabilities of the component codes, we have performed an exhaustive search of linear block codes of lengths $r \in [6, 7, 8, 15]$, including Hamming codes, Cyclic codes, Quasi Cyclic codes and Cordaro-Wagner Codes, and tabulated their corresponding description in terms of minimum distance d and $(p_1, p_2, ..., p_K)$. In all cases, we show that a large fraction of GC nodes is required in the GLDPC graph to reduce the original gap to capacity. However, the closest gap to capacity is not achieved at $\nu = 1$, but a smaller value must be used. Namely, there exists a critical ν^* value for which the gap to capacity is minimum. Furthermore, the best results are obtained for high-rate component codes, suggesting that the use of very powerful component codes does not pay off, since the gain in threshold does not compensate for the severe decrease of the GLDPC code rate.

We conclude the paper by combining the asymptotic threshold results with the weight spectral analysis of GLDPC ensembles in [10] to explore the range of ν values for which the GLDPC ensembles reduce the original gap to capacity and at the same time maintain a linear growth of the minimum distance with the block length. Finally, we illustrate how to incorporate further design techniques that can help to reduce

the gap to capacity of the code ensembles. Specifically, we discuss both random puncturing [17] and a simple class of doubly generalized LDPC (DG-LDPC) codes [18], [19]. In general, the methodology presented in this paper is flexible and decouples the problems of bounding the GLDPC coding rate and the asymptotic analysis of the ensemble. In this regard, broader classes of component codes at variable nodes and GC nodes could also be incorporated in a systematic way.

The paper is organized as follows. In Section II, we introduce GLDPC code ensembles and the notation used to characterize the DDs. Sections III and IV present the decoding algorithm and its asymptotic analysis. In Section V we bound the GLDPC code rate and analyze the rate-threshold tradeoff as a function of the fraction ν of GC nodes in the graph. The behavior of the GLDPC code ensembles with specific component codes is analyzed in VI. Finally, Sections VII and VIII consider further techniques to improve the asymptotic behavior of the code ensemble, by means of random puncturing and generalized variable nodes. We conclude the paper in Section IX with a discussion of our results.

II. GLDPC ENSEMBLES

In this section, we introduce the GLDPC code ensembles that will be analyzed in the rest of the paper and the notation used to define their DD.

A. Degree distribution

As illustrated in Fig. 1, the Tanner graph of every member in the ensemble contains $\mathtt{n}$ variable nodes (coded bits) and $\mathtt{c}$ parity-check nodes, among which a fraction ν corresponds to GC nodes while the rest corresponds to SPC nodes. We denote by $\mathtt{E}$ the number of edges in the Tanner graph and we define the degree of a node as the number of edges connected to it.

The DD of the ensemble is characterized as follows. The vector $\overline{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_J)$ is the *left* DD, where λ_i represents the fraction of edges (w.r.t. $\mathtt{E}$) connected to a variable node of degree i. Given $\overline{\lambda}$, $\mathtt{n}$ and $\mathtt{E}$ are related by [14]

$$\mathtt{n} = \mathtt{E} \sum_{i=1}^{J} \lambda_i / i. \tag{1}$$

The *right* DD is defined by two vectors $\overline{\rho}_p = (\rho_{p1}, \rho_{p2}, ..., \rho_{pK})$ and $\overline{\rho}_c = (\rho_{c1}, \rho_{c2}, ..., \rho_{cK})$, where ρ_{pj} denotes the fraction of edges (w.r.t. $\mathtt{E}$) connected to a SPC node that has degree j and ρ_{cj} denotes the fraction of edges (w.r.t. $\mathtt{E}$) connected to a GC node that has degree j. Throughout the paper, we use the subscript p for any DD component related to standard *parity* check nodes and the subscript c for any DD component related to generalized *component* codes. The DD is then characterized by the tuple
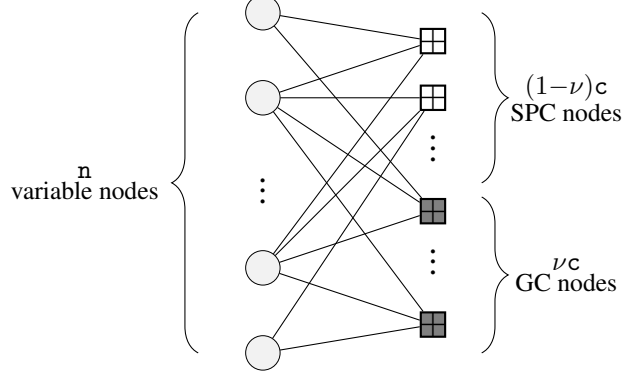
Fig. 1. Tanner graph of a GLDPC code.

$(\overline{\lambda}, \overline{\rho}_p, \overline{\rho}_c, \nu)$ and the ensemble of codes generated by this DD is denoted by $\mathcal{C}_{\overline{\lambda}, \overline{\rho}_p, \overline{\rho}_c, \nu}$. Since the fraction of GC nodes in the graph is ν, the following must hold:

$$\nu = \frac{\sum_{j=1}^{K} \rho_{cj}/j}{\sum_{u=1}^{K} (\rho_{cu} + \rho_{pu})/u}. \tag{2}$$

For simplicity, we restrict our analysis to the class of GLDPC ensembles characterized by SPC and GC nodes with constant degree J and K, respectively. The Tanner graph of any code in this ensemble contains n variable nodes, $\texttt{E} = J\texttt{n}$ edges, $\nu\frac{J}{K}\texttt{n}$ GC nodes, and $(1 - \nu)\frac{J}{K}\texttt{n}$ SPC nodes. The DD of the GLDPC codes is characterized by the triple (J, K, ν), and the ensemble of codes generated by this DD is denoted by $\mathcal{C}_{J,K,\nu}$. The DD of the LDPC ensemble obtained by taking $\nu = 0$ is defined as the *base DD*, and the corresponding LDPC code ensemble is referred to as the *base ensemble*. The coding rate of the base ensemble is denoted by $\texttt{R}_0$ and can be computed as:

$$\texttt{R}_0 = 1 - \frac{J}{K}. \tag{3}$$

Finally, we assume that the incoming edges to every degree-K GC nodes are assigned uniformly at random to each position of the component code.

B. The coding rate of the $\mathcal{C}_{J,K,\nu}$ ensemble

As discussed in the introduction of the paper, we propose tools to analyze the decoding performance of GLDPC under ML-decoded GC nodes that do not require to set in advance a specific component code to be used as the GC nodes. Instead, we consider the family of linear block codes with blocklength K and minimum distance d, and we use the classical results on linear block codes to bound the coding rate of the GLDPC code ensembles.

Let $k^{(\ell)} \in \mathbb{N}^+$, $\ell = 1, \ldots, \nu E/K$, be the number of rows in the parity-check matrix associated with the component code of the ℓ-th GC node.

Lemma 1: The design rate $R(\nu)$ of the $\mathcal{C}_{J,K,\nu}$ ensemble is

$$R(\nu) = R_0 - \nu(1 - R_0)(k_{avg} - 1), \qquad (4)$$

where $k_{avg} \triangleq (\nu \frac{E}{K})^{-1} \sum_{\ell=1}^{\nu \frac{E}{K}} k^{(\ell)}$ is the average number of rows in the parity-check matrix of the component codes.

Proof: Any SPC node in the Tanner graph accounts for a single row in the parity-check matrix of the GLDPC code, and any GC node accounts for $k^{(\ell)}$ rows. Thus, the design rate $R(\nu)$ is given by

$$R(\nu) = 1 - \frac{(1-\nu)\frac{E}{K} + \sum_{\ell=1}^{\nu \frac{E}{K}} k^{(\ell)}}{n} = 1 - \frac{(1-\nu)\frac{E}{K} + \nu \frac{E}{K}k_{avg}}{E/J} = R_0 - \nu(1-R_0)(k_{avg} - 1). \quad (5)$$

∎

Note that the second term in (4) accounts for the rate loss at GC nodes. When the component codes are linear block codes with minimum distance d, we obtain the following bounds on $R(\nu)$:

Lemma 2: If all component codes in the $\mathcal{C}_{J,K,\nu}$ ensemble are linear block codes with minimum distance $d > 2$, then

$$R(\nu) \leq R_0 - \nu(1 - R_0) \log_2 \left(\frac{1}{2} \sum_{q=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{K}{q} \right). \qquad (6)$$

Furthermore, there exists a set of linear block codes to be used as component codes such that

$$R(\nu) \geq R_0 - \nu(1 - R_0) \left\lceil \log_2 \left(\frac{1}{2} + \frac{1}{2} \sum_{q=0}^{d-2} \binom{K-1}{q} \right) \right\rceil. \qquad (7)$$

Here, we use $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ to denote the ceiling and floor functions, respectively. The two bounds coincide, for example, when $d = 3$ and $r = 2^z - 1$, where $z \in \mathbb{Z}_+$.

Proof: First, the condition $d > 2$ is required to differentiate between the rate loss at SPC nodes, which are block codes with minimum distance 2, and at GC nodes. We start by proving the converse bound in (6). By the sphere-packing bound [15, Theorem 12, p.531], any component code with blocklength K and minimum distance d must satisfy

$$2^{K-k} \leq \frac{2^K}{\sum_{q=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{K}{q}}, \qquad (8)$$

where k is the number of rows in the parity-check matrix. This implies that the term $(k_{avg} - 1)$ in (4) is bounded by

$$k_{avg} \geq \log_2 \left(\sum_{q=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{K}{q} \right), \qquad (9)$$

which proves (6). Regarding the achievable bound in (7), the Varshamov Bound [16, Theorem 2.9.3] guarantees the existence of a linear component code with blocklength K and minimum distance at least d if

$$2^{K-\mathtt{k}} \geq 2^{\mathtt{n} - \left\lceil \log_2 \left(1 + \sum_{q=0}^{\mathtt{d}-2} \binom{K-1}{q} \right) \right\rceil}. \tag{10}$$

If the above condition is satisfied, then there exists a set of linear block codes to be used as component codes with blocklength K and minimum distance at least d such that

$$\mathtt{k}_{\mathrm{avg}} - 1 \leq \left\lceil \log_2 \left(\frac{1}{2} + \frac{1}{2} \sum_{q=0}^{\mathtt{d}-2} \binom{K-1}{q} \right) \right\rceil, \tag{11}$$

which proves (7).

Finally, if we substitute $\mathtt{d} = 3$ and $K = 2^z - 1$ for some $z \in \mathbb{Z}_+$ into (6) and (7), a straightforward computation shows that the converse bound in (6) can be simplified to

$$\mathtt{R}(\nu) \leq \mathtt{R}_0 - \nu(1 - \mathtt{R}_0)(z - 1), \tag{12}$$

and, likewise, the achievable bound in (7) simplifies to

$$\mathtt{R}(\nu) \geq \mathtt{R}_0 - \nu(1 - \mathtt{R}_0)(z - 1). \tag{13}$$

∎

C. Growth rate of the weight distribution of the $\mathcal{C}_{J,K,\nu}$ ensemble

A useful tool for analysis and design of LDPC codes and their generalizations is the asymptotic exponent of the weight distribution. The growth rate of the weight distribution was introduced in [20] to show that the minimum distance of a randomly-generated regular LDPC code with a variable nodes of degree of at least three is a linear function of the codeword length with high probability. The growth rate of the weight distribution for a class of doubly generalized LDPC (D-GLDPC) codes was introduced in [10]. The $\mathcal{C}_{J,K,\nu}$ GLDPC code ensemble can be seen as a particular instance of the codes analyzed in that work. The *weight spectral shape* of the $\mathcal{C}_{J,K,\nu}$ ensemble captures the behavior of codewords whose weight is linear in the block length n and is defined by

$$G(\alpha) \triangleq \lim_{\mathtt{n} \to \infty} \frac{1}{\mathtt{n}} \log \mathbb{E}_{\mathcal{C}_{J,K,\nu}}[X_{\alpha\mathtt{n}}] \tag{14}$$

for $\alpha > 0$, where X_w denotes the number of codewords of weight-w of a randomly chosen code in the $\mathcal{C}_{J,K,\nu}$ code ensemble. This limit assumes the inclusion of only those positive integers for which $\alpha\mathtt{n} \in \mathbb{Z}$. We define the *critical exponent codeword weight ratio* as $\hat{\alpha} \triangleq \inf\{\alpha \geq 0 | G(\alpha) \geq 0\}$. If $\hat{\alpha} > 0$, then the code's minimum distance asymptotically grows as $\mathcal{O}(\hat{\alpha}\mathtt{n})$ and the ensemble is said to have good growth

rate behavior. If $\hat{\alpha} = 0$, then the minimum distance of the code may still grow with the block length $\mathtt{n}$ but at a slower rate, e.g., as $\mathcal{O}(\log(\mathtt{n}))$.

Lemma 3: If all component codes in the $\mathcal{C}_{J,K,\nu}$ ensemble are linear block codes with minimum distance $\mathtt{d} > 2$, then $\hat{\alpha} > 0$ for $J > 2$. For $J = 2$, $\hat{\alpha} > 0$ if and only if

$$\nu > \frac{K-2}{K-1} \triangleq \hat{\nu}. \tag{15}$$

Otherwise, $\hat{\alpha} = 0$.

Proof: The lemma follows directly by particularizing the results in [10] [Section II] to the $\mathcal{C}_{J,K,\nu}$ ensemble. ∎

III. PROBABILISTIC PEELING DECODING OVER THE BEC

Suppose we use a random sample of the $\mathcal{C}_{J,K,\nu}$ ensemble to transmit over a BEC(ϵ). For this channel, each of the $\mathtt{n}$ coded bits is erased with probability ϵ. Without loss of generality, we assume that the all-zero codeword is transmitted, hence the received vector $\boldsymbol{y}$ belongs to the set $\{0, ?\}^{\mathtt{n}}$, where ? denotes an erasure. Let $\Gamma_{\boldsymbol{y}} \subseteq \{1, \ldots, \mathtt{n}\}$ be the index set of the bits correctly received, namely $y_i = 0$ for all $i \in \Gamma_{\boldsymbol{y}}$. Decoding will be performed using a generalization of the PD algorithm [11] similar to that proposed for GLDPC codes in [13]. The final formulation of the decoding algorithm depends on the decoding capabilities we assume at GC nodes. For instance, if we assume BD decoding at component codes, then the generalized PD algorithm, denoted as BD-PD, proceeds as described in Algorithm 1.

Algorithm 1 BD-PD

Remove from the Tanner graph of the GLDPC code all variable nodes with indexes in $\Gamma_{\boldsymbol{y}}$.

Construct Ψ, the index set of check nodes that correspond to either degree-one SPC nodes or GC nodes of degree less or equal to $\mathtt{d} - 1$.

repeat

    1) Select at random a member of Ψ.

    2) Remove from the Tanner graph the check node with the index drawn in Step 1). Further, remove all connected variable nodes, and all attached edges.

    3) Update Ψ.

until All variable nodes have been removed (successful decoding) or $\Psi = \emptyset$ (decoding failure).

BD-PD is a suboptimal decoding method that considers decodable all GC nodes up to degree $\mathtt{d} - 1$ [9], [21]. However, it ignores the fact that any component code will be able to decode a certain fraction

of erasure patterns of weight equal to or greater than d. As already reported in various works, e.g., [8], [13], the GLPDC code performance dramatically improves if we consider ML decoding at GC nodes. In principle, to consider ML decoding at GC nodes, we have to specify a full list of decodable erasure patterns and, label each of the incoming edges at every GC node to differentiate between decodable and non-decodable GC nodes. As shown in [13], incorporating this labelling into the asymptotic analysis requires the use of multi-edge type DDs.

Algorithm 2 P-PD

Remove from the Tanner graph of the GLDPC code all variable nodes with indexes in $\Gamma_{\boldsymbol{y}}$.

for all GC nodes **do**

  If the GC has degree w, tag the check node as *decodable* with probability p_w.

end for

Construct Ψ, the index set of check nodes corresponding to either degree-one SPC nodes or GC nodes tagged as *decodable*.

repeat

  1) Select at random a member of Ψ.

  2) Remove from the Tanner graph the check node with the index drawn in Step 1). Further remove all connected variable nodes and all attached edges.

  3)

  for every non-decodable GC node that has lost one or more edges **do**

    If the GC has degree w, tag the check node as *decodable* with probability p_w.

  end for

  4) Update Ψ.

until All variable nodes have been removed (successful decoding) or $\Psi = \emptyset$ (decoding failure).

In order to incorporate beyond-BD decoding at GC nodes into our analysis, and at the same time maintain a formulation compatible with the random definition of the $\mathcal{C}_{J,K,\nu}$ ensemble, we will further constrain the family of component codes to be used at degree-K GC nodes. More specifically, we assume that the fraction of ML-decodable weight-w erasure patterns at every GC node is given by some $p_w \in [0,1], w = 1, \ldots, K$. Thus, the family of component codes under analysis is the family of blocklength-K linear block codes with minimum distance d and with decoding profile described by the vector $\mathbf{p} = (p_1, \ldots, p_K)$. Note that if the minimum distance of the component code is d, then $p_w = 1$ for $w \leq \mathtt{d} - 1$.

By exploiting the fact that incoming edges at every GC node are assigned to each position of the component code uniformly at random, we can incorporate ML-decoded GC nodes into the PD as shown in Algorithm 2, denoted as *probabilistic PD (P-PD)*. Observe that the key P-PD feature is to tag GC check nodes as decodable with probabilities given by **p** only when they lose one or more edges, which may happen either at the initialization or after a connected variable is removed. If only one decodable check node is removed per iteration, after every P-PD iteration only a few GC nodes can change its state (from non-decodable to decodable). Thus, at every iteration, P-PD emulates the ML decoding operation of a degree-w GC node by drawing the decoding capability according to a Bernoulli distribution with parameter p_w, $w \in \{1, \ldots, K\}$. Note that P-PD is a procedure that allows for simpler analysis rather than a practical decoding algorithm. Further, note that we recover the bounded distance PD (BD-PD) algorithm from P-PD if we set $p_w = 0$ for $w \geq \mathtt{d}$ and $p_w = 1$ otherwise.

The bounds on $\mathtt{R}(\nu)$, predicted in Lemma 2, could in principle be refined according to **p**. While this is an interesting open question, we will later show that the bounds are tight in certain scenarios and there is little room for refinement.

A. Comparing the P-PD and ML-PD performances by Monte Carlo simulation

If we select a specific component code, we can compare the simulation performance of the $\mathcal{C}_{J,K,\nu}$ ensemble for the corresponding parameters under P-PD with that of the practical GLDPC codes with GC nodes that are decoded via ML, using the actual parity-check matrix of the component codes. We refer to this latter case as ML-PD.

More precisely, for a given finite blocklength $\mathtt{n}$, fixed $\nu \in [0, 1]$, and base DD, we generate a member of the $\mathcal{C}_{J,K,\nu}$ ensemble as follows:

1) Generate at random a Tanner graph according to the (J, K) base DD. Then, select at random a fraction ν of check nodes to be used as GC nodes. Overall, the graph contains $\mathtt{n}$ variable nodes, $\nu \mathtt{E}/K$ GC nodes and $(1 - \nu)\mathtt{E}/K$ SPC nodes.

2) For each of the $\nu \mathtt{E}/K$ GC nodes, we generate uniformly at random a permutation of the set $\{1, 2, \ldots, K\}$, which is used to associate each of the incoming edges to the GC node to a position in the component code.

We estimate by Monte Carlo simulation the bit error rate (BER) over the BEC achieved by both P-PD, which follows Algorithm 2, and ML-PD, which uses a look-up table of decodable erasure patterns. In Fig. 2 (a), we plot the BER as a function of the channel erasure probability of P-PD and ML-PD for a $(2, 6)$-regular base DD with a rate-$1/2$ Hamming $(6, 3)$ linear block code as component code. In Fig. 2 (b), we plot the same quantities for a $(2, 8)$-regular base DD using a rate-$1/2$ $(8, 4)$ Hamming component code.
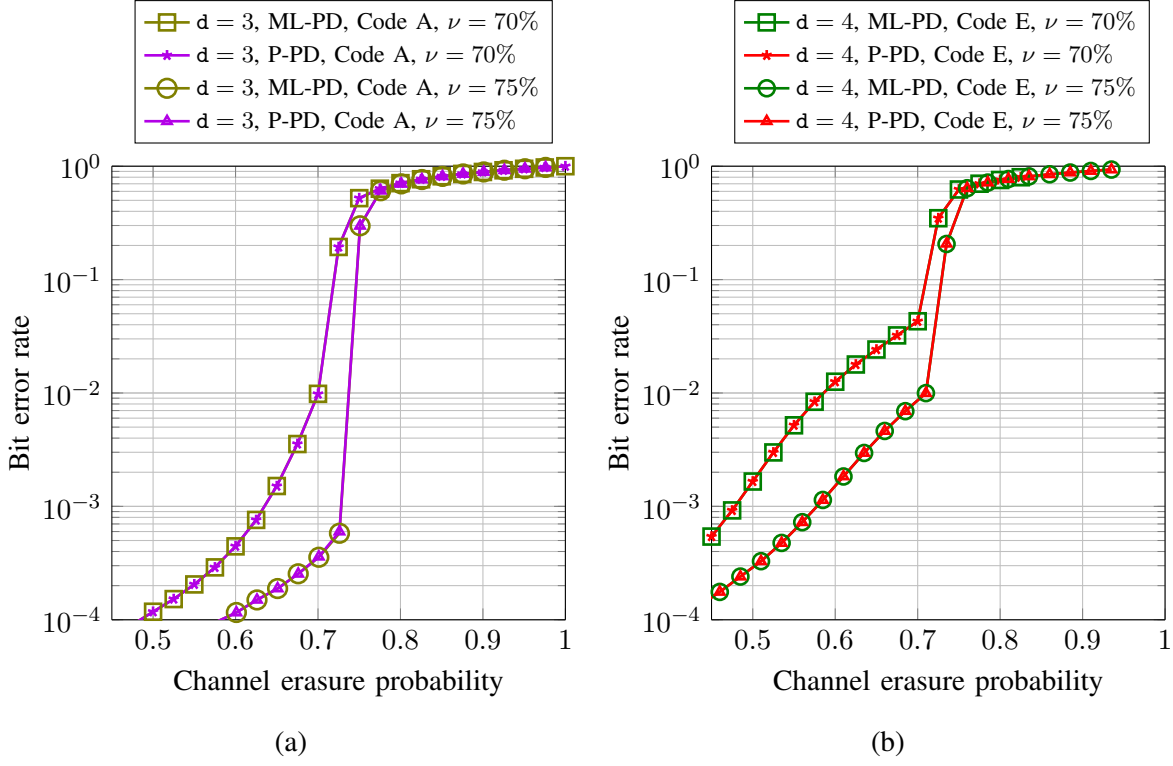
Fig. 2. In Fig. 2 (a), we plot the BER as a function of the channel erasure probability for a $(2,6)$ base DD and a rate-$1/2$ Hamming $(6,3)$ linear block code as component code. In Fig. 2 (b), we plot the BER as a function of the channel erasure probability for a $(2,8)$ base DD and a rate-$1/2$ $(8,4)$ Hamming component code. Results have been averaged over 10 generated samples from the $\mathcal{C}_{J,K,\nu}$ ensemble with a blocklength of 10000 bits.

Results have been averaged over 10 generated samples from the $\mathcal{C}_{J,K,\nu}$ ensemble. Observe the perfect match between the BERs for P-PD and ML-PD in all cases. This illustrates that we are not sacrificing accuracy with the probabilistic description of the decoder, as long as GLDPC codes are generated as described above.

IV. ASYMPTOTIC ANALYSIS

The P-PD decoder yields a sequence of residual graphs by sequentially removing degree-one SPC nodes and decodable GC nodes from the GLDPC Tanner graph. Our next goal is to predict the asymptotic behaviour of the $\mathcal{C}_{J,K,\nu}$ ensemble under P-PD by extending the methodology proposed in [11] to analyze the asymptotic behavior of LDPC ensembles under PD. In [11], it is shown that if we apply the PD to elements of an LDPC ensemble, then the expected DD of the sequence of residual graphs can be described as the solution of a set of differential equations. Furthermore, the deviation of the process w.r.t. the expected evolution decreases exponentially fast with the LDPC blocklength. This analysis is based on a result on the evolution of Markov processes due to Wormald [22]. The proof that the GLDPC

asymptotic graph evolution under P-PD can be predicted using the same result is given in Appendix A. In this section, we introduce the notation used to characterize the DDs of the residual Tanner graphs of GLDPC ensembles with P-PD decoding and then present the system of differential equations that describes the asymptotic GLDPC graph evolution.

In order to characterize the DDs of the residual Tanner graphs of GLDPC ensembles is to augment the DD notation introduced in Section II to differentiate between GC nodes that have been tagged as decodable and those tagged as non-decodable. In order to simplify the formulation, we restrict ourselves to the case $p_w = 0$ for $w \geq \mathtt{d} + 2$, i.e., we consider component codes can only decode a certain fraction of erasure patterns of degrees $\mathtt{d}$ and $\mathtt{d} + 1$ and all erasure patterns of degree below $\mathtt{d}$. This may not be an strong assumption. After exhaustive search of short linear block component codes (blocklengths up to 15 bits), we have not found any component code with $p_w > 0$ for $w \geq \mathtt{d} + 2$. In any case, the analysis provided here directly generalizes to any arbitrary p_w.

As introduced in Section II, any edge adjacent to a degree i variable node is said to have left degree i. Similarly, any edge adjacent to a degree j SPC (GC) node is said to have right SPC (GC) degree j. Given the residual graph at the ℓ-th iteration of the P-PD algorithm, let $L_i^{(\ell)}$ denote the number of edges with left degree i at iteration ℓ, $i = 1, ..., J$. Similarly, let $R_{pj}^{(\ell)}$ denote the number of edges with right SPC degree j and $R_{cj}^{(\ell)}$ denote the number of edges with right GC degree j at iteration ℓ, $j = 1, ..., K$. For $j \in \{\mathtt{d}, \mathtt{d} + 1\}$, we split $R_{cj}^{(\ell)}$ into two terms, $\hat{R}_{cj}^{(\ell)}$ and $\bar{R}_{cj}^{(\ell)}$, where $\hat{R}_{cj}^{(\ell)}$, $j \in \{\mathtt{d}, \mathtt{d} + 1\}$ denotes the number of edges with right GC degree j connected to GC nodes tagged as *decodable*, and $\bar{R}_{cj}^{(\ell)}$ denotes the number of edges with right GC degree j connected to GC nodes tagged as *not-decodable*. Clearly, we have $R_{cj}^{(\ell)} = \hat{R}_{cj}^{(\ell)} + \bar{R}_{cj}^{(\ell)}, j = \mathtt{d}, \mathtt{d} + 1$. Recall that $\mathtt{E}$ denotes the number of edges in the original GLPDC graph.

Theorem 4: Consider a BEC with erasure probability ϵ and assume we use elements of the $\mathcal{C}_{\overline{\lambda}, \overline{\rho}_p, \overline{\rho}_c, \nu}$ code ensemble for transmission. If we use P-PD with parameters $(\mathtt{d}, p_\mathtt{d}, p_{\mathtt{d}+1})$, in the limit as $\mathtt{E} \to \infty$ the DD of the residual graph at iteration ℓ can be computed as follows:

$$\lim_{\mathtt{E}\to\infty} L_i^{(\ell)} = \mathtt{E}l_i^{(\ell/\mathtt{E})}, \quad i \in \{1, \dots, J\} \tag{16}$$

$$\lim_{\mathtt{E}\to\infty} R_{pj}^{(\ell)} = \mathtt{E}r_{pj}^{(\ell/\mathtt{E})}, \quad j \in \{1, \dots, K\} \tag{17}$$

$$\lim_{\mathtt{E}\to\infty} R_{cj}^{(\ell)} = \mathtt{E}r_{cj}^{(\ell/\mathtt{E})}, \quad j \in \{1, \dots, K\} \text{ and } j \notin \{\mathtt{d}, \mathtt{d} + 1\} \tag{18}$$

$$\lim_{\mathtt{E}\to\infty} \hat{R}_{cj}^{(\ell)} = \mathtt{E}\hat{r}_{cj}^{(\ell/\mathtt{E})}, \quad j \in \{\mathtt{d}, \mathtt{d} + 1\} \tag{19}$$

$$\lim_{\mathtt{E}\to\infty} \bar{R}_{cj}^{(\ell)} = \mathtt{E}\bar{r}_{cj}^{(\ell/\mathtt{E})}, \quad j \in \{\mathtt{d}, \mathtt{d} + 1\} \tag{20}$$

where $l_i^{(\tau)}, r_{pj}^{(\tau)}, r_{cj}^{(\tau)}, \hat{r}_{cj}^{(\tau)}, \bar{r}_{cj}^{(\tau)} \ \tau \in [0, \sum_{i=1}^{J} l_i^{(\tau)}/i]$ are the solutions to the following system of differential equations:

$$\frac{\mathrm{d}l_i^{(\tau)}}{\mathrm{d}\tau} = -\frac{il_i^{(\tau)}}{e^{(\tau)}}\left(P_{p1}^{(\tau)} + \sum_{w=1}^{\mathrm{d}+1} wP_{cw}^{(\tau)}\right),\tag{21}$$

$$\frac{\mathrm{d}r_{pj}^{(\tau)}}{\mathrm{d}\tau} = P_{p1}^{(\tau)}\left((r_{p(j+1)}^{(\tau)} - r_{pj}^{(\tau)})\frac{j(a^{(\tau)}-1)}{e^{(\tau)}} - \mathbb{I}[j=1]\right)$$

$$+ \sum_{w=1}^{\mathrm{d}+1} P_{cw}^{(\tau)}(r_{p(j+1)}^{(\tau)} - r_{pj}^{(\tau)})\frac{jw(a^{(\tau)}-1)}{e^{(\tau)}},\tag{22}$$

$$\frac{\mathrm{d}r_{cj}(\tau)}{\mathrm{d}\tau} = P_{p1}^{(\tau)}\left((r_{c(j+1)}^{(\tau)} - r_{cj}^{(\tau)})\frac{j(a^{(\tau)}-1)}{e^{(\tau)}}\right)$$

$$+ \sum_{w=1}^{\mathrm{d}+1} P_{cw}^{(\tau)}\left((r_{c(j+1)}^{(\tau)} - r_{cj}^{(\tau)})\frac{jw(a^{(\tau)}-1)}{e^{(\tau)}} - w\mathbb{I}[j=w]\right), \quad j \notin \{\mathrm{d}, \mathrm{d}+1\}\tag{23}$$

$$\frac{\mathrm{d}\hat{r}_{cj}^{(\tau)}}{\mathrm{d}\tau} = P_{p1}^{(\tau)}\left((p_j\bar{r}_{c(j+1)}^{(\tau)} + \hat{r}_{c(j+1)}^{(\tau)} - \hat{r}_{cj}^{(\tau)})\frac{j(a^{(\tau)}-1)}{e^{(\tau)}}\right)$$

$$+ \sum_{w=1}^{j+1} P_{cw}^{(\tau)}\left((p_j\bar{r}_{c(j+1)}^{(\tau)} + \hat{r}_{c(j+1)}^{(\tau)} - \hat{r}_{cj}^{(\tau)})\frac{jw(a^{(\tau)}-1)}{e^{(\tau)}} - w\mathbb{I}[w=j]\right), \quad j \in \{\mathrm{d}, \mathrm{d}+1\}\tag{24}$$

$$\frac{\mathrm{d}\bar{r}_{cj}^{(\tau)}}{\mathrm{d}\tau} = P_{p1}^{(\tau)}\left(((1-p_j)\bar{r}_{c(j+1)}^{(\tau)} - \bar{r}_{cj}^{(\tau)})\frac{j(a^{(\tau)}-1)}{e^{(\tau)}}\right)$$

$$+ \sum_{w=1}^{j+1} P_{cw}^{(\tau)}\left(((1-p_j)\bar{r}_{c(j+1)}^{(\tau)} - \bar{r}_{cj}^{(\tau)})\frac{jw(a^{(\tau)}-1)}{e^{(\tau)}} - w\mathbb{I}[w=j]\right), \quad j \in \{\mathrm{d}, \mathrm{d}+1\}\tag{25}$$

In (21)-(25), $\mathbb{I}[\cdot]$ denotes the indicator function, and

$$e^{(\tau)} = \sum_{i=1}^{J} l_i^{(\tau)} = \sum_{j=1}^{K}[r_{pj}^{(\tau)} + r_{cj}^{(\tau)}],\tag{26}$$

$$a^{(\tau)} = \sum_i il_i^{(\tau)}/e^{(\tau)},\tag{27}$$

$$P_{p1}^{(\tau)} = \frac{r_{p1}^{(\tau)}}{s^{(\tau)}},\tag{28}$$

$$P_{cj}(\tau) = \begin{cases} \dfrac{r_{cj}^{(\tau)}/j}{s^{(\tau)}} & j < \mathrm{d} \\[3mm] \dfrac{\hat{r}_{cj}^{(\tau)}/j}{s^{(\tau)}} & j \in \{\mathrm{d}, \mathrm{d}+1\} \end{cases}\tag{29}$$

$$s^{(\tau)} = r_{p1}^{(\tau)} + \sum_{w=1}^{\mathrm{d}-1} \frac{r_{cw}^{(\tau)}}{w} + \frac{\hat{r}_{cd}^{(\tau)}}{\mathrm{d}} + \frac{\hat{r}_{c(\mathrm{d}+1)}^{(\tau)}}{\mathrm{d}+1}.\tag{30}$$

The initial conditions of the system of differential equations (21)-(25) are given by

$$l_i^{(0)} = \epsilon \lambda_i, \tag{31}$$

$$r_{pj}^{(0)} = \sum_{\alpha \geq j} \rho_{p\alpha} \binom{\alpha - 1}{j - 1} \epsilon^j (1 - \epsilon)^{\alpha - j}, \tag{32}$$

$$r_{cj}^{(0)} = \sum_{\alpha \geq j} \rho_{c\alpha} \binom{\alpha - 1}{j - 1} \epsilon^j (1 - \epsilon)^{\alpha - j}, \tag{33}$$

$$\hat{r}_{c\nu}^{(0)} = p_\nu r_{c\nu}^{(0)}, \tag{34}$$

$$\bar{r}_{c\nu}^{(0)} = (1 - p_\nu) r_{c\nu}^{(0)} \tag{35}$$

for $i = 1, \ldots J$, $j = 1, \ldots, K$, and $\nu = \mathtt{d}, \mathtt{d} + 1$.

Proof: See Appendix A. ∎

Using Theorem 4, we can predict the P-PD threshold for the $\mathcal{C}_{J,K,\nu}$ code ensemble by setting $\lambda_i = \mathbb{I}[i = J]$ in (31), $\rho_{p\alpha} = (1 - \nu)\mathbb{I}[\alpha = K]$ in (32), and $\rho_{c\alpha} = \nu \mathbb{I}[\alpha = K]$ in (33). We then numerically search for the highest ϵ value for which the function $r_{p1}^{(\tau)} + \sum_{w=1}^{\mathtt{d}-1} r_{cw}^{(\tau)}/w + \hat{r}_{c\mathtt{d}}^{(\tau)}/\mathtt{d} + \hat{r}_{c(\mathtt{d}+1)}^{(\tau)}/(\mathtt{d} + 1)$ remains strictly positive for any $\tau \in [0, \sum_{i=1}^{J} l_i^{(\tau)}/i]$ such that $e^{(\tau)} > 0$.

V. ANALYSIS OF THE $\mathcal{C}_{J,K,\nu}$ ENSEMBLE UNDER P-PD

In this section, we study the asymptotic performance of the $\mathcal{C}_{J,K,\nu}$ ensemble for different base DDs as we vary the fraction ν of GC nodes in the graph. We use high rate base DDs that correspond to regular LDPC code ensembles with variable degree equal to $J = 2$. Further examples with $J > 2$ are discussed in Section VIII. We summarize the parameter of the base DD considered here in Table I. We denote by ϵ_0 the PD threshold of the base LDPC ensemble. Recall that $p_w = 1$ for $w \leq \mathtt{d} - 1$ and $p_w = 0$ for $w \geq \mathtt{d} + 2$. In order to determine $p_\mathtt{d}, p_{\mathtt{d}+1}$, we performed an exhaustive search over the database [23], [24], which implements MAGMA [25] to design block codes with the largest minimum distance. For every K, we search for the code with the largest minimum distance $\mathtt{d}$, and we use the corresponding $p_\mathtt{d}$ and $p_{\mathtt{d}+1}$ parameters. Like this, we ensure that there exists at least one linear block code that satisfies these requirements. We use this specific block code as the reference of a family of linear block codes with the same decoding capabilities. The values found are listed in Table II and used as a reference for a whole family of linear block codes. The corresponding reference block codes are listed in Appendix C. Note that despite having different blocklength and rate, many reference block codes share the same $p_\mathtt{d}$, $p_{\mathtt{d}+1}$ parameters.

We construct $\mathcal{C}_{J,K,\nu}$ ensembles by combining various base DDs with the component code families summarized in Table II. For each code ensemble, we compute the P-PD threshold ϵ^* as a function of ν.

TABLE I

BASE DDS, THEIR DESIGN RATES AND ITERATIVE DECODING THRESHOLDS UNDER PD

Base DD	K	$\mathtt{R}_0$	ϵ_0	Gap to capacity $(1 - \mathtt{R}_0 - \epsilon_0)$
$(2,6)$-regular	6	2/3	0.206	0.127
$(2,7)$-regular	7	5/7	0.167	0.119
$(2,8)$-regular	8	3/4	0.147	0.103
$(2,15)$-regular	15	13/15	0.071	0.062

TABLE II

FAMILIES OF COMPONENT LINEAR BLOCK CODES.

Code Family Index	blocklength K	$\mathtt{d}$	$p_{\mathtt{d}}$	$p_{\mathtt{d}+1}$
I	6	3	0.8	0
II	6	4	0.8	0
III	7	3	0.8	0
IV	7	4	0.8	0
V	8	4	0.8	0
VI	8	4	0.9143	0.5714
VII	8	5	0.9643	0.75
VIII	15	3	0.9231	0.6154
IX	15	4	0.9231	0.6154

A. Results for $(2,6)$ and $(2,7)$ base DDs

Fig. 3 shows the computed P-PD threshold ϵ^* of the $\mathcal{C}_{J,K,\nu}$ ensemble for a base DD $(2,6)$-regular as a function of ν. We consider GC nodes with minimum distance $\mathtt{d}$ equal to 3 and 4 and parameters given by Families I and II in Table II. We also include the BD-PD threshold, which only depends on the minimum distance $\mathtt{d}$ of the component codes and can be computed by solving the system of differential equations in Theorem 4 by setting $p_{\mathtt{d}} = p_{\mathtt{d}+1} = 0$. First of all, observe that the P-PD gains in threshold w.r.t. BD-PD are only significant for large values of ν. Furthermore, for both P-PD and BD-PD, using component codes with larger minimum distance ($\mathtt{d} = 4$ instead of $\mathtt{d} = 3$) pays off only for very large values of ν.

Since increasing ν also modifies the code rate $\mathtt{R}(\nu)$ in (4), the comparison in Fig. 3 can be misleading, as we cannot directly evaluate the distance to the channel capacity. In fact, not all values of ν are achievable, since they would give rise to a negative rate $\mathtt{R}(\nu)$. We overcome this issue by directly comparing the asymptotic threshold and code rate, both defined as parametric curves w.r.t. ν. Denote by $\epsilon^*(\nu)$ the
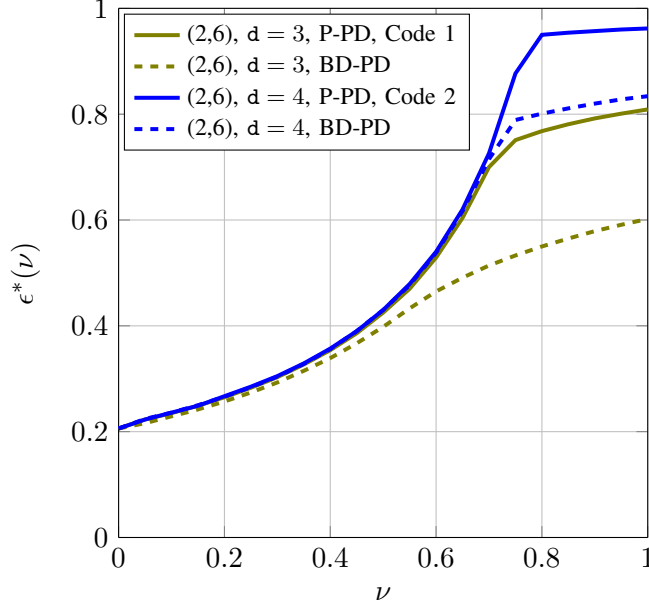
Fig. 3. P-PD and BD-PD thresholds as a function of ν for the $(2,6)$ base DD.

threshold ϵ^* as a function of $\nu \in [0,1]$. From Fig. 3 we see that $\epsilon^*(\nu)$ is a continuous, strictly increasing function of ν and that for $\nu = 0$ its value is equal to ϵ_0, the threshold of the base LDPC ensemble. The inverse of this function, which can be obtained numerically, is denoted by $\nu(\epsilon^*)$ and provides the minimum fraction of GC nodes in the graph required to achieve an ensemble threshold at least ϵ^*. Given the function $\nu(\epsilon^*)$ described above, we use Lemma 2 to determine bounds on $\texttt{R}(\nu)$ for a given targeted decoding threshold ϵ^*. More precisely, by using $\nu(\epsilon^*)$ in (6), we obtain a *converse bound* on the coding rate required to achieve a P-PD decoding threshold equal to ϵ^* using component codes with minimum distance $\texttt{d}$. Similarly, using $\nu(\epsilon^*)$ in (7), we obtain an *achievable bound* on the coding rate required to achieve a P-PD decoding threshold equal to ϵ^* using linear component codes with minimum distance $\texttt{d}$. We proceed along the same lines to obtain bounds on the $\mathcal{C}_{J,K,\nu}$ rate for the BD-PD thresholds.

In Fig. 4 (a) we plot these bounds as a function of ϵ^*, both for P-PD and BD-PD, using Code Family I component codes with minimum distance $\texttt{d} = 3$. In Fig. 4 (b), we show the gap to channel capacity computed for each case, and indicate the threshold $\epsilon^*(\hat{\nu})$ with $\hat{\nu}$ given in (15). Since $\epsilon^*(\nu)$ is monotonically increasing in ν, any configuration with threshold larger than $\epsilon^*(\hat{\nu})$ has a minimum distance that grows linearly with the block length $\texttt{n}$. Observe that the performance of both BD-PD and P-PD overlaps for coding rates close to the original rate of the base DD, i.e., for small values of ν. However, as $\epsilon^*(\nu)$ increases, P-PD significantly outperforms BD-PD. Furthermore, there are values of ν for which the gap to capacity of P-PD is smaller than that for the base LDPC ensemble under PD. For the $(2,6)$ base DD,
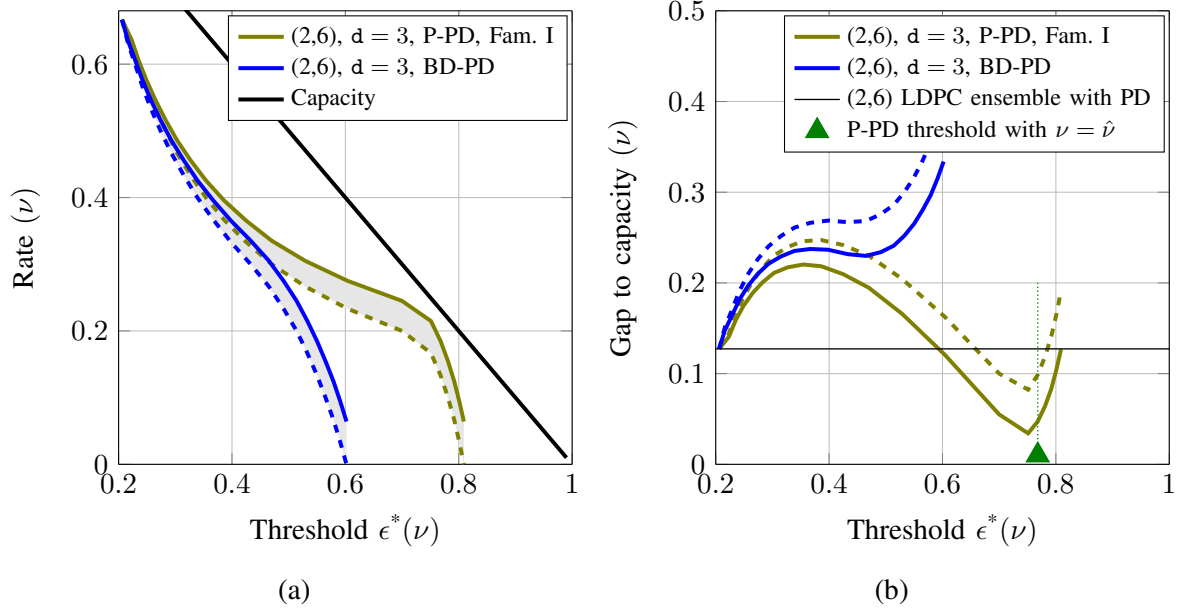
Fig. 4. In Fig. 4 (a), we plot the bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for the base DD $(2,6)$ and component codes of minimum distance $\mathtt{d} = 3$ as a function of the P-PD and BD-PD thresholds. In Fig. 4 (b), we show their gap to channel capacity. We also indicate the P-PD threshold for $\nu = \hat{\nu}$.

the minimum gap to capacity of P-PD, measured using the achievable rate bound, is 0.0823 for a coding rate of 0.1667. For $\nu = \hat{\nu}$, the gap to capacity grows to 0.0987 but it is still below the base LDPC gap to capacity, which is 0.1273 according to Table I. Thus, for ν slightly above $\hat{\nu}$ we are able to reduce the original gap to capacity and at the same time obtain a good ensemble from minimum distance point of view. Observe also that the region where the $\mathcal{C}_{J,K,\nu}$ ensemble outperforms the base LDPC ensemble is very narrow, and it does not include the case where all check nodes are GC nodes $(\nu = 1)$.

Fig. 5 reproduces the results for the Code Family II with minimum distance $\mathtt{d} = 4$. However, in this case the two bounds are loose and it is uncertain whether we can find an specific block component code in the family that is able to operate close to the converse bound. In fact, for large $\epsilon^*(\nu)$, the P-PD converse bound coincides with the capacity. Furthermore, the bounds for P-PD and BD-PD overlap in a large region despite the fact that P-PD using component codes from Family II resolves degree-$\mathtt{d}$ erasure patterns with high probability (0.8).

In Fig. 6 we show the asymptotic behaviour of the $\mathcal{C}_{J,K,\nu}$ ensemble constructed using a $(2,7)$ base DD with $\mathtt{d} = 3$ component codes. As predicted by Lemma 2, when using component codes of blocklength $r = 7$ with minimum distance $\mathtt{d} = 3$, the converse and achievable bound on the $\mathcal{C}_{J,K,\nu}$ coding rate coincide. Thus, the existence of a linear block component code that satisfies the properties of Code Family III and for which the $\mathcal{C}_{J,K,\nu}$ ensemble asymptotically achieves the results in Fig. 6 is guaranteed.
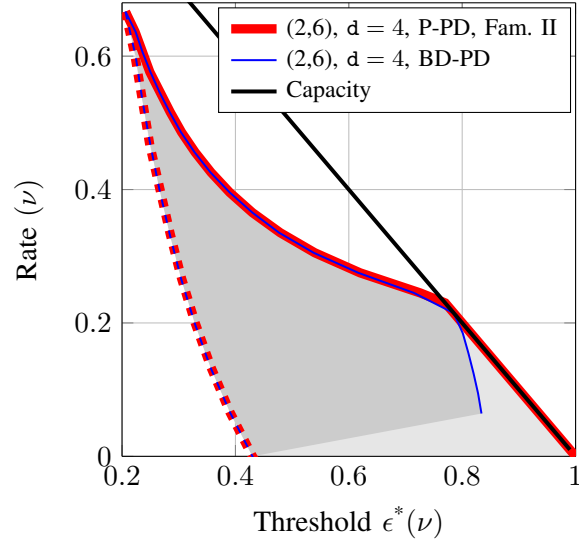
Fig. 5. Bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for a base DD $(2,6)$ and $\mathtt{d} = 4$ component codes as a function of the P-PD and BD-PD thresholds.

Again, there is a region where the gap to capacity of P-PD can be reduced with respect to that of the base LDPC ensemble.
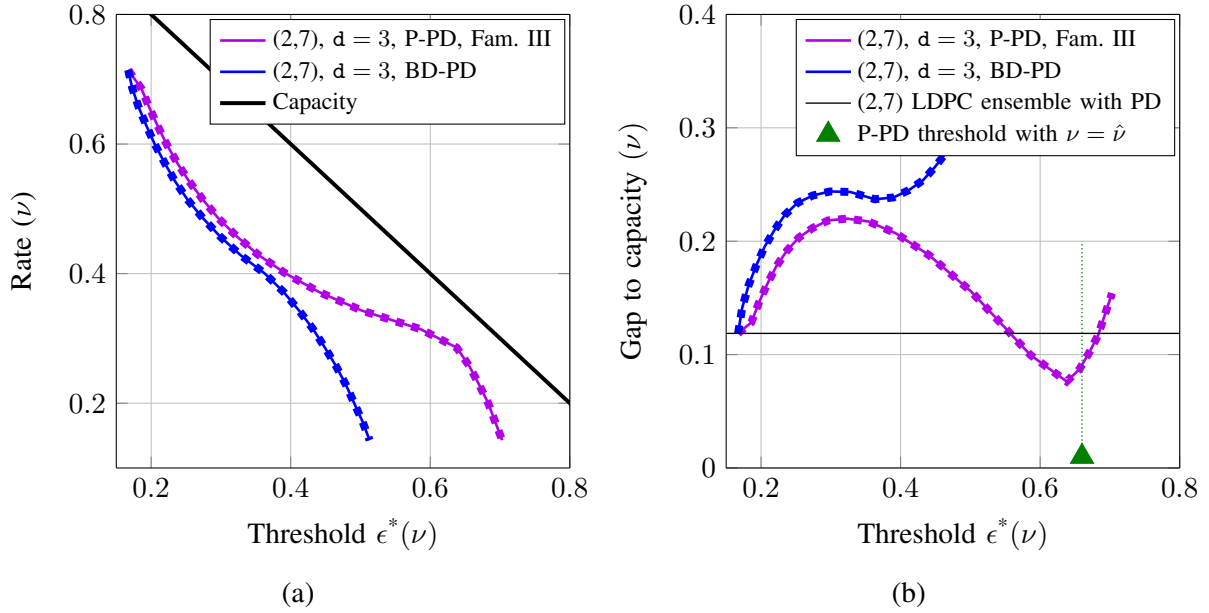

(a)

(b)

Fig. 6. In (a), we plot the bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for a base DD $(2,7)$ as a function of the P-PD and BD-PD thresholds. Note that the bounds overlap in this case. In (b), we show the gap to channel capacity for each case. We also indicate the P-PD threshold for $\nu = \hat{\nu}$.

B. Results for higher-density base DDs

We finish this section by extending the above results to base DDs with higher check degree and, thus higher ensemble density. In Fig. 7(a), we show the asymptotic behavior of the $\mathcal{C}_{J,K,\nu}$ ensemble constructed using a $(2,8)$ base DD with component codes in Code Families V, VI and VII (See Table II). Observe first that the rate bounds for Code Families V and VI coincide, even though Code Family VI has better decoding capabilities. In both cases the bounds are loose, but we can still observe a significant improvement w.r.t. the Code Family VII, which has but very large ($\mathtt{d} = 5$) minimum distance and, hence, and small coding rate. This again demonstrates that the use of more robust and powerful component codes does not compensate for the quick decay in the GLDPC ensemble coding rate. In Fig. 7(b), we consider a $(2,15)$ base DDs with a component code of Code Families VIII ($\mathtt{d} = 3$). In this case, as predicted by lemma 2, the bounds coincide and the gap to capacity is minimized at a coding rate $\mathtt{R} \approx 0.54$ and threshold $\epsilon^* \approx 0.379$, resulting in a gap capacity equal to 0.074. This is slightly above the gap to capacity for the base LDPC ensemble (0.062). Also, at this point the GLDPC ensemble does not have linear growth of the minimum distance, since for this ensemble, $\epsilon^*(\hat{\nu}) = 0.493$.
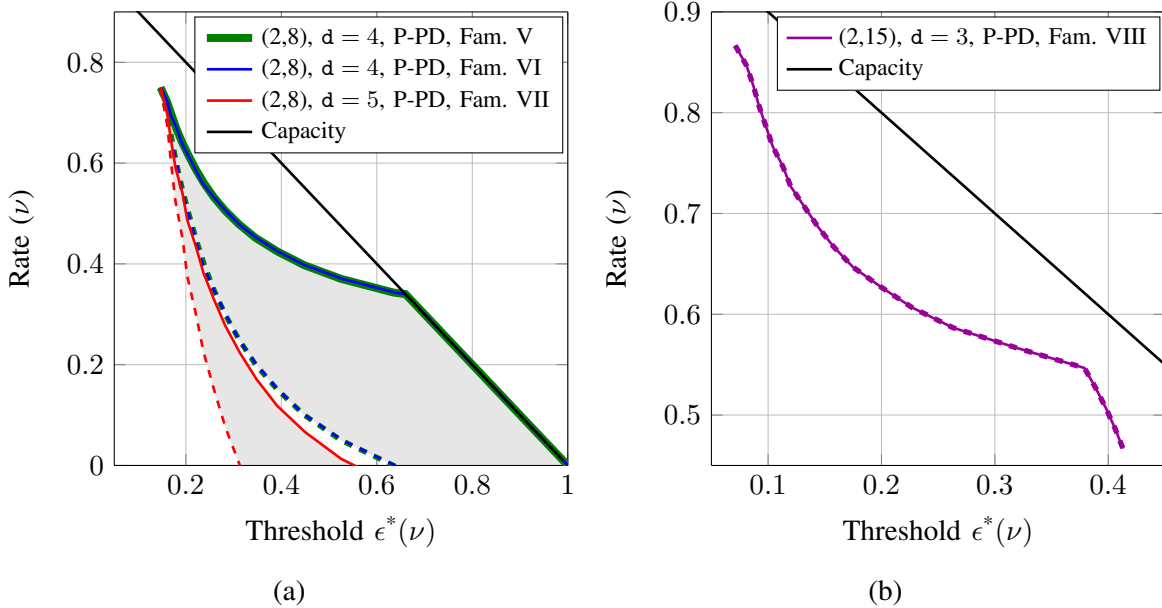

Fig. 7. We plot the bounds on the $\mathcal{C}_{J,K,\nu}$ coding rate in (6) and (7) for a base DD $(2,8)$ (Fig. 7 (a)) and $(2,15)$ (Fig. 7 (b)) as a function of the P-PD threshold.

VI. SELECTING SPECIFIC COMPONENT CODES

By using the bounds on the $\mathcal{C}_{J,K,\nu}$ code rate, we have been able to assess the performance of $\mathcal{C}_{J,K,\nu}$ ensembles for a family of linear component codes. In certain scenarios the proposed bounds on the $\mathcal{C}_{J,K,\nu}$

code rate provide meaningful design information about the asymptotic behavior of the ensemble. The natural question that arises at this point is whether we can find specific component codes within the family that outperform the achievable bound in (7), reducing the gap to the rate converse bound in (6). In this section, we analyze the asymptotic performance of $\mathcal{C}_{J,K,\nu}$ when component codes are chosen from the the list of reference linear block component codes summarized in Table III. The construction of these linear block codes is detailed in [23], and their generator matrix is given in Appendix C. We use the notation R-I to denote the reference linear block code of Code Family I.

TABLE III

REFERENCE COMPONENT CODES. THE PARAMETER k DESCRIBES THE NUMBER OF ROWS IN THE PARITY-CHECK MATRIX.

Code index	Blocklength K	k	Rate	Code family in Table II
R-I	6	3	1/2	I
R-II	6	4	1/3	II
R-III	7	3	4/7	III
R-IV	7	4	3/7	IV
R-V	8	4	1/2	V
R-VI	8	5	3/8	VI
R-VII	8	6	1/4	VII
R-VIII	15	4	11/15	VIII
R-IX	15	5	2/3	IX

Once we fix a particular class of component codes to be used at GC nodes, we can replace the $\mathcal{C}_{J,K,\nu}$ code bounds by the actual code rate in (4). In Fig. 8 we plot the $\mathcal{C}_{J,K,\nu}$ coding rate and the achievable bound of the corresponding family of codes for $(2,6)$ and $(2,7)$ base DDs. Results for $(2,8)$ and $(2,15)$ base DDs can be found in Fig. 9. Observe that, with the proposed component codes, we are able to perform at least as good as the achievable bound of the corresponding family of block component codes. In some cases, e.g. the $(2,8)$ base DD, the achievable bound is significantly outperformed. Recall that for the $(2,8)$ base DD the rate bounds in Fig. 7(a) are lose. Also, we highlight those points for which, asymptotically, the $\mathcal{C}_{J,K,\nu}$ ensemble with the proposed linear component codes under P-PD operates closer to channel capacity than the base LDPC code ensemble under PD. For both the $(2,6)$, $(2,7)$, and the $(2,8)$ base DDs we were able to find such points. For the $(2,15)$ ensemble, the minimum gap to capacity obtained is slightly above the one of the base LDPC code ensemble under PD (0.0743 and 0.0623 respectively). As already discussed in Section V, observe that for every base DD the use of component codes with higher minimum distance does not improve the GLDPC asymptotic performance.
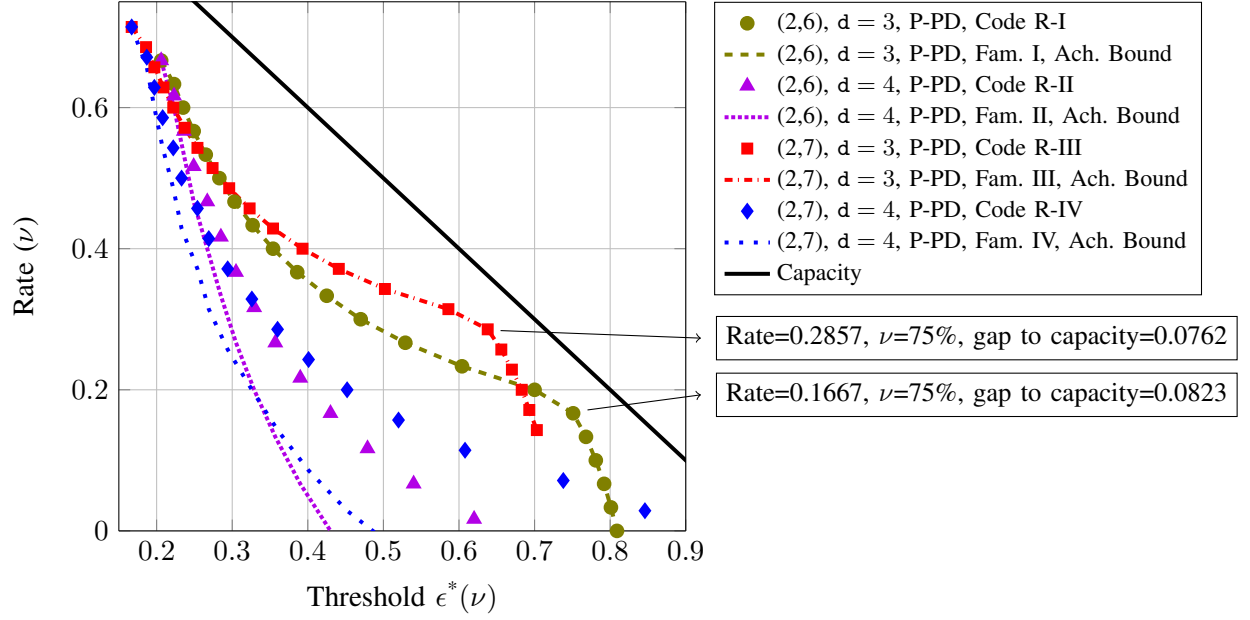
Fig. 8. $\mathcal{C}_{J,K,\nu}$ coding rate and achievable bound in (7) for $(2,6)$ and $(2,7)$ base DDs and component codes from Table II and III as a function of the P-PD decoding threshold.
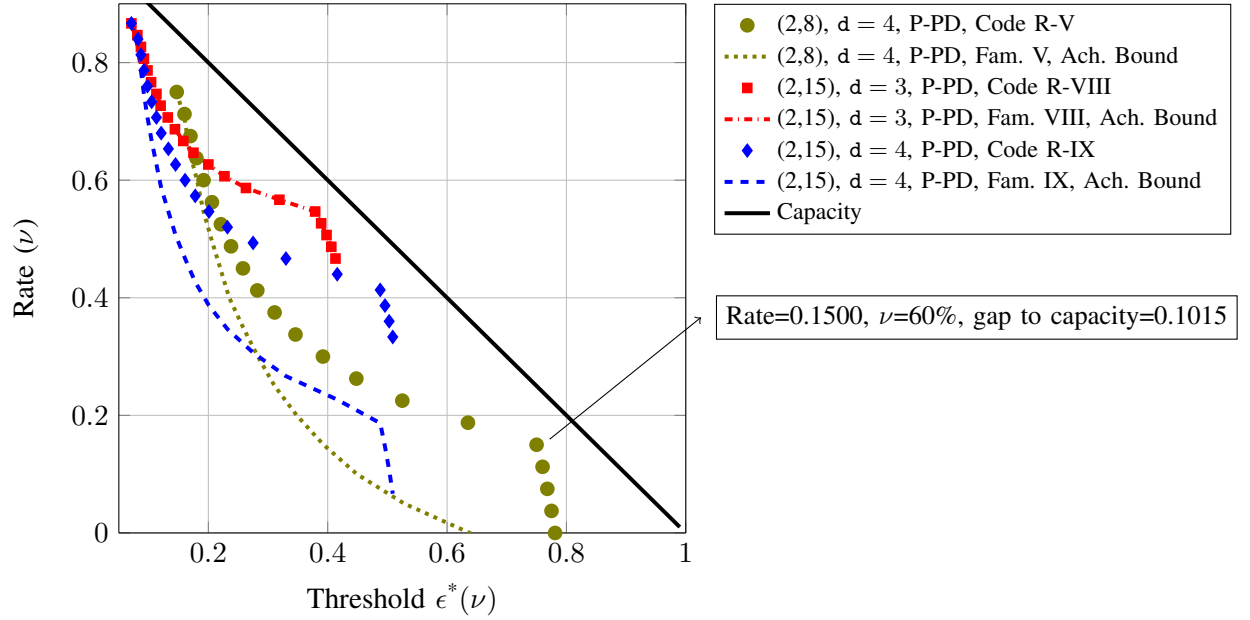

Fig. 9. $\mathcal{C}_{J,K,\nu}$ coding rate and achievable bound in (7) for $(2,8)$ and $(2,15)$ base DDs and component codes from Tables II and III as a function of the P-PD decoding threshold.

A. Growth Rate of the Weight Distribution

Upon selecting a specific block code, we can compute the weight spectral shape $G(\alpha)$ in (14) using the tools proposed in [10]. In Fig. 10, we plot $G(\alpha)$ for different values of ν, computed for the $(2,6)$

base DD with Code R-I as component code (Fig. 10 (a)) and the $(2,7)$-regular base DD with Code R-III as component code (Fig. 10 (b)). Recall that the critical exponent codeword weight ratio is defined as $\hat{\alpha} \triangleq \inf\{\alpha \geq 0 | G(\alpha) \geq 0\}$. In the plots, we highlight $\hat{\alpha}$ with a star. By Lemma 3, we have $\hat{\alpha} = 0$ at $\nu = \hat{\nu}$. As ν grows, $\hat{\alpha}$ grows, too, and it achieves its maximum at $\nu = 1$. These results indicate that there is a trade-off between the gap to capacity and $\hat{\alpha}(\nu)$, the critical exponent codeword weight ratio. As an example, we include values of both quantities in Table IV for the $(2,6)$-regular base DD with Code R-I as component code.
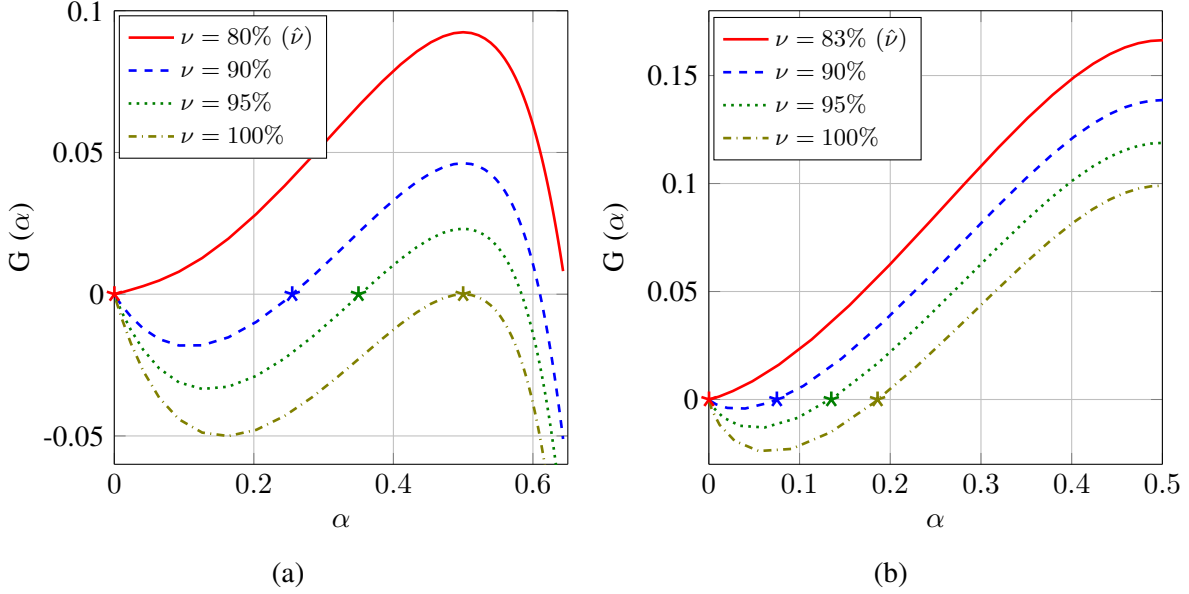

Fig. 10. In Fig. 10 (a), we plot the weight spectral shape $G(\alpha)$ in (14) of the $\mathcal{C}_{J,K,\nu}$ ensemble for a $(2,6)$ base DD and with Code R-I as component code. In Fig. 10 (b), we plot the same quantity for the $\mathcal{C}_{J,K,\nu}$ ensemble for a $(2,7)$ base DD and with Code R-II as component code (b).

TABLE IV

$\hat{\alpha}$, ϵ^* AND GAP TO CAPACITY FOR DIFFERENT VALUES OF ν, COMPUTED FOR THE $(2,6)$-BASE DD WITH CODE R-I COMPONENT CODES

ν	α	ϵ^*	Gap to capacity
80%	0	0.768	0.0987
87.5%	0.2049	0.788	0.1287
90%	0.2556	0.792	0.1413
92.5%	0.3038	0.797	0.1530
95%	0.3526	0.801	0.1657
97.5%	0.4056	0.806	0.1773
100%	0.6078	0.809	0.1910

VII. Random puncturing

We have proposed the P-PD algorithm as a flexible model to analyze beyond-BD decoding algorithm at GC nodes. Observe that for the P-PD algorithm, the evaluation of the coding rate and the iterative decoding threshold are decoupled problems. This provides a flexible analysis framework that allows the exploration of additional techniques to modify the designs presented above and further reduce the gap to capacity. In this section and the following one, we consider two relevant examples. Specifically, in this section we consider the use of random puncturing to accommodate the coding rate by dropping the transmission of a fraction of coded bits [17]. In the next section, a simple model of doubly-generalized LDPC (DG-LDPC) code ensembles is analyzed [18], [19], [5].

As illustrated in [17], a linear code is *punctured* by removing a set of columns from its generator matrix. After puncturing at random a fraction ξ of the coded bits in the $\mathcal{C}_{J,K,\nu}$ ensemble, the resulting coding rate is

$$\text{R}(\nu, \xi) = \frac{\text{R}(\nu)}{1 - \xi}, \quad \xi \in [0, 1), \tag{36}$$

where we recall that $\text{R}(\nu)$ denotes the coding rate of the original $\mathcal{C}_{J,K,\nu}$ ensemble. In [17], the authors derive a simple analytic expression for the iterative belief propagation (BP) decoding threshold of a randomly punctured LDPC code ensemble on the binary erasure channel (BEC). Following their proof, it can be verified that the same results apply to a randomly punctured GLDPC code ensemble. The result reads as follows. Given a $\mathcal{C}_{J,K,\nu}$ ensemble with iterative decoding threshold $\epsilon^*(\nu)$, the threshold $\epsilon^*(\nu, \xi)$ of the GLDPC ensemble that follows by randomly puncturing a fraction ξ of the coded bits is related to the unpunctured case as follows:

$$\epsilon^*(\nu, \xi) = 1 - \frac{1 - \epsilon^*(\nu)}{1 - \xi}. \tag{37}$$

Observe that the larger the unpunctured threshold $\epsilon^*(\nu)$ is, the larger the threshold of the punctured ensemble will be. In this regard, we can think of the design of a punctured GLDPC ensemble as a two stage process: First, the GLDPC code ensemble can be designed by choosing ν to minimize the gap to capacity. Second, for a fixed ν, we can analyze the overall gap to capacity as we increasing the code rate by combining (36) and (37). We perform this experiment in Fig. 11 (a) for the $(2, 6)$ and the $(2, 7)$ base DDs and component codes R-I and R-III, respectively. With markers we show the $\mathcal{C}_{J,K,\nu}$ threshold-rate curve as we increase the fraction of GC nodes in the graph. Solid lines indicate the evolution of the rate and threshold of the punctured ensemble for fixed ν as we increase the puncturing fraction ξ. Observe that with puncturing it is possible to increase the coding rate and obtain an iterative decoding threshold that is closer to capacity than those obtained by the original $\mathcal{C}_{J,K,\nu}$ ensemble. The accuracy of the predicted

threshold can be observed in Fig. 11 (b), where we include both the threshold predicted by (37) (dashed lines) and the simulated P-PD performance for the $(2,6)$ base DD with component code R-I, $\mathtt{n} = 10000$ bits, and different values of the puncturing rate ξ (solid lines).
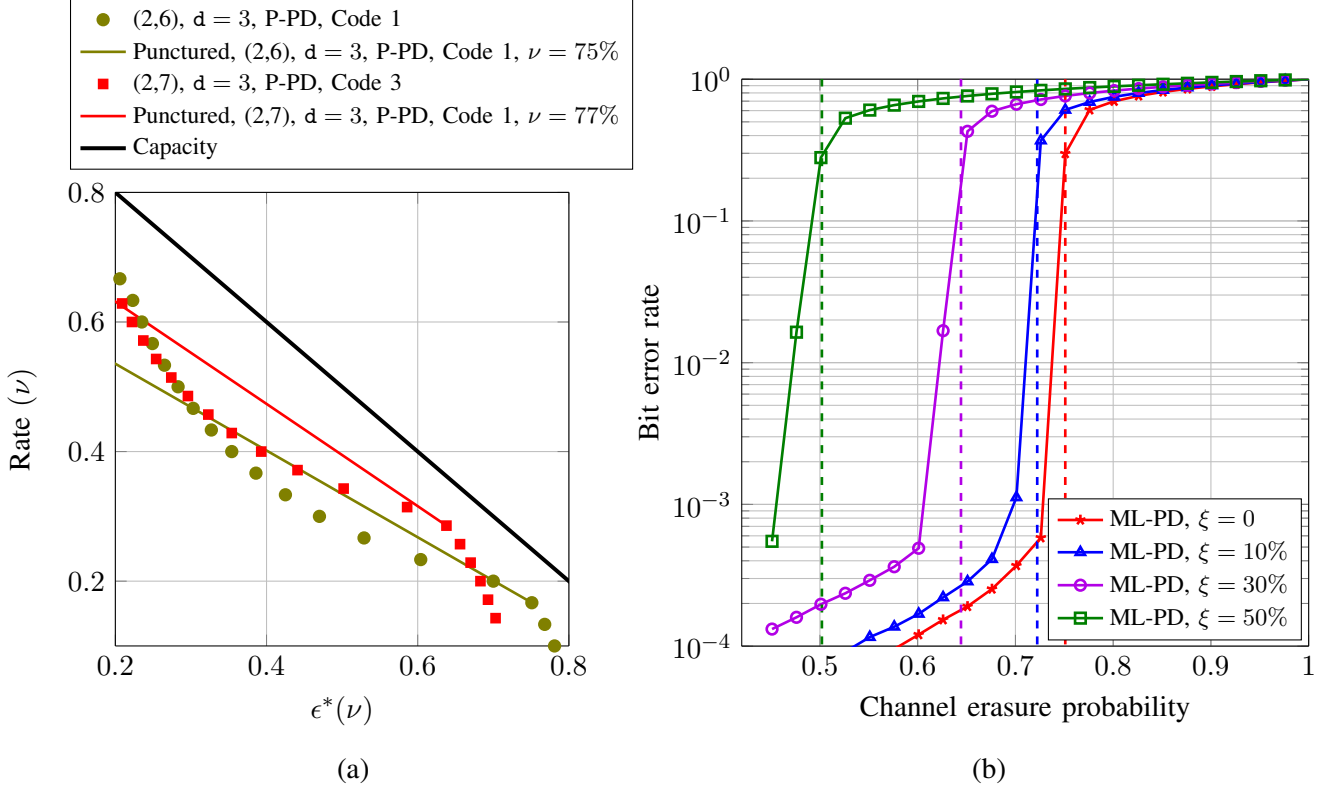

Fig. 11. In Fig. 11 (a), we show with markers the $\mathcal{C}_{J,K,\nu}$ threshold-rate curve for the $(2,6)$ and the $(2,7)$ base DDs and component codes A and C, respectively. Solid lines indicate the evolution of the rates and thresholds of the punctured ensemble for a fixed ν as we increase the puncturing fraction ξ. In Fig. 11 (b), we show the simulated P-PD performance for the $(2,6)$ base DD with component code A, $\mathtt{n} = 10000$ bits, and different values of the puncturing rate ξ. Vertical dashed lines indicate the thresholds predicted by (37).

VIII. DOUBLY-GENERALIZED LDPC CODES

A different technique that can potentially help to find a better balance between coding rate and threshold is the inclusion of generalized variable nodes, giving rise to a doubly-generalized LDPC code ensemble [18]. In this section we develop an example with a simple class of a DG-LDPC ensemble. We modify the $\mathcal{C}_{J,K,\nu}$ ensemble by replacing a certain fraction β of regular variable (RV) nodes by generalized variable (GV) nodes, see Fig. 12 (a). Degree-J RV nodes in the $\mathcal{C}_{J,K,\nu}$ graph can be seen as rate $1/J$ repetition code of block length J, where the input to the repetition code represents one bit of the DG-LDPC codeword. On the other hand, degree-J GV nodes are characterized by a $(J, \mathtt{m})$ linear block code, where
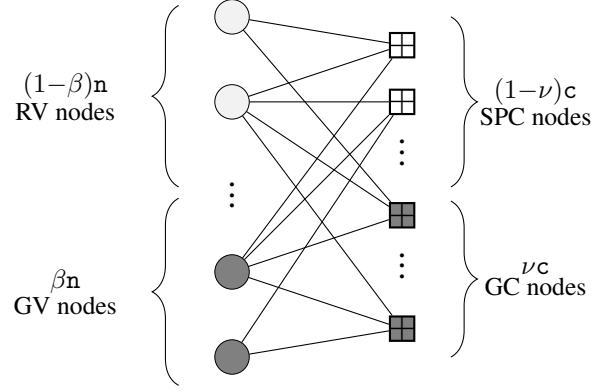
Fig. 12. Tanner graph of the DG-LDPC code ensemble.

the input to the variable component code represents $\mathtt{m}$ bits of the DG-LDPC codeword. Thus, the total block length of the DG-LDPC code ensemble is $\mathtt{n}' = (1-\beta)\mathtt{n} + \beta\mathtt{nm}$, where $\mathtt{n}$ is the number of variable nodes (both RV and GV) in the graph. In the following, we will assume $J = 3, \mathtt{m} = 2$ and the following generator matrix for GV nodes:

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \tag{38}$$

Thus, each GV node encodes two bits of the DG-LDPC codeword. Denote this ensemble by $\mathcal{C}_{3,K,\nu,\beta}$. If the component codes at GC nodes are linear block codes with a $\mathtt{k}$-row parity check matrix, an easy calculation shows that the coding rate of the ensemble is

$$\mathtt{R}(\alpha, \beta) = 1 - (1 - \mathtt{R_0})\left(\frac{1 + (\mathtt{k}-1)\nu}{1+\beta}\right). \tag{39}$$

As before, we characterize the component codes at GC nodes by the triple $(\mathtt{d}, p_\mathtt{d}, p_{\mathtt{d}+1})$. Furthermore, the code associated with the generator matrix (38) has minimum distance 2 and can only decode erasure patterns of weight one.

A. Decoding via P-PD

Suppose we use a random sample of the $\mathcal{C}_{3,K,\nu,\beta}$ code ensemble to transmit over a BEC(ϵ). RV nodes are removed from the graph with probability $1-\epsilon$. Regarding GV nodes, we have to consider the following three scenarios:

- With probability $(1-\epsilon)^2$ the two DG-LDPC coded bits are correctly received and the GV node can be removed from the graph.
- With probability $2\epsilon(1-\epsilon)$, only of the two coded bits is received. Since the node is only encoding one unknown bit, note that we can replace the GV node in the graph by a degree-2 RV node.

- With probability ϵ^2 the GV node remains in the graph as a degree-3 GV node.

Decoding will be performed via P-PD. Since the code spanned by (38) can only decode one error, during the P-PD procedure every GV node needs to lose at least two edges before it can be removed from the graph. Further, once it loses one edge, it can be replaced by a degree-2 RV node. Hence, a small modification is required at step 2) in the P-PD Algorithm in Section III. Now, it reads as follows:

2) Remove from the Tanner graph the check node with the index drawn in Step 1). Further remove all connected RV nodes, connected degree-2 GV nodes and all attached edges.

B. Degree Distribution and Asymptotic Analysis

While no change is needed to describe the evolution of the check nodes of the residual DG-LDPC code ensemble during P-PD, additional definitions at the variable side are needed to tackle both RV nodes and GV nodes. Let $L_{r2}^{(\ell)}$ and $L_{r3}^{(\ell)}$ represent the total number of edges in the graph connected to RV nodes of degree 2 and 3, respectively, after iteration ℓ of the decoder. Further let $L_{g3}^{(\ell)}$ be the total number of edges in the graph connected to GV nodes of degree 3.

Theorem 5: Consider a BEC with erasure probability ϵ and assume we use elements of the $\mathcal{C}_{3,K,\nu,\beta}$ code ensemble for transmission. If we use P-PD with parameters $(\mathtt{d}, p_\mathtt{d}, p_{\mathtt{d}+1})$, in the limit as $\mathtt{E} \to \infty$ the DD of the residual graph at iteration ℓ can be computed as follows:

$$\lim_{\mathtt{E}\to\infty} L_{r2}^{(\ell)} = \mathtt{E} l_{r2}^{(\ell/\mathtt{E})}, \tag{40}$$

$$\lim_{\mathtt{E}\to\infty} L_{r3}^{(\ell)} = \mathtt{E} l_{r3}^{(\ell/\mathtt{E})}, \tag{41}$$

$$\lim_{\mathtt{E}\to\infty} L_{g3}^{(\ell)} = \mathtt{E} l_{g3}^{(\ell/\mathtt{E})}, \tag{42}$$

$$\lim_{\mathtt{E}\to\infty} R_{pj}^{(\ell)} = \mathtt{E} r_{pj}^{(\ell/\mathtt{E})}, \quad j \in \{1, \ldots, K\} \tag{43}$$

$$\lim_{\mathtt{E}\to\infty} R_{cj}^{(\ell)} = \mathtt{E} r_{cj}^{(\ell/\mathtt{E})}, \quad j \in \{1, \ldots, K\} \text{ and } j \notin \{\mathtt{d}, \mathtt{d}+1\} \tag{44}$$

$$\lim_{\mathtt{E}\to\infty} \hat{R}_{cj}^{(\ell)} = \mathtt{E} \hat{r}_{cj}^{(\ell/\mathtt{E})}, \quad j \in \{\mathtt{d}, \mathtt{d}+1\} \tag{45}$$

$$\lim_{\mathtt{E}\to\infty} \bar{R}_{cj}^{(\ell)} = \mathtt{E} \bar{r}_{cj}^{(\ell/\mathtt{E})}, \quad j \in \{\mathtt{d}, \mathtt{d}+1\} \tag{46}$$

where $l_{r2}^{(\tau)}$, $l_{r3}^{(\tau)}$, $l_{g3}^{(\tau)}$ $r_{pj}^{(\tau)}$, $r_{cj}^{(\tau)}$, $\hat{r}_{cj}^{(\tau)}$, $\bar{r}_{cj}^{(\tau)}$ $\tau \in [0, 1/J]$ are the solutions to the system of differential

equations given by (21)-(25) using $a^{(\tau)} = (3l_{r3}^{(\tau)} + 2l_{r2}^{(\tau)} + l_{g3}^{(\tau)})/e^{(\tau)}$ and

$$\frac{\mathrm{d}l_{r2}^{(\tau)}}{\mathrm{d}\tau} = 2\left(\frac{l_{g3}^{(\tau)} - l_{r2}^{(\tau)}}{e^{(\tau)}}\right)\left(P_{p1}^{(\tau)} + \sum_{w=1}^{\mathrm{d}+1} wP_{cw}^{(\tau)}\right) \tag{47}$$

$$\frac{\mathrm{d}l_{r3}^{(\tau)}}{\mathrm{d}\tau} = -\frac{3l_{r3}^{(\tau)}}{e^{(\tau)}}\left(P_{p1}^{(\tau)} + \sum_{w=1}^{\mathrm{d}+1} wP_{cw}^{(\tau)}\right) \tag{48}$$

$$\frac{\mathrm{d}l_{g3}^{(\tau)}}{\mathrm{d}\tau} = -\frac{3l_{g3}^{(\tau)}}{e^{(\tau)}}\left(P_{p1}^{(\tau)} + \sum_{w=1}^{\mathrm{d}+1} wP_{cw}^{(\tau)}\right), \tag{49}$$

Here, $e^{(\tau)}$, $P_{p1}^{(\tau)}$ and $P_{cw}^{(\tau)}$ are defined in (26), (28), and (29), respectively. The initial conditions of the system of differential equations in (21)-(25) and (90)-(92) are given by

$$l_{g3}^{(0)} = \epsilon^2\beta, \tag{50}$$

$$l_{r3}^{(0)} = \epsilon(1-\beta), \tag{51}$$

$$l_{r2}^{(0)} = 4\beta\epsilon(1-\epsilon)/3 \tag{52}$$

and by (32)-(35) evaluated at $\epsilon' = \epsilon(1 + \beta(1-\epsilon)/3)$.

Proof: See Appendix B. ∎

C. Results for the $(3,6)$ and $(3,7)$ base DDs

Fig. 13 shows the computed rate-threshold curve parametrized by ν for both the $\mathcal{C}_{3,K,\nu,\beta}$ ensembles, both with $\beta = 0$, i.e., when the code graph has no generalized variable nodes, and with $\beta = 0.3$. We use a $(3,6)$ base DD with code R-I (see Table III) as component code. While in the former case the minimun gap to capacity is achieved for the base LDPC code ensemble (with a gap to capacity of 0.0710), by using a certain amount of generalized variable nodes we are able to reduce this gap to 0.0592. Further, since all variable nodes in the graph have degree 3, by Lemma 3, for any value of ν the code ensemble has a minimum distance that grows linearly with the block length. Fig. 14 shows similar results for a $(3,7)$ base DD with Code R-III as component code.

IX. CONCLUSIONS AND FUTURE WORK

We proposed the P-PD algorithm as a flexible and efficient decoding algorithm that allows us to easily incorporate ML-decoded GC nodes with specific properties into the asymptotic analysis and still maintain a random definition of the graph degree distribution. Using P-PD, asymptotic analysis of the GLDPC ensemble is carried out by a simple generalization of the original PD analysis by Luby et al. in [11]. The only information required about the component code and its decoding method is the fraction of decodable
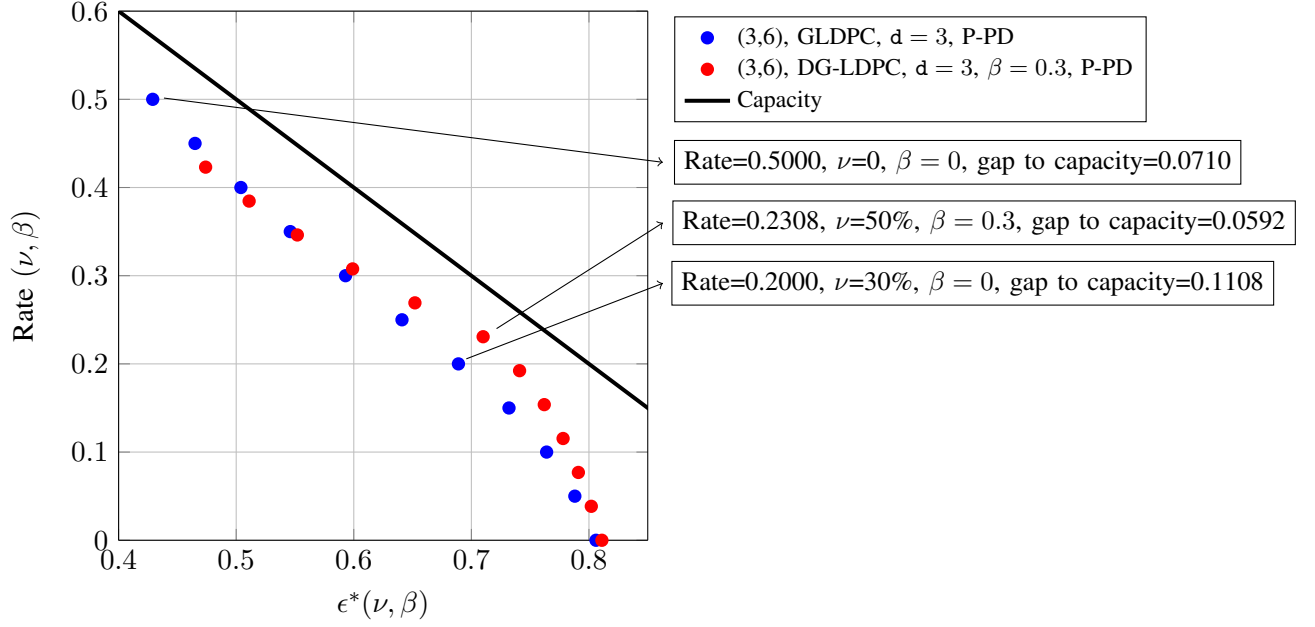
Fig. 13. $\mathcal{C}_{3,K,\nu,\beta}$ coding rate for a (3,6) base DD with Code R-I as component code, GV nodes constructed using the generator matrix in (38), and $\beta = 0.3$.
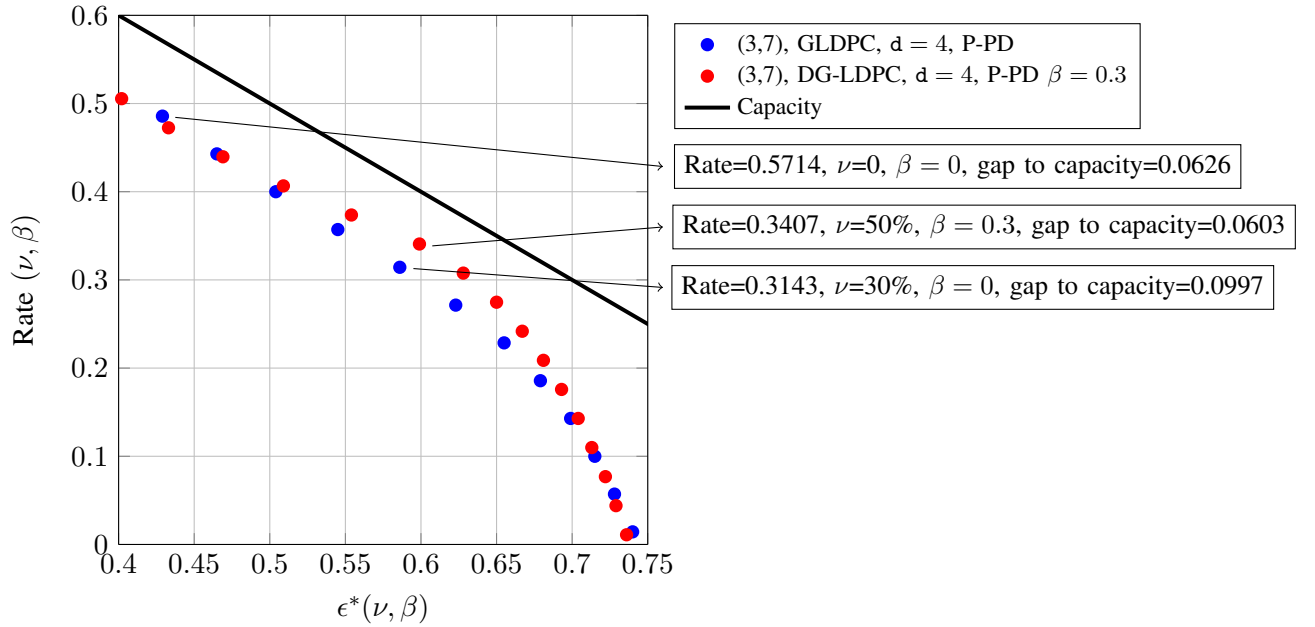

Fig. 14. $\mathcal{C}_{3,K,\nu,\beta}$ coding rate for (3,7) base DD, Code R-III component code, GV nodes constructed using the generator matrix in (38), and $\beta = 0.3$.

erasure patterns of a certain weight. We consider a class of GLDPC code ensembles characterized by a regular base DD where we include a certain fraction of GC nodes, and we study the tradeoff between iterative decoding threshold, coding rate and minimum distance. We have shown that one can find a fraction of GC nodes required that reduces the original gap to capacity and yields a GLDPC ensemble with linear growth of the minimum distance w.r.t. the block length. Finally, we show how the P-PD analysis can be combined with additional techniques to find a better balance between coding rate and asymptotic gap to capacity. In particular, we consider random puncturing and the use of generalized variable nodes. We would like to emphasize that, in the proposed analysis framework, the evaluation of both coding rate and of iterative decoding threshold are decoupled problems. Consequently, broader classes of component codes or improved decoding methods at GC nodes can be incorporated in a systematic way.

Future lines of work include the analysis of GLDPC codes with regular base DD and a certain fraction of GC nodes in the finite-length regime. Due to their regularity of the DD, we expect such codes to possess a robust finite-length behavior compared to GLDPC code designs proposed in the literature, characterized by capacity-achieving DDs. We will extend the finite-length analysis of regular LDPC ensembles in [26] to GLDPC codes under P-PD decoding and assess its accuracy in the finite-length regime. Following [27], we will also incorporate techniques to improve the code minimum distance. We further intend to extend our analysis to the design of spatially-coupled GLDPC (SC-GLDPC) code ensembles [28]. The combination of capacity achieving SC-GLDPC code ensembles at a certain coding rate and puncturing techniques is a promising design approach to construct SC-GLDPC code ensembles with improved finite-length scaling properties [29], [30].

APPENDIX A
WORMALD'S THEOREM AND THE PROOF OF THEOREM 4

Proving Theorem 4 is tantamount to showing that the conditions of Wormald's theorem are satisfied [22]. In this case, Theorem 4 follows directly from (55) and (56) below.

A. Wormald's theorem [22]

Let $\{Z^{(\ell)}(a)\}_{a\geq 1}$ be a d-dimensional discrete-time Markov random process with state space $\{0, 1, \ldots, \lfloor a\alpha \rfloor\}^d$ for $\alpha > 0$ and $\ell \in \mathbb{N}_+$ denotes the time index. Further let $Z_i^{(\ell)}(a)$, $i = 1, \ldots, d$ denote the i-th component of $Z^{(\ell)}(a)$. Let $\mathcal{D}$ be some open connected bounded set containing the closure of

$$\left\{ (z_1, ..., z_d) : \ P\left(\frac{Z_i^{(0)}(a)}{a} = z_i, 1 \leq i \leq d \right) > 0 \text{ for some } a \right\}. \tag{53}$$

We define the stopping time ℓ_D to be the smallest time index ℓ such that

$$(Z_1^{(\ell_D)}(a)/a, ..., Z_d^{(\ell_D)}(a)/a) \notin \mathcal{D} \tag{54}$$

Furthermore, let $f_i(\cdot)$, $i = 1, \ldots, d$, be functions from $\mathbb{R}^{d+1}$ to $\mathbb{R}$. Assume that the following conditions are satisfied:

1) (Boundedness) There exists a constant ν such that for all $i = 1, \ldots d$, $\ell = 0, \ldots, \ell_{\mathcal{D}} - 1$ and $a \geq 1$,

$$\left| Z_i^{(\ell+1)}(a) - Z_i^{(\ell)}(a) \right| \leq \nu.$$

2) (Trend functions) For all $i = 1, \ldots, d$, $\ell = 0, \ldots, \ell_{\mathcal{D}} - 1$ and $a \geq 1$,

$$\mathbb{E}\left[Z_i^{(\ell+1)}(a)/a - Z_i^{(\ell)}(a)/a \Big| Z^{(\ell)}(a)/a \right] = f_i\left(\ell/a, Z_1^{(\ell)}(a)/a, ..., Z_d^{(\ell)}(a)/a \right) + \mathcal{O}(1/a).$$

3) (Lipschitz continuity) Each function $f_i(\cdot)$, $i = 1, \ldots, d$, is Lipschitz continuous on $\mathcal{D}$. Namely, for any pair $b, c \in \mathcal{D}$ that belongs to such intersection, there exists a constant κ such that

$$|f_i(b) - f_i(c)| \leq \kappa \sum_{j=1}^{d+1} |b_j - c_j|.$$

Under these conditions, the following holds:

- The system of differential equations

$$\frac{\partial z_i}{\partial \tau} = f_i(\tau, z_1, ..., z_d), \quad i = 1, ..., d, \tag{55}$$

  has a unique solution for any initial condition $(b_1, ..., b_d) \in \mathcal{D}$.

- There exists a strictly positive constant ζ such that

$$P\left(\left| Z_i^{(\ell)}(a)/a - z_i(\ell/a) \right| > \zeta a^{-\frac{1}{6}} \right) = \mathcal{O}\left(e^{-\sqrt{a}} \right) \tag{56}$$

  for $i = 1, ..., d$ and $0 \leq t \leq t_{\mathcal{D}}$, where $z_i(\ell/a)$ is the solution to (55) for

$$b_i = \mathbb{E}[Z_i^{(0)}(a)]/a, \quad i = 1, \ldots, d. \tag{57}$$

The result in (56) states that any realization of the process $Z_i^{(t)}(a)$ concentrates around the solution predicted by (55) in the limit as $a \to \infty$. In the next subsection we show that this theorem is suitable to describe the expected GLDPC graph evolution of the P-PD.

B. Expected graph evolution under P-PD

To analyze the asymptotic behavior of the $\mathcal{C}_{J,K,\nu}$ ensemble under P-PD using Wormald's theorem, we identify the Markov random process $Z^{(\ell)}(a)$ in the previous section by the random process $\mathcal{G}^{(\ell)}(\mathrm{E})$, where

$$\mathcal{G}^{(\ell)}(\mathrm{E}) = \left\{ L_i^{(\ell)}, R_{pj}^{(\ell)}, R_{cj}^{(\ell)}, \hat{R}_{c\mathrm{d}}^{(\ell)}, \bar{R}_{c\mathrm{d}}^{(\ell)}, \hat{R}_{c(\mathrm{d}+1)}^{(\ell)}, \bar{R}_{c(\mathrm{d}+1)}^{(\ell)} \right\}_{\substack{i=1,\dots,J \\ j=1,\dots,\mathrm{d}-1,\mathrm{d}+2,\dots,K}} \tag{58}$$

namely $\mathcal{G}^{(\ell)}(\mathrm{E})$ is the random process that contains all terms in the DD of the residual graph after $\ell - 1$ iterations. Note that any component in $\mathcal{G}^{(\ell)}(\mathrm{E})$ belongs to the set $\{0, 1, \dots, \mathrm{E}\}$, and recall that E is the number of edges in the original GLPDC graph. Thus, E will play the role of the parameter a. In this subsection we prove that the evolution of $\mathcal{G}^{(\ell)}(\mathrm{E})$ under P-PD satisfies the three conditions of Wormald's theorem stated in the previous subsection. We start by computing the conditional expected evolution of all elements in $\mathcal{G}^{(\ell)}(\mathrm{E})$ after one P-PD iteration. We define the following normalized quantities:

$$\tau \triangleq \frac{\ell}{\mathrm{E}}, \quad l_i^{(\ell)} \triangleq \frac{L_i^{(\ell)}}{\mathrm{E}}, \quad r_{pj}^{(\ell)} \triangleq \frac{R_{pj}^{(\ell)}}{\mathrm{E}}, \quad r_{cj}^{(\ell)} \triangleq \frac{R_{cj}^{(\ell)}}{\mathrm{E}}, \quad \hat{r}_{c\nu}^{(\ell)} \triangleq \frac{\hat{R}_{c\nu}^{(\ell)}}{\mathrm{E}}, \quad \bar{r}_{c\nu}^{(\ell)} \triangleq \frac{\bar{R}_{c\nu}^{(\ell)}}{\mathrm{E}}, \tag{59}$$

for $i \in \{1, \dots, J\}, j \in \{1, \dots, \mathrm{d}-1, \mathrm{d}+2, \dots, K\}$ and $\nu \in \{\mathrm{d}, \mathrm{d}+1\}$. We have that

$$r_{c\nu}^{(\ell)} = \hat{r}_{c\nu}^{(\ell)} + \bar{r}_{c\nu}^{(\ell)}, \quad \nu = \mathrm{d}, \mathrm{d}+1, \tag{60}$$

$$e^{(\ell)} \triangleq \sum_{i=1}^{J} l_i^{(\ell)} = \sum_{j=1}^{K} [r_{pj}^{(\ell)} + r_{cj}^{(\ell)}], \tag{61}$$

and $e^{(\tau)}$ is the fraction of edges remaining in the residual graph at time ℓ. The P-PD process starts at $\ell = 0$, after BEC transmission and initialization. The following relation holds between the quantities defined above at $\ell = 0$ and the $\mathcal{C}_{J,K,\nu}$ DD described in Section II:

$$\mathbb{E}[l_i^{(0)}] = \epsilon \lambda_i, \tag{62}$$

$$\mathbb{E}[r_{pj}^{(0)}] = \sum_{\alpha \geq j} \rho_{p\alpha} \binom{\alpha - 1}{j - 1} \epsilon^j (1 - \epsilon)^{\alpha - j}, \tag{63}$$

$$\mathbb{E}[r_{cj}^{(0)}] = \sum_{\alpha \geq j} \rho_{c\alpha} \binom{\alpha - 1}{j - 1} \epsilon^j (1 - \epsilon)^{\alpha - j}, \tag{64}$$

for $i = 1, \dots J$ and $j = 1, \dots, K$, where the expectation is computed w.r.t. the $\mathcal{C}_{J,K,\nu}$ ensemble and the channel output. Upon initialization, every degree-d GC node is tagged as decodable with probability p_d, and every degree-$(\mathrm{d}+1)$ GC node is tagged as decodable with probability $p_{\mathrm{d}+1}$. Recall that all GC nodes

with degree less than d are decodable and, by assumption, all GC nodes with degree more than $d+1$ are not decodable. We thus have the following initial conditions

$$\mathbb{E}[\hat{r}_{cj}^{(0)}] = p_j \mathbb{E}[r_{cj}^{(0)}],$$

$$\mathbb{E}[\bar{r}_{cj}^{(0)}] = (1-p_j)\mathbb{E}[r_{cj}^{(0)}], \quad j = d, d+1. \tag{65}$$

The equations (62)-(65) correspond to the initial conditions in (57). Observe that since the largest GC degree is K and the largest variable node degree is J, the graph loses at most JK edges per iteration. This is an upper bound on the absolute variation of any component in $\mathcal{G}^{(\ell)}(E)$ between two consecutive iterations. Hence, Condition 1) of Wormald's theorem is satisfied.

Suppose we observe $\mathcal{G}^{(\ell)}(E)$. To derive the conditional expectations in Condition 2) of Wormald's Theorem, the so-called trend functions, we have to average among every possible scenario that we can observe after a P-PD iteration. According to Step 1) in Algorithm 2, we chose at random a decodable check node. Let $P_{p1}^{(\ell)}$ be the probability of selecting a degree-one SPC node, and let $P_{cj}^{(\ell)}$ denote the probability of selecting a decodable degree-j GC node, $j = 1, \dots, d+1$. By a simple counting argument, if the check node is selected uniformly at random then

$$P_{p1}^{(\ell)} = \frac{r_{p1}^{(\ell)}}{s^{(\tau)}}, \tag{66}$$

$$P_{cj}^{(\ell)} = \frac{r_{cj}^{(\ell)}/j}{s^{(\tau)}}, \quad j < d, \tag{67}$$

$$P_{cj}^{(\ell)} = \frac{\hat{r}_{cj}^{(\ell)}/j}{s^{(\tau)}}, \quad j \in \{d, d+1\}. \tag{68}$$

In (66)-(68),

$$s^{(\tau)} = r_{p1}^{(\ell)} + \sum_{w=1}^{d-1} \frac{r_{cw}^{(\ell)}}{w} + \frac{\hat{r}_{cd}^{(\ell)}}{d} + \frac{\hat{r}_{c(d+1)}^{(\ell)}}{d+1} \tag{69}$$

is the normalized sum of decodable check nodes at the ℓ-th iteration.

1) Evolution of left edge degrees in the Tanner graph after one P-PD iteration: Suppose we observe the residual graph $\mathcal{G}^{(\ell)}$ at iteration ℓ. Our aim is to evaluate

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}(E)\right], \tag{70}$$

for $i = 1, 2, ..., J$. Given the graph DD $\mathcal{G}^{(\ell)}$, recall that $P_{p1}^{(\ell)}$ denotes the probability of P-PD selecting a degree-one SPC node in the current iteration, and $P_{cj}^{(\ell)}$ denotes the probability of selecting a degree-j

decodable GC node. We can decompose the expectation in (70) according to each possible type of check node to be removed, namely,

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)}\Big|\mathcal{G}^{(\ell)}(\text{E})\right] = P_{p1}^{(\ell)}\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)}\Big|\mathcal{G}^{(\ell)}(\text{E}), \text{Deg}_{p1}\right] + \sum_{w=1}^{\text{d}+1} P_{cw}^{(\ell)}\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)}\Big|\mathcal{G}^{(\ell)}(\text{E}), \text{Deg}_{cw}\right],$$
(71)

where Deg_{p1} indicates that the P-PD removes a degree-one SPC node from the graph, and Deg_{cw} indicates that P-PD removes a degree-w decodable GC node from the graph. Computing the expectation in the first case is similar to the derivation carried out in [11] for PD with LDPC ensembles. Indeed probability that the edge adjacent to the removed degree-one SPC node has left degree i is $l_i^{(\tau)}/e^{(\tau)}$. In such a case, after deleting this variable node, the graph loses $i-1$ additional edges adjacent to this variable node, so

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)}\Big|\mathcal{G}^{(\ell)}(\text{E}), \text{Deg}_{p1}\right] = -\frac{i l_i^{(\ell)}}{e^{(\ell)}}.$$
(72)

When the P-PD decoder removes a decodable degree-w GC node, this node is connected to w variable nodes that are also removed from the residual Tanner graph, along with their connected edges (assuming the graph does not have double edges). Note that left degrees of the w edges connected to the removed GC node are, in general, not independent. Let $X_u \in \{1,\ldots,J\}$ the RV that indicates the left degree of the u-th edge, $u = 1,\ldots,w$. Arbitrarily, we can decompose the joint probability of $X_1,\ldots,X_w$ as follows

$$P(X_1,\ldots,X_w) = P(X_1)P(X_2|X_1)P(X_3|X_1,X_2)\ldots P(X_w|X_1,\ldots,X_{w-1}).$$
(73)

While $P(X_1 = x_1) = l_{x_1}^{(\ell)}/e^{(\ell)}$, $x_1 = 1,\ldots,J$, the conditional distribution of X_2 given X_1 is given by

$$P(X_2 = x_2|X_1 = x_1) = \begin{cases} \dfrac{l_{x_2}^{(\tau)}}{e^{(\ell)} - 1/\text{E}} & x_2 \neq x_1 \\[3ex] \dfrac{l_{x_2}^{(\ell)} - 1/\text{E}}{e^{(\ell)} - 1/\text{E}} & x_2 = x_1 \end{cases},$$
(74)

for $x_1, x_2 \in \{1,\ldots,J\}$, where the $1/\text{E}$ terms appear due to the fact that the DD has to be reparameterized after we condition on $X_1 = x_1$. The above expression can be generalized to any of the factors in (73) as follows:

$$\begin{aligned}
P(X_u = x_u|X_1 = x_1,\ldots,X_{u-1} = x_{u-1}) &= \frac{l_{x_u}^{(\ell)} - \dfrac{\sum_{u'=1}^{u} \mathbb{I}[x'_u = x_u]}{\text{E}}}{e^{(\tau)} - \dfrac{u-1}{\text{E}}} \\[3ex]
&= \left(\frac{l_{x_u}^{(\ell)}}{e^{(\ell)}} - \frac{\sum_{u'=1}^{u} \mathbb{I}[x'_u = x_u]}{e^{(\ell)}\text{E}}\right) \frac{e^{(\ell)}\text{E}}{e^{(\ell)}\text{E} - (u-1)}.
\end{aligned}$$
(75)

Note that $e^{(\tau)}\mathsf{E}$ is the number of edges in the graph at time ℓ. Since $u \le w < J$ and J is a constant independent of E, the second factor in (75) is of order $1 - \mathcal{O}(1/\mathsf{E})$. Thus

$$P(X_1 = x_1, \ldots, X_w = x_w) = \prod_{u=1}^{w} \left(\frac{l_{x_u}^{(\ell)}}{e^{(\ell)}} - \frac{\sum_{u'=1}^{u} \mathbb{I}[x'_u = x_u]}{e^{(\ell)}\mathsf{E}} \right) + \mathcal{O}(1/\mathsf{E}), \qquad (76)$$

using again that $w \le \mathtt{d}+1 \le J$ where J is a constant independent of E, and that $l_{x_u}^{(\ell)}/e^{(\ell)}$ is independent of E, we can write (73) as follows

$$P(X_1 = x_1, \ldots, X_w = x_w) = \prod_{u=1}^{w} \frac{l_{x_u}^{(\ell)}}{e^{(\ell)}} + \mathcal{O}(1/\mathsf{E}). \qquad (77)$$

Thus, the joint probability distribution of the left degrees of w edges connected to a degree-w GC node asymptotically factorizes as $\mathsf{E} \to \infty$ and the number of edges with left degree-i connected to the removed GC node can be roughly described by a binomial RV with parameter $l_i^{(\ell)}/e^{(\ell)}$. Hence, we obtain

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}(\mathsf{E}), \mathrm{Deg}_{cw} \right] = -\frac{iwl_i^{(\ell)}}{e^{(\tau)}} + \mathcal{O}(1/\mathsf{E}). \qquad (78)$$

Combining (78) and (72) with (71), we obtain

$$\mathbb{E}\left[L_i^{(\ell+1)} - L_i^{(\ell)} \middle| \mathcal{G}^{(\ell)}(\mathsf{E}) \right] = -\frac{il_i^{(\ell)}}{e^{(\tau)}} \left(P_{p1}^{(\ell)} + \sum_{w=1}^{\mathtt{d}+1} wP_{cw}^{(\ell)} \right) + \mathcal{O}(1/\mathsf{E}) \triangleq f_i(\mathcal{G}^{(\ell)}(\mathsf{E})/\mathsf{E}) + \mathcal{O}(1/\mathsf{E}). \qquad (79)$$

Note that $f_i(\mathcal{G}^{(\ell)}(\mathsf{E})/\mathsf{E})$ depends on every component in $\mathcal{G}^{(\ell)}$, normalized by E. Observe that $f_i(\mathcal{G}^{(\ell)}(\mathsf{E})/\mathsf{E})$ in (79) is of the form required by Condition 2) of Wormald's theorem.

2) Evolution of right edge degrees in the Tanner graph after one P-PD iteration: Our goal now is to evaluate

$$\mathbb{E}\left[R_{pj}^{(\ell+1)} - R_{pj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}(\mathsf{E}) \right], \quad j = 1, \ldots, K,$$

$$\mathbb{E}\left[R_{cj}^{(\ell+1)} - R_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}(\mathsf{E}) \right], \quad j = 1, \ldots, K \text{ and } j \notin \{\mathtt{d}, \mathtt{d}+1\}$$

$$\mathbb{E}\left[\hat{R}_{cj}^{(\ell+1)} - \hat{R}_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}(\mathsf{E}) \right], \quad j \in \{\mathtt{d}, \mathtt{d}+1\}$$

$$\mathbb{E}\left[\bar{R}_{cj}^{(\ell+1)} - \bar{R}_{cj}^{(\ell)} \middle| \mathcal{G}^{(\ell)}(\mathsf{E}) \right], \quad j \in \{\mathtt{d}, \mathtt{d}+1\}$$

As before, we evaluated these terms by conditioning on the type of check node to be removed at the current P-PD iteration. Using (77), the average number of edges removed from the graph after a degree-w GC node is removed is given by $\Delta_w^{(\tau)} \triangleq wa^{(\ell)} + \mathcal{O}(1/\mathsf{E})$, where $a^{(\ell)} = \sum il_i^{(\ell)}/e^{(\ell)}$. Among those, w are connected to the same degree-w GC node, i.e. they have right degree w. Consider the remaining $\Delta_w - w$ edges. Following a similar argument as in (77), it can be shown that the joint probability distribution of their right degree asymptotically factorizes as $\mathsf{E} \to \infty$ and that the deviation in the finite case is dominated by $\mathcal{O}(1/\mathsf{E})$ terms. By taking $w = 1$, the same arguments hold for the case where decoder

removes a degree-1 SPC node. In addition to this results, in order to evaluate the expected variation in the number of edges of certain right degree we also have to take into account that, when we remove one edge from the graph, we modify the right degree of the rest of edges still connected to the same SPC/GC node. For example, if one of the edges that are removed from the graph has right SPC degree j, after deleting such edge the graph loses j edges with right SPC degree j and gains $j-1$ edges with right SPC degree $j-1$.

Following the above arguments, conditioned on $\mathcal{G}^{(\ell)}(\mathrm{E})$, the expected change in the number of edges with right SPC degree j is given by the following expression

$$
\mathbb{E}\left[R_{pj}^{(\ell+1)} - R_{pj}^{(\ell)}\middle|\mathcal{G}^{(\ell)}(\mathrm{E})\right]
$$

$$
= P_{p1}^{(\ell)}\left((r_{p(j+1)}^{(\ell)} - r_{pj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)} - \mathbb{I}[j=1]\right) + \sum_{w=1}^{\mathtt{d}+1} P_{cw}^{(\ell)}(r_{p(j+1)}^{(\ell)} - r_{pj}^{(\ell)})\frac{j(wa(\ell)-w)}{e(\ell)} + \mathcal{O}(1/\mathrm{E})
$$

$$
\triangleq g_{pj}(\mathcal{G}^{(\ell)}/\mathrm{E}) + \mathcal{O}(1/\mathrm{E}). \tag{80}
$$

It can be further shown that the expected variation in the number of edges of right GC degree j with $j \neq \mathtt{d}, \mathtt{d}+1$ satisfies

$$
\mathbb{E}\left[R_{cj}^{(\ell+1)} - R_{cj}^{(\ell)}\middle|\mathcal{G}^{(\ell)}(\mathrm{E})\right]
$$

$$
= P_{p1}^{(\ell)}\left((r_{c(j+1)}^{(\ell)} - r_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)}\right) + \sum_{w=1}^{\mathtt{d}+1} P_{cw}^{(\ell)}\left((r_{c(j+1)}^{(\ell)} - r_{cj}^{(\ell)})\frac{j(wa(\ell)-w)}{e(\ell)} - w\mathbb{I}[j=w]\right) + \mathcal{O}(1/\mathrm{E})
$$

$$
\triangleq g_{cj}(\mathcal{G}^{(\ell)}/\mathrm{E}) + \mathcal{O}(1/\mathrm{E}). \tag{81}
$$

To analyze the expected change in the number of edges connected to decodable and not decodable GC nodes of degree $\mathtt{d}$ and $\mathtt{d}+1$, we have to take into account that if a non-decodable degree-$(d+2)$ GC node loses one edge, it becomes decodable with probability $p_{\mathtt{d}+1}$. Similarly, if a non-decodable degree-$(d+1)$ GC node loses one edge, it becomes decodable with probability $p_{\mathtt{d}}$. Also note that if a decodable GC node of degree $\mathtt{d}+1$ loses one edge, it becomes a decodable GC node of degree $\mathtt{d}$ with probability 1. It follows that the expected change in the fraction of edges connected to decodable and not decodable GC nodes of degree $j = \mathtt{d}, \mathtt{d}+1$, are given by

$$
\mathbb{E}\left[\hat{R}_{cj}^{(\ell+1)} - \hat{R}_{cj}^{(\ell)}\middle|\mathcal{G}^{(\ell)}(\mathrm{E})\right]
$$

$$
= P_{p1}^{(\ell)}\left((p_j\bar{r}_{c(j+1)}^{(\ell)} + \hat{r}_{c(j+1)}^{(\ell)} - \hat{r}_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e(\ell)}\right)
$$

$$
+ \sum_{w=1}^{j+1} P_{cw}^{(\ell)}\left((p_j\bar{r}_{c(j+1)}^{(\ell)} + \hat{r}_{c(j+1)}^{(\ell)} - \hat{r}_{cj}^{(\ell)})\frac{j(wa(\ell)-w)}{e(\ell)} - w\mathbb{I}[w=j]\right) + \mathcal{O}(1/\mathrm{E})
$$

$$
\triangleq \hat{g}_{cj}(\mathcal{G}^{(\ell)}/\mathrm{E}) + \mathcal{O}(1/\mathrm{E}) \tag{82}
$$

$$\mathbb{E}\left[\bar{R}_{cj}^{(\ell+1)} - \bar{R}_{cj}^{(\ell)}\Big|\mathcal{G}^{(\ell)}(\mathtt{E})\right]$$

$$= P_{p1}^{(\ell)}\left(((1-p_j)\bar{r}_{c(j+1)}^{(\ell)} - \bar{r}_{cj}^{(\ell)})\frac{j(a(\ell)-1)}{e^{(\ell)}}\right)$$

$$+ \sum_{w=1}^{j+1} P_{cw}^{(\ell)}\left(((1-p_j)\bar{r}_{c(j+1)}^{(\ell)} - \bar{r}_{cj}^{(\ell)})\frac{j(wa(\ell)-w)}{e^{(\ell)}} - w\mathbb{I}[w=j]\right) + \mathcal{O}(1/\mathtt{E})$$

$$\triangleq \bar{g}_{cj}(\mathcal{G}^{(\ell)}/\mathtt{E}) + \mathcal{O}(1/\mathtt{E}) \tag{83}$$

Note that $\bar{R}_{c(\mathtt{d}+2)}^{(\ell)} = R_{c(\mathtt{d}+2)}^{(\ell)}$ and $\hat{R}_{c(\mathtt{d}+2)}^{(\ell)} = 0$. Further, observe that (79)-(83) are of the form required by Condition 2) of Wormald's theorem.

3) On the Lipschitz continuity of the trend functions in (79)-(83)*:* Condition 3) of Wormald's theorem requires that the trend functions in (79)-(83) are Lipschitz in the set of all possible DDs. First, we note that if we would restrict the P-PD to remove only decodable check nodes (either degree-1 SPC nodes or GC nodes of one particular degree), then (79)-(83) are still valid by simply setting the corresponding probabilities $P_{p1}^{(\ell)}$ and $P_{cj}^{(\ell)}$, $j = 1, \ldots, \mathtt{d}+1$ to either zero or one. In such a case, (79)-(83) are equal up to a multiplicative constant to the PD trend functions for LDPC codes in [11], hence they are Lipschitz continuous. When we drop the restriction to remove one particular type of decodable check node, then the trend functions in (79)-(83) are convex the combinations of Lipschitz continuous functions, with the coefficients given by the functions $P_{p1}^{(\ell)}$ and $P_{cj}^{(\ell)}$, $j = 1, \ldots, \mathtt{d}+1$ in (66)-(68), which are also Lipschitz continuous (note their similarity in form with (72), which is Lipschitz continuous [11]). Since they are all bounded functions, we conclude that Condition 3) of Wormald's theorem is also satisfied.

APPENDIX B

PROOF OF THEOREM 5

The proof of Theorem 5 closely follows that of Theorem 4 given in Appendix A. As before, it is sufficient to show that the conditions of Wormald's theorem introduced in Appendix A-A are satisfied.

Following the definitions given in Section VIII-B, the left DD of the residual graph of the $\mathcal{C}_{3,K,\nu,\beta}$ code ensemble during P-PD has three components: the number of edges connected to degree-2 or degree-3 RV nodes ($L_{r2}^{(\ell)}$ and $L_{r3}^{(\ell)}$ respectively), and the number of edges connected to degree-3 GV nodes ($L_{g3}^{(\ell)}$). The right DD of the residual graph has the same elements as those defined for the $\mathcal{C}_{J,K,\nu}$ ensemble in Appendix A-B. Thus, the DD of the residual graph is defined by the random process

$$\mathcal{G}^{(\ell)}(\mathtt{E}) = \left\{L_{r2}^{(\ell)}, L_{r3}^{(\ell)}, L_{3g}^{(\ell)}, R_{pj}^{(\ell)}, R_{cj}^{(\ell)}, \hat{R}_{c\mathtt{d}}^{(\ell)}, \bar{R}_{c\mathtt{d}}^{(\ell)}, \hat{R}_{c(\mathtt{d}+1)}^{(\ell)}, \bar{R}_{c(\mathtt{d}+1)}^{(\ell)}\right\}_{j=1,\ldots,\mathtt{d}-1,\mathtt{d}+2,\ldots,K}. \tag{84}$$

We define

$$l_{r2}^{(\ell)} \triangleq \frac{L_{r2}^{(\ell)}}{\mathtt{E}}, \quad l_{r3}^{(\ell)} \triangleq \frac{L_{r3}^{(\ell)}}{\mathtt{E}}, \quad l_{g3}^{(\ell)} \triangleq \frac{L_{g3}^{(\ell)}}{\mathtt{E}}. \tag{85}$$

After P-PD initialization, i.e. $\ell = 0$, it can be shown that

$$\mathbb{E}\left[l_{g3}^{(0)}\right] = \epsilon^2 \beta, \tag{86}$$

$$\mathbb{E}\left[l_{r3}^{(0)}\right] = \epsilon(1 - \beta), \tag{87}$$

$$\mathbb{E}\left[l_{r2}^{(0)}\right] = 4\beta\epsilon(1 - \epsilon)/3. \tag{88}$$

To evaluate (88), we compute the average number of GV nodes for which one of the two DG-LDPC coded bits is received. According to the generator matrix in (38), GV nodes can be viewed as degree-2 variable nodes. Based on (86)-(88), the average fraction of edges remaining in the graph after P-PD initialization is

$$\epsilon' = \epsilon(1 - \beta) + 4\beta\epsilon(1 - \epsilon)/3 + \epsilon^2\beta = \epsilon\left(1 + \frac{\beta(1 - \epsilon)}{3}\right). \tag{89}$$

We can further determine expected initial conditions of the right DD of the residual graph after P-PD initialization by using (32) and (34) and replacing ϵ by ϵ'.

By following a similar procedure as in Appendix A-B, it can be shown that conditioned, on $\mathcal{G}^{(\ell)}(\mathrm{E})$, the expected variation in $L_{r2}^{(\ell)}, L_{r3}^{(\ell)}$, and $L_{3g}^{(\ell)}$ after one P-PD iteration is given by

$$\mathbb{E}\left[L_{r3}^{(\ell+1)} - L_{r3}^{(\ell)}\Big|\mathcal{G}^{(\ell)}\right] = -\frac{3l_{r3}^{(\ell)}}{e^{(\ell)}}\left(P_{p1}^{(\ell)} + \sum_{w=1}^{\mathrm{d}+1} wP_{cw}^{(\ell)}\right) + \mathcal{O}(1/\mathrm{E}), \tag{90}$$

$$\mathbb{E}\left[L_{r2}^{(\ell+1)} - L_{r2}^{(\ell)}\Big|\mathcal{G}^{(\ell)}\right] = \left(\frac{2l_{g3}^{(\ell)}}{e^{(\ell)}} - \frac{2l_{r2}^{(\ell)}}{e^{(\ell)}}\right)\left(P_{p1}^{(\ell)} + \sum_{w=1}^{\mathrm{d}+1} wP_{cw}^{(\ell)}\right) + \mathcal{O}(1/\mathrm{E}), \tag{91}$$

$$\mathbb{E}\left[L_{g3}^{(\ell+1)} - L_{g3}^{(\ell)}\Big|\mathcal{G}^{(\ell)}\right] = -\frac{3l_{g3}^{(\ell)}}{e^{(\ell)}}\left(P_{p1}^{(\ell)} + \sum_{w=1}^{\mathrm{d}+1} wP_{cw}^{(\ell)}\right) + \mathcal{O}(1/\mathrm{E}), \tag{92}$$

where $e(\ell) = l_{r3}^{(\ell)} + l_{g3}^{(\ell)} + l_{g3}^{(\ell)}$ and $P_{p1}^{(\ell)}$ and $P_{cw}^{(\ell)}$ are given in (28) and (29) respectively. In (91), we have used that that if a degree-3 GV node loses one edge, then the graph loses 3 edges with left GV degree 3 and gains 2 edges with left RV degree 2. The conditional expected variation of the right DD of the residual graph can be computed using (80)-(83) by taking $a^{(\ell)} = (3l_{r3}^{(\ell)} + 2l_{r2}^{(\ell)} + l_{g3}^{(\ell)})/e^{(\ell)}$. Finally, proving that the conditions in Wormald's Theorem "follows by the same arguments as in the proof of Theorem 4 in Appendix A.

APPENDIX C
GENERATOR MATRICES OF REFERENCE CODES

Reference codes have been found by performing an exhaustive search over the database [23], [24], which implements MAGMA [25] to design block codes with the largest minimum distance.

Code R-I: Rate-1/2 Hamming $(6,3)$ linear block code with generator matrix

$$
G_{\text{R-I}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{93}
$$

Code R-II: Rate-1/3 Cordaro-Wagner 2-dimensional repetition code of length 6 with generator matrix

$$
G_{\text{R-II}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{94}
$$

Code R-III: Rate-4/7 Hamming (7,4) code with generator matrix

$$
G_{\text{R-III}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{95}
$$

Code R-IV: Rate-3/7 linear block code with generator matrix

$$
G_{\text{R-IV}} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{96}
$$

Code R-V: Rate-1/2 extended $(7,4)$-Hamming code with extra parity bit, i.e., $(8,4)$ Hamming code. Another example is a Quasi-Cyclic $(8,4,4)$ code with generator matrix

$$
G_{\text{R-V}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \tag{97}
$$

Code R-VI: Rate-3/8 cyclic linear block code with generator matrix

$$G_{\text{R-VI}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{98}$$

Code R-VII: Rate-1/4 Cordaro-Wagner 2-dimensional repetition code length 8 with generator matrix

$$G_{\text{R-VII}} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \tag{99}$$

Code R-VIII: Rate-11/15 linear block code with generator matrix

$$G_{\text{R-VIII}} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{100}$$

Code R-IX: Rate-2/3 linear block code with generator matrix

$$G_{\text{R-IX}} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{101}$$

REFERENCES

[1] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533 – 547, Sept. 1981.

[2] M. Lentmaier and K. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," *IEEE Communications Letters*, vol. 3, no. 8, pp. 248–250, Aug 1999.

[3] G. Yue, L. Ping, and X. Wang, "Generalized Low-Density Parity-Check Codes Based on Hadamard Constraints," *IEEE Transactions on Information Theory,*, vol. 53, no. 3, pp. 1058–1079, March 2007.

[4] G. Liva, W. Ryan, and M. Chiani, "Quasi-cyclic generalized LDPC codes with low error floors," *IEEE Transactions on Communications*, vol. 56, no. 1, pp. 49–57, January 2008.

[5] Paolini, E. and Fossorier, M.P.C. and Chiani, M., "Generalized and Doubly Generalized LDPC Codes With Random Component Codes for the Binary Erasure Channel," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1651–1672, April 2010.

[6] D. Mitchell, M. Lentmaier, and D. Costello, "On the minimum distance of generalized spatially coupled LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT), Istanbul, Turkey*, July 2013, pp. 1874–1878.

[7] S. Abu-Surra, D. Divsalar, and W. E. Ryan, "Enumerators for protograph-based ensembles of LDPC and generalized LDPC codes," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 858–886, Feb. 2011.

[8] M. Lentmaier and G. Fettweis, "On the thresholds of generalized LDPC convolutional codes based on protographs," in *Proc. IEEE International Symposium on Information Theory Proceedings (ISIT), Austin, USA.*, June 2010, pp. 709–713.

[9] Y. Y. Jian, H. D. Pfister, and K. R. Narayanan, "Approaching capacity at high rates with iterative hard-decision decoding," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, July 2012, pp. 2696–2700.

[10] M. F. Flanagan, E. Paolini, M. Chiani, and M. P. C. Fossorier, "Spectral shape of doubly-generalized ldpc codes: Efficient and exact evaluation," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7212–7228, Nov 2013.

[11] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569 –584, Feb. 2001.

[12] C. Measson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5277 –5307, Dec. 2008.

[13] P. Olmos, D. Mitchell, and J. Costello, D.J., "Analyzing the finite-length performance of generalized LDPC codes," in *Proc. 2015 IEEE International Symposium on Information Theory (ISIT), Hong Kong, China*, June 2015, pp. 2683–2687.

[14] T. J. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, Mar. 2008.

[15] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*. North-Holland Pub. Co. New York, 1977.

[16] W. C. Huffman and V. Pless, *Fundamentals of error-correcting codes*. Cambridge, U.K., New York: Cambridge University Press, 2003.

[17] D. Mitchell, M. Lentmaier, A. Pusane, and D. Costello, "Randomly Punctured LDPC Codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 408–421, Feb 2016.

[18] Y. Wang and M. Fossorier, "Doubly generalized ldpc codes," in *2006 IEEE International Symposium on Information Theory*, July 2006, pp. 669–673.

[19] ——, "Doubly Generalized LDPC Codes over the AWGN Channel," *IEEE Transactions on Communications*, vol. 57, no. 5, pp. 1312–1319, May 2009.

[20] R. G. Gallager, *Low Density Parity Check Codes*. MIT Press, 1963.

[21] N. Miladinovic and M. Fossorier, "Generalized LDPC codes and generalized stopping sets," *IEEE Transactions on Communications*, vol. 56, no. 2, pp. 201–212, February 2008.

[22] N. C. Wormald, "Differential equations for random processes and random graphs," *Annals of Applied Probability*, vol. 5, no. 4, pp. 1217–1235, 1995.

[23] M. Grassl, "Bounds on the minimum distance of linear codes and quantum codes," Online available at http://www.codetables.de, 2007, accessed on 2017-01-07.

[24] ——, "Searching for linear codes with large minimum distance," in *Discovering Mathematics with Magma — Reducing the Abstract to the Concrete*, ser. Algorithms and Computation in Mathematics, W. Bosma and J. Cannon, Eds. Heidelberg: Springer, 2006, vol. 19, pp. 287–313.

[25] W. Bosma, J. Cannon, and C. Playoust, "The Magma Algebra System I: The User Language," *Journal of Symbolic Computation*, vol. 24, no. 3-4, pp. 235–265, Oct. 1997.

[26] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 473 –498, Feb. 2009.

[27] I. Andriyanova and J. P. Tillich, "Designing a good low-rate sparse-graph code," *IEEE Transactions on Communications*, vol. 60, no. 11, pp. 3181–3190, November 2012.

[28] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. Sh. Zigangirov, "Iterative Decoding Threshold Analysis for LDPC Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5274–5289, 2010.

[29] P. M. Olmos and R. Urbanke, "A Scaling Law to Predict the Finite-length Performance of Spatially Coupled LDPC Codes." *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3164 – 3184, 2015.

[30] M. Stinner and P. M. Olmos, "On the Waterfall Performance of Finite-Length SC-LDPC Codes Constructed From Protographs," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 345–361, Feb 2016.