

Degree-degree Correlated Low-density Parity-check Codes and Their Extensions

Hsiao-Wen Yu, Cheng-En Lee, Ruhui Zhang, Cheng-Shang Chang, and Duan-Shin Lee

Abstract—Most existing work on analyzing the performance of a random ensemble of low-density parity-check (LDPC) codes assumes that the degree distributions of the two ends of a randomly selected edge are independent. In this paper, we go one step further by considering ensembles of LDPC codes with degree-degree correlations. We propose two methods to construct such an ensemble of degree-degree correlated LDPC codes and derive a system of density evolution equations for these codes over a binary erasure channel (BEC). By conducting extensive numerical experiments, we demonstrate how the degree-degree correlation affects the performance of LDPC codes. Our numerical results suggest that LDPC codes with negative degree-degree correlation could enhance the maximum tolerable erasure probability. Moreover, increasing the negative degree-degree correlation could facilitate better unequal error protection (UEP) design. In the final part of our extension efforts, we extend degree-degree correlated LDPC codes to multi-edge type LDPC codes and leverage these to construct convolutional LDPC codes.

Index Terms—Low-density parity-check codes, unequal error protection.

I. INTRODUCTION

LOW-density parity-check (LDPC) codes, first introduced by R. Gallager in 1962 [2], have been widely used in practice, including the 5G new radio wireless communication standard (see, e.g., [3], [4]), application-specific integrated circuit (ASIC) implementation, and flash-memory systems (see, e.g., [5], [6]). Crucial for efficient and reliable communication systems, especially in 5G applications, LDPC codes offer improved error correction capabilities and higher throughput. Their flexible structure enables efficient implementation in ASICs, supporting the diverse and high-performance demands of 5G networks [7]. A recent application of LDPC codes for rate-diverse multiple access is detailed in [8].

LDPC codes are linear block codes that can be characterized by either a generator matrix or a parity-check matrix. It is well-known that there is a one-to-one mapping between the generator matrix and the parity-check matrix (see, e.g.,

Section II-B of the survey paper [9]). In particular, for an (n, k) -LDPC code, there are k information bits in a codeword of n bits. The $n \times k$ generator matrix then maps the k information bits to the n codeword bits by multiplying the vector of the information bits with the generator matrix. On the other hand, the $n \times (n - k)$ parity-check matrix can be used for checking whether an n -bit binary vector is a legitimate codeword and decoding the k information bits from that codeword.

The $n \times (n - k)$ parity-check matrix associated with an (n, k) -LDPC code can be represented by a bipartite graph, commonly known as the Tanner graph [10]. For such a bipartite graph, there are n variable nodes on one side and $n - k$ check nodes on the other side. A check node represents a constraint on the values of variable nodes connected to that check node. In particular, for binary-valued variables, the value of a check node is simply the checksum of the values of the variable nodes connected to that check node. For a legitimate codeword, the checksums of the $n - k$ check nodes are all 0's. In other words, multiplying a codeword with the parity-check matrix yields a vector with all 0's.

Performance analysis of LDPC codes over a binary erasure channel (BEC) has been studied extensively in the literature (see, e.g., [11]–[14]). For a BEC with the erasure probability δ , each bit in a codeword is erased *independently* with probability δ . To recover the erased code bits, one can apply an iterative decoding (peeling) algorithm described below. One first adds the values of the non-erased variable nodes to the check nodes connected to them. Then remove all these non-erased nodes and their edges. The remaining bipartite graph only consists of erased variable nodes. An erased variable node connected to a check node with degree 1 can be decoded by the value of that check node. Once decoded, it becomes a non-erased variable node and can be removed in the same way as the initial non-erased variable nodes. The process is repeated until no more erased variable nodes can be decoded.

A probabilistic framework for analyzing the iterative algorithm is known as the *density evolution* method [12]–[14]. Such a method tracks the probability that the variable end of a randomly selected edge is not decoded after each iteration. The evolution of such a probability is characterized by the degree distribution of a randomly selected variable node and that of a randomly selected check node. The density evolution method can also be extended to provide unequal error protection of code bits [15]–[19]. For the setting with unequal error protection, code bits are classified into several classes, and the density evolution method tracks the probability that the variable end of a randomly selected edge *in each class* is not

Manuscript received July 24, 2023; revised April 9, 2024; approved for publication by Tarable, Alberto, Division I Editor, June 4, 2024.

Part of this work was presented in 2023 IEEE International Symposium on Information Theory (ISIT) [1]. This work was supported in part by the National Science and Technology, Taiwan, under Grant 111-2221-E-007-038-MY3, and in part by Qualcomm Technologies under Grant SOW NAT-487844-2.

The authors are with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 300044, Taiwan R.O.C. email: s109064701@m109.nthu.edu.tw, {benny_110065508, huibrana}@gapp.nthu.edu.tw, cschang@ee.nthu.edu.tw, and lds@cs.nthu.edu.tw.

H.-W. Yu is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2024.000030

Creative Commons Attribution-NonCommercial (CC BY-NC).

This is an Open Access article distributed under the terms of Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided that the original work is properly cited.

decoded after each iteration.

One important ensemble of random LDPC codes is the class of *irregular* LDPC codes that are characterized by the generating function of the (excess) degree distribution of the variable nodes $\lambda(z)$ and the generating function of the (excess) degree distribution of the check nodes $\rho(z)$. From these two generating functions, the number of stubs (or known as sockets) for each node is specified, and a random bipartite graph is then generated by randomly pairing these stubs. Such a method of generating a random bipartite graph is known as the *configuration model* in the book [20]. For an irregular LDPC code over a BEC with the erasure probability δ , it is known (see, e.g., [13]) that every code bit can be recovered (with high probability) if

$$\delta\lambda(1 - \rho(1 - z)) < z \quad (1)$$

for all $z \in (0, \delta)$. The largest δ satisfying (1) is called the maximum tolerable erasure probability δ^* (also known as the percolation threshold). The optimization problem is to find the two degree distributions $\lambda(z)$ and $\rho(z)$ so that δ^* can be maximized with a given code rate k/n .

LDPC codes are known to be (nearly) capacity-achieving codes when the degree distributions are optimized [21]–[23]. However, the optimized degree distributions are in general highly irregular [13], [23]–[25], and the throughput performance of regular LDPC codes is worse than irregular LDPC codes. The relaxation from *independent* degree distributions to a general bivariate distribution allows us to improve the performance of LDPC codes in the literature. To further improve the performance of LDPC codes, the framework of multi-edge type LDPC (MET-LDPC) codes was proposed in [26]. Instead of using a single edge type in the irregular LDPC codes, there are multiple edge types in the MET-LDPC codes. Stubs of the same edge type are paired together to form edges in the bipartite graph. For an MET-LDPC codes with n_e edge types, one has the freedom to specify the probability of a node to have d_k type k edges, $k = 1, 2, \dots, n_e$. However, these probabilities need to be carefully chosen so that the number of stubs of the same type at the variable nodes is the same as that at the check nodes (the socket count equality constraint in [26]). Moreover, such a generalization also complicates the analysis as both the degree distributions of the variable nodes and the check nodes are now functions with n_e variables. As the densities of various edge types are convolved [26], the density evolution method is generally difficult to be applied to the MET-LDPC codes, and one has to resort to approximation algorithms (see, e.g., [27]). However, the optimal percolation thresholds found by approximation algorithms are often different from the true ones [27]. Another drawback of MET-LDPC codes is that it is not straightforward to improve the performance directly from a good irregular LDPC code with two given degree distributions $\lambda(z)$ and $\rho(z)$.

Our idea to alleviate the problems of MET-LDPC codes is to exploit degree-degree correlations of bipartite graphs. In an irregular LDPC code, the two degree distributions $\lambda(z)$ and $\rho(z)$ of the two ends of a randomly selected edge are assumed to be *independent*. Instead of characterizing a random

ensemble of LDPC codes by two independent degree distributions, we characterize it by a general degree-degree bivariate distribution (of the two ends of a randomly selected edge). An ensemble of LDPC codes with a given degree-degree bivariate distribution is called a *degree-degree correlated LDPC code* in this paper.

By classifying edges with the same degrees in the variable end and the check end into one specific type of edges, a degree-degree correlated LDPC code can be viewed as a variant of MET-LDPC codes with multiple variable node types and multiple check node types. Note that the original MET-LDPC codes in [26] has only one check node type. Degree-degree correlated LDPC codes have the following advantages over the MET-LDPC codes: (i) The bivariate distribution automatically satisfies the socket count equality constraints, (ii) the density evolution analysis is a simple extension of that for irregular LDPC codes, (iii) the exact percolation threshold can be easily computed for a BEC, and (iv) it is straightforward to improve the performance directly from a good irregular LDPC code with two given degree distributions $\lambda(z)$ and $\rho(z)$.

The main objective of this paper is to study the effect of the degree-degree correlation of a randomly selected edge on LDPC codes over a BEC. We summarize our contributions as follows:

- **Construction:** As a generalization of the configuration model (with independent degree distributions) [20], we propose two constructions of random LDPC codes (bipartite graphs) with degree-degree correlations. The first construction, called the block construction in this paper, is an extension of the construction for uni-partite graphs in [28]. For such a model, we derive a closed-form expression of the degree-degree bivariate distribution of the two ends of a randomly selected edge. The second construction, called the general construction, is more general than the first one, and it can construct random LDPC codes with a specified degree-degree bivariate distribution. By classifying edges with the same degrees in the variable end and the check end into one specific type of edges, the second construction can be viewed as a special case of the multi-edge type LDPC codes in [26].
- **Analysis:** It is known that the density evolution method is generally difficult to be applied to the multi-edge type LDPC codes as the densities of various types are convolved. However, by using the degree-degree bivariate distribution, we can average over the (convolved) densities of various *edge* types into a single *node* type. As such, we derive a system of density evolution equations over a BEC. Such an extension covers the independent case as a special case. From these systems of density evolution equations, we derive lower bounds on the maximum tolerable erasure probability (such that every code bit is recovered with high probability).
- **Effect of correlation:** By conducting extensive numerical experiments, we show the effect of the degree-degree correlation on the performance of LDPC codes. Our numerical results show that optimizing *negative* correlation can achieve a much higher maximum tolerable erasure proba-

bility than LDPC codes with independent degree distributions. This implies that *partially regular* LDPC codes [16] with degree-degree correlation might be good enough for practical use.

- **Performance improvement:** By taking the degree-degree correlation into consideration, we can further improve the best-known result for the maximum tolerable erasure probability in the literature. In particular, we show that under the same marginal degree distributions of the variable ends and the check ends in [29], the maximum tolerable erasure probability can be extended from 0.49553 to 0.49568.
- **Generalization:** We propose generalizing degree-degree correlated LDPC codes to MET-LDPC codes with multiple variable node types and multiple check node types. Unlike the original construction of MET-LDPC codes that have fixed edge types in [26], the ensemble of our MET-LDPC codes is generated with independently selected edge types. Since edge types are independently selected, they can be averaged using conditional probabilities, as in our analysis for degree-degree correlated LDPC codes. As illustrative examples, we show how convolutional LDPC codes (see, e.g., [30], [31]) and the spatially coupled IRSA with multiple classes of users and receivers [32], [33] can be constructed using this extension.

The rest of the paper is organized as follows. In Section II, we propose two constructions for degree-degree correlated random LDPC codes. In Section III, we conduct the density evolution analysis for LDPC codes with degree-degree correlations. In Section IV, we generalize degree-degree correlated LDPC codes to MET-LDPC codes with independently selected edge types, and this includes the convolutional LDPC codes and spatially coupled IRSA with multiple classes of users and receivers. We evaluate the performance of the degree-degree correlated LDPC codes in Section V. The paper concludes in Section VI, where we discuss potential future research directions.

II. CONSTRUCTION OF DEGREE-DEGREE CORRELATED RANDOM LDPC CODES

Instead of assuming that the degree of the variable end and the degree of the check end of a randomly selected edge are independent in [13]. In this section, we construct a random bipartite graph with a specific degree-degree correlation. Suppose that there are n variable nodes and $n - k$ check nodes. Let $p_X(x)$ (resp. $p_Y(y)$) be the degree distribution of a randomly selected variable node X (resp. check node Y). Also, let $E[X] = \sum_x xp_X(x)$ (resp. $E[Y] = \sum_y yp_Y(y)$) be the average degree of a randomly selected variable node (resp. check node). Following the argument for the configuration model in [20], we generate $np_X(x)$ variable nodes with degree (stub) x and $(n - k)p_Y(y)$ check nodes with degree (stub) y . Then the total number of stubs for variable nodes is $\sum_x nxp_X(x)$, and the total number of stubs for check nodes is $\sum_y (n - k)yp_Y(y)$. Suppose that

$$nE[X] = (n - k)E[Y]. \quad (2)$$

In this paper, we let

$$G = \frac{n}{n - k} = \frac{E[Y]}{E[X]} \quad (3)$$

when the identity in (2) is satisfied. Note that the code rate k/n is simply $1 - 1/G$.

A. Block Construction

In this section, our construction for such a bipartite graph is similar to the block construction of degree-degree correlated random networks in [28]. Assume that (2) holds. We arrange the $nE[X]$ stubs of the variable nodes in descending order and partition the $nE[X]$ stubs into b blocks evenly, each with $nE[X]/b$ stubs. In each block of $nE[X]/b$ stubs, we *randomly* select $qnE[X]/b$ stubs as type 1 stubs, where the parameter q is a design parameter for controlling the degree-degree correlation. The remaining $(1 - q)nE[X]/b$ stubs in that block are classified as type 2 stubs. Similarly, we arrange the $(n - k)E[Y]$ stubs of the check nodes in descending order and partition the $(n - k)E[Y]$ stubs into b blocks evenly, each with $(n - k)E[Y]/b$ stubs. In each block of $(n - k)E[Y]/b$ stubs, we *randomly* select $q(n - k)E[Y]/b$ stubs as type 1 stubs and the remaining $(1 - q)(n - k)E[Y]/b$ stubs as type 2 stubs. These two types of stubs will be connected differently to form degree-degree correlation.

Consider a permutation π of $\{1, 2, \dots, b\}$. For $i = 1, 2, \dots, b$, we randomly select a type 1 stub from the i th block of variable nodes and another type 1 stub from the $\pi(i)$ th block of check nodes and connect these two stubs to form an edge. Repeat the process until all the type 1 stubs are connected. For type 2 stubs, the connection is independent, i.e., we randomly select a type 2 stub of variable nodes and another type 2 stub of check nodes and connect these two stubs to form an edge. It is clear that the marginal distribution of the degree of the variable end (of a randomly selected edge) is

$$P(X_e = x) = \frac{xp_X(x)}{E[X]}, \quad (4)$$

and the marginal distribution of the degree of the check end (of a randomly selected edge) is

$$P(Y_e = y) = \frac{yp_Y(y)}{E[Y]}. \quad (5)$$

Assume that all the nodes with the same degree are in one common block. Let $S_X(i)$ (resp. $S_Y(i)$), $i = 1, 2, \dots, b$, be the set of degrees in the i th block of variable nodes (resp. check nodes). Now we compute the conditional probability $P(Y_e = y | X_e = x)$. Suppose that $x \in S_X(i)$ for some i . If $y \notin S_Y(\pi(i))$, then an edge with degree x at the variable end and degree y at the check end can only be formed by type 2 stubs. The number of type 2 stubs of the check nodes with degree y is $(1 - q)(n - k)yp_Y(y)$, and the total number of type 2 stubs is $(1 - q)(n - k)E[Y]$. The probability that a stub of

the variable nodes is of type 2 is $1 - q$. Thus, for $x \in S_X(i)$ and $y \notin S_Y(\pi(i))$,

$$\begin{aligned} P(Y_e = y | X_e = x) &= (1 - q) \frac{(1 - q)(n - k)yp_Y(y)}{(1 - q)(n - k)E[Y]} \\ &= \frac{(1 - q)yp_Y(y)}{E[Y]}. \end{aligned} \quad (6)$$

If $y \in S_Y(\pi(i))$, then an edge with degree x at the variable end and degree y at the check end can also be formed by type 1 stubs. The number of type 1 stubs of the check nodes with degree y is $q(n - k)yp_Y(y)$, and the total number of type 1 stubs in the $\pi(i)$ th block of check nodes is $q(n - k)E[Y]/b$. The probability that a stub of the variable nodes is of type 1 is q . Adding the probability formed by type 1 stubs in (6), we have for $x \in S_X(i)$ and $y \in S_Y(\pi(i))$,

$$\begin{aligned} P(Y_e = y | X_e = x) &= \frac{q(n - k)yp_Y(y)}{q(n - k)E[Y]/b} + \frac{(1 - q)yp_Y(y)}{E[Y]} \\ &= \frac{(bq + (1 - q))yp_Y(y)}{E[Y]}. \end{aligned} \quad (7)$$

Then, we have from (4), (5), (6), and (7) that for $x \in S_X(i)$,

$$\begin{aligned} P(X_e = x, Y_e = y) &= \begin{cases} (bq + (1 - q))p_{X_e}(x)p_{Y_e}(y), & y \in S_Y(\pi(i)), \\ (1 - q)p_{X_e}(x)p_{Y_e}(y), & y \notin S_Y(\pi(i)). \end{cases} \end{aligned} \quad (8)$$

Example 1: (Two types of degrees) Suppose that there are $n/3$ variable nodes with degree $2d$, and $2n/3$ variable nodes with degree d . Also, there are $(n - k)/3$ check nodes with degree $2Gd$, and $2(n - k)/3$ check nodes with degree Gd , where $G = n/(n - k)$ is a positive integer. Using the construction in Section II-A, we have $E[X] = 4d/3$, $E[Y] = 4Gd/3$ so that the condition in (2) is satisfied. Now we partition the stubs into two blocks, i.e., $b = 2$. The first block consists of variable nodes with degree $2d$ (resp. check nodes with degree $2Gd$), and the second block consists of variable nodes with degree d (resp. check nodes with degree Gd). For the permutation π with $\pi(1) = 2$ and $\pi(2) = 1$, we have from (4) and (8) that

$$\begin{aligned} P(Y_e = 2Gd | X_e = 2d) &= \frac{1 - q}{2}, \\ P(Y_e = 2Gd | X_e = d) &= \frac{1 + q}{2}, \\ P(Y_e = Gd | X_e = 2d) &= \frac{1 + q}{2}, \\ P(Y_e = Gd | X_e = d) &= \frac{1 - q}{2}. \end{aligned} \quad (9)$$

Note that the degree-degree correlation can be computed as follows:

$$\rho(X_e, Y_e) = \frac{E(X_e Y_e) - E(X_e)E(Y_e)}{\sqrt{\text{Var}(X_e)}\sqrt{\text{Var}(Y_e)}} = -q. \quad (10)$$

Thus, the degree-degree correlation is negatively correlated for the permutation π with $\pi(1) = 2$ and $\pi(2) = 1$. On the other hand, it is positively correlated for the permutation

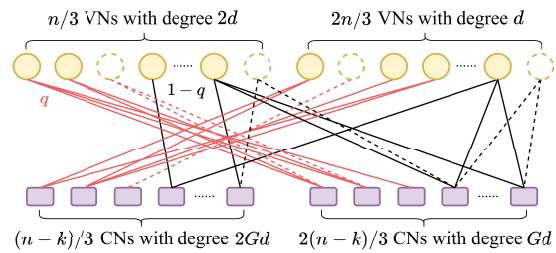


Fig. 1. An illustration of the construction of the bipartite graph in Example 1, where the red (resp. black) edges are formed by type 1 (resp. 2) stubs.

π with $\pi(1) = 1$ and $\pi(2) = 2$. An illustration of the construction of the bipartite graph is shown in Fig. 1, where the red (resp. black) edges are formed by type 1 (resp. 2) stubs.

B. General Construction

In this section, we propose a general construction from the degree-degree bivariate distribution $P(X_e = x, Y_e = y)$. Given such a bivariate distribution, we first compute the two marginal distributions:

$$p_{X_e}(x) = \sum_y P(X_e = x, Y_e = y), \quad (11)$$

and

$$p_{Y_e}(y) = \sum_x P(X_e = x, Y_e = y). \quad (12)$$

Then we use the marginal distribution in (11) and (4) to compute the degree distribution of X :

$$p_X(x) = \frac{p_{X_e}(x)/x}{\sum_{x'} p_{X_e}(x')/x'}. \quad (13)$$

Similarly, we have from the marginal distribution in (12) and (5) that

$$p_Y(y) = \frac{p_{Y_e}(y)/y}{\sum_{y'} p_{Y_e}(y')/y'}. \quad (14)$$

From (13) and (14), we have

$$E[X] = \frac{1}{\sum_{x'} p_{X_e}(x')/x'}, \quad (15)$$

$$E[Y] = \frac{1}{\sum_{y'} p_{Y_e}(y')/y'}. \quad (16)$$

Choose n and k so that the condition in (3) is satisfied. It then follows from (3), (4), and (5) that

$$\begin{aligned} n x p_X(x) P(Y_e = y | X_e = x) &= n E[X] p_{X_e}(x) P(Y_e = y | X_e = x) \\ &= n E[X] P(X_e = x, Y_e = y) \end{aligned} \quad (17)$$

$$\begin{aligned} &= (n - k) E[Y] p_{Y_e}(y) P(X_e = x | Y_e = y) \\ &= (n - k) y p_Y(y) P(X_e = x | Y_e = y). \end{aligned} \quad (18)$$

Note that $n x p_X(x)$ is the number of stubs of variable nodes with degree x , and $(n - k) y p_Y(y)$ is the number of stubs of check nodes with degree y . Among these stubs, we can

randomly select $nxp_X(x)P(Y_e = y|X_e = x)$ stubs from the variable nodes and $(n - k)yp_Y(y)P(X_e = x|Y_e = y)$ stubs from the check nodes, and connect them at random (as in the configuration model). As the number of edges between a variable node with degree x and a check node with degree y is $nxp_X(x)P(Y_e = y|X_e = x)$ and the total number of edges is $nE[X]$, the probability that a randomly selected edge has degree x in its variable end and degree y in its check end is (cf. (17))

$$P(X_e = x, Y_e = y). \quad (19)$$

It is of interest to point out the connections between our construction and the multi-edge type LDPC codes in [26]. By classifying edges with degree x in the variable end and degree y in the check end into one specific type of edges, our construction is a special case of the multi-edge type LDPC codes in [26]. However, as pointed out in [26], [34], the density evolution method is generally difficult to be applied to the multi-edge type LDPC codes (as the densities of various types are convolved). By using the degree-degree bivariate distribution, we will show in Section III that one can average over the (convolved) densities of various *edge* types into a single *node* type (that only depends on its node degree). Thus, the computational complexity of the density evolution method can be greatly reduced.

Example 2: (Two types of degrees) Consider the following bivariate distribution:

$$\begin{aligned} P(X_e = d_1, Y_e = Gd_1) &= p_1, \\ P(X_e = d_1, Y_e = Gd_2) &= p_2, \\ P(X_e = d_2, Y_e = Gd_1) &= p_2, \\ P(X_e = d_2, Y_e = Gd_2) &= 1 - p_1 - 2p_2, \end{aligned} \quad (20)$$

where $0 \leq p_2 \leq 1/2$ and $0 \leq p_1 \leq 1 - 2p_2$.

From (15), it follows that

$$\begin{aligned} E[X] &= \frac{1}{\frac{p_1+p_2}{d_1} + \frac{1-p_1-2p_2}{d_2}}, \\ E[Y] &= \frac{1}{\frac{p_1+p_2}{Gd_1} + \frac{1-p_1-2p_2}{Gd_2}}. \end{aligned} \quad (21)$$

Thus, the condition in (3) is satisfied.

From (20), we know that

$$\begin{aligned} P(Y_e = Gd_2|X_e = d_2) &= \frac{1 - p_1 - 2p_2}{1 - p_1 - p_2}, \\ P(Y_e = Gd_2|X_e = d_1) &= \frac{p_2}{p_1 + p_2}, \\ P(Y_e = Gd_1|X_e = d_2) &= \frac{p_2}{1 - p_1 - p_2}, \\ P(Y_e = Gd_1|X_e = d_1) &= \frac{p_1}{p_1 + p_2}. \end{aligned} \quad (22)$$

It is easy to see that (22) recovers (9) by letting $p_1 = \frac{1-q}{4}$, $p_2 = \frac{1+q}{4}$, $d_1 = 2d$ and $d_2 = d$. As such, the construction in this example is more general than that in Example 1.

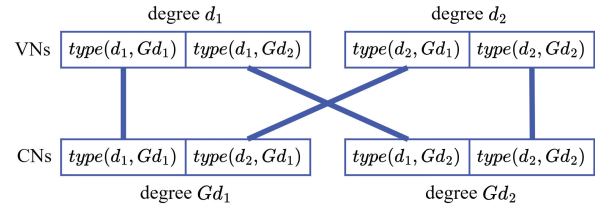


Fig. 2. An illustration of the construction of the bipartite graph, where the bold lines represent edges connected among the four edge types.

Similarly, from (20), it follows that

$$\begin{aligned} P(X_e = d_2|Y_e = Gd_2) &= \frac{1 - p_1 - 2p_2}{1 - p_1 - p_2}, \\ P(X_e = d_2|Y_e = Gd_1) &= \frac{p_2}{p_1 + p_2}, \\ P(X_e = d_1|Y_e = Gd_2) &= \frac{p_2}{1 - p_1 - p_2}, \\ P(X_e = d_1|Y_e = Gd_1) &= \frac{p_1}{p_1 + p_2}. \end{aligned} \quad (23)$$

To construct a random LDPC code with the six input parameters p_1 , p_2 , d_1 , d_2 , G , and n in this example, one can classify the $nE[X]$ edges (with $E[X]$ in (21)) into the four edge types (x, y) (with degree x in the variable end and degree y in the check end): (d_1, Gd_1) , (d_1, Gd_2) , (d_2, Gd_1) and (d_2, Gd_2) . For each edge type, connect the stubs in the variable nodes and the stubs in check nodes by using the configuration model. An illustration of the construction is shown in Fig. 2, where the bold lines represent edges connected among the four edge types.

III. DENSITY EVOLUTION IN CORRELATED LDPC CODES

In this section, we extend the density evolution analysis for LDPC codes with degree-degree correlations. It is known that the density evolution analysis is generally difficult to be applied to the multi-edge type LDPC codes as the densities of various types are convolved. However, by using the conditional probabilities derived in the previous section, we can average over the (convolved) densities of various *edge* types into a single *node* type.

Now we derive the density evolution equations in the setting where G is fixed and $n \rightarrow \infty$. Consider using a degree-degree correlated LDPC code over a binary erasure channel with the erasure probability δ . Let $\alpha_x^{(i)}$ be the probability that the variable end of a randomly selected edge with degree x is not decoded after the i th iteration, and $\beta_y^{(i)}$ be the probability that the check end of a randomly selected edge with degree y is not decoded after the i th iteration. Analogous to the AND-OR argument in [11], [12], we have

$$\alpha_x^{(i)} = \delta \left(\sum_y \beta_y^{(i)} P(Y_e = y|X_e = x) \right)^{x-1}. \quad (24)$$

Clearly, $\alpha_x^{(0)} = \delta$ for all x (as every variable bit is erased independently with probability δ through the erase channel). Similarly,

$$\beta_y^{(i)} = 1 - \left(1 - \sum_x \alpha_x^{(i-1)} P(X_e = x|Y_e = y) \right)^{y-1}. \quad (25)$$

Combining these two equations, we have a system of nonlinear recursive equations:

$$\alpha_x^{(i)} = \delta \left(\sum_y (1 - (1 - \sum_{x'} \alpha_{x'}^{(i-1)}) P(X_e = x' | Y_e = y))^{y-1} \right) \times P(Y_e = y | X_e = x)^{x-1}, \quad (26)$$

with $\alpha_x^{(0)} = \delta$ for all x .

Let $\gamma_x^{(i)}$ be the probability that a randomly selected variable node with degree x is successfully decoded after the i th iteration. Note that a randomly selected variable node is not decoded after the i th iteration if and only if the variable node is erased and all the *check* ends of its x edges are not decoded after the i th iteration. Thus,

$$\gamma_x^{(i)} = 1 - \delta \left(\sum_y \beta_y^{(i)} P(Y_e = y | X_e = x) \right)^x. \quad (27)$$

Let $\gamma^{(i)}$ be the probability that a randomly selected variable node is successfully decoded after the i th iteration. Then

$$\begin{aligned} \gamma^{(i)} &= \sum_x \gamma_x^{(i)} p_X(x) \\ &= 1 - \delta \sum_x \left(\sum_y \beta_y^{(i)} P(Y_e = y | X_e = x) \right)^x p_X(x). \end{aligned} \quad (28)$$

As in [13], we are interested in the maximum tolerable erasure probability δ^* such that for all $\delta < \delta^*$,

$$\lim_{i \rightarrow \infty} \gamma^{(i)} \rightarrow 1. \quad (29)$$

Since the capacity of the BEC with the erasure probability δ is known to be $1 - \delta$ [35], we know that $\frac{k}{n} \leq 1 - \delta^*$. In view of (3), we have

$$\delta^* \leq \frac{1}{G}. \quad (30)$$

In view of (28) and (25), a sufficient condition for (29) is

$$\lim_{i \rightarrow \infty} \alpha_x^{(i)} \rightarrow 0 \quad (31)$$

for all x . In the following theorem, we show a lower bound for δ^* .

Theorem 1: Let $d_{v,\min}$ be the minimum degree of a variable node and $d_{c,\max}$ be the maximum degree of a check node. If $d_{v,\min} \geq 2$ and $\delta < 1/(d_{c,\max} - 1)$, then $\lim_{i \rightarrow \infty} \alpha_x^{(i)} \rightarrow 0$ for all x .

Proof. Note from (25) that

$$\beta_y^{(i)} \leq 1. \quad (32)$$

Since $\alpha_x^{(0)} = \delta$, one can easily show by induction (from (25) and (24)) that for all i

$$\alpha_x^{(i)} \leq \delta. \quad (33)$$

Let $\alpha_{\max}^{(i)} = \max_x \alpha_x^{(i)}$ and $\beta_{\max}^{(i)} = \max_y \beta_y^{(i)}$. Using the inequality that $(1 - z)^k \geq 1 - kz$ for all $z \geq 0$, we have from (25) that

$$\begin{aligned} \beta_y^{(i)} &\leq (y - 1) \sum_x \alpha_x^{(i-1)} P(X_e = x | Y_e = y) \\ &\leq (y - 1) \alpha_{\max}^{(i-1)} \sum_x P(X_e = x | Y_e = y) \\ &\leq (d_{c,\max} - 1) \alpha_{\max}^{(i-1)}. \end{aligned} \quad (34)$$

It follows from (33) that

$$\beta_{\max}^{(i)} \leq (d_{c,\max} - 1) \alpha_{\max}^{(i-1)} \leq (d_{c,\max} - 1) \delta. \quad (35)$$

On the other hand, we have from (24) and (35) that

$$\begin{aligned} \alpha_x^{(i)} &\leq \delta (\beta_{\max}^{(i)} \sum_y P(Y_e = y | X_e = x))^{x-1} \\ &\leq \delta (\beta_{\max}^{(i)})^{x-2} \beta_{\max}^{(i)} \\ &\leq \delta ((d_{c,\max} - 1) \delta)^{x-2} (d_{c,\max} - 1) \alpha_{\max}^{(i-1)} \\ &\leq ((d_{c,\max} - 1) \delta)^{x-1} \alpha_{\max}^{(i-1)}. \end{aligned} \quad (36)$$

Since we assume that $(d_{c,\max} - 1) \delta < 1$, it follows from (36) that

$$\alpha_x^{(i)} \leq ((d_{c,\max} - 1) \delta)^{d_{v,\min}-1} \alpha_{\max}^{(i-1)}. \quad (37)$$

Thus,

$$\alpha_{\max}^{(i)} \leq ((d_{c,\max} - 1) \delta)^{d_{v,\min}-1} \alpha_{\max}^{(i-1)}. \quad (38)$$

Since $d_{v,\min} \geq 2$ and $(d_{c,\max} - 1) \delta < 1$, we have from (38) and $\alpha_x^{(0)} = \delta$ that

$$\alpha_{\max}^{(i)} \leq (((d_{c,\max} - 1) \delta)^{d_{v,\min}-1})^i \delta. \quad (39)$$

Thus, $\alpha_{\max}^{(i)}$ converges to 0 when $i \rightarrow \infty$. \blacksquare

IV. EXTENSIONS

A. MET-LDPC Codes with Independently Selected Edge Types

In this section, we extend the framework for DDC-LDPC codes. For a DDC-LDPC code, we can classify (i) the variable nodes with degree x as type x variable nodes, (ii) the check nodes with degree y as type y check nodes, and (iii) the edges between a variable node with degree x and a check node with degree y as type (x, y) edges. Viewing it this way, the DDC-LDPC code can be considered as a MET-LDPC code with multiple types of variable nodes and check nodes. Inspired by this perspective, we propose to generalize DDC-LDPC codes to MET-LDPC codes with multiple variable node types and multiple check node types. Unlike the original construction of MET-LDPC codes that have fixed edge types in [26], the ensemble of our MET-LDPC codes is generated with independently selected edge types. As edge types are independently selected, they can be averaged using conditional probabilities as described in Section III, thereby significantly reducing the number of recursive equations required for density evolution.

Specifically, we assume there are n_V types of variable nodes, n_C types of check nodes, and n_E types of edges. We specify the following probabilities:

- $p_0(v)$: The probability that a randomly selected variable node is of type v , $v = 1, 2, \dots, n_V$.
- $p_1(d|v)$: The probability that a type v variable node has degree d .
- $p_2(e|v)$: The probability that an edge of a type v variable node is a type e edges, $e = 1, 2, \dots, n_E$.
- $q_0(c)$: The probability that a randomly selected check node is of type c , $c = 1, 2, \dots, n_C$.

- $q_1(d|c)$: The probability that a type c check node has degree d .
- $q_2(e|c)$: The probability that an edge of a type c check node is a type e edges, $e = 1, 2, \dots, n_E$.

When n and $n - k$ are large, there are roughly $np_0(v)$ type v variable nodes and $(n - k)q_0(c)$ type c check nodes. The expected degree of a type v variable node (resp. a type c check node) is $\sum_d dp_1(d|v)$ (resp. $\sum_d dq_1(d|c)$). The expected number of type e edges of a type v variable node is then $\sum_d dp_1(d|v)p_2(e|v)$. Similarly, the expected number of type e edges of a type c check node is $\sum_d dq_1(d|c)q_2(e|c)$. To construct a (n, k) -LDPC code, the number of type e edges of the n variable nodes must be the same as that of the $n - k$ check nodes for all $e = 1, 2, \dots, n_E$. This leads to the following socket (stub) count equality constraints:

$$\begin{aligned} n \sum_{v=1}^{n_V} p_0(v) \sum_d dp_1(d|v)p_2(e|v) \\ = (n - k) \sum_{c=1}^{n_C} q_0(c) \sum_d dq_1(d|c)q_2(e|c), \end{aligned} \quad (40)$$

$e = 1, 2, \dots, n_E$. As in [26], stubs of the same edge type are *randomly* paired together (as in the configuration model) to form edges in the bipartite graph.

One key property that does not exist in the original MET-LDPC codes in [26] is the conditional probabilities defined in the following lemma. We will use the results from this lemma to simplify the derivation of the density evolution equations.

Lemma 2: Let $p(c|v)$ be the probability that a randomly selected edge of a type v variable node is connected to a type c check node. Then

$$p(c|v) = \sum_{e=1}^{n_E} p_2(e|v) \frac{q_0(c) \sum_d dq_1(d|c)q_2(e|c)}{\sum_{c'=1}^{n_C} q_0(c') \sum_d dq_1(d|c')q_2(e|c')}. \quad (41)$$

Let $p(v|c)$ be the probability that a randomly selected edge of a type c check node is connected to a type v variable node. Then

$$p(v|c) = \sum_{e=1}^{n_E} q_2(e|c) \frac{p_0(v) \sum_d dp_1(d|v)p_2(e|v)}{\sum_{v'=1}^{n_V} p_0(v') \sum_d dp_1(d|v')p_2(e|v')}. \quad (42)$$

Proof. We follow the standard argument for the configuration model [20]. Note that the expected number of type e stubs of type c check nodes is

$$(n - k)q_0(c) \sum_d dq_1(d|c)q_2(e|c).$$

Summing over all the n_C types of check nodes, we know that the expected total number of type e stubs of check nodes is

$$(n - k) \sum_{c'=1}^{n_C} q_0(c') \sum_d dq_1(d|c')q_2(e|c').$$

Given that a randomly selected edge of a type v variable node is a type e edge, the probability that it is connected to a type c check node is

$$\frac{q_0(c) \sum_d dq_1(d|c)q_2(e|c)}{\sum_{c'=1}^{n_C} q_0(c') \sum_d dq_1(d|c')q_2(e|c')}.$$

Thus, summing over all n_E types, we derive (41).

The argument for (42) is the same and thus omitted. \blacksquare

Now we extend the density evolution analysis in Section III. Note that once we obtain the connection probabilities in Lemma 2, edge types no longer play a role in the analysis, and we do not need to track the densities of edge types.

Consider using such a random LDPC code over a binary erasure channel with the erasure probability δ . Denote by $\alpha_v^{(i)}$ the probability that the variable end of a randomly selected edge is a type v variable node and it is not decoded after the i th iteration. Clearly, $\alpha_v^{(0)} = \delta$ for all v (as every variable bit is erased independently with probability δ through the erasure channel). Similarly, denote by $\beta_c^{(i)}$ the probability that the check end of a randomly selected edge is a type c check node and it is not decoded after the i th iteration. Let

$$\tilde{p}_1(d|v) = \frac{(d + 1)p_1(d + 1|v)}{\sum_{d'} (d' + 1)p_1(d' + 1|v)} \quad (43)$$

be the excess degree distribution of a type v variable node and

$$\lambda_v(z) = \sum_d \tilde{p}_1(d|v)z^d \quad (44)$$

be its generating function. Also, let

$$\tilde{q}_1(d|c) = \frac{(d + 1)q_1(d + 1|c)}{\sum_{d'} (d' + 1)q_1(d' + 1|c)} \quad (45)$$

be the excess degree distribution of a type c check node and

$$\rho_c(z) = \sum_d \tilde{q}_1(d|c)z^d. \quad (46)$$

Analogous to the density evolution analysis in Section III, we can extend (24) and (25) as follows:

$$\begin{aligned} \alpha_v^{(i)} &= \delta \sum_d \tilde{p}_1(d|v) \left(\sum_c \beta_c^{(i-1)} p(c|v) \right)^d \\ &= \delta \lambda_v \left(\sum_c \beta_c^{(i-1)} p(c|v) \right), \end{aligned} \quad (47)$$

and

$$\begin{aligned} \beta_c^{(i)} &= 1 - \sum_d \tilde{q}_1(d|c) \left(1 - \sum_v \alpha_v^{(i-1)} p(v|c) \right)^d \\ &= 1 - \rho_c \left(1 - \sum_v \alpha_v^{(i-1)} p(v|c) \right). \end{aligned} \quad (48)$$

B. Convolutional LDPC Codes

Convolutional LDPC codes that exploit spatial coupling were shown to have better performance than regular LDPC codes (see, e.g., [30], [31]). In this section, we show how to construct a convolutional LDPC code from the MET-LDPC code described in Section IV-A.

Consider an irregular (n, k) -LDPC code characterized by the generation functions of the two degree distributions $\lambda(z)$ and $\rho(z)$. Note that $\lambda(z)$ and $\rho(z)$ need to be chosen to meet the socket count equality constraint in (2). This irregular LDPC code is viewed as a *base* random bipartite graph for the construction of a convolutional (nL, kL) -LDPC code.

For the MET construction of the (nL, kL) -LDPC code, there are L types of variable nodes, L types of check nodes, and L types of edges. We specify the following probability.

- $p_0(v)$: The probability that a randomly selected variable node is of type v is equal to $1/L$, for $v = 1, 2, \dots, L$.
- $p_1(d|v)$: The probability that a type v variable node has degree d is characterized by the generating function of the excess degree distribution $\lambda(z)$, for $v = 1, 2, \dots, L$.
- $p_2(e|v)$: The probability that an edge of a type v variable node is a type e edges is $1 - \eta$ if $e = v$, and η if $e = ((v + 1) \bmod L)$, for $v = 1, 2, \dots, L$.
- $q_0(c)$: The probability that a randomly selected check node is of type c is equal to $1/L$, for $c = 1, 2, \dots, L$.
- $q_1(d|c)$: The probability that a type c check node has degree d is characterized by the generating function of the excess degree distribution $\rho(z)$, for $c = 1, 2, \dots, L$.
- $q_2(e|c)$: The probability that an edge of a type c check node is a type e edges is 1 if $e = c$, for $c = 1, 2, \dots, L$.

That is, type e edges are connected to type e check nodes.

When n and $n - k$ are very large, there are roughly n type v variable nodes and $n - k$ type c check nodes. Intuitively, one may view type v variable nodes are the set of variable nodes indexed from $n(v - 1) + 1$ to nv , $v = 1, 2, \dots, L$, and type c check nodes are the set of check nodes indexed from $(n - k)(c - 1) + 1$ to $(n - k)c$, $c = 1, 2, \dots, L$. As such, we can view the construction of the (nL, kL) -LDPC code as a concatenation of L copies of the irregular (n, k) -LDPC codes. However, the trick is to rewire some edges to make these L copies convoluted. With probability η , an edge in the v th copy is rewired randomly to the $((v + 1) \bmod L)$ th copy.

It is easy to see that all the socket count equality constraints in (40) are satisfied from the socket count equality constraint in (2).

To make such a MET-LDPC code a convolutional LDPC code, we puncture type L variable nodes and remove all the edges connected to type L variable nodes. Such a puncturing yields a $n(L - 1) \times (n - k)L$ parity-check matrix. Since type 1 edges between type L variable nodes and type 1 check nodes are removed, it is easy to see that for $v = 1, 2, \dots, L - 1$,

$$p(c|v) = \begin{cases} 1 - \eta, & \text{if } c = v, \\ \eta, & \text{if } c = v + 1. \end{cases} \quad (49)$$

On the other hand, for $c = 2, \dots, L - 1$,

$$p(v|c) = \begin{cases} 1 - \eta, & \text{if } v = c, \\ \eta, & \text{if } v = c - 1. \end{cases} \quad (50)$$

For $c = v = 1$, we have $p(v|c) = 1$ (since type 1 edges between type L variable nodes and type 1 check nodes are removed). For $c = L, v = L - 1$, we also have $p(v|c) = 1$ (since type L edges between type L variable nodes and type L check nodes are removed). The puncturing step is crucial in improving the performance. The insight for that is like the double-zipper ZigZag decoding algorithm (see, e.g., [36], [37]) that starts the decoding from the two ends and then propagates toward to the central part of the code.

For $v = 1, 2, \dots, L - 1$, we still have $\lambda_v(z) = \lambda(z)$. With the type 1 edges between type L variable nodes and type 1 check nodes being removed, we have $\rho_1(z) = \rho((1 - \eta)z + \eta)$. On the other hand, since the type 1 edges between type L variable nodes and type L check nodes are removed, we have $\rho_L(z) = \rho(\eta z + (1 - \eta))$. For $c = 2, 3, \dots, L - 1$, we still have $\rho_c(z) = \rho(z)$. For $v = 1, 2, \dots, L - 1$, let $\alpha_v^{(i)}$ be the probability that the variable end of a randomly selected edge is a type v variable node and it is not decoded after the i th iteration. For $c = 1, 2, \dots, L$, let $\beta_c^{(i)}$ be the probability that the check end of a randomly selected edge is a type c check node and it is not decoded after the i th iteration. Now the density evolution equations in (47) and (48) can be written as follows:

$$\alpha_v^{(i)} = \delta \lambda((1 - \eta)\beta_v^{(i)} + \eta\beta_{v+1}^{(i)}), \quad 1 \leq v \leq L - 1, \quad (51)$$

and

$$\begin{aligned} \beta_1^{(i)} &= 1 - \rho(1 - (1 - \eta)\alpha_1^{(i-1)}), \\ \beta_c^{(i)} &= 1 - \rho(1 - (1 - \eta)\alpha_c^{(i-1)} - \eta\alpha_{c-1}^{(i-1)}), \\ &\quad 2 \leq c \leq L - 1, \\ \beta_L^{(i)} &= 1 - \rho(1 - \eta\alpha_{L-1}^{(i-1)}). \end{aligned} \quad (52)$$

The base graphs for the convolutional LDPC codes can also be extended to the degree-degree correlated LDPC codes in Section II-B. Specifically, we can construct a convolutional LDPC code with L copies by viewing (i) the variable nodes at the v th copy with degree x as type (v, x) variable nodes, (ii) the check nodes at the c th copy with degree y as type (c, y) check nodes, and (iii) the edges between a variable node at the v th copy with degree x and a check node at the c th copy with degree y as type $((v, x), (c, y))$ edges. Denote by $\alpha_{v,x}^{(i)}$ the probability that the variable end of a randomly selected edge is a type (v, x) variable node and it is not decoded after the i th iteration. Also, $\alpha_{v,x}^{(0)} = \delta$ for all v, x (as every variable bit is erased independently with probability δ through the erasure channel). Similarly, denote by $\beta_{c,y}^{(i)}$ the probability that the check end of a randomly selected edge is a type (c, y) check node and it is not decoded after the i th iteration. The generating function of the excess degree distribution of a type (v, x) variable node is

$$\lambda_{v,x}(z) = z^{x-1}. \quad (53)$$

Also, the generating function of the excess degree distribution of a type (c, y) check node is

$$\rho_{c,y}(z) = z^{y-1}. \quad (54)$$

Analogous to the previous density evolution analysis for convolutional LDPC codes, we have

$$\begin{aligned} \alpha_{v,x}^{(i)} &= \delta \left(\sum_y \sum_c \mathbb{P}(Y_e = y | X_e = x) \beta_{c,y}^{(i)} p(c|v) \right)^{x-1} \\ &= \delta \lambda_{v,x} \left(\sum_y \sum_c \mathbb{P}(Y_e = y | X_e = x) \beta_{c,y}^{(i)} p(c|v) \right), \end{aligned} \quad (55)$$

and

$$\begin{aligned} \beta_{c,y}^{(i)} &= 1 - \left(1 - \sum_x \sum_v P(X_e = x | Y_e = y) \alpha_{v,x}^{(i-1)} p(v|c)\right)^{y-1} \\ &= 1 - \rho_{c,y} \left(1 - \sum_x \sum_v P(X_e = x | Y_e = y) \alpha_{v,x}^{(i-1)} p(v|c)\right), \end{aligned} \quad (56)$$

where $p(c|v)$ and $p(v|c)$ are specified in (49) and (50), respectively.

As discussed in the survey paper [38], protograph-based LDPC codes (see, e.g., [39]–[41]) can be viewed as a sub-ensemble within the ensemble of convolutional LDPC codes, sharing the same joint degree distribution on variable nodes and check nodes. This similarity arises because a protograph-based LDPC code is constructed through a ‘copy-and-permute’ operation on a given bipartite graph with multiple edges, known as the protograph. This operation is akin to the construction of convolutional LDPC codes discussed in this paper. The key distinction between convolutional LDPC codes and protograph-based LDPC codes is that the former comprises an ensemble of random bipartite graphs, whereas the latter is a structured deterministic bipartite graph. Given that different protographs can result in distinct thresholds, choosing the most effective protograph can improve threshold performance. This method is likely to achieve a superior threshold compared to the convolutional LDPC code, whose threshold represents the average over all possible protographs. Intuitively, a protograph-based LDPC code can be considered an *optimized* realization selected from the ensemble of convolutional LDPC codes. Consequently, protograph-based LDPC codes are more suited for practical applications and have thus become the de facto standard in LDPC code design. We note that our DDC-LDPC code’s design process, utilizing density evolution analysis, can serve as a beneficial preliminary step in the following methodology: (i) Identify the optimal marginal degree distributions for variable nodes and check nodes to maximize the threshold, as outlined in [42]. (ii) With these marginal degree distributions established, design the best joint degree distribution. (iii) Using the joint degree distribution, design the best (or at least a reasonably good) protograph. In each step, the threshold may improve.

C. Poisson Degree Distributions of the Check Nodes

In this section, we consider a special class of the MET-LDPC code in Section IV-A, where the degree distributions of the check nodes are Poisson. Consider a MET-LDPC code with n_V types of variable nodes, n_C types of check nodes, and n_C types of edges. For this, we specify the following probabilities.

- $p_0(v)$: The probability that a randomly selected variable node is of type v , $v = 1, 2, \dots, n_V$.
- $p_1(d|v)$: The probability that a type v variable node has degree d is characterized by the generating function of the excess degree distribution $\lambda_v(z)$, $v = 1, 2, \dots, n_V$.
- $p_2(e|v)$: The probability that an edge of a type v variable node is a type e edges, $e = 1, 2, \dots, n_C$.

- $q_0(c)$: The probability that a randomly selected check node is of type c , $c = 1, 2, \dots, n_C$.
- $q_1(d|c)$: The probability that a type c check node has degree d is $e^{-\nu_c} \frac{\nu_c^d}{d!}$, i.e., it is a Poisson distribution with mean ν_c .
- $q_2(e|c)$: The probability that an edge of a type c check node is a type e edges is 1 if $e = c$, $e = 1, 2, \dots, n_C$. That is, type e edges are connected to type e check nodes.

Note that the expected number of type c edges connected to variable nodes is $n \sum_{v=1}^{n_V} p_0(v) \sum_d dp_1(d|v) p_2(c|v)$. On the other hand, the expected number of type c edges connected to check nodes is $(n-k)q_0(c)\nu_c$. To satisfy the socket count equality constraints in (40), we choose

$$\nu_c = G \frac{\sum_{v=1}^{n_V} p_0(v) \sum_d dp_1(d|v) p_2(c|v)}{q_0(c)}, \quad (57)$$

where $G = n/(n-k)$ is defined in (3). The generating function of a Poisson degree distribution with mean ν_c is $e^{-\nu_c(1-z)}$. It is well-known (see, e.g., [20]) that the excess degree distribution of a Poisson degree distribution is also Poisson with the same mean. Thus, we have

$$\rho_c(z) = e^{-\nu_c(1-z)}. \quad (58)$$

Note from Lemma 2 that

$$p(c|v) = p_2(c|v), \quad (59)$$

and

$$\begin{aligned} p(v|c) &= \frac{p_0(v) \sum_d dp_1(d|v) p_2(c|v)}{\sum_{v'=1}^{n_V} p_0(v') \sum_d dp_1(d|v') p_2(c|v')} \\ &= \frac{G p_0(v) \sum_d dp_1(d|v) p_2(c|v)}{q_0(c) \nu_c}, \end{aligned} \quad (60)$$

where we use (57) in the last identity.

Now the density evolution equations in (47) and (48) can be written as follows:

$$\alpha_v^{(i)} = \delta \lambda_v \left(\sum_{c=1}^{n_C} \beta_c^{(i)} p_2(c|v) \right), \quad (61)$$

and

$$\beta_c^{(i)} = 1 - \exp \left(- \frac{1}{q_0(c)} \sum_{v=1}^{n_V} \alpha_v^{(i-1)} G p_0(v) \xi_v p_2(c|v) \right), \quad (62)$$

where $\xi_v = \sum_d dp_1(d|v)$ is the mean degree of a type v variable node. The recursive equations in (61) and (62) were previously derived in [33] for irregular repetition slotted ALOHA (IRSA) [32] with multiple classes of users and multiple classes of receivers.

V. NUMERICAL RESULTS

A. Block Construction

In this section, we evaluate the performance of random LDPC codes generated by the block construction in Section II-A. Consider using the randomly generated LDPC code in Example 1 for a BEC with the erasure probability δ . In this numerical experiment, we set $G = 3$, $d = 3$.

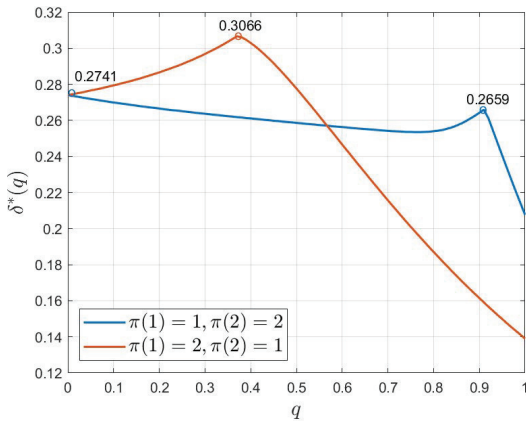


Fig. 3. The maximum tolerable erasure probability $\delta^*(q)$ as a function of the parameter q (with the step size of 0.01) for two permutations: the negatively correlated one with $\pi(1) = 2, \pi(2) = 1$ (the orange curve), and the positively correlated one with $\pi(1) = 1, \pi(2) = 2$ (the blue curve).

In Fig. 3, we plot the maximum tolerable erasure probability $\delta^*(q)$ as a function of the parameter q (with the step size of 0.01) for two permutations: the negatively correlated one with $\pi(1) = 2, \pi(2) = 1$ (the orange curve), and the positively correlated one with $\pi(1) = 1, \pi(2) = 2$ (the blue curve). Recall that when $q = 0$, it reduces to the independent case. As shown in Fig. 3, adding a negative degree-degree correlation can lead to a much larger maximum tolerable erasure probability than that of the independent case. In particular, when $q = 0.37$, we find that the maximum tolerable erasure probability $\delta^*(q)$ is 0.3066, which is larger than $\delta^*(q) = 0.2741$ for $q = 0$. However, adding a positive degree-degree correlation does not improve the maximum tolerable erasure probability in this numerical experiment.

To explain these numerical results, we note that LDPC codes with positive degree-degree correlations tend to form a giant component in the bipartite graph (as nodes with large degrees tend to connect to nodes with large degrees), and that makes the decoding of the LDPC code difficult. On the other hand, LDPC codes with negative degree-degree correlations are less likely to form a giant component. As such, one can exploit the negative degree-degree correlation by increasing q to improve the maximum tolerable erasure probability. However, when q is increased to 1, the bipartite graph gradually decouples into two separate bipartite graphs (from the block construction). Each has its own maximum tolerable erasure probability. When q is close to 1, the maximum tolerable erasure probability is dominated by the bipartite graph with a smaller maximum tolerable erasure probability. Due to these two effects, the orange curve for the case with a negative degree-degree correlation is unimodal (with only one peak).

In order to verify the effectiveness of the asymptotic results derived from the density evolution equations, we conduct extensive simulations. For our simulations, we generate degree-degree correlated LDPC codes with $n = 18,000$ variable nodes and $n - k = 6,000$ check nodes by using the construction in Example 1. Then we simulate the LDPC codes over a BEC with an erasure probability δ . After that, we perform

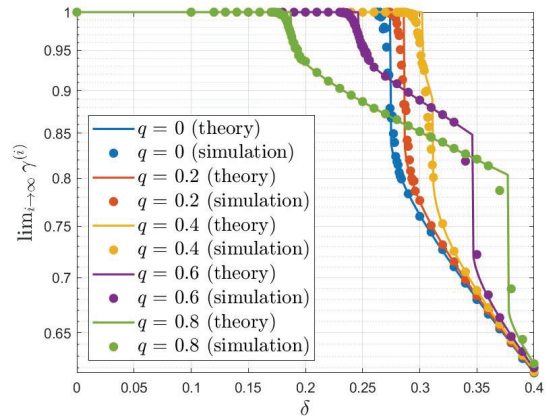


Fig. 4. The probability that a randomly selected variable node is successfully decoded, i.e., $\lim_{i \rightarrow \infty} \gamma^{(i)}$, as a function of the erasure probability δ from 0 to 0.4 for $q = 0, 0.2, 0.4, 0.6, 0.8$.

iterative decoding (peeling decoder) for the LDPC codes (until there are no check nodes with degree 1). In Fig. 4, we plot the probability that a randomly selected variable node is successfully decoded, i.e., $\lim_{i \rightarrow \infty} \gamma^{(i)}$ with $\gamma^{(i)}$ in (28), as a function of the erasure probability δ from 0 to 0.4 with respect to various parameters $q = 0, 0.2, 0.4, 0.6, 0.8$. This is done under the condition of a negatively correlated permutation $\pi(1) = 2, \pi(2) = 1$. Each data point is the average of 100 experiments. The five solid curves represent the theoretical results of $\lim_{i \rightarrow \infty} \gamma^{(i)}$, and the five dotted curves represent the corresponding simulation results. As shown in Fig. 4, the simulation results and the asymptotic results match very well. Also, we observe the largest maximum tolerable erasure probability occurs when $q = 0.4$ (the yellow curve). This result is consistent with that of Fig. 3. Additionally, we made a new discovery: for $q \geq 0.4$, the curve exhibits two turning points. One occurs as $\lim_{i \rightarrow \infty} \gamma^{(i)}$ drops from 1, representing the largest maximum tolerable erasure probability. Another turning point arises after reaching this maximum tolerable erasure probability. This nonlinear nature of the density evolution is not observed in the independent case ($q = 0$). The underlying reasons for this phenomenon are further elucidated in Fig. 5.

Finally, we examine the effect of degree on the decoding probabilities of variable nodes. The variable nodes in Example 1 exhibit two types of degrees: d and $2d$. In Fig. 5 and Fig. 6, we plot the probability that a randomly selected variable node with degree x is successfully decoded, i.e., $\lim_{i \rightarrow \infty} \gamma_x^{(i)}$, as a function of the erasure probability δ for different values of q . These plots are generated under the conditions of a negatively correlated permutation $\pi(1) = 2, \pi(2) = 1$ (in Fig. 5) and a positively correlated permutation $\pi(1) = 1, \pi(2) = 2$ (in Fig. 6), respectively. The solid curve (resp. dashed curve) represents the decoding probability for a variable node with degree d (resp. $2d$).

In Fig. 5, it is well known that the decoding probability for variable nodes with a large degree is higher than that for nodes with a small degree under negatively correlated conditions. This characteristic is referred to as the unequal

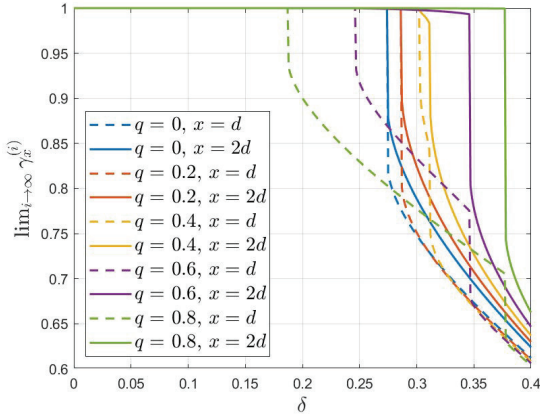


Fig. 5. The probability that a randomly selected variable node with degree x (for $x = d, 2d$) is successfully decoded, i.e., $\lim_{i \rightarrow \infty} \gamma_x^{(i)}$, as a function of the erasure probability δ for $q = 0, 0.2, 0.4, 0.6, 0.8$.

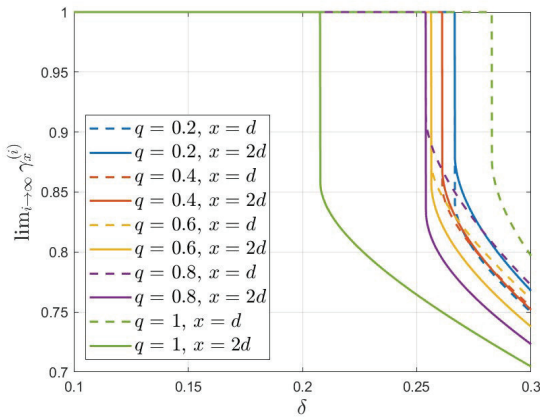


Fig. 6. The probability that a randomly selected variable node with degree x (for $x = d, 2d$) is successfully decoded, i.e., $\lim_{i \rightarrow \infty} \gamma_x^{(i)}$, as a function of the erasure probability δ from 0.1 to 0.3 for $q = 0.2, 0.4, 0.6, 0.8, 1$.

error protection (UEP) property. One significant finding from our experiments is that the degree-degree correlation plays a critical role in UEP. Recall that the degree-degree correlation in Example 1 is $-q$. As shown in Fig. 5, increasing q increases the gap between the curve for $\lim_{i \rightarrow \infty} \gamma_d^{(i)}$ and the curve for $\lim_{i \rightarrow \infty} \gamma_{2d}^{(i)}$. In particular, for $q = 0.8$ (the two green curves), the gap is the largest among the five choices of q . Even when the erasure probability exceeds the upper bound $1/G = 1/3$, variable nodes with degree $2d$ can still have a very high decoding probability. This is at the cost of sacrificing the decoding probability of variable nodes with degree d . Given that the two blocks of degree d and degree $2d$ are coupled, the trends of the two curves for $\lim_{i \rightarrow \infty} \gamma_d^{(i)}$ and $\lim_{i \rightarrow \infty} \gamma_{2d}^{(i)}$ influence each other. As demonstrated in Fig. 4, this coupling leads to the formation of curves with two turning points. On the other hand, the two curves for $q = 0.4$ are very close to each other. Even though they have the largest maximum tolerable erasure probability δ^* , they are not suitable for LDPC codes with the UEP property.

In Fig. 6, we note that for all δ values (this figure focuses on the corresponding decoding probabilities $\lim_{i \rightarrow \infty} \gamma_d^{(i)}$ and

$\lim_{i \rightarrow \infty} \gamma_{2d}^{(i)}$ for δ from 0.1 to 0.3), when $q \leq 0.4$, the solid curves are consistently above the dashed curves for the same q . This implies that when positively correlated, the decoding probability for variable nodes with a large degree is typically higher than that with a small degree. Interestingly, for all δ , the dashed curves shift above the solid curves when $q > 0.4$. As illustrated in Fig. 6, beyond the maximum tolerable erasure probability δ^* , variable nodes with degree d can exhibit higher decoding probabilities. This is because nodes with smaller degrees are less likely to form a giant component, thereby facilitating LDPC code decoding. This effect is due to the positive degree-degree correlation in LDPC codes. Furthermore, observe that when $q > 0.8$, increasing q widens the gap between the curve for $\lim_{i \rightarrow \infty} \gamma_d^{(i)}$ and the curve for $\lim_{i \rightarrow \infty} \gamma_{2d}^{(i)}$, indicative of LDPC codes with the UEP property.

B. General Construction for Performance Improvement

In this section, we show that the LDPC codes from the general construction in Section II-B can lead to further performance improvement.

In [29], Shokrollahi and Storn proposed a construction of the LDPC code with the following (independent) bivariate distribution:

$$P(X_e = x, Y_e = y) = p_{X_e}(x)p_{Y_e}(y), \quad (63)$$

where $p_{X_e}(2) = 0.2633$, $p_{X_e}(3) = 0.1802$, $p_{X_e}(7) = 0.2700$, $p_{X_e}(30) = 0.2865$, and $p_{Y_e}(8) = 0.6341$, $p_{Y_e}(9) = 0.3659$.

It is easy to verify that $G = 2$. It was shown in [29] that the maximum tolerable erasure probability $\delta^* = 0.49553$. For this example, we find the bivariate distribution $P(X_e = 2, Y_e = 8) = 0.1534$, $P(X_e = 3, Y_e = 8) = 0.1789$, $P(X_e = 7, Y_e = 8) = 0.1035$ such that it has the same marginal distributions $p_{X_e}(x)$ and $p_{Y_e}(y)$. The maximum tolerable erasure probability δ^* for such a bivariate distribution is 0.49568, which is larger than 0.49553 for the LDPC code in [29].

Another example is the LDPC code in Example 3.64 of the book [42]. The maximum tolerable erasure probability δ^* for that LDPC code is 0.47410. Using the general construction in Section II-B, we find the bivariate distribution $P(X_e = 2, Y_e = 8) = 0$, $P(X_e = 3, Y_e = 8) = 0.274$, $P(X_e = 7, Y_e = 8) = 0.007$ such that it has the same marginal distributions $p_{X_e}(x)$ and $p_{Y_e}(y)$. We can extend the maximum tolerable erasure probability δ^* to 0.48077.

C. Convolutional LDPC Codes

In this section, we evaluate the performance of random convolutional LDPC codes. We consider a randomly generated convolutional LDPC code constructed according to Section IV-B, which is used for a BEC with the erasure probability δ . In this numerical experiment, we set $G = 2$, $d = 3$, and $\eta = 0.5$ in (51) and (52), allowing us to derive the following density evolution equations:

$$\alpha_v^{(i)} = \delta \lambda(0.5\beta_v^{(i)} + 0.5\beta_{v+1}^{(i)}), \quad 1 \leq v \leq L - 1, \quad (64)$$

TABLE I

THE MAXIMUM TOLERABLE ERASURE PROBABILITY δ^* OF DENSITY EVOLUTION EQUATIONS FOR $L = 2, 5, 10, 20, 50, 200$ IN FIVE CASES.

L	Case (i)	Case (ii)	Case (iii)	Case (iv)	Case (v)
2	0.8588	0.8256	0.9299	0.9482	0.9910
5	0.5137	0.5197	0.5527	0.5513	0.5529
10	0.4883	0.4916	0.4957	0.5048	0.5098
20	0.4880	0.4914	0.4912	0.4978	0.4996
50	0.4880	0.4914	0.4912	0.4978	0.4970
200	0.4880	0.4914	0.4912	0.4978	0.4968
conventional	0.4294	0.4128	0.4649	0.4741	0.4955

and

$$\begin{aligned}
 \beta_1^{(i)} &= 1 - \rho(1 - 0.5\alpha_1^{(i-1)}), \\
 \beta_c^{(i)} &= 1 - \rho(1 - 0.5\alpha_c^{(i-1)} - 0.5\alpha_{c-1}^{(i-1)}), \\
 &2 \leq c \leq L - 1, \\
 \beta_L^{(i)} &= 1 - \rho(1 - 0.5\alpha_{L-1}^{(i-1)}).
 \end{aligned} \tag{65}$$

Corresponding to different L values, we select five instances of LDPC codes for our numerical experiments. These codes are constructed as per Section IV-B, incorporating various degree distribution pairs and considering the degree-degree correlation as defined in Example 2. For $G = 2$, we find that when using a bivariate distribution with $p_1 = 0.155, p_2 = 0.345$, the irregular LDPC code achieves the maximum tolerable erasure probability. This is included in Table I below for comparison. We also employ the conventional LDPC code as a benchmark. The five degree distribution pairs are:

- (i) The (3,6)-Regular LDPC code: $\lambda(z) = z^2, \rho(z) = z^5$.
- (ii) The irregular LDPC code in Example 2 with the bivariate distribution specified by $p_1 = 0.25, p_2 = 0.25$ (with independent degree distribution): $\lambda(z) = 0.5z^2 + 0.5z^5, \rho(z) = 0.5z^5 + 0.5z^{11}$.
- (iii) The irregular LDPC code in Example 2 with the bivariate distribution specified by $p_1 = 0.155, p_2 = 0.345$: $\lambda(z) = 0.5z^2 + 0.5z^5, \rho(z) = 0.5z^5 + 0.5z^{11}$.
- (iv) The LDPC code by Shokrollahi and Storn [29]: $\lambda(z) = 0.26328z + 0.18020z^2 + 0.27000z^6 + 0.28649z^{29}, \rho(z) = 0.63407z^7 + 0.36593z^8$.
- (v) The LDPC code in Example 3.64 of the book [42]: $\lambda(z) = 0.106257z + 0.486659z^2 + 0.010390z^{10} + 0.396694z^{19}, \rho(z) = 0.5z^7 + 0.5z^8$.

The maximum tolerable erasure probabilities δ^* as determined by numerical solutions to the density evolution equations for $L = 2, 5, 10, 20, 50, 200$ in the above five cases are presented in Table I. Numerical results indicate that irrespective of the degree distribution set, the maximum tolerable erasure probabilities δ^* for the convolutional LDPC code consistently outperform those for the conventional LDPC code.

To validate the asymptotic results derived from the density evolution equations, we carry out extensive simulations. In these simulations, we generate three MET constructions of (nL, kL) -LDPC codes for $L = 2, 5, 10$, where all base random bipartite graphs are irregular LDPC codes with $n = 6000$ variable nodes and $n - k = 3000$ check nodes. All base

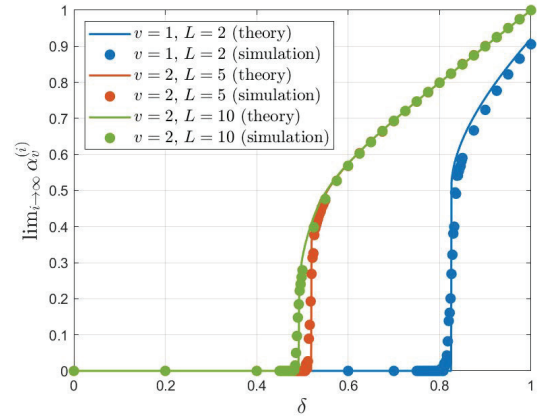


Fig. 7. The probability that the variable end of a randomly selected edge is a type v variable nodes and it is not decoded after the i th iteration, i.e., $\lim_{i \rightarrow \infty} \alpha_v^{(i)}$, as a function of the erasure probability δ from 0 to 1 with respect to $L = 2, 5, 10$ when $p_1 = p_2 = 0.25$.

irregular LDPC codes are characterized by the generating functions of Case (ii). To create L copies of (nL, kL) -LDPC code convolutionally for $L = 2, 5, 10$, we randomly rewire some edges from the v th copy to the $((v + 1) \bmod L)$ th copy with probability $\eta = 0.5$, and puncture type L variable nodes along with their connected edges. After constructing these graphs, we simulate each one over a binary erasure channel with an erasure probability δ , and then we perform iterative decoding (using the peeling decoder) for these LDPC codes until there are no check nodes with degree 1 remaining.

In Fig. 7, we plot the probability that the variable end of a randomly selected edge corresponds to a type v variable node and that it is not decoded after the i th iteration, i.e., $\lim_{i \rightarrow \infty} \alpha_v^{(i)}$, as a function of the erasure probability δ ranging from 0 to 1, with respect to $L = 2, 5, 10$ when $p_1 = p_2 = 0.25$. All the base irregular LDPC codes are characterized by the generating functions of Case (ii), i.e., $\lambda(z) = 0.5z^2 + 0.5z^5, \rho(z) = 0.5z^5 + 0.5z^{11}$. Each data point represents the average of 100 experiments. The three solid curves depict the theoretical results of $\lim_{i \rightarrow \infty} \alpha_v^{(i)}$, and the three dotted curves represent the corresponding simulation results. In the $L = 2$ case, we plot the decoding result of type 1 variable nodes, while in the $L = 5, 10$ cases, we plot the decoding results of type 2 variable nodes. As shown in Fig. 7, there is a strong alignment between the simulation results and the asymptotic results, thereby confirming the effectiveness of the asymptotic results derived from the density evolution equations.

Next, we use the bivariate distribution specified by $p_1 = 0.155$ and $p_2 = 0.345$ to evaluate the performance of random convolutional LDPC codes with degree-degree correlation. Also, we set $G = 2, x = 6, 3, y = 12, 6, \eta = 0.5$ in (22), (23), (55) and (56), and display the results of the density evolution equations in Fig. 8.

In Fig. 8, all the base irregular LDPC codes are characterized by the generating functions of Case (iii), i.e., $\lambda(z) = 0.5z^2 + 0.5z^5, \rho(z) = 0.5z^5 + 0.5z^{11}$. In this numerical experiment, we plot the probability that the variable end of a randomly selected edge is a type v variable node, and it

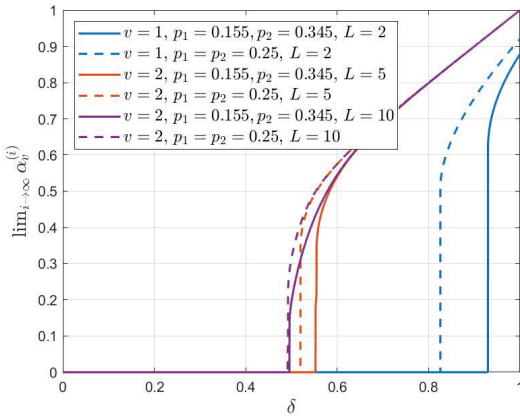


Fig. 8. The probability that the variable end of a randomly selected edge is a type v variable nodes and it is not decoded after the i th iteration, i.e., $\lim_{i \rightarrow \infty} \alpha_v^{(i)}$, as a function of the erasure probability δ from 0 to 1, $p_1 = 0.155$, $p_2 = 0.345$ (the solid curve) and $p_1 = p_2 = 0.25$ (the dashed curve), with respect to $L = 2, 5, 10$.

is not decoded after the i th iteration, i.e., $\lim_{i \rightarrow \infty} \alpha_v^{(i)}$, as a function of the erasure probability δ ranging from 0 to 1. For $L = 2, 5, 10$, we obtain numerical results comparing the bivariate distribution specified by $p_1 = 0.155, p_2 = 0.345$ (the solid curve) to that specified by $p_1 = p_2 = 0.25$ (the dashed curve, representing independent degree distribution). In the $L = 2$ case, we plot the decoding result of type 1 variable nodes, while in the $L = 5, 10$ cases, we plot the decoding results of type 2 variable nodes. As observed in Fig. 8 and Table I, when $L \geq 10$, the maximum tolerable erasure probability δ^* under Cases (ii) and (iii) are almost identical. This indicates that as L increases, the influence of the degree-degree correlation of the base graph gradually diminishes. The phenomenon can be attributed to the potent influence that the correlation, driven by the upper-level convolutional (“copy-and-permute”) structure, exerts on the decoding probability. The convolutional structure exhibits a “wave”-like decoding process, progressing from the outer stages toward the inner stages. This influence significantly outweighs the impact of the base graph’s degree-degree correlation (at the lower level). Consequently, optimizing the degree-degree correlation within the base graph does not significantly enhance the performance of convolutional LDPC codes.

VI. CONCLUSION

In this paper, we proposed two constructions of degree-degree correlated LDPC codes and derived the density evolution equations for such LDPC codes. For the block construction, our numerical results show that adding a negative degree-degree correlation can achieve a much higher maximum tolerable erasure probability than LDPC codes with independent degree distributions. Moreover, adding a negative degree-degree correlation could lead to better designs of LDPC codes with the UEP property. The general construction (with an arbitrary bivariate degree-degree distribution) provides a much larger ensemble of LDPC codes than block construction. It can lead to performance improvement over the existing

LDPC codes with independent degree-degree distributions. In our extensions, we expanded the degree-degree correlated LDPC codes to multi-edge type LDPC codes and further to convolutional LDPC codes. Our numerical results demonstrate that, regardless of the degree-degree correlation controlling parameter q , utilizing spatial coupling offers improved performance when the number of copies L approaches infinity (i.e., the code rate of LDPC codes is close to k/n).

Interpreting the numerical results, such as thresholds, of degree-degree correlated LDPC codes, is challenging due to two main reasons: (i) The equations derived from the density evolution analysis are highly nonlinear, and (ii) these equations are interconnected through the degree-degree correlation or edge types. One possible approach to tackle this issue is to use the mathematical framework in [43], [44]. Results related to this line of research will be reported separately.

REFERENCES

- [1] H.-W. Yu, C.-E. Lee, R. Zhang, C.-S. Chang, and D.-S. Lee, “Degree-degree correlated low-density parity-check codes over a binary erasure channel,” in *Proc. IEEE ISIT*, 2023.
- [2] R. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [3] J. H. Bae, A. Abotabl, H.-P. Lin, K.-B. Song, and J. Lee, “An overview of channel coding for 5G NR cellular communications,” *APSIPA Trans. Signal Inf. Process.*, vol. 8, 2019.
- [4] T. Thi Bao Nguyen, T. Nguyen Tan, and H. Lee, “Low-complexity high-throughput QC-LDPC decoder for 5G new radio wireless communication,” *Electronics*, vol. 10, no. 4, p. 516, 2021.
- [5] H.-C. Lee, J.-H. Shy, Y.-M. Chen, and Y.-L. Ueng, “LDPC coded modulation for TLC flash memory,” in *Proc. IEEE ITW*, 2017.
- [6] Y. Fang, Y. Bu, P. Chen, F. C. Lau, and S. Al Otaibi, “Irregular-mapped protograph LDPC-coded modulation: A bandwidth-efficient solution for 6G-enabled mobile networks,” *IEEE Trans. Intell. Transportation Syst.*, vol. 24, no. 2, pp. 2060–2073, 2021.
- [7] S. Shao *et al.*, “Survey of turbo, LDPC, and polar decoder ASIC implementations,” *IEEE Commun. Surveys Tuts*, vol. 21, no. 3, pp. 2309–2333, 2019.
- [8] P. Chen, L. Shi, Y. Fang, F. C. Lau, and J. Cheng, “Rate-diverse multiple access over gaussian channels,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5399–5413, 2023.
- [9] N. Bonello, S. Chen, and L. Hanzo, “Low-density parity-check codes and their rateless relatives,” *IEEE Commun. Surveys Tuts*, vol. 13, no. 1, pp. 3–26, 2010.
- [10] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [11] M. Luby, M. Mitzenmacher, and M. A. Shokrollahi, “Analysis of random processes via and-or tree evaluation,” in *Proc. SODA*, 1998.
- [12] M. Luby, M. Mitzenmacher, A. Shokrollah, and D. Spielman, “Analysis of low density codes and improved designs using irregular graphs,” in *Proc. ACM STOC*, 1998.
- [13] M. A. Shokrollahi, “New sequences of linear time erasure codes approaching the channel capacity,” in *Proc. AAECC*, 1999.
- [14] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [15] N. Rahnavard and F. Fekri, “New results on unequal error protection using LDPC codes,” *IEEE Commun. Lett.*, vol. 10, no. 1, pp. 43–45, 2006.
- [16] N. Rahnavard, H. Pishro-Nik, and F. Fekri, “Unequal error protection using partially regular LDPC codes,” *IEEE Trans. Commun.*, vol. 55, no. 3, pp. 387–391, 2007.
- [17] N. Rahnavard, B. N. Vellambi, and F. Fekri, “Rateless codes with unequal error protection property,” *IEEE Trans. Inf. Theory*, vol. 53, no. 4, pp. 1521–1532, 2007.
- [18] Y.-H. Chen, Y.-T. Liu, C.-H. Wang, and C.-C. Chao, “Analysis of UEP QC-LDPC codes using density evolution,” in *Proc. IEEE ISITA*, 2020.
- [19] Y. Zhao, Y. Zhang, F. C. Lau, Z. Zhu, and H. Yu, “Duplicated zigzag decodable fountain codes with the unequal error protection property,” *Computer Commun.*, vol. 185, pp. 66–78, 2022.

- [20] M. Newman, *Networks: an introduction*. OUP Oxford, 2009.
- [21] D. J. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics letters*, vol. 33, no. 6, pp. 457–458, 1997.
- [22] D. J. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, 1999.
- [23] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, 2001.
- [24] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. ACM STOC*, 1997.
- [25] S. Giddens, M. A. Gomes, J. P. Vilela, J. L. Santos, and W. K. Harrison, "Enumeration of the degree distribution space for finite block length LDPC codes," in *Proc. IEEE ICC*, 2021.
- [26] T. Richardson, R. Urbanke *et al.*, "Multi-edge type LDPC codes," in *Workshop honoring Prof. Bob McEliece on his 60th birthday, California Institute of Technology, Pasadena, California*, 2002, pp. 24–25.
- [27] S. Jayasooriya, M. Shirvanimoghaddam, L. Ong, G. Lechner, and S. J. Johnson, "A new density evolution approximation for LDPC and multi-edge type LDPC codes," *IEEE Trans. Commun.*, vol. 64, no. 10, pp. 4044–4056, 2016.
- [28] D.-S. Lee, C.-S. Chang, M. Zhu, and H.-C. Li, "A generalized configuration model with degree correlations and its percolation analysis," *Applied Netw. Sci.*, vol. 4, no. 1, pp. 1–21, 2019.
- [29] A. Shokrollahi and R. Storn, "Design of efficient erasure codes with differential evolution," in *Proc. IEEE ISIT*, 2000.
- [30] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, 2011.
- [31] D. G. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, 2015.
- [32] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, 2011.
- [33] C.-M. Chang, Y.-J. Lin, C.-S. Chang, and D.-S. Lee, "On the stability regions of coded Poisson receivers with multiple classes of users and receivers," *IEEE/ACM Trans. Netw.*, vol. 31, no. 1, pp. 234–247, 2023.
- [34] G. Liva and M. Chiani, "Protograph LDPC codes design based on EXIT analysis," in *Proc. IEEE GLOBECOM*, 2007.
- [35] P. Elias, "Coding for two noisy channels," in *Proc. IEEE Information Theory: 3rd London Symp.*, 1955.
- [36] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM*, 2008.
- [37] M. Kazemi, T. M. Duman, and M. Médard, "Collision resolution for random access," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3464–3477, 2022.
- [38] Y. Fang, G. Bi, Y. L. Guan, and F. C. Lau, "A survey on protograph LDPC codes and their applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1989–2016, 2015.
- [39] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, 2009.
- [40] T. Van Nguyen, *Design of capacity-approaching protograph-based LDPC coding systems*. The University of Texas at Dallas, 2012.
- [41] L. Dai, Y. Fang, Z. Yang, P. Chen, and Y. Li, "Protograph LDPC-coded BICM-ID with irregular CSK mapping in visible light communication systems," *IEEE Trans Veh. Technol.*, vol. 70, no. 10, pp. 11 033–11 038, 2021.
- [42] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge university press, 2008.
- [43] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled scalar recursions," in *Proc. IEEE ISTC*, 2012.
- [44] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of threshold saturation for coupled vector recursions," in *Proc. IEEE ITW*, 2012.



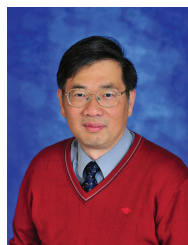
Hsiao-Wen Yu received the B.S. degree in Electrical Engineering from Chang Gung University, Taoyuan, Taiwan, in 2021. She is currently pursuing the M.S. degree in the Institute of Communications Engineering, National Tsing Hua University, Hsinchu, Taiwan. Her research interest is in 5G and beyond wireless communication.



Cheng-En Lee received the B.S. degree in Electrical Engineering from National Tsing Hua University, Hsinchu, Taiwan in 2021. He is currently a graduate student in Institute of Information Systems and Applications, National Tsing Hua University, Hsinchu, Taiwan, in 2023. His research interests are in network science, deep learning, and wireless communications.



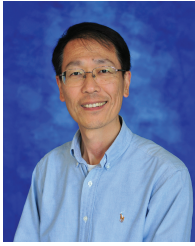
Ru-Hui Zhang received the B.S. degree in Advertisement from Xiamen University, Xiamen, China, in 2014, and the Ph.D. degree with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, in 2023. Her research interests are in network science, data analytics, and wireless communications.



Cheng-Shang Chang (S'85-M'86-M'89-SM'93-F'04) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1983, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, USA, in 1986 and 1989, respectively, all in Electrical Engineering.

From 1989 to 1993, he was employed as a Research Staff Member with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Since 1993, he has been with the Department of Electrical Engineering, National Tsing Hua University, Taiwan, where he is a Tsing Hua Distinguished Chair Professor. He is the Author of the book *Performance Guarantees in Communication Networks* (Springer, 2000) and the Coauthor of the book *Principles, Architectures and Mathematical Theory of High Performance Packet Switches* (Ministry of Education, R.O.C., 2006). His current research interests are concerned with network science, big data analytics, mathematical modeling of the Internet, and high-speed switching.

Dr. Chang served as an Editor for *Operations Research* from 1992 to 1999, an Editor for the *IEEE/ACM TRANSACTIONS ON NETWORKING* from 2007 to 2009, and an Editor for the *IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING* from 2014 to 2017. He is currently serving as an Editor-at-Large for the *IEEE/ACM TRANSACTIONS ON NETWORKING*. He is a Member of IFIP Working Group 7.3. He received an IBM Outstanding Innovation Award in 1992, an IBM Faculty Partnership Award in 2001, and Outstanding Research Awards from the National Science Council, Taiwan, in 1998, 2000, and 2002, respectively. He also received Outstanding Teaching Awards from both the College of EECS and the university itself in 2003. He was appointed as the first Y. Z. Hsu Scientific Chair Professor in 2002. He received the Merit NSC Research Fellow Award from the National Science Council, R.O.C. in 2011. He also received the Academic Award in 2011 and the National Chair Professorship in 2017 and in 2023 from the Ministry of Education, R.O.C. He is the Recipient of the 2017 IEEE INFOCOM Achievement Award.



Duan-Shin Lee (S'89-M'90-SM'98) received the B.S. degree from National Tsing Hua University, Taiwan, in 1983, and the MS and Ph.D. degrees from Columbia University, New York, in 1987 and 1990, all in Electrical Engineering. He worked as a Research Staff Member at the C&C Research Laboratory of NEC USA, Inc. in Princeton, New Jersey from 1990 to 1998. He joined the Department of Computer Science of National Tsing Hua University in Hsinchu, Taiwan, in 1998. Since August 2003, he has been a Professor. He received the Best Paper Award from the Y.Z. Hsu Foundation in 2006. He served as an editor for the *Journal of Information Science and Engineering* between 2013 and 2015. He served as an associate editor for *Performance Evaluation* from January 2018 to October 2021. Dr. Lee's current research interests are network science, game theory, machine learning and high-speed networks. He is a Senior IEEE Member.