

Design of Bilayer and Multi-layer LDPC Ensembles from Individual Degree Distributions

Eshed Ram and Yuval Cassuto, *Senior Member, IEEE*

Abstract—A new approach for designing bilayer and multi-layer LDPC codes is proposed and studied in the asymptotic regime. The ensembles are defined through individual uni-variate degree distributions, one for each layer. We present a construction that: 1) enables low-complexity decoding for high-SNR channel instances, 2) provably approaches capacity for low-SNR instances, 3) scales linearly (in terms of design complexity) in the number of layers. For the setup where decoding the second layer is significantly more costly than the first layer, we propose an optimal-cost decoding schedule and study the trade-off between code rate and decoding cost.

I. INTRODUCTION

Low-Density Parity-Check (LDPC) codes and their message-passing decoding algorithms [1] are an efficient way to achieve Shannon’s limit on various channels. LDPC codes are extremely powerful because they can attain competitive performance with low-complexity decoding (message passing on sparse bi-partite graphs) and simple code design (random drawing from explicit code ensembles). Many extensions of the basic LDPC construction have been proposed to enhance the code’s functionality through more structured code graphs, while maintaining the convenience of randomly drawing codes from ensembles. Some prominent examples are repeat-accumulate codes [2], [3], protograph-based codes [4], spatially-coupled codes [5], and multi-edge-type (MET) codes [6] (which can be viewed as a meta-class containing all the others). An especially useful class of structured LDPC codes is *bilayer codes* [7], [8], [9], [10] (and more generally multi-layer codes), which allow decoding the same codeword with two (or more) different decoders. A recent interest in bilayer LDPC codes is raised by storage applications, in which the code needs to be designed for both extreme and average channel conditions [11], [12].

In this paper we develop new tools for the construction and (asymptotic) analysis of bilayer LDPC code ensembles. An important feature of these tools is that they lend themselves well for extending beyond two layers, which we do later in the paper. A bilayer LDPC code has a code graph in which the variable nodes are connected to two types of check nodes,

allowing different connectivity to each type. There are several applications that motivate codes with bilayer structure: communication over the relay channel [7], decoder parallelization [13], multi-block coding [11], incremental redundancy [14], [15], [16], to name a few. Bilayer codes are a special case of MET codes that are defined at full generality in [6]. However, to gain tools and insight for construction and analysis, we find it beneficial to use a more compact specification of the codes than needed when viewed as MET codes. In particular, this compact specification is the key to our ability to extend the results to more than two layers without complexity blow-up.

A. Contributions

Bilayer LDPC codes are designed to simultaneously guarantee correction capabilities in two decoding modes: layer 1 only, and layers 1+2. More generally, L -layer codes are designed for L decoding modes: layer 1, layers 1+2, ..., layers 1+2+...+ L . The design of such codes in this paper is pursued through a new approach: *the codes are defined by specifying each layer separately as a standard degree-distribution pair*. This approach may at first seem less natural given that we decode layers 1+2 jointly, and not layer 2 separately. Indeed, prior work [7] designed the joint layer-1+2 code directly. The advantage of the new approach is that working with standard degree distributions (specified as uni-variate polynomials), rather than product degree distributions (specified as multi-variate polynomials), enables tractable design of explicit code ensembles with provable asymptotic performance.

The basis of this approach is laid in Section III, where the correction capability of the layer-1+2 code is characterized mathematically given the separate layer-1 and layer-2 degree distributions. This is achieved by deriving a two-dimensional density-evolution framework, where decoding thresholds are found as certain fixed points in two variables (each variable tracks the density on edges of one layer).

Section IV provides a general construction for bilayer codes with any desired thresholds for layer-1 and layer-1+2 decoding. The resulting codes are given as explicit degree distributions (building on known properties of standard single-layer codes), without need to employ optimization tools such as linear programming. In particular, the construction is used to construct code sequences that approach capacity for layer-1+2 decoding, while guaranteeing any desired threshold for layer-1 decoding. Furthermore, the additive gap to capacity of the layer-1+2 code is characterized (and bounded) given the gaps to capacity of the individual layer-1 and layer-2 degree distributions. Section V generalizes the results of Sections III,IV to L -layer codes, for any $L \geq 2$.

In Section VI we treat a model in which layer-2 decoding iterations are more costly than layer-1, and thus we seek codes

E. Ram and Y. Cassuto are with the Andrew and Erna Viterbi Department of Electrical and Computer Engineering, Technion – Israel Institute of Technology, Haifa 320003, Israel. E-mails: {s6eshedr@campus, ycasuto@ee}.technion.ac.il .

This work has been supported by the Israel Science Foundation (ISF) under Grant 2525/19 and by the US-Israel Binational Science Foundation (BSF) under Grant 2016703.

Part of the results of this paper was presented at the 2018 International Symposium on Information Theory.

Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

that successfully decode layers 1+2 with few layer-2 iterations. For this model we propose optimal-cost decoding schedules, and study the trade-off between rate (through layer-1 and layer-2 gaps to capacity) and decoding cost.

All of our results are given for coding over the binary erasure channel (BEC), however, the same analysis and constructions extend to other channels (such as the AWGN channel) using the EXIT method.

B. Related Work

Bilayer LDPC codes are used to implement binning for the relay channel in [7], [8], [9] ([10] also suggests bilayer codes with one layer being a low-density generator-matrix (LDGM) code). The design approach in these works is to first optimize layer 1 for the source-relay channel (higher SNR), and then optimize layers 1+2 for the source-destination channel (lower SNR) constrained to be consistent with layer 1. This approach extends the classical single-layer linear-programming ensemble-design framework by taking the layer 1+2 optimization variables to be coefficients of a *bi-variate* degree-distribution polynomial, and adding the layer-1 consistency constraints. This extension, however, entails solving an optimization problem with many variables: the *product* of the maximal degrees in layer 1 and layer 2. Furthermore, extending this approach to L -layer codes would make the number of variables grow exponentially with L (viewed as MET codes [6], the canonical representation of an L -layer ensemble is by an L -variate polynomial, whose number of coefficients grows exponentially with L for a given maximum layer degree.) Our results in this paper provide a design alternative that avoids this multiplicative/exponential growth of complexity, and also extend beyond optimization frameworks to offer analytical insight.

Relevant to our techniques is the work in [17] where (single-layer) LDPC codes are used for a BEC with an erasure rate that can take one of two values: full decoding is sought for the better channel, while for the worse channel some partial correction performance is specified. More related work includes non-asymptotic design of bilayer LDPC codes such as codes that enable decoder parallelism [13] and codes for incremental redundancy [14], [15], [16] (whose layer 2 is an LDGM code). Another indication of interest in bilayer/multi-layer codes is a parallel work on (non-LDPC) algebraic bilayer/multi-layer codes [18], [19], [20], [21]. Finally, we note a different framework that constructs codes with multiple thresholds, using a puncturing technique [22], [23], [24], which is useful for communication-with-feedback applications. The key difference is that bi- and multi-layer codes allow decoding the same variable nodes with successively more powerful decoders, while punctured codes need to add variable nodes to improve decoding.

II. PRELIMINARIES AND NOTATIONS

A. LDPC Codes

A linear block code is an LDPC code if it has at least one parity-check matrix that is sparse, i.e., the number of 1's in

H is linear in the block length. Every parity-check matrix H can be represented by a bipartite graph, called a Tanner graph, with nodes partitioned to variable nodes and check nodes; there exists an edge between check node i and variable node j , if and only if $H_{ij} = 1$ (this paper focuses on binary linear codes, but this representation can be generalized). In single-edge-type LDPC codes, the fraction of variable (resp. check) nodes in a Tanner graph with degree i is denoted by Λ_i (resp. Ω_i), and the fraction of edges connected to variable (resp. check) nodes of degree i is denoted by λ_i (resp. ρ_i); Λ_i and Ω_i are called node-perspective degree distributions, and λ_i and ρ_i are called edge-perspective degree distributions. The degree-distribution polynomials associated to a Tanner graph are given by

$$\Lambda(x) = \sum_i \Lambda_i x^i, \quad \lambda(x) = \sum_i \lambda_i x^{i-1}, \quad x \in [0, 1], \quad (1)$$

$$\Omega(x) = \sum_i \Omega_i x^i, \quad \rho(x) = \sum_i \rho_i x^{i-1}, \quad x \in [0, 1]. \quad (2)$$

B. Bilayer LDPC Codes

1) *Graph Structure*: In the bilayer Tanner graph, the check nodes are divided into two disjoint sets: *type-1* and *type-2* check nodes. The edges of the graph are partitioned into two sets as well: edges connecting variable nodes to type-1 (resp. type-2) check nodes are type-1 (resp. type-2) edges (see Figure 1). Finally, each variable node has two degrees, corresponding to the two edge types.

Bilayer codes can also be constructed with a *sub-block structure* [11], that is, the variable nodes are partitioned to disjoint sets called sub-blocks, and each set connects to a separate layer-1 graph with disjoint type-1 check nodes, while the layer-2 graph is shared among all sub-blocks as in Figure 1. Our results in the sequel carry over to this structure, but for simplicity we present them for the basic structure depicted in Figure 1.

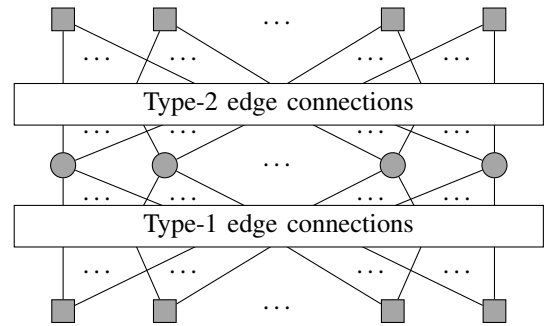


Fig. 1: An illustration of a bilayer Tanner graph.

We denote by $\Lambda_i^{(1)}$ (resp. $\Lambda_i^{(2)}$) the fraction of variable nodes with type-1 (resp. type-2) degree i . Similarly, $\Omega_i^{(1)}$ (resp. $\Omega_i^{(2)}$) is the fraction of type-1 (resp. type-2) check nodes of degree i . $\lambda_i^{(1)}$ (resp. $\lambda_i^{(2)}$) designates the fraction of type-1 (resp. type-2) edges connected to a variable node with type-1 (resp. type-2) degree i , and $\rho_i^{(1)}$ (resp. $\rho_i^{(2)}$)

designates the fraction of type-1 (resp. type-2) edges connected to a type-1 (resp. type-2) check node of degree i . Note that since the type-1 sub-graph (i.e., variable nodes, type-1 check nodes, and type-1 edges) is supposed to be used in decoding without the type-2 check nodes and edges, then $\Lambda_0^{(1)} = \Lambda_1^{(1)} = 0$. However, this is not the case with the type-2 sub-graph (which is assumed to be used together with the type-1 sub-graph), and we do allow $\Lambda_0^{(2)}, \Lambda_1^{(2)} > 0$ (see also [7]). In sections III–IV, we use P_0 to denote the coefficient $\Lambda_0^{(2)}$. The type-1 and type-2 degree-distribution polynomials $\Lambda^{(i)}(\cdot), \lambda^{(i)}(\cdot), \Omega^{(i)}(\cdot), \rho^{(i)}(\cdot), i \in \{1, 2\}$ are defined similarly to the degree-distribution polynomials for ordinary LDPC codes in (1)–(2).

The ensembles induced by the above description of bilayer graphs are characterized by the block length n , and the above degree distributions. The design rate of this ensemble is given by

$$R = 1 - \frac{\int_0^1 \rho^{(1)}(x) dx}{\int_0^1 \lambda^{(1)}(x) dx} - \frac{\int_0^1 \rho^{(2)}(x) dx}{\int_0^1 \lambda^{(2)}(x) dx} (1 - P_0). \quad (3)$$

We can see in (3), that setting $P_0 > 0$ allows increasing the code rate, which we later find crucial in our constructions.

2) *Density Evolution*: We distinguish between decoding using only layer 1, i.e., the type-1 sub-graph, and decoding using the entire graph (both layers). This separation yields a two-level threshold behavior: when using only layer 1, the asymptotic threshold is denoted by ϵ_1^* , and when using both layers the threshold is marked as ϵ_2^* . Since the second layer can only help, then $\epsilon_2^* \geq \epsilon_1^*$. However, decoding using only layer 1 has complexity advantages, and if the signal-to-noise ratio (SNR) is high enough, then layer 1 suffices. Moreover, in applications where bilayer codes have sub-block structure, layer-1 decoding is performed on fewer variable nodes than the full code block (see [11]), further reducing the complexity.

When decoding the type-1 sub-graph, one can use the known density evolution method for LDPC codes to analyze the performance. Specifically, the fraction of variable-to-check erasure messages after l BP iterations over the BEC(ϵ), $x_l(\epsilon)$, is given by the recursive equation [6]

$$x_l(\epsilon) = \epsilon \lambda^{(1)}(1 - \rho^{(1)}(1 - x_{l-1}(\epsilon))).$$

From this, ϵ_1^* can be calculated via

$$\epsilon_1^* = \inf_{x \in (0,1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))}. \quad (4)$$

When decoding both layers, one should consider both degree-distribution pairs for the analysis, since, as a specific instance of MET codes, the graph structure plays a crucial role in the decoding analysis. While MET codes can be specified in full generality using degree-distributions multinomials [6], more compact representations are typically helpful for obtaining effective analysis and design tools for particular classes of MET codes. For example, in [7], it is shown that for a variable node with type-1 and type-2 degrees i and j , respectively, and incoming type-1 and type-2 densities (erasure rates) $u \in [0, 1]$

and $w \in [0, 1]$, respectively, the outgoing type-1 and type-2 densities x and y , respectively, are given by [7, Eq. (15)–(16)]

$$x = \epsilon \cdot w^j \cdot u^{i-1}, \quad y = \epsilon \cdot w^{j-1} \cdot u^i.$$

The authors in [7] define the bilayer code through a product variable-node degree distribution specified by a bivariate polynomial with coefficients $\lambda_{i,j}$ (i for layer 1 and j for layer 2) and regular check node degrees. They then pursue code design using linear-programming optimization of the product degree distribution $\lambda_{i,j}$, under the constraint that it is consistent with a given (capacity approaching) degree distribution for layer 1.

We take a different approach and specify (and then design) the code ensembles through the individual degree distributions $(\lambda^{(1)}(\cdot), \rho^{(1)}(\cdot))$ and $(\lambda^{(2)}(\cdot), \rho^{(2)}(\cdot), P_0)$. As we will later see, this approach offers analysis and construction advantages compared to [7].

III. THRESHOLD

In this section, we study the asymptotic decoding threshold of bilayer ensembles, as defined in Section II, when using both layers. Our ultimate goal (in Section IV) is to provide a design tool for constructing bilayer LDPC codes: given two noise levels, ϵ_1 and ϵ_2 , choose $(\lambda^{(1)}(\cdot), \rho^{(1)}(\cdot))$ such that layer 1 provides the correction capability to tolerate ϵ_1 , and then set $(\lambda^{(2)}(\cdot), \rho^{(2)}(\cdot), P_0)$ such that the entire graph provides the correction capability to tolerate ϵ_2 . Working with $(\lambda^{(2)}(\cdot), \rho^{(2)}(\cdot), P_0)$ instead of the product degree distribution as in [7] enables, among other benefits, finding capacity-approaching sequences for the full-graph code. The derivations in this section lay the theoretical infrastructure needed to show the optimality of our constructions (i.e., capacity achieving in Section IV).

We assume that the message scheduling when decoding the entire graph is a flooding schedule: the variable nodes send messages to the type-1 and type-2 check nodes in parallel (later in Section VI, we change the schedule from flooding to be more complexity aware). Consider a random instance from the bilayer ensemble characterized by block length n and degree distributions $\Lambda^{(1)}, \Lambda^{(2)}, \Omega^{(1)}, \Omega^{(2)}$ (recall that $\Lambda^{(2)}$ includes P_0). Let $x_l(\epsilon)$ and $y_l(\epsilon)$ denote the probability that a type-1 and type-2 edge, respectively, carry a variable-to-check erasure message after l BP iterations over the BEC(ϵ) as $n \rightarrow \infty$. In view of the MET density-evolution equations in [6], we have

$$x_l(\epsilon) = \epsilon \cdot \lambda^{(1)}(1 - \rho^{(1)}(1 - x_{l-1}(\epsilon))) \cdot \Lambda^{(2)}(1 - \rho^{(2)}(1 - y_{l-1}(\epsilon))), \quad l \geq 0, \quad (5a)$$

$$y_l(\epsilon) = \epsilon \cdot \Lambda^{(1)}(1 - \rho^{(1)}(1 - x_{l-1}(\epsilon))) \cdot \lambda^{(2)}(1 - \rho^{(2)}(1 - y_{l-1}(\epsilon))), \quad l \geq 0, \quad (5b)$$

$$x_{-1}(\epsilon) = y_{-1}(\epsilon) = 1. \quad (5c)$$

Figure 2 graphically illustrates equations (5a)–(5b): in the center diagram the right outgoing edge carries the message in (5a) to a type-1 check and the left outgoing edge carries the message in (5b) to a type-2 check. To simplify notations, ϵ will be omitted from now on if it is clear from the context.

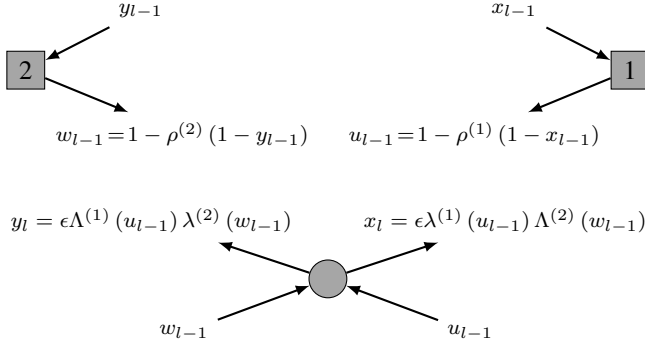


Fig. 2: Illustration of the DE equations (5a)–(5b).

Remark 1. Although x_l and y_l in (5a)–(5b) seem symmetric to each other, it is not necessarily true since we allow variable nodes to have type-2 degrees 0 ($P_0 > 0$) or 1 ($\lambda^{(2)}(0) > 0$), while their type-1 degrees are forced to be strictly greater than 1. This asymmetry has a crucial effect on the decoding process which is explained and detailed later in this section. Symmetry does hold in the special and less interesting case where $\rho^{(1)}(x) = \rho^{(2)}(x)$ and $\lambda^{(1)}(x) = x^{l_1-1}$, $\lambda^{(2)}(x) = x^{l_2-1}$, in which case, for every iteration $l \geq 0$, $y_l = x_l = \epsilon \lambda(1 - \rho(1 - x_{l-1}))$, where, $\rho(x) \triangleq \rho^{(1)}(x)$, $\lambda(x) \triangleq x^{l_1+l_2-1}$. Thus, if we use identical degree-distributions for type-1 and type-2 check nodes and we force all variable nodes to have regular degrees, then the 2D-DE equations in (5a)–(5b) degenerate to the already known 1D-DE equation. However, codes falling under this special case are less interesting because they are sub-optimal in their rates and restricted in their thresholds.

Define

$$f(\epsilon, x, y) = \epsilon \cdot \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \cdot \Lambda^{(2)}(1 - \rho^{(2)}(1 - y)), \quad x, y, \epsilon \in [0, 1] \quad (6a)$$

$$g(\epsilon, x, y) = \epsilon \cdot \Lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \cdot \lambda^{(2)}(1 - \rho^{(2)}(1 - y)), \quad x, y, \epsilon \in [0, 1] \quad (6b)$$

such that (5a)–(5c) can be re-written as

$$x_l = f(\epsilon, x_{l-1}, y_{l-1}), \quad l \geq 0 \quad (7a)$$

$$y_l = g(\epsilon, x_{l-1}, y_{l-1}), \quad l \geq 0 \quad (7b)$$

$$x_{-1} = y_{-1} = 1. \quad (7c)$$

Lemma 1. The functions f and g are monotonically non-decreasing in all of their variables.

Proof: Since the images of $\lambda^{(1)}(\cdot)$, $\Lambda^{(1)}(\cdot)$, $\lambda^{(2)}(\cdot)$, $\Lambda^{(2)}(\cdot)$, $\rho^{(1)}(\cdot)$ and $\rho^{(2)}(\cdot)$ lie in $[0, 1]$, then f and g are monotonically non-decreasing in $\epsilon \in [0, 1]$. The proof for x, y is similar. ■

Definition 1. Let $\epsilon \in (0, 1)$. We say that $(x, y) \in [0, 1]^2$ is an

(f, g) -fixed point if

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f(\epsilon, x, y) \\ g(\epsilon, x, y) \end{pmatrix}. \quad (8)$$

Clearly, for every $\epsilon \in (0, 1)$, $(x, y) = (0, 0)$ is a trivial (f, g) -fixed point. However, it is not clear yet if there exists a non-trivial (f, g) -fixed point. In particular, we ask: for which choices of ϵ , $\lambda^{(1)}$, $\rho^{(1)}$, $\lambda^{(2)}$, $\rho^{(2)}$ and P_0 there exists a non-trivial (f, g) -fixed point? The following lemmas help answering this question.

Lemma 2. Let $\epsilon \in (0, 1)$, and let $(x, y) \in [0, 1]^2$ be an (f, g) -fixed point. Then,

- 1) $x = 0$ implies $y = 0$, and if $P_0 = 0$ or $\lambda^{(2)}(0) > 0$, then $y = 0$ implies $x = 0$.
- 2) $(x, y) \in [0, \epsilon]^2$.
- 3) If $\{x_l\}_{l=0}^{\infty}$ and $\{y_l\}_{l=0}^{\infty}$ are defined by (7a)–(7c), then

$$x_l \geq x, \quad y_l \geq y, \quad \forall l \geq 0. \quad (9)$$

Proof: See Appendix A. ■

Remark 2. Item 1 in Lemma 2 expresses the asymmetry (discussed in Remark 1) between the type-1 and type-2 densities. Note that the ensembles where symmetry does not hold are those that have $\Lambda_0^{(2)} > 0$ and $\Lambda_1^{(2)} = 0$, a common combination in our constructions later in the paper.

Lemma 3. Let x_l and y_l be defined by (7a)–(7c) and let $0 < \epsilon \leq \epsilon' < 1$. Then,

$$x_{l+1}(\epsilon) \leq x_l(\epsilon), \quad y_{l+1}(\epsilon) \leq y_l(\epsilon), \quad \forall l \geq 0, \quad (10a)$$

and

$$x_l(\epsilon) \leq x_l(\epsilon'), \quad y_l(\epsilon) \leq y_l(\epsilon'), \quad \forall l \geq 0. \quad (10b)$$

Proof: By mathematical induction on l and by Lemma 1. ■

In view of (7a)–(7c), it can be verified that for every iteration $l \geq 0$, $x_l(0) = y_l(0) = 0$, $x_l(1) = y_l(1) = 1$. Since x_l and y_l are bounded from below by 0, then Lemma 3 implies that the limits $\lim_{l \rightarrow \infty} x_l(\epsilon)$ and $\lim_{l \rightarrow \infty} y_l(\epsilon)$ exist. Thus we can define a decoding threshold by

$$\epsilon_2^* = \sup \left\{ \epsilon \in [0, 1]: \lim_{l \rightarrow \infty} y_l(\epsilon) = \lim_{l \rightarrow \infty} x_l(\epsilon) = 0 \right\}. \quad (11)$$

Note that from the continuity of g in (6b), item 1 in Lemma 2 implies that if $\lim_{l \rightarrow \infty} x_l(\epsilon) = 0$, then $\lim_{l \rightarrow \infty} y_l(\epsilon) = 0$. Thus, (11) can be re-written as

$$\epsilon_2^* = \sup \left\{ \epsilon \in [0, 1]: \lim_{l \rightarrow \infty} x_l(\epsilon) = 0 \right\}. \quad (12)$$

Theorem 4. $\epsilon_2^* = \sup \{ \epsilon \in [0, 1]: (8) \text{ has no solution with } (x, y) \in (0, 1] \times [0, 1] \}$.

Proof: Mark $\hat{\epsilon}$ as the supremum of all $\epsilon \in [0, 1]$ such that (8) has no solution with $(x, y) \in (0, 1] \times [0, 1]$, let $\epsilon < \hat{\epsilon}$, and let $x(\epsilon) = \lim_{l \rightarrow \infty} x_l(\epsilon)$, $y(\epsilon) = \lim_{l \rightarrow \infty} y_l(\epsilon)$. Taking the limit $l \rightarrow \infty$ in (7a)–(7c) yields that $(x(\epsilon), y(\epsilon))$ is an (f, g) -fixed point. Since $\epsilon < \hat{\epsilon}$, it follows that $x(\epsilon) = 0$. From (12) we have $\epsilon < \epsilon_2^*$, for every $\epsilon < \hat{\epsilon}$; this implies that $\hat{\epsilon} \leq \epsilon_2^*$.

For the other direction, let $\epsilon > \hat{\epsilon}$ and let (z_1, z_2) be an (f, g) -fixed point such that $z_1 > 0$. Lemma 2-item 3 implies that for every iteration l , $x_l(\epsilon) \geq z_1 > 0$, $\forall l \geq 0$, thus $\lim_{l \rightarrow \infty} x_l(\epsilon) > 0$, where the existence of this limit is assured due to Lemma 3; hence, $\epsilon > \epsilon_2^*$. Since this is true for all $\epsilon > \hat{\epsilon}$, then we deduce that $\hat{\epsilon} \geq \epsilon_2^*$ and complete the proof. ■

We proceed by providing a numerical way to calculate the threshold of a given set of degree distributions. Define

$$\begin{aligned} q_1(x) &\triangleq x \cdot \frac{\Lambda^{(1)}(1 - \rho^{(1)}(1 - x))}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))}, \quad x \in (0, 1], \\ q_2(x) &\triangleq x \cdot \frac{\Lambda^{(2)}(1 - \rho^{(2)}(1 - x))}{\lambda^{(2)}(1 - \rho^{(2)}(1 - x))}, \quad x \in (0, 1]. \end{aligned} \quad (13)$$

Lemma 5. $\lim_{x \rightarrow 0} q_1(x) = 0$.

Proof: See Appendix B. ■

Since $q_1(1) = 1$, Lemma 5 and the intermediate-value theorem imply that for every $w \in (0, 1]$, there exists $x \in (0, 1]$ such that $q_1(x) = w$. Note that it is not true in general that $\lim_{x \rightarrow 0} q_2(x) = 0$ (another evidence of asymmetry); this limit may be infinite (for example the case $P_0 > 0$, $\rho^{(2)}(x) = x^3$ and $\lambda^{(2)}(x) = x^2$).

Definition 2. For every $y > 0$ such that $q_2(y) \leq 1$ define

$$q(y) \triangleq \max\{x : q_1(x) = q_2(y)\}. \quad (14)$$

Theorem 6. Let $\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}$ be degree-distribution polynomials, let $P_0 \in [0, 1]$, and let ϵ_2^* be the decoding threshold of the bilayer ensemble characterized by these degree distributions as $n \rightarrow \infty$.

If $P_0 = 0$ or $\lambda^{(2)}(0) > 0$, then

$$\epsilon_2^* = \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)}. \quad (15)$$

Else,

$$\epsilon_2^* = \min \left\{ \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)}, \frac{1}{P_0} \inf_{(0, 1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))} \right\}. \quad (16)$$

Proof: See Appendix C. ■

Remark 3. Although the right-hand side of (15) and the first argument in the min operator of (16) only have the variable y in them and thus may appear to only depend on layer 2, in fact their values depend on both layers through the function $q(\cdot)$. Moreover, it is not clear, a priori, which argument of the min operator of (16) will be smaller, and one must use the above procedure to calculate the values.

Example 1. Consider an ensemble characterized by $\lambda^{(1)}(x) = x$, $\rho^{(1)}(x) = x^9$, $\lambda^{(2)}(x) = 0.3396x + 0.6604x^4$, $P_0 = 0.2667$, $\rho^{(2)}(x) = x^9$. Using (3) and (4), the design rate is $R = 0.5571$ and the type-1 decoding threshold is $\epsilon_1^* = 0.1112$. In view

of (16), the decoding threshold¹ when using both layers is $\epsilon_2^* = \min\{0.35, 0.4168\} = 0.35$. Figure 3 illustrates the 2D-DE equations in (5a)-(5c) for three different erasure probabilities: 0.33, 0.35, 0.37. When the channel's erasure probability is $\epsilon = 0.33$, there are no non-trivial (f, g) -fixed points – the decoding process ends successfully, and when $\epsilon = 0.37$, there are two (f, g) -fixed points, $(0.335, 0.3202)$ and $(0.2266, 0.1795)$ – the decoding process gets stuck at $(0.335, 0.3202)$. When $\epsilon = 0.35 = \epsilon_2^*$, there is exactly one (f, g) -fixed point at $(0.27, 0.237)$, and the dashed and dotted lines are tangent.

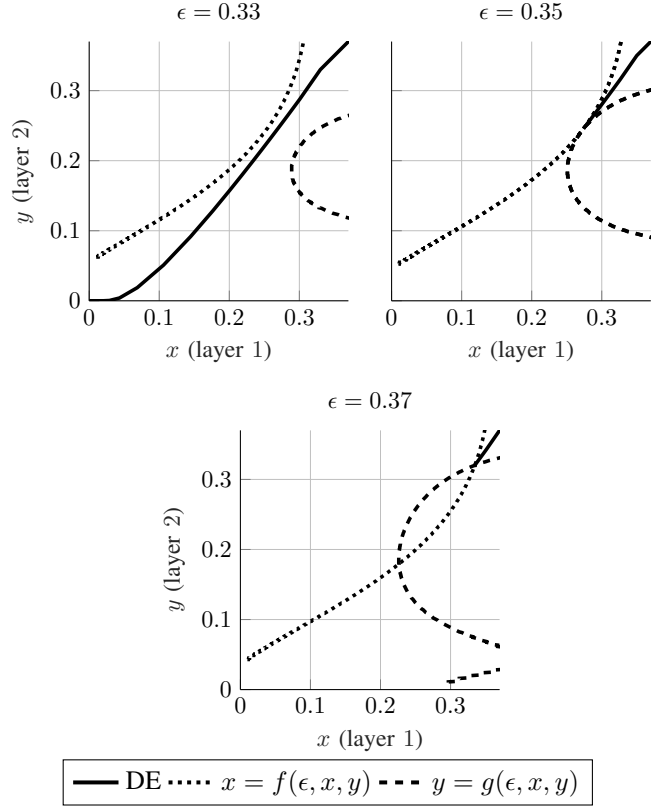


Fig. 3: Illustration of the density-evolution equations in (5a)-(5c) for the ensemble in Example 1, which induce a decoding threshold of $\epsilon_2^* = 0.35$. The evolved channel erasure probabilities are $\epsilon = 0.33, 0.35, 0.37$.

IV. CODE CONSTRUCTION AND APPROACHING CAPACITY

In this section, we present a code construction, and show how to use this construction to optimally combine two degree distributions (type 1 and type 2) in order to approach capacity. The inputs for the construction are the desired decoding

¹better thresholds for that rate are achieved in the next section, and these degree distributions are given to graphically exemplify the results derived so far.

thresholds, ϵ_1 and ϵ_2 , and the outputs are degree-distributions $(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0)$ such that

$$\epsilon_1^* (\lambda^{(1)}, \rho^{(1)}) = \epsilon_1 \quad \epsilon_2^* (\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0) = \epsilon_2.$$

The specified parameters ϵ_1 and ϵ_2 can be arbitrarily chosen as fit for the specific application using the codes. ϵ_2 is logically chosen to meet the “worst-case” noise level in extreme channel instances, while ϵ_1 should specify a lower noise tolerance that is sufficient for a significant fraction of channel instances.

In principle, setting $P_0 = 0$, and picking any two LDPC ensembles $(\lambda^{(1)}, \rho^{(1)})$ and $(\lambda^{(2)}, \rho^{(2)})$ that induce thresholds ϵ_1 and ϵ_2 , respectively, would suffice, but this choice yields poor rates (intuitively, with that choice the type-1 and type-2 codes do not “cooperate”). Another solution is not using the second layer at all, i.e., choosing $(\lambda^{(1)}, \rho^{(1)})$ that induce a threshold that equals to ϵ_2 , and setting $P_0 = 1$. However, this solution is undesired since it would miss the opportunity to have a low-complexity decoder for the majority of decoding instances where the erasure probabilities are below ϵ_1 .

Definition 3. Let $(\lambda^{(1)}, \rho^{(1)})$ be type-1 degree-distribution (DD) polynomials, and let ϵ_1 be their decoding threshold. For $\epsilon \in (\epsilon_1, 1)$, let

- 1) $h_\epsilon(x) = \epsilon \cdot \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) - x$, $x \in [0, 1]$
- 2) $x_s(\epsilon) = \max\{x \in [0, 1] : h_\epsilon(x) \geq 0\}$
- 3) $a_s(\epsilon) = \Lambda^{(1)}(1 - \rho^{(1)}(1 - x_s(\epsilon)))$

For every $x \in [0, 1]$, $h_\epsilon(x)$ is the erasure-probability change in one BP iteration on the type-1 sub-graph, if the current erasure probability is x . By definition, since $\epsilon > \epsilon_1$, the type-1 sub-graph fails to decode, and $h_\epsilon(x) > 0$ for some $x \in [0, 1]$. In addition, for every $x > \epsilon$, $h_\epsilon(x) < 0$, so $x_s(\epsilon)$ is well defined. Operationally, $x_s(\epsilon)$ is the type-1 erasure probability when the decoder gets stuck (using only type-1 edges). Items 1 and 2 have appeared in [5]; we add $a_s(\epsilon)$ as a function of $x_s(\epsilon)$ that encapsulates the erasure probability passed from layer 1 to layer 2.

A. Code Construction

Construction 1.

Input: thresholds ϵ_1 and $\epsilon_2 > \epsilon_1$.

- 1) Choose any type-1 DD $(\lambda^{(1)}, \rho^{(1)})$ that induce a threshold ϵ_1 .
- 2) Calculate $a_s(\epsilon_2)$.
- 3) Choose any type-2 DD $(\lambda^{(2)}, \rho^{(2)})$ that induce a threshold $\epsilon_2 \cdot a_s(\epsilon_2)$.
- 4) Set $P_0 = \epsilon_1/\epsilon_2$.

Remark 4. The main feature of the above construction is that one can use off-the-shelf codes for the two layers and no further optimization is needed. Moreover, if the component codes are efficient (in rate) then so is their combination. We will later investigate how the type-1 and type-2 gaps to capacity affect the overall gap to capacity.

Theorem 7. Let $(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0)$ be degree distributions constructed by Construction 1. Then $\epsilon_2^*(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0) = \epsilon_2$.

Proof: From (4) and Theorem 6, since $P_0 = \epsilon_1/\epsilon_2 > 0$, we have

$$\epsilon_2^* \leq \frac{1}{P_0} \cdot \inf_{(0,1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))} = \frac{\epsilon_1}{P_0} = \epsilon_2. \quad (17)$$

For the opposite direction, let $\epsilon_1 < \epsilon < \epsilon_2$. In view of Theorem 4, it suffices to show that (8) has no solution for $(x, y) \in (0, 1] \times [0, 1]$. In view of (6a), for every $x \in (0, 1]$,

$$\begin{aligned} f(\epsilon, x, 0) &= \epsilon \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) P_0 \\ &< P_0 \epsilon_2 \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \\ &= \epsilon_1 \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \\ &\leq x. \end{aligned} \quad (18)$$

Furthermore, Definition 3 implies that for every $(x, y) \in (x_s(\epsilon), 1) \times [0, 1]$,

$$\begin{aligned} f(\epsilon, x, y) &= \epsilon \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \Lambda^{(2)}(1 - \rho^{(2)}(1 - y)) \\ &< x \Lambda^{(2)}(1 - \rho^{(2)}(1 - y)) \\ &\leq x, \end{aligned} \quad (19)$$

and from Lemma 1, if $(x, y) \in (0, x_s(\epsilon)] \times (0, 1]$,

$$\begin{aligned} g(\epsilon, x, y) &\leq g(\epsilon, x_s(\epsilon), y) \\ &= \epsilon \lambda^{(2)}(1 - \rho^{(2)}(1 - y)) \Lambda^{(1)}(1 - \rho^{(1)}(1 - x_s(\epsilon))) \\ &= \epsilon \lambda^{(2)}(1 - \rho^{(2)}(1 - y)) a_s(\epsilon). \end{aligned} \quad (20)$$

Since $\epsilon < \epsilon_2$, then $\epsilon_2 \cdot a_s(\epsilon_2) > \epsilon \cdot a_s(\epsilon)$; thus (20) yields

$$g(\epsilon, x, y) < y, \quad \forall (x, y) \in (0, x_s(\epsilon)] \times (0, 1]. \quad (21)$$

Combining (18), (19), and (21) implies that (8) has no solution in $(0, 1] \times [0, 1]$. Thus $\epsilon_2^* \geq \epsilon$. Since this is true for any $\epsilon < \epsilon_2$, we conclude that

$$\epsilon_2^* \geq \epsilon_2,$$

which combined with (17) completes the proof. \blacksquare

Remark 5. In most cases, it is hard to produce an analytical expression for $x_s(\epsilon)$, but if we limit the type-1 degrees of the ensemble to be small, then a closed-form expression could be derived for $x_s(\epsilon)$ and $a_s(\epsilon)$.

Example 2. Consider type-1 degree distributions taking the form:

$$\lambda^{(1)}(x) = x, \quad \rho^{(1)}(x) = \rho_2^{(1)}x + \rho_3^{(1)}x^2 + \rho_4^{(1)}x^3, \quad (22)$$

where $\rho_i^{(1)} \geq 0$, $\sum_{i=2}^4 \rho_i^{(1)} = 1$. In view of (4), for the family of ensembles given in (22), $\epsilon_1 = \frac{1}{1 + \rho_3^{(1)} + 2\rho_4^{(1)}}$. In addition, for every $\epsilon_2 \in (\epsilon_1, 1)$,

$$x_s(\epsilon_2) = \begin{cases} 1 - \frac{1}{\rho_3^{(1)}}(1/\epsilon_2 - 1), & \rho_4^{(1)} = 0 \\ \frac{\rho_3^{(1)} + 3\rho_4^{(1)} - \sqrt{(\rho_3^{(1)} + \rho_4^{(1)})^2 + 4\rho_4^{(1)}(1/\epsilon_2 - 1)}}{2\rho_4^{(1)}}, & \rho_4^{(1)} > 0 \end{cases}.$$

Finally, since $\Lambda^{(1)}(x) = x^2$, and in view of Definition 3,

$$a_s(\epsilon_2) = \Lambda^{(1)}(1 - \rho^{(1)}(1 - x_s(\epsilon_2))) = \left(\frac{x_s(\epsilon_2)}{\epsilon_2} \right)^2.$$

These closed-form expressions of x_s and a_s can be used for constructing a layer-1 code designed for certain parameters ϵ_1, ϵ_2 , using a simple optimization of the parameters $\rho_2^{(1)}, \rho_3^{(1)}, \rho_4^{(1)}$. This optimization maximizes the fraction of bits layer 1 uncovers for layer 2 in case the channel parameter is ϵ_2 , while guaranteeing its own noise resilience ϵ_1 (we omit the details here).

B. Approaching Capacity

In this sub-section, we show how to approach capacity in the bilayer framework, using the construction proposed in Section IV-A. More specifically, we upper bound the bilayer additive gap to capacity with a certain linear combination of the gaps to capacity of the two component codes. During the derivation, we refer to $\delta(\lambda, \rho)$ as the additive gap to capacity of the (single layer) LDPC(λ, ρ) ensemble, i.e., $\delta(\lambda, \rho) = 1 - \epsilon^*(\lambda, \rho) - R(\lambda, \rho)$. Similarly, we define $\delta(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0) = 1 - \epsilon^*(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0) - R(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0)$ as the bilayer gap to capacity.

Definition 4. A sequence of bilayer degree distributions $\{\lambda^{(1;k)}(\cdot), \rho^{(1;k)}(\cdot), \lambda^{(2;k)}(\cdot), \rho^{(2;k)}(\cdot), P_{0k}\}_{k \geq 1}$ with associated rates $\{R_k\}_{k \geq 1}$ is said to approach capacity on a BEC with two channel parameters $0 < \epsilon_1 < \epsilon_2 < 1$ if:

- 1) The threshold of layer 1 approaches ϵ_1 as $k \rightarrow \infty$.
- 2) The threshold with both layers approaches ϵ_2 as $k \rightarrow \infty$.
- 3) R_k approaches $1 - \epsilon_2$ as $k \rightarrow \infty$.

Note that items 2 and 3 imply that $\lim_{k \rightarrow \infty} \delta(\lambda^{(1;k)}(\cdot), \rho^{(1;k)}(\cdot), \lambda^{(2;k)}(\cdot), \rho^{(2;k)}(\cdot), P_{0k}) = 0$.

Lemma 8. Let $(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0)$ be bilayer degree-distribution polynomials constructed according to Construction 1, and let $\delta_1 \triangleq \delta(\lambda^{(1)}, \rho^{(1)})$ and $\delta_2 \triangleq \delta(\lambda^{(2)}, \rho^{(2)})$ be the type-1 and type-2 (additive) gaps to capacity, respectively. Then,

$$\delta(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0) \leq \delta_1 + \delta_2 \cdot (1 - P_0). \quad (23)$$

Proof: See Appendix D for the proof for the general case with $L \geq 2$ layers. ■

Remark 6. In principle, the bound in Lemma 8 is not tight since $a_s(\epsilon_2)\epsilon_2 < \epsilon_2$, and the gap to capacity is smaller. However, since the left-hand side of the last inequality depends on the particular degree distributions used in Construction 1, the bound (23) has the advantage of applying in full generality.

At this point, it should be clear how to construct a capacity-approaching sequence of bilayer ensembles with two thresholds $0 < \epsilon_1 < \epsilon_2 < 1$. Choose any two sequences of ‘‘ordinary’’ LDPC ensembles $\{\lambda^{(1;k)}(\cdot), \rho^{(1;k)}(\cdot)\}_{k \geq 1}$ and $\{\lambda^{(2;k)}(\cdot), \rho^{(2;k)}(\cdot)\}_{k \geq 1}$ that achieve capacity on the BEC(ϵ_1)

and BEC(ϵ_2), respectively, and set $P_{0k} = (1 - \epsilon_1/\epsilon_2)$, for all $k \geq 1$. Item 1 in Definition 4 clearly holds for this sequence, and in view of Theorem 7, item 2 in Definition 4 holds as well. Finally, Lemma 8 implies that

$$\begin{aligned} & \lim_{k \rightarrow \infty} \delta(\lambda^{(1;k)}(\cdot), \rho^{(1;k)}(\cdot), \lambda^{(2;k)}(\cdot), \rho^{(2;k)}(\cdot), P_{0k}) \\ & \leq \lim_{k \rightarrow \infty} \delta(\lambda^{(1;k)}(\cdot), \rho^{(1;k)}(\cdot)) \\ & \quad + \lim_{k \rightarrow \infty} \delta(\lambda^{(2;k)}(\cdot), \rho^{(2;k)}(\cdot)) (1 - \epsilon_1/\epsilon_2) \\ & = 0. \end{aligned}$$

Example 3. We construct a bilayer capacity-achieving sequence with thresholds $\epsilon_1 = 0.05$ and $\epsilon_2 = 0.2$. We set $P_0 = \epsilon_1/\epsilon_2 = 0.25$ and use the Tornado capacity-approaching sequence [25],

$$\begin{aligned} \lambda^{(1)}(x) &= \frac{1}{H(D_1)} \sum_{i=1}^{D_1} \frac{x^i}{i}, & \lambda^{(2)}(x) &= \frac{1}{H(D_2)} \sum_{i=1}^{D_2} \frac{x^i}{i}, \\ \rho^{(1)}(x) &= e^{-\alpha_1} \sum_{i=0}^{\infty} \frac{(\alpha_1 x)^i}{i!}, & \rho^{(2)}(x) &= e^{-\alpha_2} \sum_{i=0}^{\infty} \frac{(\alpha_2 x)^i}{i!}, \end{aligned} \quad (24)$$

where $H(\cdot)$ is the harmonic sum, $\alpha_j = \frac{H(D_j)}{\epsilon_j}$, $j \in \{1, 2\}$ (the check degree-distribution series are truncated to get degree-distribution polynomials with finite degrees). D_1 (resp. D_2) controls the type-1 (resp. type-2) gap to capacity δ_1 (resp. δ_2); the bigger it is, the smaller the gap is.

Figure 4 exemplifies how the sequence $\{\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0\}$ approaches capacity as $D_1 \rightarrow \infty$, $D_2 \rightarrow \infty$: Theorem 7 implies that for every value of D_1 and D_2 , the global decoding threshold is $\epsilon_2^* \geq 0.2$; the type-1 gap to capacity δ_1 and type-2 gap to capacity δ_2 both vanish as $D_1 \rightarrow \infty$ and $D_2 \rightarrow \infty$, which in view of (23), implies that the overall gap to capacity δ vanishes as well. In addition, as demonstrated in Figure 4, δ_1 vanishes much faster with D_1 thanks to the lower ϵ_1 .

Remark 7. Figure 4 (in particular the second from bottom row in the table) shows the advantage of the bilayer scheme: one can get very close to capacity with type-1 ensembles that are extremely low complexity thanks to their low D_1 values.

C. Complexity Advantage

As mentioned in Section II-B2, decoding bilayer LDPC codes using only layer 1 has complexity advantages over decoding ordinary LDPC codes. This comes from the fact that since layer 1 is designed for lower noise levels, for the same gap to capacity its node degrees are lower than those of the ordinary LDPC. For the BEC, counting edges in the Tanner graph is a first-order approximation of the decoding complexity. We now perform a comparison between two codes: Code 1 is a bilayer LDPC code, and Code 2 an ordinary LDPC code. Code 1 is constructed by Construction 1 for erasure levels $0 < \epsilon_1 < \epsilon_2 < 1$, and Code 2 is designed for the worst-case channel parameter ϵ_2 . In particular, we take Code 1 degree

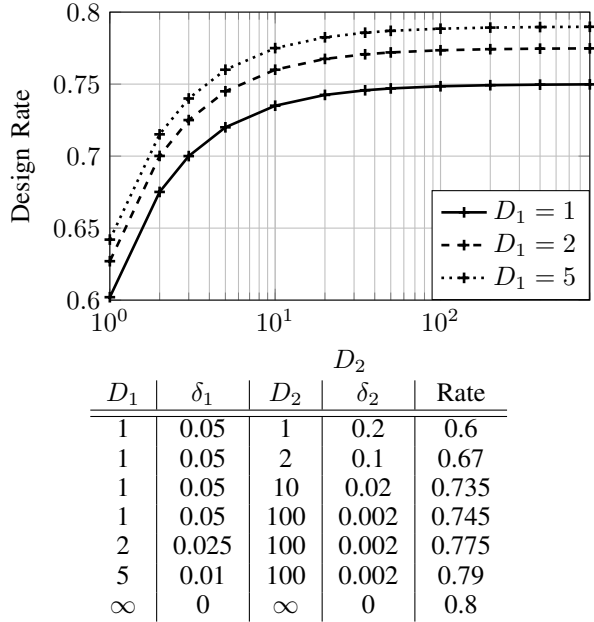


Fig. 4: Top: design rate of a the bilayer ensembles constructed by Construction 1 with two Tornado component codes from Example 3. The horizontal axis D_2 and the different plots D_1 are the degree of the polynomials $\lambda^{(2)}(\cdot)$ and $\lambda^{(1)}(\cdot)$, respectively. Bottom: type-1 gap to capacity δ_1 , type-2 gap to capacity δ_2 , and rate of the bilayer capacity-achieving sequence from Example 3.

distributions $(\lambda^{(1)}, \rho^{(1)}, \lambda^{(2)}, \rho^{(2)}, P_0)$ from Example 3 with $D_1 = 2, D_2 = 10$. Code 2's degree distributions are $\lambda^{(2)}, \rho^{(2)}$. The variable-node type-1 and type-2 average degrees in Code 1 (bilayer) are denoted by d_1 and d_2 , respectively. The variable-node average degree in Code 2 is denoted by d . It is known that if the variable-node degree distribution is given by $\lambda(\cdot)$, then their average degree is given by $1/\int_0^1 \lambda$. Hence

$$d_1 = \left(\int_0^1 \lambda^{(1)}(x) dx \right)^{-1} = 2.25,$$

$$d_2 = (1 - P_0) \cdot \left(\int_0^1 \lambda^{(2)}(x) dx \right)^{-1} = 2.41$$

$$d = \left(\int_0^1 \lambda^{(2)}(x) dx \right)^{-1} = 3.22.$$

The complexity reduction when decoding layer 1 is 30.16% compared to the ordinary LDPC code. When decoding both layers we pay with an increase of 45.37% in the average degree, but this applies only to the 2-layers decoder which we assume to be used either infrequently (if most decoding instances have erasure rates below ϵ_1) or by nodes where computational parsimony is less critical (compared to nodes performing layer-1 decoding).

V. GENERAL MULTI-LAYER CONSTRUCTION

In this section, we show how to generalize Construction 1 to more than two layers. This generalization is motivated, for example, by multi-block coding [11] with a hierarchical structure where a number of sub-blocks are joined to form a super-block, and a number of super-blocks are joined further, etc. Another example is a multiple-relay channel, with a source, $L - 1$ relays, and a destination (see [7] for the relay channel with $L = 2$). The source sends a message to all relays and destination, and for every $i \in \{1, 2, \dots, L - 1\}$, the i -th relay decodes its incoming message and forwards parity bits to relays $j \in \{i + 1, \dots, L - 1\}$, and to the destination.

The advantage of the framework developed in this paper toward the multi-layer extension is that the number of parameters of the ensemble grows linearly with L . An extension of [7] to multi-layer codes through multi-variate DD polynomials would imply exponential growth of the number of ensemble parameters.

A. Code Structure & Density Evolution

Let $L > 1$ be an integer. The L -layer ensemble is characterized by the block length n , and L degree-distributions polynomials $\{(\Lambda^{(i)}(\cdot), \Omega^{(i)}(\cdot))\}_{i=1}^L$. Each variable node has L types of edges emanating from it with degrees specified by $\{\Lambda^{(i)}(\cdot)\}_{i=1}^L$, and check nodes are divided into L types with degrees specified by $\{\Omega^{(i)}(\cdot)\}_{i=1}^L$ where check nodes can connect only to edges of the same type. For every layer $i \in \{1, 2, \dots, L\}$, we denote by $P_0^{(i)} = \Lambda^{(i)}(0)$ the fraction of variable-nodes with no type- i edges. From edge perspective the degree-distribution polynomials are given by $\{(\lambda^{(i)}(\cdot), \rho^{(i)}(\cdot), P_0^{(i)})\}_{i=1}^L$. Since layer 1 should have a positive threshold, then we require that $P_0^{(1)} = 0$. For $i \in \{2, 3, \dots, L\}$ we allow $P_0^{(i)} > 0$.

The generalization of the density-evolution equations in (5a)–(5c) for the multi-layer ensemble are given by

$$x_l^{(i)}(\epsilon) = \epsilon \cdot \lambda^{(i)} \left(1 - \rho^{(i)} \left(1 - x_{l-1}^{(i)}(\epsilon) \right) \right) \cdot \prod_{j \neq i} \Lambda_j \left(1 - \rho_j \left(1 - x_{l-1}^{(j)}(\epsilon) \right) \right) \quad (25)$$

$$\triangleq f_i \left(\epsilon, \mathbf{x}_{l-1}^{(L)}(\epsilon) \right), \quad l \geq 0, \quad \forall 1 \leq i \leq L,$$

$$x_{-1}^{(i)}(\epsilon) = 1, \quad \forall 1 \leq i \leq L,$$

where for every $i \in \{1, 2, \dots, L\}$, $x_l^{(i)}(\epsilon)$ is the probability that a type- i edge carries a variable-to-check erasure messages after l BP iterations over the BEC(ϵ), and $\mathbf{x}_l^{(i)}(\epsilon) = (x_l^{(1)}(\epsilon), x_l^{(2)}(\epsilon), \dots, x_l^{(i)}(\epsilon))$. In what follows, we omit ϵ from $\mathbf{x}_l^{(i)}(\epsilon)$, and for brevity we re-write (25) as $\mathbf{x}_l^{(i)} = F^{(i)} \left(\epsilon, \mathbf{x}_{l-1}^{(L)} \right)$ where $F^{(i)}: [0, 1]^{L+1} \rightarrow [0, 1]^i$ encapsulates the first i density-evolution equations.

B. Code Construction

Construction 2.

- *Input:* thresholds $0 < \epsilon_1 < \epsilon_2 < \dots < \epsilon_L < 1$.
- *Output:*

degree	distributions
--------	---------------

$$\left\{ \left(\lambda^{(i)}(\cdot), \rho^{(i)}(\cdot), P_0^{(i)} \right) \right\}_{i=1}^L$$
such that for every $i \in \{1, 2, \dots, L\}$, the decoding threshold of the first i layers equals ϵ_i .
 - 1) Choose any degree distributions $(\lambda^{(1)}, \rho^{(1)})$ such that $\epsilon^*(\lambda^{(1)}, \rho^{(1)}) = \epsilon_1$.
 - 2) For each $i \in \{1, 2, \dots, L-1\}$ do:
 - a) Calculate $(x_s^{(1)}, x_s^{(2)}, \dots, x_s^{(i)})$ as the largest (element-wise) fixed point of $F^{(i)}(\epsilon_{i+1}, x^{(1)}, x^{(2)}, \dots, x^{(i)}, 1, 1, \dots, 1)$.
 - b) Calculate $a_s^{(i)} = \prod_{j=1}^i \Lambda_j (1 - \rho_j(1 - x_s^{(j)}))$ (quantifies the amount by which layers $1, 2, \dots, i$ help when decoding $i+1$ layers).
 - c) Choose any degree distributions $(\lambda_{i+1}, \rho_{i+1})$ that induce a threshold $\epsilon_{i+1} \cdot a_s^{(i)}$ (since $a_s^{(i)} \leq 1$, it is possible to find such degree distributions [6, Section 3.15]).
 - d) Set $P_0^{(i+1)} = \epsilon_i / \epsilon_{i+1}$.

Note that any set of thresholds $0 < \epsilon_1 < \epsilon_2 < \dots < \epsilon_L < 1$ is achievable using Construction 2 since $a_s^{(i)} \leq 1$ for all $i \in \{1, 2, \dots, L\}$. As pointed out in Section IV-C for the two-layer case, there is a trade-off between the layers thresholds and their decoding complexity. For example, if we increase ϵ_i for some $1 \leq i < L$, then the complexity (in terms of edge density) of decoding layers $1, 2, \dots, i$ will increase as well.

Lemma 9. *Let δ be the additive gap to capacity of the above ensemble. Then,*

$$\delta \leq \sum_{i=1}^L \delta_i \cdot (1 - P_0^{(i)}), \quad (26)$$

where δ_i is the individual additive gap to capacity of the i -th layer $(\lambda^{(i)}, \rho^{(i)})$.

Proof: See Appendix D. ■

The advantage of the design approach suggested in this paper is made clearer in view of Construction 2. For each layer i we need only to calculate steps 2a–2b, and then choose any code that meets the criteria in step 2c. On the other hand, in the construction of [7], linear programming is used to optimize the product degree distribution of all layers; thus the complexity of Construction 2 is much smaller.

VI. REDUCING THE NUMBER OF TYPE-2 ITERATIONS

We now return to the specific case of two layers.

It has not been emphasized earlier in the paper, but in practical settings, the type-1 and type-2 decoding iterations may be very different in terms of cost. For example, the hardware that implements the layer-2 checks may be more costly to operate due to higher code complexity. That means that even when decoding the two layers, we would like to reduce the

number of layer-2 iterations. We define the number of layer-2 iterations performed during decoding by N_2 , where a layer-2 decoding iteration is a round of variable-to-type-2-check messages and type-2-check-to-variable messages. Ideally, the decoder successfully decodes on the type-1 sub-graph, and no type-2 iterations are needed ($N_2 = 0$); in the asymptotic regime, this happens when the fraction of erased bits ϵ is equal or less than the type-1 threshold, i.e., $\epsilon \leq \epsilon_1$. However, if $\epsilon > \epsilon_1$, then at least one type-2 iteration is necessary ($N_2 \geq 1$).

In this section, we suggest a scheduling scheme for updating layer 2 during the decoding of the entire graph. We prove that our scheduling scheme is optimal in the sense of minimizing N_2 . It is known that there is a trade-off between rate and the number of decoding iterations (see, for example, [16, Table III]). We extend this observation and study how the parameters of the type-1 and type-2 degree distributions affect N_2 when using the optimal scheduling scheme. Note that our notion of scheduling differs from the standard meaning of scheduling algorithms for iterative-decoding (see [26], [27], [28]). We consider scheduling of type-2 decoding iterations, while previous work considered the order of message passing between nodes in the Tanner graph.

A. An N_2 -optimal scheduling scheme

Recall that in the bilayer density-evolution equations, (7a) and (7b) express a type-1 and a type-2 iteration, respectively. A scheduling scheme prescribes decoder access to the type-2 check nodes in only part of the iterations, and thus replaces (7b) with

$$y_l = \begin{cases} g(\epsilon, x_{l-1}, y_{l-1}) & l \in A \\ y_{l-1} & l \notin A \end{cases} \quad (27)$$

for some $A \subseteq \mathbb{N}$ representing the iteration numbers where type-2 checks are accessed; in this case we have $N_2 = |A|$. Since Lemma 3 (monotonicity) still holds when (7b) is replaced with (27), the limits $\lim_{l \rightarrow \infty} x_l$ and $\lim_{l \rightarrow \infty} y_l$ exist for every scheduling scheme.

Given type-1 and type-2 degree distributions, a scheduling scheme is called *valid* if for every ϵ less than the ensemble's threshold ϵ_2 , $\lim_{l \rightarrow \infty} x_l(\epsilon) = 0$ (successful decoding). Our goal is to find an *optimal* scheduling scheme: a valid scheduling scheme that minimizes N_2 . For example, if $A = \emptyset$, then $N_2 = 0$ but $\lim_{l \rightarrow \infty} x_l(\epsilon) > 0$ if $\epsilon \in (\epsilon_1, \epsilon_2)$; thus, the scheduling scheme is not valid. If, on the other hand, type-2 checks are accessed in every iteration (as assumed in Sections III–IV), then the scheduling scheme is valid, but N_2 equals the total number of iterations, which is the worst case. We do not require the scheduling scheme to be pre-determined, and it can use “on-line” information about the decoding process. For example, it can use the current fraction of erasure messages or the change in this fraction between two consecutive iterations.

Definition 5. *Let $(\Lambda^{(2)}, \rho^{(2)})$ be type-2 degree-distribution polynomials, let $\epsilon \in (0, 1)$ be the erasure probability of a BEC, and let $y \in [0, 1]$ be an instantaneous erasure probability of a type-2 edge. We define the effective erasure probability from*

layer 1's perspective as

$$\epsilon_{\text{eff}}^{(1)}(\epsilon, y) = \epsilon \cdot \Lambda^{(2)} \left(1 - \rho^{(2)}(1 - y) \right). \quad (28)$$

In view of (6a) and (28), we have

$$x_l = f(\epsilon, x_{l-1}, y_{l-1}) = \epsilon_{\text{eff}}^{(1)}(\epsilon, y_{l-1}) \lambda^{(1)}(1 - \rho^{(1)}(1 - x_{l-1})). \quad (29)$$

$\epsilon_{\text{eff}}^{(1)}(\epsilon, y_{l-1})$ takes the role of ϵ when layer 1 is viewed as a standard LDPC code, hence the term ‘‘effective erasure probability from layer 1's perspective’’.

Our proposed scheduling scheme is parameterized by $\eta > 0$, and is given by

$$x_l = f(\epsilon, x_{l-1}, y_{l-1}) \\ y_l = \begin{cases} g(\epsilon, x_{l-1}, y_{l-1}) & |x_{l-2} - x_{l-1}| \leq \eta, \epsilon_{\text{eff}}^{(1)}(\epsilon, y_{l-1}) \geq \epsilon_1 \\ y_{l-1} & \text{else} \end{cases} \quad (30)$$

Lemma 10. *For every $\eta > 0$, the scheduling scheme described in (30) is valid.*

Proof: See Appendix E. ■

Note that if $\eta = 0$, the scheduling scheme described in (30) is not valid. However, since type-1 iterations have zero cost in our model, we can assume that we can apply arbitrarily many type-1 iterations to get arbitrarily close to $\eta = 0$. Numerical simulations show that $\eta = 10^{-4}$ suffices for achieving minimal N_2 . For the following analysis we will assume that $\eta = 0$, and that the scheduling scheme is still valid. In this scheduling scheme, the decoder tries to decode the type-1 sub-graph until it gets ‘‘stuck’’, which refers to not being able to reduce the erasure probability while it is still strictly greater than zero. This happens first when $x_{l_1} = x_s(\epsilon)$, for some iteration l_1 , where $x_s(\epsilon)$ is given in Definition 3. So, in the first type-2 update we have

$$x_{l_1} = x_s(\epsilon), \quad y_{l_1} = 1 \\ x_{l_1+1} = x_s(\epsilon), \quad y_{l_1+1} = g(\epsilon, x_s(\epsilon), 1).$$

In view of (29), the type-1 sub-graph now ‘‘sees’’ $\epsilon_{\text{eff}}^{(1)}(\epsilon, y_{l_1+1}) < \epsilon$ as an effective erasure probability, and it can continue the decoding algorithm without accessing type-2 edges. It may get ‘‘stuck’’ again and another type-2 update will be invoked; this procedure continues until $\epsilon_{\text{eff}}^{(1)}(\epsilon, y_{l_p+1}) < \epsilon_1$ in the p -th (and last) update, which enables successful decoding (i.e., $N_2 = p$). In general, let $\{l_k\}_{k=1}^{N_2}$ be the type-2 update iterations of the scheduling scheme described above and let ϵ_k be the effective erasure probability from layer-1 perspective between type-2 updates $k-1$ and k . Then,

$$y_{l_1} = 1, \quad \epsilon_1 = \epsilon, \quad x_{l_1} = x_s(\epsilon), \quad (31a)$$

$$y_{l_k} = g(\epsilon, x_{l_{k-1}}, y_{l_{k-1}}), \quad 2 \leq k \leq N_2, \quad (31b)$$

$$\epsilon_k = \epsilon_{\text{eff}}^{(1)}(\epsilon, y_{l_k}), \quad 2 \leq k \leq N_2, \quad (31c)$$

$$x_{l_k} = x_s(\epsilon_k), \quad 2 \leq k \leq N_2, \quad (31d)$$

where

$$\epsilon = \epsilon_1 > \epsilon_2 > \dots > \epsilon_{N_2-1} \geq \epsilon_1 > \epsilon_{N_2}. \quad (32)$$

Lemma 11. *The scheduling scheme described above is optimal.*

Proof: See Appendix F. ■

We assume from now on that the decoder applies the optimal scheduling scheme suggested above.

B. The Rate-vs.- N_2 Trade-Off

We will now see that the smaller the gap to capacity is, the higher N_2 is; therefore, to decrease N_2 we have to pay with rate, and there are several ways to do so. In this section we study how the parameters of the component layers affect N_2 . In particular, we focus on how the type-1 and type-2 additive gaps to capacity δ_1 and δ_2 , receptively, affect N_2 .

It is well known that if $\{\lambda^{(1;k)}, \rho^{(1;k)}\}_{k=1}^{\infty}$ is a (ordinary) capacity-approaching sequence for the BEC(ϵ_1), then

$$\lim_{k \rightarrow \infty} \epsilon_1 \lambda^{(1;k)}(1 - \rho^{(1;k)}(1 - x)) = x, \quad x \in [0, \epsilon_1] \quad (33)$$

(see [29]). This leads to the following lemma.

Lemma 12. *Let $0 < \epsilon_1 < \epsilon < 1$, and let $\{\lambda^{(1;k)}, \rho^{(1;k)}\}_{k=1}^{\infty}$ be a capacity-approaching sequence for the BEC(ϵ_1). Then,*

$$x_s(\epsilon) \triangleq \lim_{k \rightarrow \infty} x_s^{(1;k)}(\epsilon) = \epsilon,$$

where $x_s^{(1;k)}(\epsilon)$ corresponds to Definition 3 with $(\lambda^{(1;k)}, \rho^{(1;k)})$.

Proof: See Appendix G. ■

Lemma 12 asserts that if the type-1 degree-distribution polynomials imply a threshold ϵ_1 and a design rate that is very close to capacity $(1 - \epsilon_1)$, and the channel erasure probability ϵ is greater than ϵ_1 , then the BP decoding algorithm on the type-1 sub-graph gets ‘‘stuck’’ immediately after correcting only a small fraction of the erasures. This leads, in view of (31b), to a small change in the erasure-message probability on the type-2 update, which in turn yields a minor progress in the type-1 side. Therefore, choosing close to capacity type-1 degree-distribution polynomials implies high N_2 . Another consequence of (33) is that the change in the erasure-message probability in one iteration of the BP decoding algorithm is small. Thus, close to capacity type-2 degree-distribution polynomials yield high N_2 , regardless of the type-1 degree-distribution polynomials.

Example 4. *Let $\epsilon_1 = 0.05$ and $\epsilon_2 = 0.2$. We use the capacity-achieving sequence given in Example 3. A computer program simulated (31a)-(31d) with $\epsilon = 0.1998$ (99.9% of ϵ_2) and degree distributions from (24) with the same values of D_1 and D_2 as in Figure 4. The results are presented in Figure 5. Figure 4 and Figure 5 exemplify the trade-off between rate and N_2 : when the ensemble is close to capacity with $\delta_1 = 10^{-2}$, $\delta_2 = 2.5 \cdot 10^{-4}$ ($D_1 = 5$, $D_2 = 800$, $R = 0.79$), we get $N_2 = 570$, and to reduce N_2 we have to pay with rate. However, there are several ways to do so. For example, changing the type-1 gap to $\delta_1 = 5 \cdot 10^{-2}$ ($D_1 = 1$) while the type-2 gap stays $\delta_2 = 2.5 \cdot 10^{-4}$ (red circle labeled A in the plot) yields $R = 0.75$ and $N_2 = 26$, and changing the type-1 and type-2 gap to $\delta_1 = 2.5 \cdot 10^{-2}$ ($D_1 = 2$) and $\delta_2 = 4 \cdot 10^{-2}$*

($D_2 = 5$), respectively (blue circle labeled B), yields the same $R = 0.75$ but a smaller $N_2 = 11$.

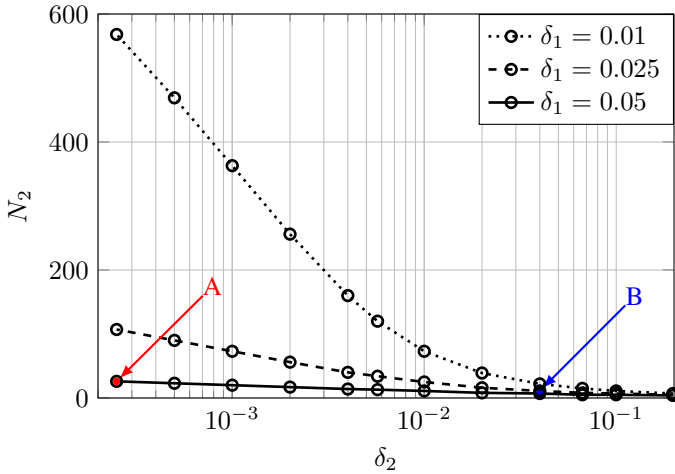


Fig. 5: Plot of N_2 as a function of the type-2 additive gap to capacity δ_2 for different values of the type-1 additive gap to capacity δ_1 . The simulated erasure rate is $\epsilon = 0.1998$ (99.9% of $\epsilon_2 = 0.2$).

VII. SUMMARY

This paper develops new tools for the construction and analysis of bilayer (and more generally multi-layer) LDPC code ensembles. In particular, we derived a code-analysis framework which resulted in a simple way to construct capacity-achieving sequences. Our design approach lends itself well for an extension to multi-layer code construction without complexity blow-up. We showed that using this design framework can give codes that enjoy low-complexity layer-1 decoding in low error rates, while still having small gaps to capacity for decoding layers 1+2 in high error rates. Since in some applications it is of interest that the decoding algorithm will avoid layer-2 messages as much as possible, we studied another trade-off regarding the number of layer-2 iterations and the gap to capacity.

Interesting future work includes combining the asymptotic design techniques with finite-block design techniques for the BEC and other channels. In addition, one can study generalizations of the decoding modes in the L -layer framework. Instead of considering L modes: layer 1, layers 1+2, ..., layers 1+2+...+ L , one can consider other subsets, for example layers 2+3+5+8. In this case, it would be interesting to use the design approach suggested in this paper to optimize the rate and decoding thresholds.

VIII. ACKNOWLEDGMENT

We thank the AE and anonymous referees of this and a previous version of this paper for valuable comments that improved the presentation considerably.

APPENDIX A PROOF OF LEMMA 2

- 1) Assume that $x = 0$. Since $\lambda^{(1)}(0) = 0$, (6b) implies that $y = g(\epsilon, 0, y) = 0$. Moreover, if $y = 0$ and $P_0 = 0$, then (6a) yields $x = f(\epsilon, x, 0) = 0$. Finally, if $y = 0$ and $\lambda^{(2)}(0) > 0$, then (6b) implies that $0 = \lambda^{(1)}(1 - \rho^{(1)}(1 - x))$, hence $\rho^{(1)}(1 - x) = 1$ and $x = 0$.
- 2) Follows immediately from (6a), (6b) and (8).
- 3) We prove (9) by a mathematical induction. For $l = 0$, (9) holds due to Item 2 and the fact that $x_0 = y_0 = \epsilon$. Assume correctness of (9) for some $l \geq 0$ and consider iteration $l + 1$. In view of Lemma 1, (7a)–(7c) and the induction assumption, it follows that

$$\begin{aligned} x_{l+1} &= f(\epsilon, x_l, y_l) \geq f(\epsilon, x, y) = x, \\ y_{l+1} &= g(\epsilon, x_l, y_l) \geq g(\epsilon, x, y) = y. \end{aligned} \quad (34)$$

This prove correctness of (9) for $l + 1$ and by mathematical induction proves (9) for all $l \geq 0$.

APPENDIX B PROOF OF LEMMA 5

Let $I = \min\{i: \Lambda_i^{(1)} > 0\}$ be the first non-zero coefficient of $\Lambda^{(1)}(\cdot)$, and let $C = \frac{d}{du} \Lambda^{(1)}(u)|_{u=1}$. Clearly, $I \geq 2$. Since $\lambda^{(1)}(u) = \frac{1}{C} \cdot \frac{d}{du} \Lambda^{(1)}(u)$, then

$$\begin{aligned} \lim_{u \rightarrow 0} \frac{\Lambda^{(1)}(u)}{\lambda^{(1)}(u)} &= \lim_{u \rightarrow 0} \frac{\Lambda^{(1)}(u)}{\frac{d}{du} \Lambda^{(1)}(u)} \cdot C \\ &= C \cdot \lim_{u \rightarrow 0} \frac{\sum_{i \geq I} \Lambda_i^{(1)} u^i}{\sum_{i \geq I} i \Lambda_i^{(1)} u^{i-1}} \\ &= C \cdot \lim_{u \rightarrow 0} \frac{u^I \sum_{i \geq I} \Lambda_i^{(1)} u^{i-I}}{u^{I-1} \sum_{i \geq I} i \Lambda_i^{(1)} u^{i-I}} \\ &= C \cdot \frac{\Lambda_I^{(1)}}{I \Lambda_I^{(1)}} \cdot \lim_{u \rightarrow 0} u \\ &= 0. \end{aligned} \quad (35)$$

Further, let $u(x) = 1 - \rho^{(1)}(1 - x)$ and note that $\lim_{x \rightarrow 0} u(x) = 0$. Thus,

$$\begin{aligned} \lim_{x \rightarrow 0} x \cdot \frac{\Lambda^{(1)}(1 - \rho^{(1)}(1 - x))}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))} &= \lim_{x \rightarrow 0} x \cdot \lim_{x \rightarrow 0} \frac{\Lambda^{(1)}(u(x))}{\lambda^{(1)}(u(x))} \\ &= \lim_{x \rightarrow 0} x \cdot \lim_{u \rightarrow 0} \frac{\Lambda^{(1)}(u)}{\lambda^{(1)}(u)} \\ &= 0. \end{aligned}$$

APPENDIX C PROOF OF THEOREM 6

Lemma 13. *If (x, y) is an (f, g) -fixed point with $y > 0$, then $x \leq q(y)$.*

Proof: Let $\epsilon \in (0, 1)$ and let (x, y) be a solution to (8) with $y > 0$. In view of (6a) and (6b), dividing the first equation

of (8) with the second one yields

$$\frac{x}{y} = \frac{\lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \cdot \Lambda^{(2)}(1 - \rho^{(2)}(1 - y))}{\lambda^{(2)}(1 - \rho^{(2)}(1 - y)) \cdot \Lambda^{(1)}(1 - \rho^{(1)}(1 - x))} \quad (36)$$

which after some rearrangements implies

$$q_2(y) = q_1(x), \quad (37)$$

where $q_2(\cdot)$ and $q_1(\cdot)$ are defined in (13). In view of (13), since (x, y) is an (f, g) -fixed point, then

$$\begin{aligned} q_2(y) &= y \cdot \frac{\Lambda^{(2)}(1 - \rho^{(2)}(1 - y))}{\lambda^{(2)}(1 - \rho^{(2)}(1 - y))} \\ &= g(\epsilon, x, y) \cdot \frac{\Lambda^{(2)}(1 - \rho^{(2)}(1 - y))}{\lambda^{(2)}(1 - \rho^{(2)}(1 - y))} \\ &= \underbrace{\leq 1}_{\epsilon} \cdot \overbrace{\Lambda^{(2)}(1 - \rho^{(2)}(1 - y))}^{\leq 1} \cdot \overbrace{\Lambda^{(1)}(1 - \rho^{(1)}(1 - x))}^{\leq 1} \\ &\leq 1, \end{aligned} \quad (38)$$

which together with Definition 2 and (37) completes the proof. \blacksquare

Let

$$1 > \epsilon > \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)}. \quad (39)$$

From the continuity of $\frac{y}{g(1, q(y), y)}$ (for $y \in (0, 1]$, $q_2(y) \leq 1$), and from the fact that for $y = 1$, $\frac{y}{g(1, q(y), y)} = 1$, there exists $y_0 \in (0, 1]$ such that $q_2(y_0) \leq 1$ and

$$\begin{aligned} y_0 &= \epsilon \cdot g(1, q(y_0), y_0) \\ &= \epsilon \cdot \lambda^{(2)}(1 - \rho^{(2)}(1 - y_0)) \cdot \Lambda^{(1)}(1 - \rho^{(1)}(1 - q(y_0))). \end{aligned} \quad (40)$$

In view of (14),

$$q_1(q(y_0)) = q_2(y_0), \quad (41)$$

which combined with (6a), (13) and (40) yields

$$\begin{aligned} q(y_0) &= \frac{\lambda^{(1)}(1 - \rho^{(1)}(1 - q(y_0)))}{\Lambda^{(1)}(1 - \rho^{(1)}(1 - q(y_0)))} \cdot y_0 \cdot \frac{\Lambda^{(2)}(1 - \rho^{(2)}(1 - y_0))}{\lambda^{(2)}(1 - \rho^{(2)}(1 - y_0))} \\ &= \epsilon \cdot \lambda^{(1)}(1 - \rho^{(1)}(1 - q(y_0))) \cdot \Lambda^{(2)}(1 - \rho^{(2)}(1 - y_0)) \\ &= f(\epsilon, q(y_0), y_0). \end{aligned} \quad (42)$$

Thus, $(q(y_0), y_0)$ is a non-zero (f, g) -fixed point, which in view of Theorem 4 implies that $\epsilon > \epsilon_2^*$. Hence,

$$\epsilon_2^* \leq \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)}. \quad (43)$$

Next, let

$$\epsilon < \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)} \quad (44)$$

or equivalently, for every $y \in (0, 1]$ such that $q_2(y) \leq 1$,

$$g(\epsilon, q(y), y) = \epsilon \cdot g(1, q(y), y) < y. \quad (45)$$

Let (x, y) be a solution to (8); in what follows, we prove that $y = 0$. Assume to the contrary that $y > 0$. From Lemma 13 it follows that $x \leq q(y)$, which in view Lemma 1, (38) and (45) implies

$$y = g(\epsilon, x, y) \leq g(\epsilon, q(y), y) < y, \quad (46)$$

in contradiction; thus, $y = 0$. Next, consider two cases:

- 1) If $P_0 = 0$ or $\lambda^{(2)}(0) > 0$, then Item 1 of Lemma 2 implies that $x = 0$. Hence, every (f, g) -fixed point satisfies $y = x = 0$. In view of Theorem 4, it follows that if (44) holds, then $\epsilon < \epsilon_2^*$, so

$$\epsilon_2^* \geq \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)} \quad (47)$$

which with (43) completes the proof when $P_0 = 0$ or $\lambda^{(2)}(0) > 0$.

- 2) If $P_0 > 0$ and $\lambda^{(2)}(0) = 0$, it is not true in general that for every fixed point (x, y) , $y = 0$ implies $x = 0$. However, if in addition to (44),

$$\epsilon < \frac{1}{P_0} \cdot \inf_{(0, 1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))}, \quad (48)$$

and $y = 0$ for some fixed point (x, y) , then $x = 0$. To see this, assume to the contrary that $x > 0$. In view of (6a) and (48) it follows that

$$x = f(\epsilon, x, 0) = \epsilon \cdot P_0 \cdot \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) < x \quad (49)$$

in contradiction; hence, if (44) and (48) hold, $x = 0$ thus $\epsilon < \epsilon_2^*$. This means that

$$\epsilon_2^* \geq \min \left\{ \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)}, \frac{1}{P_0} \cdot \inf_{(0, 1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))} \right\}. \quad (50)$$

To complete the proof, we must show that when $P_0 > 0$ and $\lambda^{(2)}(0) = 0$, then

$$\epsilon_2^* \leq \min \left\{ \inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)}, \frac{1}{P_0} \cdot \inf_{(0, 1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))} \right\}. \quad (51)$$

If

$$\inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)} \leq \frac{1}{P_0} \cdot \inf_{(0, 1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))},$$

then (51) follows immediately from (43); hence we can assume that

$$\inf_{\substack{y \in (0, 1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)} > \frac{1}{P_0} \cdot \inf_{(0, 1]} \frac{x}{\lambda^{(1)}(1 - \rho^{(1)}(1 - x))}. \quad (52)$$

Let

$$\inf_{\substack{y \in (0,1] \\ q_2(y) \leq 1}} \frac{y}{g(1, q(y), y)} > \epsilon > \frac{1}{P_0} \inf_{(0,1]} \frac{y}{\lambda^{(1)}(1 - \rho^{(1)}(1 - y))}, \quad (53)$$

and let $x_0 \in (0, 1]$, such that $x_0 = \epsilon \cdot P_0 \cdot \lambda^{(1)}(1 - \rho^{(1)}(1 - x_0))$. Since $\lambda^{(2)}(0) = 0$, it follows that $(x_0, 0)$ is a fixed point with $x_0 > 0$, thus $\epsilon > \epsilon_2^*$. Since this is true for every $\epsilon > \frac{1}{P_0} \inf_{(0,1]} \frac{y}{\lambda^{(1)}(1 - \rho^{(1)}(1 - y))}$, then $\epsilon_2^* \leq \frac{1}{P_0} \inf_{(0,1]} \frac{y}{\lambda^{(1)}(1 - \rho^{(1)}(1 - y))}$. In view of (52), it follows that (51) holds. This completes the proof for the $P_0 > 0$ and $\lambda^{(2)}(0) = 0$ case.

APPENDIX D PROOF OF LEMMA 9

The design rate of the L -layer ensemble is given by

$$R = 1 - \sum_{i=1}^L \frac{\int_0^1 \rho^{(i)}(x) dx}{\int_0^1 \lambda^{(i)}(x) dx} \left(1 - P_0^{(i)}\right),$$

and the threshold is ϵ_L . Thus, the overall gap to capacity is given by

$$\begin{aligned} \delta &= 1 - R - \epsilon_L \\ &= \sum_{i=1}^L \frac{\int_0^1 \rho^{(i)}(x) dx}{\int_0^1 \lambda^{(i)}(x) dx} \left(1 - P_0^{(i)}\right) - \epsilon_L \\ &= \epsilon_1 + \delta_1 + \sum_{i=2}^L \left(\epsilon_i a_s^{(i-1)} + \delta_i\right) \left(1 - P_0^{(i)}\right) - \epsilon_L \\ &= \delta_1 + \sum_{i=2}^L \delta_i \left(1 - P_0^{(i)}\right) + \epsilon_1 + \sum_{i=2}^L \epsilon_i a_s^{(i-1)} \left(1 - P_0^{(i)}\right) - \epsilon_L. \end{aligned} \quad (54)$$

In view of Construction 2, $P_0^{(i)} = \epsilon_{i-1}/\epsilon_i$ and $a_s^{(i)} \leq 1$. Hence,

$$\begin{aligned} &\epsilon_1 + \sum_{i=2}^L \left(\epsilon_i a_s^{(i-1)}\right) \left(1 - P_0^{(i)}\right) - \epsilon_L \\ &= \epsilon_1 + \sum_{i=2}^L a_s^{(i-1)} (\epsilon_i - \epsilon_{i-1}) - \epsilon_L \\ &= \epsilon_1 + \sum_{i=2}^L a_s^{(i-1)} (\epsilon_i - \epsilon_{i-1}) - \epsilon_L \\ &\quad - \epsilon_2 - \sum_{i=3}^{L-1} (\epsilon_i - \epsilon_{i-1}) + \epsilon_{L-1} \\ &= \sum_{i=2}^L \left(a_s^{(i-1)} - 1\right) (\epsilon_i - \epsilon_{i-1}) \\ &\leq 0, \end{aligned}$$

which combined with (54) completes the proof.

APPENDIX E PROOF OF LEMMA 10

To prove Lemma 10 we need the following lemma.

Lemma 14. *A scheduling scheme is valid if and only if, $\epsilon_{\text{eff}}^{(1)}(y_l) < \epsilon_1$, for some iteration l .*

Proof: Recall the definition of the type-1 threshold,

$$\epsilon_1 = \sup\{\epsilon: x = \epsilon \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \text{ has no solution in } (0, 1]\}, \quad (55)$$

and let $x = \lim_{l \rightarrow \infty} x_l$ and $y = \lim_{l \rightarrow \infty} y_l$. Since under every scheduling scheme y_l is monotonically non-increasing in l , then in view of (29),

$$\exists l \in \mathbb{N}, \epsilon_{\text{eff}}^{(1)}(y_l) < \epsilon_1$$

$$\Downarrow$$

$$\epsilon_{\text{eff}}^{(1)}(y) < \epsilon_1$$

$$x = \epsilon_{\text{eff}}^{(1)}(y) \lambda^{(1)}(1 - \rho^{(1)}(1 - x)) \text{ has no solution for } x \in (0, 1]$$

$$\Downarrow$$

$$\lim_{l \rightarrow \infty} x_l = 0.$$

■

We proceed with the proof of Lemma 10. Let (x_l, y_l) be defined as in (30), let $(x, y) = \lim_{l \rightarrow \infty} (x_l, y_l)$, and assume in contradiction that $\epsilon_{\text{eff}}^{(1)}(y) \geq \epsilon_1$. Since $\eta > 0$, letting $l \rightarrow \infty$ in (30) implies that (x, y) is a non-trivial (f, g) -fixed point. However, in view of Theorem 4, if $\epsilon < \epsilon_2$, then every (f, g) -fixed point is the trivial point, in contradiction. Thus, $\epsilon_{\text{eff}}^{(1)}(y) < \epsilon_1$ which, due to Lemma 14, completes the proof.

APPENDIX F PROOF OF LEMMA 11

Let $\left\{l_k^{[1]}\right\}_{k=1}^{N_2^{[1]}}$ and $\left\{l_k^{[2]}\right\}_{k=1}^{N_2^{[2]}}$ be the type-2-update iterations of the scheduling scheme described in (31d) and in some arbitrary valid scheduling scheme, respectively. We need to show that $N_2^{[1]} \leq N_2^{[2]}$. To proceed we need the following lemmas:

Lemma 15. *$x_s(\epsilon)$ as defined in Definition 3 is monotone non-decreasing in ϵ .*

Proof: Let $\epsilon_1 \leq \epsilon_2$, and consider $x_s(\epsilon_1)$, $x_s(\epsilon_2)$. In view of Definition 3,

$$\begin{aligned} h_{\epsilon_2}(x_s(\epsilon_1)) &\triangleq \epsilon_2 \lambda^{(1)}(1 - \rho^{(1)}(1 - x_s(\epsilon_1))) - x_s(\epsilon_1) \\ &\geq \epsilon_1 \lambda^{(1)}(1 - \rho^{(1)}(1 - x_s(\epsilon_1))) - x_s(\epsilon_1) \\ &\triangleq h_{\epsilon_1}(x_s(\epsilon_1)) \\ &\geq 0. \end{aligned}$$

Thus, $x_s(\epsilon_2) \triangleq \max\{x \in [0, 1] : h_{\epsilon_2}(x) \geq 0\} \geq x_s(\epsilon_1)$. ■

Lemma 16. *Let*

$$\begin{aligned} \epsilon_k^{[1]} &= \epsilon_{\text{eff}}^{(1)}\left(y_{l_k^{[1]}}\right), \quad 1 \leq k \leq N_2^{[1]}, \\ \epsilon_k^{[2]} &= \epsilon_{\text{eff}}^{(1)}\left(y_{l_k^{[2]}}\right), \quad 1 \leq k \leq N_2^{[2]}. \end{aligned} \quad (56)$$

Then, for every $1 \leq k \leq \min(N_2^{[1]}, N_2^{[2]})$,

$$y_{l_k^{[1]}} \leq y_{l_k^{[2]}}, \quad \text{and} \quad \varepsilon_k^{[1]} \leq \varepsilon_k^{[2]}, \quad \text{and} \quad x_{l_k^{[1]}} \leq x_{l_k^{[2]}}. \quad (57)$$

Proof: By induction on $1 \leq k \leq \min(N_2^{[1]}, N_2^{[2]})$. In the first type-2 update, we have $y_{l_1^{[1]}} = 1 = y_{l_1^{[2]}}$ and $\varepsilon_1 \triangleq \varepsilon_{\text{eff}}^{[1]}(y_{l_1^{[1]}}) = \varepsilon$. Thus, in view of Definition 3, in the first type-2 update $= x_{l_1^{[2]}} \geq x_s(\varepsilon) = x_{l_1^{[1]}}$. Hence (57) holds for $k = 1$. Assume correctness for some type-2 update $k < \min(N_2^{[1]}, N_2^{[2]})$, and consider update $k + 1$. In view of Lemma 1, (31b)-(31d), and the induction assumption, $y_{l_{k+1}^{[1]}} = g(\varepsilon, x_{l_k^{[1]}}, y_{l_k^{[1]}}) \leq g(\varepsilon, x_{l_k^{[2]}}, y_{l_k^{[2]}}) = y_{l_{k+1}^{[2]}}$, which together with (31c) implies that $\varepsilon_{k+1}^{[1]} \triangleq \varepsilon_{\text{eff}}^{[1]}(y_{l_{k+1}^{[1]}}) \leq \varepsilon_{\text{eff}}^{[1]}(y_{l_{k+1}^{[2]}}) \triangleq \varepsilon_{k+1}^{[2]}$. In view of Lemma 15, it follows that $x_{l_{k+1}^{[1]}} \triangleq x_s(\varepsilon_{k+1}^{[1]}) \leq x_s(\varepsilon_{k+1}^{[2]}) \leq x_{l_{k+1}^{[2]}}$. By induction, we complete the proof. ■

We proceed with the proof of Lemma 11. Assume, on the contrary, that $N_2^{[1]} > N_2^{[2]}$. Lemma 16 and the monotonicity of ε_k in k imply that

$$\varepsilon_{N_2^{[2]}}^{[2]} \geq \varepsilon_{N_2^{[2]}}^{[1]} \geq \varepsilon_{N_2^{[1]}-1}^{[1]} \geq \varepsilon_1, \quad (58)$$

which, in view of Lemma 14 yields that the scheduling scheme indexed by $\left\{l_k^{[2]}\right\}_{k=1}^{N_2^{[2]}}$ is not valid, in contradiction. Thus, $N_2^{[1]} \leq N_2^{[2]}$.

APPENDIX G PROOF OF LEMMA 12

In view of Definition 3, let $h_\varepsilon^{(k)}(x) = \varepsilon \lambda^{(k)}(1 - \rho^{(k)}(1 - x)) - x$. Eq. (33) yields

$$h_\varepsilon(x) = \lim_{k \rightarrow \infty} h_\varepsilon^{(k)}(x) = \begin{cases} \left(\frac{\varepsilon}{\varepsilon_1} - 1\right)x & 0 \leq x \leq \varepsilon_1 \\ \varepsilon - x & \varepsilon_1 \leq x \leq \varepsilon \end{cases} \quad (59)$$

For every $k \in \mathbb{N}$, and $x \in (\varepsilon, 1]$,

$$\begin{aligned} h_\varepsilon^{(k)}(x) &= \varepsilon \lambda^{(k)}(1 - \rho^{(k)}(1 - x)) - x \\ &\leq \varepsilon - x \\ &< 0, \end{aligned}$$

Thus

$$x_s^{(k)}(\varepsilon) \leq \varepsilon, \quad \forall k \in \mathbb{N}. \quad (60)$$

In addition, for every $0 < a < \varepsilon$ there exists K_0 such that

$$h_\varepsilon^{(k)}(\varepsilon - a) > 0, \quad \forall k \geq K_0,$$

so $x_s^{(k)}(\varepsilon) \geq \varepsilon - a$, for every $k \geq K_0$; hence,

$$\liminf_{k \rightarrow \infty} x_s^{(k)}(\varepsilon) \geq \varepsilon. \quad (61)$$

Combining (60) and (61) implies that $\lim_{k \rightarrow \infty} x_s^{(k)}(\varepsilon)$ exists, and completes the proof.

REFERENCES

- [1] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [2] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for "turbo-like" codes", *Proc. of the Allerton Conf. on Comm., Control, and Computing*, pp. 201–210, Monticello, IL, USA, Sept. 1998.
- [3] H. Jin, A. K. Khandekar, and R. J. McEliece, "Irregular repeat-accumulate codes", *Proc. of the Int. Conf. on Turbo Codes and Related Topics*, pp. 1–8, Brest, France, Sept. 2000.
- [4] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs", *Jet Propuls. Lab. Tech. Rep.*, Pasadena, CA, USA, Aug. 2003.
- [5] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via coupling: why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
- [6] T. Richardson and R. Urbanke, *Modern Coding Theory*, New York, NY, USA: Cambridge Univ. Press, 2008.
- [7] P. Razaghi and W. Yu, "Bilayer low-density parity-check codes for decode-and-forward in relay channels," *IEEE Trans. on Inf. Theory*, vol. 53, no. 10, pp. 3723–3739, Sep. 2007.
- [8] M. H. Azmi, Jinhong Yuan, Jun Ning and H. Q. Huynh, "Improved bilayer LDPC codes using irregular check node degree distribution," *Proc. 2008 IEEE Int Symp. on Inf. Theory*, pp. 141–145, Toronto, ON, CA, July 2008.
- [9] P. Razaghi and W. Yu, "Bilayer LDPC Codes for the Relay Channel," *Proc. 2006 IEEE International Conference on Comm.*, pp. 1574–1579, Istanbul, Turkey, June 2006.
- [10] M. J. Wainwright and E. Martinian, "Low-density graph codes that are optimal for binning with side information," *IEEE Trans. on Inf. Theory*, vol. 55, no. 3, pp. 1061–1079, March. 2009.
- [11] E. Ram and Y. Cassuto, "LDPC codes with local and global decoding," *Proc. 2018 IEEE Int. Symp. on Inf. Theory*, pp. 1151–1155, Vail, CO, USA, June 2018.
- [12] E. Sharon, I. Alrod, R. Zamir, O. Fainzilber, I. Ilani, A. Bazarzsky, I. Goldenberg, "Low cost and power LDPC for commodity NAND products", *The 11th Annual Non-Volatile Memories Workshop*, available online at <http://nvmw.ucsd.edu/nvmw2020-program/unzip/current2/nvmw2020-final29.pdf>.
- [13] J. Li, K. Liu, S. Lin, and K. Abdel-Ghaffar, "Reed-solomon based nonbinary globally coupled LDPC codes: correction of random errors and bursts of erasures," *Proc. 2017 IEEE Int. Symp. on Inf. Theory*, pp. 381–385, Aachen, Germany, June 2017.
- [14] H. Wang, S. V. S. Ranganathan and R. D. Wesel, "Approaching capacity using incremental redundancy without feedback," *Proc. 2017 IEEE Int. Symp. on Inf. Theory*, pp. 161–165, Aachen, Germany, June 2017.
- [15] H. Wang and R. D. Wesel, "Channel code analysis and design using multiple variable-length codes in parallel without feedback," *2018 IEEE Glob. Comm. Conference*, pp. 1–7, Abu Dhabi, United Arab Emirates, Dec. 2018.
- [16] H. Wang, S. V. S. Ranganathan and R. D. Wesel, "Variable-length coding with shared incremental redundancy: design methods and examples," *IEEE Trans. on Comm.*, vol. 67, no. 9, pp. 5981–5995, Sept. 2019.
- [17] O. Barak, U. Erez and D. Burshtein, "Bounds on rates of LDPC codes for BEC with varying erasure rate," *IEEE Trans. on Comm.*, vol. 58, no. 10, pp. 2835–2844, October 2010.
- [18] M. Hassner, K. Abdel-Ghaffar, A. Patel, R. Koetter, and B. Trager, "Integrated interleaving – a novel ECC architecture," *IEEE Trans. Magn.*, vol. 37, no. 3, pp. 773–775, Feb. 2001.
- [19] J. Han and L. A. Lastras-Montano, "Reliable memories with subline accesses," *Proc. 2007 IEEE Int. Symp. on Inf. Theory*, pp. 2531–2535, Nice, France, June 2007.

- [20] M. Blaum and S. Hetzler, "Integrated interleaved codes as locally recoverable codes: properties and performance," *Int. J. Inf. Coding Theory*, vol. 3, no. 4, pp. 324–344, Jan. 2016.
- [21] Y. Cassuto, E. Hemo, S. Puchinger and M. Bossert, "Multi-block interleaved codes for local and global read access," *Proc. 2017 IEEE Int. Symp. on Inf. Theory*, pp. 1758–1762, Aachen, Germany, June 2017.
- [22] J. Ha, J. Kim and S.W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. on Inf. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.
- [23] J. Ha, J. Kim, D. Klinc and S.W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes with short block lengths," *IEEE Trans. on Inf. Theory*, vol. 52, no. 2, pp. 728–738, Feb. 2006.
- [24] I. Andriyanova and E. Soljanin, "Optimized IR-HARQ schemes based on punctured LDPC codes over the BEC," *IEEE Trans. on Inf. Theory*, vol. 58, no. 10, pp. 6433–6445, Oct. 2012.
- [25] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.
- [26] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 8–15, Pacific Grove, CA, USA, 2002.
- [27] H. Xiao and A. H. Banihashemi, "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Trans. on Comm.*, vol. 52, no. 12, pp. 2098–2105, Dec. 2004.
- [28] A. I. V. Casado, M. Griot and R. D. Wesel, "LDPC decoders with informed dynamic scheduling," *IEEE Trans. on Comm.*, vol. 58, no. 12, pp. 3470–3479, Dec. 2010.
- [29] M. A. Shokrollahi, "Capacity-achieving sequences," *Codes, Systems, and Graphical Models (The IMA Vols. in Math. and its App.)*, B. Marcus, J. Rosenthal (eds), New-York: Springer, vol. 123, pp. 153–166, 2001.

Eshed Ram received the B.Sc., summa cum laude, the M.Sc and Ph.D. degrees in Electrical Engineering from the Technion in 2015, 2017 and 2021, respectively. In the years between 2012-2015 he worked at the IBM Research Labs in Haifa. He is currently with Nvidia, Israel, where he works on algorithmic research and development for video compression. His research interests are information theory, coding theory, signal processing, and quantum information processing.

Yuval Cassuto (S'02–M'08–SM'14) is a faculty member at the Viterbi Department of Electrical Engineering, Technion – Israel Institute of Technology. His research interests lie at the intersection of the theoretical information sciences and the engineering of practical computing and storage systems. During 2010-2011 he has been a Scientist at EPFL, the Swiss Federal Institute of Technology in Lausanne. From 2008 to 2010 he was a Research Staff Member at Hitachi Global Storage Technologies, San Jose Research Center. He received the B.Sc degree in Electrical Engineering, summa cum laude, from the Technion in 2001, and the M.S. and Ph.D. degrees in Electrical Engineering from the California Institute of Technology, in 2004 and 2008, respectively. From 2000 to 2002, he was with Qualcomm, Israel R&D Center, where he worked on modeling, design and analysis in wireless communications. Dr. Cassuto has won the Best Student Paper Award in data storage from the IEEE Communications Society in 2010 as a student, and in 2019 as an adviser. Before that, he won the 2001 Texas Instruments DSP and Analog Challenge \$100,000 prize.