How to Find Good Finite-Length Codes: From Art Towards Science

Abdelaziz Amraoui[†] , Andrea Montanari[*] and Ruediger Urbanke[‡]

***Abstract*— We explain how to optimize finite-length LDPC codes for transmission over the binary erasure channel. Our approach relies on an analytic approximation of the erasure probability. This is in turn based on a finite-length scaling result to model large scale erasures and a union bound involving minimal stopping sets to take into account small error events. We show that the performances of optimized ensembles as observed in simulations are well described by our approximation. Although we only address the case of transmission over the binary erasure channel, our method should be applicable to a more general setting.**

I. Introduction

In this paper, we consider transmission using random elements from the standard ensemble of low-density parity-check (LDPC) codes defined by the degree distribution pair (λ, ρ). For an introduction to LDPC codes and the standard notation see [1]. In [2], one of the authors (AM) suggested that the probability of error of iterative coding systems follows a scaling law. In [3]–[5], it was shown that this is indeed true for LDPC codes, assuming that transmission takes place over the BEC. Strictly speaking, scaling laws describe the asymptotic behavior of the error probability close to the threshold for increasing blocklengths. However, as observed empirically in the papers mentioned above, scaling laws provide good approximations to the error probability also away from the threshold and already for modest blocklengths. This is the starting point for our finite-length optimization.

In [3], [5] the *form* of the scaling law for transmission over the BEC was derived and it was shown how to compute the *scaling parameters* by solving a system of ordinary differential equations. This system was called *covariance evolution* in analogy to density evolution. Density evolution concerns the evolution of the *average* number of erasures still contained in the graph during the decoding process, whereas covariance evolution concerns the evolution of its *variance*. Whereas Luby et al. [6] found an explicit solution to the density evolution equations, to date no such solution is known for the system of covariance equations. Covariance evolution must therefore be integrated numerically. Unfortunately the dimension of the ODE's system ranges from hundreds to thousand for typical examples. As a consequence, numerical integration can be quite time consuming. This is a serious problem if we want to use scaling laws in the context of optimization, where the computation of scaling parameters must be repeated for many different ensembles during the optimization process.

In this paper, we make two main contributions. First, we derive explicit analytic expressions for the scaling parameters as a function of the degree distribution pair and quantities which appear in density evolution. Second, we provide an accurate approximation to the erasure probability stemming from small stopping sets and resulting in the erasure floor.

The paper is organized as follows. Section II describes our approximation for the error probability, the scaling law being discussed in Section II-B and the error floor in Section II-C. We combine these results and give in Section II-D an approximation to the erasure probability curve, denoted by $\mathrm{P}(n, \lambda, \rho, \epsilon)$, that can be computed efficiently for any blocklength, degree distribution pair, and channel parameter. The basic ideas behind the explicit determination of the scaling parameters (together with the resulting expressions) are collected in Section III. Finally, the most technical (and tricky) part of this computation is deferred to Section IV.

As a motivation for some of the rather technical points to come, we start in Section I-A by showing how $\mathrm{P}(n, \lambda, \rho, \epsilon)$ can be used to perform an efficient finite-length optimization.

A. Optimization

The optimization procedure takes as input a blocklength n, the BEC erasure probability ϵ, and a target probability of erasure, call it $\mathrm{P}_{\text{target}}$. Both bit or block probability can be considered. We want to find a degree distribution pair (λ, ρ) of maximum rate so that $\mathrm{P}(n, \lambda, \rho, \epsilon) \leq \mathrm{P}_{\text{target}}$, where $\mathrm{P}(n, \lambda, \rho, \epsilon)$ is the approximation discussed in the introduction.

Let us describe an efficient procedure to accomplish this optimization locally (however, many equivalent approaches are possible). Although providing a global optimization scheme goes beyond the scope of this paper, the local procedure was found empirically to converge often to the global optimum.

It is well known [1] that the *design rate* $r(\lambda, \rho)$ associated to a degree distribution pair (λ, ρ) is equal to

$$r(\lambda, \rho) = 1 - \frac{\sum_i \frac{\rho_i}{i}}{\sum_j \frac{\lambda_j}{j}}. \tag{1}$$

This invited paper is an enhanced version of the work presented at the 4^{th} International Symposium on Turbo Codes and Related Topics, Münich, Germany, 2006.

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

AM has been partially supported by the EU under the integrated project EVERGROW.

[†] A. Amraoui is with EPFL, School of Computer and Communication Sciences, Lausanne CH-1015, Switzerland, abdelaziz.amraoui@epfl.ch

[*] A. Montanari is with Ecole Normale Supérieure, Laboratoire de Physique Théorique, 75231 Paris Cedex 05, France, montanar@lpt.ens.fr

[‡] R. Urbanke is with EPFL, School of Computer and Communication Sciences, Lausanne CH-1015, Switzerland, ruediger.urbanke@epfl.ch

For "most" ensembles the actual rate of a randomly chosen element of the ensemble LDPC(n, λ, ρ) is close to this design rate [7]. In any case, $r(\lambda, \rho)$ is *always* a lower bound. Assume we change the degree distribution pair slightly by $\Delta\lambda(x) = \sum_i \Delta\lambda_i x^{i-1}$ and $\Delta\rho(x) = \sum_i \Delta\rho_i x^{i-1}$, where $\Delta\lambda(1) = 0 = \Delta\rho(1)$ and assume that the change is sufficiently small so that $\lambda + \Delta\lambda$ as well as $\rho + \Delta\rho$ are still valid degree distributions (non-negative coefficients). A quick calculation then shows that the design rate changes by

$$\begin{aligned} &r(\lambda + \Delta\lambda, \rho + \Delta\rho) - r(\lambda, \rho) \\ &\simeq \sum_i \Delta\lambda_i \frac{(1-r)}{i \sum_j \frac{\lambda_j}{j}} - \sum_i \frac{\Delta\rho_i}{i \sum_j \frac{\lambda_j}{j}}. \end{aligned} \tag{2}$$

In the same way, the erasure probability changes (according to the approximation) by

$$\begin{aligned} &\mathrm{P}(n, \lambda + \Delta\lambda, \rho + \Delta\rho, \epsilon) - \mathrm{P}(n, \lambda, \rho, \epsilon) \\ &\simeq \sum_i \Delta\lambda_i \frac{\partial \mathrm{P}}{\partial \lambda_i} + \sum_i \Delta\rho_i \frac{\partial \mathrm{P}}{\partial \rho_i}. \end{aligned} \tag{3}$$

Equations (2) and (3) give rise to a simple linear program to optimize locally the degree distribution: Start with some initial degree distribution pair (λ, ρ). If $\mathrm{P}(n, \lambda, \rho, \epsilon) \leq \mathrm{P}_{\text{target}}$, then increase the rate by a repeated application of the following linear program.

LP *1:* [Linear program to increase the rate]

$$\begin{aligned} \max\{(1-r)\sum_i \Delta\lambda_i/i - \sum_i \Delta\rho_i/i \mid \\ \sum_i \Delta\lambda_i = 0; \qquad -\min\{\delta, \lambda_i\} \leq \Delta\lambda_i \leq \delta; \\ \sum_i \Delta\rho_i = 0; \qquad -\min\{\delta, \rho_i\} \leq \Delta\rho_i \leq \delta; \\ \sum_i \frac{\partial \mathrm{P}}{\partial \lambda_i}\Delta\lambda_i + \sum_i \frac{\partial \mathrm{P}}{\partial \rho_i}\Delta\rho_i \leq \mathrm{P}_{\text{target}} - \mathrm{P}(n, \lambda, \rho, \epsilon)\}. \end{aligned}$$

Hereby, δ is a sufficiently small non-negative number to ensure that the degree distribution pair changes only slightly at each step so that changes of the rate and of the probability of erasure are accurately described by the linear approximation. The value δ is best adapted dynamically to ensure convergence. One can start with a large value and decrease it the closer we get to the final answer. The objective function in LP 1 is equal to the total derivative of the rate as a function of the change of the degree distribution. Several rounds of this linear program will gradually improve the rate of the code ensemble, while keeping the erasure probability below the target (last inequality).

Sometimes it is necessary to initialize the optimization procedure with degree distribution pairs that do not fulfill the target erasure probability constraint. This is for instance the case if the optimization is repeated for a large number of "randomly" chosen initial conditions. In this way, we can check whether the procedure always converges to the same point (thus suggesting that a global optimum was found), or otherwise, pick the best outcome of many trials. To this end we define a linear program that decreases the erasure probability.

LP *2:* [Linear program to decrease $\mathrm{P}(n, \lambda, \rho, \epsilon)$]

$$\begin{aligned} \min\{\sum \frac{\partial \mathrm{P}}{\partial \lambda_i}\Delta\lambda_i + \sum \frac{\partial \mathrm{P}}{\partial \rho_i}\Delta\rho_i \mid \\ \sum_i \Delta\lambda_i = 0; \qquad -\min\{\delta, \lambda_i\} \leq \Delta\lambda_i \leq \delta; \\ \sum_i \Delta\rho_i = 0; \qquad -\min\{\delta, \rho_i\} \leq \Delta\rho_i \leq \delta\}. \end{aligned}$$

Example 1: [Sample Optimization] Let us show a sample optimization. Assume we transmit over a BEC with channel erasure probability $\epsilon = 0.5$. We are interested in a block length of $n = 5000$ bits and the maximum variable and check degree we allow are $\mathtt{l}_{\max} = 13$ and $\mathtt{r}_{\max} = 10$, respectively. We constrain the *block* erasure probability to be smaller than $\mathrm{P}_{\text{target}} = 10^{-4}$. We further count only erasures larger or equal to $s_{\min} = 6$ bits. This corresponds to looking at an *expurgated* ensemble, i.e., we are looking at the subset of codes of the ensemble that do not contain stopping sets of sizes smaller than 6. Alternatively, we can interpret this constraint in the sense that we use an outer code which "cleans up" the remaining small erasures. Using the techniques discussed in Section II-C, we can compute the probability that a randomly chosen element of an ensemble does not contain stopping sets of size smaller than 6. If this probability is not too small then we have a good chance of finding such a code in the ensemble by sampling a sufficient number of random elements. This can be checked at the end of the optimization procedure.

We start with an arbitrary degree distribution pair:

$$\begin{aligned} \lambda(x) =& 0.139976x + 0.149265x^2 + 0.174615x^3 \\ &+ 0.110137x^4 + 0.0184844x^5 + 0.0775212x^6 \\ &+ 0.0166585x^7 + 0.00832646x^8 + 0.0760256x^9 \\ &+ 0.0838369x^{10} + 0.0833654x^{11} + 0.0617885x^{12}, \end{aligned} \tag{4}$$

$$\begin{aligned} \rho(x) =& 0.0532687x + 0.0749403x^2 + 0.11504x^3 \\ &+ 0.0511266x^4 + 0.170892x^5 + 0.17678x^6 \\ &+ 0.0444454x^7 + 0.152618x^8 + 0.160889x^9. \end{aligned} \tag{5}$$

This pair was generated randomly by choosing each coefficient uniformly in $[0, 1]$ and then normalizing so that $\lambda(1) = \rho(1) = 1$. The approximation of the block erasure probability curve of this code (as given in Section II-D) is shown in Fig. 1. For this

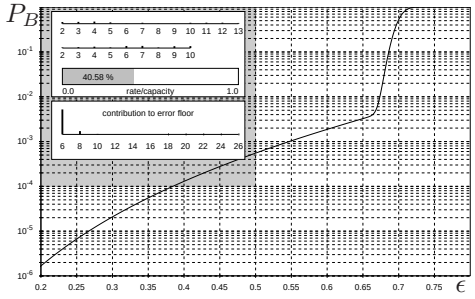

Fig. 1: Approximation of the block erasure probability for the initial ensemble with degree distribution pair given in (4) and (5).

initial degree distribution pair we have $r(\lambda, \rho) = 0.2029$ and $\mathrm{P}_B(n = 5000, \lambda, \rho, \epsilon = 0.5) = 0.000552 > \mathrm{P}_{\text{target}}$. Therefore,

we start by reducing $\mathrm{P}_B(n = 5000, \lambda, \rho, \epsilon = 0.5)$ (over the choice of λ and ρ) using LP 2 until it becomes lower than $\mathrm{P}_{\text{target}}$. After a number of LP 2 rounds we obtain the degree

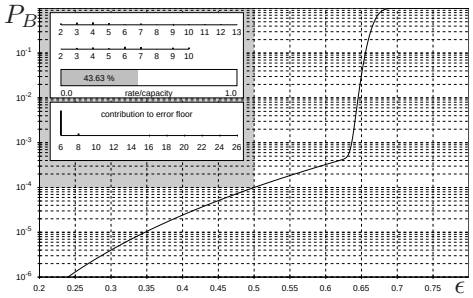

Fig. 2: Approximation of the block erasure probability for the ensemble obtained after the first part of the optimization (see (6) and (7)). The erasure probability has been lowered below the target.

distribution pair:

$$\begin{aligned}\lambda(x) =& 0.111913x + 0.178291x^2 + 0.203641x^3 \\ &+ 0.139163x^4 + 0.0475105x^5 + 0.106547x^6 \\ &+ 0.0240221x^7 + 0.0469994x^9 + 0.0548108x^{10} \\ &+ 0.0543393x^{11} + 0.0327624x^{12},\end{aligned} \tag{6}$$

$$\begin{aligned}\rho(x) =& 0.0242426x + 0.101914x^2 + 0.142014x^3 \\ &+ 0.0781005x^4 + 0.198892x^5 + 0.177806x^6 \\ &+ 0.0174716x^7 + 0.125644x^8 + 0.133916x^9.\end{aligned} \tag{7}$$

For this degree distribution pair we have $\mathrm{P}_B(n = 5000, \lambda, \rho, \epsilon = 0.5) = 0.0000997 \leq \mathrm{P}_{\text{target}}$ and $r(\lambda, \rho) = 0.218$. We show the corresponding approximation in Fig. 2.

Now, we start the second phase of the optimization and optimize the rate while insuring that the block erasure probability remains below the target, using LP 1. The resulting degree distribution pair is:

$$\lambda(x) = 0.0739196x + 0.657891x^2 + 0.268189x^{12}, \tag{8}$$

$$\rho(x) = 0.390753x^4 + 0.361589x^5 + 0.247658x^9, \tag{9}$$

where $r(\lambda, \rho) = 0.41065$. The block erasure probability plot for the result of the optimization is shown in Fig 3.

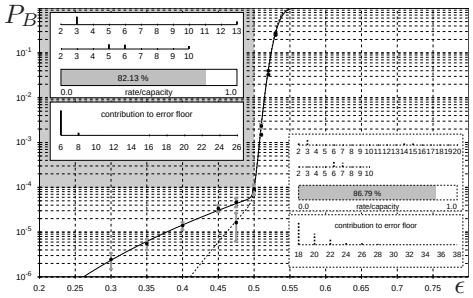

Fig. 3: Error probability curve for the result of the optimization (see (8) and (9)). The solid curve is $\mathrm{P}_B(n = 5000, \lambda, \rho, \epsilon = 0.5)$ while the small dots correspond to simulation points. In dotted are the results with a more aggressive expurgation.

Each LP step takes on the order of seconds on a standard PC. In total, the optimization for a given set of parameters $(n, \epsilon, \mathrm{P}_{\text{target}}, \mathtt{l}_{\max}, \mathtt{r}_{\max}, s_{\min})$ takes on the order of minutes.

Recall that the whole optimization procedure was based on $\mathrm{P}_B(n, \lambda, \rho, \epsilon)$ which is only an *approximation* of the *true* block erasure probability. In principle, the actual performances of the optimized ensemble could be worse (or better) than predicted by $\mathrm{P}_B(n, \lambda, \rho, \epsilon)$. To validate the procedure we computed the block erasure probability for the optimized degree distribution also by means of simulations and compare the two. The simulation results are shown in Fig 3 (dots with 95% confidence intervals) : analytical (approximate) and numerical results are in almost perfect agreement!

How hard is it to find a code without stopping sets of size smaller than 6 within the ensemble LDPC$(5000, \lambda, \rho)$ with (λ, ρ) given by Eqs. (8) and (9)? As discussed in more detail in Section II-C, in the limit of large blocklengths the number of small stopping sets has a joint Poisson distribution. As a consequence, if $\tilde{A}_i$ denotes the expected number of minimal stopping sets of size i in a random element from LDPC$(5000, \lambda, \rho)$, the probability that it contains *no* stopping set of size smaller than 6 is approximately $\exp\{-\sum_{i=1}^{5} \tilde{A}_i\}$. For the optimized ensemble we get $\exp\{-(0.2073+0.04688+0.01676+0.007874+0.0043335)\} \approx 0.753$, a quite large probability. We repeated the optimization procedure with various different random initial conditions and always ended up with essentially the same degree distribution. Therefore, we can be quite confident that the result of our local optimization is close to the global optimal degree distribution pair for the given constraints $(n, \epsilon, \mathrm{P}_{\text{target}}, \mathtt{l}_{\max}, \mathtt{r}_{\max}, s_{\min})$.

There are many ways of improving the result. E.g., if we allow higher degrees or apply a more aggressive expurgation, we can obtain degree distribution pairs with higher rate. E.g., for the choice $\mathtt{l}_{\max} = 15$ and $s_{\min} = 18$ the resulting degree distribution pair is

$$\begin{aligned}\lambda(x) =& 0.205031x + 0.455716x^2, \\ &+ 0.193248x^{13} + 0.146004x^{14}\end{aligned} \tag{10}$$

$$\rho(x) = 0.608291x^5 + 0.391709x^6, \tag{11}$$

where $r(\lambda, \rho) = 0.433942$. The corresponding curve is depicted in Fig 3, as a dotted line. However, this time the probability that a random element from LDPC$(5000, \lambda, \rho)$ has *no* stopping set of size smaller than 18 is approximately 6.10^{-6}. It will therefore be harder to find a code that fulfills the expurgation requirement.

It is worth stressing that our results could be improved further by applying the same approach to more powerful ensembles, e.g., multi-edge type ensembles, or ensembles defined by protographs. The steps to be accomplished are: (i) derive the scaling laws and define scaling parameters for such ensembles; (ii) find efficiently computable expressions for the scaling parameters; (iii) optimize the ensemble with respect to its defining parameters (e.g. the degree distribution) as above. Each of these steps is a manageable task – albeit not a trivial one.

Another generalization of our approach which is slated for future work is the extension to general binary memoryless

symmetric channels. Empirical evidence suggests that scaling laws should also hold in this case, see [2], [3]. How to prove this fact or how to compute the required parameters, however, is an open issue.

In the rest of this paper, we describe in detail the approximation $\mathrm{P}(n, \lambda, \rho, \epsilon)$ for the BEC.

II. Approximation $\mathrm{P}_B(n, \lambda, \rho, \epsilon)$ and $\mathrm{P}_b(n, \lambda, \rho, \epsilon)$

In order to derive approximations for the erasure probability we separate the contributions to this erasure probability into two parts – the contributions due to large erasure events and the ones due to small erasure events. The large erasure events give rise to the so-called *waterfall* curve, whereas the small erasure events are responsible for the *erasure floor*.

In Section II-B, we recall that the water fall curve follows a scaling law and we discuss how to compute the scaling parameters. We denote this approximation of the water fall curve by $\mathrm{P}^W_{B/b}(n, \lambda, \rho, \epsilon)$. We next show in Section II-C how to approximate the erasure floor. We call this approximation $\mathrm{P}^E_{B/b,s_{\min}}(n, \lambda, \rho, \epsilon)$. Hereby, $s_{\min}$ denotes the *expurgation* parameter, i.e, we only count error events involving at least $s_{\min}$ erasures. Finally, we collect in Section II-D our results and give an approximation to the total erasure probability. We start in Section II-A with a short review of density evolution.

A. Density Evolution

The initial analysis of the performance of LDPC codes assuming that transmission takes place of the BEC is due to Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann, see [6], and it is based on the so-called *peeling* algorithm. In this algorithm we "peel-off" one variable node at a time (and all its adjacent check nodes and edges) creating a sequence of residual graphs. Decoding is successful if and only if the final residual graph is the empty graph. A variable node can be peeled off if it is connected to at least one check node which has residual degree one. Initially, we start with the complete Tanner graph representing the code and in the first step we delete all variable nodes from the graph which have been received (have not been erased), all connected check nodes, and all connected edges.

From the description of the algorithm it should be clear that the number of degree-one check nodes plays a crucial role. The algorithm stops if and only if no degree-one check node remains in the residual graph. Luby et al. were able to give analytic expressions for the expected number of degree-one check nodes as a function of the size of the residual graph in the limit of large blocklengths. They further showed that most instances of the graph and the channel follow closely this ensemble average. More precisely, let r_1 denote the fraction of degree-one check nodes in the decoder. (This means that the actual number of degree-one check nodes is equal to $n(1-r)r_1$, where n is the blocklength and r is the design rate of the code.) Then, as shown in [6], r_1 is given parametrically by

$$r_1(y) = \epsilon\lambda(y)[y - 1 + \rho(1 - \epsilon\lambda(y))]. \tag{12}$$

where y is determined so that $\epsilon L(y)$ is the fractional (with respect to n) size of the residual graph. Hereby, $L(x) = \sum_i L_i x^i = \frac{\int_0^x \lambda(u)du}{\int_0^1 \lambda(u)du}$ is the node perspective variable node distribution, i.e. L_i is the fraction of variable nodes of degree i in the Tanner graph. Analogously, we let R_i denote the fraction of degree i check nodes, and set $R(x) = \sum_i R_i x^i$. With an abuse of notation we shall sometimes denote the irregular LDPC ensemble as $\mathrm{LDPC}(n, L, R)$.

The threshold noise parameter $\epsilon^* = \epsilon^*(\lambda, \rho)$ is the supremum value of ϵ such that $r_1(y) > 0$ for all $y \in (0, 1]$, (and therefore iterative decoding is successful with high probability). In Fig. 4, we show the function $r_1(y)$ depicted for the ensemble with $\lambda(x) = x^2$ and $\rho(x) = x^5$ for $\epsilon = \epsilon^*$. As

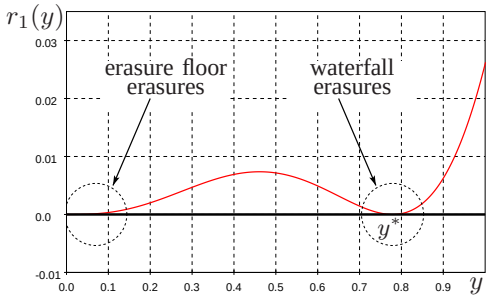

Fig. 4: $r_1(y)$ for $y \in [0, 1]$ at the threshold. The degree distribution pair is $\lambda(x) = x^2$ and $\rho(y) = x^5$ and the threshold is $\epsilon^* = 0.4294381$.

the fraction of degree-one check nodes concentrates around $r_1(y)$, the decoder will fail with high probability only in two possible ways. The first relates to $y \approx 0$ and corresponds to small erasure events. The second one corresponds to the value y^* such that $r_1(y^*) = 0$. In this case the fraction of variable nodes that can not be decoded concentrates around $\nu^* = \epsilon^* L(y^*)$.

We call a point y^* where the function $y - 1 + \rho(1 - \epsilon\lambda(y))$ and its derivative both vanish a *critical point*. At threshold, i.e. for $\epsilon = \epsilon^*$, there is at least one critical point, but there may be more than one. (Notice that the function $r_1(y)$ always vanishes together with its derivative at $y = 0$, cf. Fig. 4. However, this does not imply that $y = 0$ is a critical point because of the extra factor $\lambda(y)$ in the definition of $r_1(y)$.) Note that if an ensemble has a single critical point and this point is strictly positive, then the number of remaining erasures conditioned on decoding failure, concentrates around $\nu^* \triangleq \epsilon^* L(y^*)$.

In the rest of this paper, we will consider ensembles with a single critical point and separate the two above contributions. We will consider in Section II-B erasures of size at least $n\gamma\nu^*$ with $\gamma \in (0, 1)$. In Section II-C we will instead focus on erasures of size smaller than $n\gamma\nu^*$. We will finally combine the two results in Section II-D

B. Waterfall Region

It was proved in [3], that the erasure probability due to large failures obeys a well defined scaling law. For our purpose it is best to consider a refined scaling law which was conjectured in the same paper. For convenience of the reader we restate it here.

Conjecture 1: [Refined Scaling Law] Consider transmission over a BEC of erasure probability ϵ using random elements from the ensemble $\text{LDPC}(n,\lambda,\rho)=\text{LDPC}(n,L,R)$. Assume that the ensemble has a single critical point $y^*>0$ and let $\nu^*=\epsilon^* L(y^*)$, where ϵ^* is the threshold erasure probability. Let $\text{P}_{\text{b}}^W(n,\lambda,\rho,\epsilon)$ (respectively, $\text{P}_{\text{B}}^W(n,\lambda,\rho,\epsilon)$) denote the expected bit (block) erasure probability *due to erasures of size at least* $n\gamma\nu^*$, where $\gamma\in(0,1)$. Fix $z:=\sqrt{n}(\epsilon^*-\beta n^{-\frac{2}{3}}-\epsilon)$. Then as n tends to infinity,

$$\text{P}_{\text{B}}^W(n,\lambda,\rho,\epsilon)=Q\left(\frac{z}{\alpha}\right)\left(1+O(n^{-1/3})\right),$$
$$\text{P}_{\text{b}}^W(n,\lambda,\rho,\epsilon)=\nu^* Q\left(\frac{z}{\alpha}\right)\left(1+O(n^{-1/3})\right),$$

where $\alpha=\alpha(\lambda,\rho)$ and $\beta=\beta(\lambda,\rho)$ are constants which depend on the ensemble.

In [3], [5], a procedure called covariance evolution was defined to compute the scaling parameter α through the solution of a system of ordinary differential equations. The number of equations in the system is equal to the square of the number of variable node degrees plus the largest check node degree minus one. As an example, for an ensemble with 5 different variable node degrees and $\mathtt{r}_{\max}=30$, the number of coupled equations in covariance evolution is $(5+29)^2=1156$. The computation of the scaling parameter can therefore become a challenging task. The main result in this paper is to show that it is possible to compute the scaling parameter α without explicitly solving covariance evolution. This is the crucial ingredient allowing for efficient code optimization.

Lemma 1: [Expression for α] Consider transmission over a BEC with erasure probability ϵ using random elements from the ensemble $\text{LDPC}(n,\lambda,\rho)=\text{LDPC}(n,L,R)$. Assume that the ensemble has a single critical point $y^*>0$, and let ϵ^* denote the threshold erasure probability. Then the scaling parameter α in Conjecture 1 is given by

$$\alpha=\left(\frac{\rho(\bar{x}^*)^2-\rho(\bar{x}^{*2})+\rho'(\bar{x}^*)(1-2x^*\rho(\bar{x}^*))-\bar{x}^{*2}\rho'(\bar{x}^{*2})}{L'(1)\lambda(y^*)^2\rho'(\bar{x}^*)^2}\right.$$
$$\left.+\frac{\epsilon^{*2}\lambda(y^*)^2-\epsilon^{*2}\lambda(y^{*2})-y^{*2}\epsilon^{*2}\lambda'(y^{*2})}{L'(1)\lambda(y^*)^2}\right)^{1/2},$$

where $x^*=\epsilon^*\lambda(y^*)$, $\bar{x}^*=1-x^*$.

The derivation of this expression is explained in Section III

For completeness and the convenience of the reader, we repeat here also an explicit characterization of the shift parameter β which appeared already (in a slightly different form) in [3], [5].

Conjecture 2: [Scaling Parameter β] Consider transmission over a BEC of erasure probability ϵ using random elements from the ensemble $\text{LDPC}(n,\lambda,\rho)=\text{LDPC}(n,L,R)$. Assume that the ensemble has a single critical point $y^*>0$, and let ϵ^* denote the threshold erasure probability. Then the scaling parameter β in Conjecture 1 is given by

$$\beta/\Omega=\left(\frac{\epsilon^{*4}r_2^{*2}(\epsilon^*\lambda'(y^*)^2r_2^*-x^*(\lambda''(y^*)r_2^*+\lambda'(y^*)x^*))^2}{L'(1)^2\rho'(\bar{x}^*)^3x^{*10}(2\epsilon^*\lambda'(y^*)^2r_3^*-\lambda''(y^*)r_2^*x^*)}\right)^{1/3} \quad (13)$$

where $x^*=\epsilon^*\lambda(y^*)$ and $\bar{x}^*=1-x^*$, and for $i\geq 2$

$$r_i^*=\sum_{m\geq j\geq i}(-1)^{i+j}\binom{j-1}{i-1}\binom{m-1}{j-1}\rho_m(\epsilon^*\lambda(y^*))^j.$$

Further, Ω is a universal (code independent) constant defined in Ref. [3], [5].

We also recall that Ω is numerically quite close to 1. In the rest of this paper, we shall always adopt the approximate Ω by 1.

C. Error Floor

Lemma 2: [Error Floor] Consider transmission over a BEC of erasure probability ϵ using random elements from an ensemble $\text{LDPC}(n,\lambda,\rho)=\text{LDPC}(n,L,R)$. Assume that the ensemble has a single critical point $y^*>0$. Let $\nu^*=\epsilon^*L(y^*)$, where ϵ^* is the threshold erasure probability. Let $\text{P}_{\text{b},s_{\min}}^E(n,\lambda,\rho,\epsilon)$ (respectively $\text{P}_{\text{B},s_{\min}}^E(n,\lambda,\rho,\epsilon)$) denote the expected bit (block) erasure probability *due to stopping sets of size between* s_{min} *and* $n\gamma\nu^*$, where $\gamma\in(0,1)$. Then, for any $\epsilon<\epsilon^*$,

$$\text{P}_{\text{b},s_{\min}}^E(n,\lambda,\rho,\epsilon)=\sum_{s\geq s_{\min}}s\tilde{A}_s\epsilon^s\left(1+o(1)\right), \quad (14)$$

$$\text{P}_{\text{B},s_{\min}}^E(n,\lambda,\rho,\epsilon)=1-e^{-\sum_{s\geq s_{\min}}\tilde{A}_s\epsilon^s}\left(1+o(1)\right), \quad (15)$$

where $\tilde{A}_s=\text{coef}\left\{\log\left(A(x)\right),x^s\right\}$ for $s\geq 1$, with $A(x)=\sum_{s\geq 0}A_sx^s$ and

$$A_s=\sum_e\left(\text{coef}\left\{\prod_i(1+xy^i)^{nL_i},x^sy^e\right\}\times\right. \quad (16)$$
$$\left.\frac{\text{coef}\left\{\prod_i((1+x)^i-ix)^{n(1-r)R_i},x^e\right\}}{\binom{nL'(1)}{e}}\right).$$

Discussion: In the lemma we only claim a multiplicative error term of the form $o(1)$ since this is easy to prove. This weak statement would remain valid if we replaced the expression for A_s given in (16) with the explicit and much easier to compute asymptotic expression derived in [1]. In practice however the approximation is much better than the stated $o(1)$ error term if we use the finite-length averages given by (16). The hurdle in proving stronger error terms is due to the fact that for a given length it is not clear how to relate the number of stopping sets to the number of *minimal* stopping sets. However, this relationship becomes easy in the limit of large blocklengths.

Proof: The key in deriving this erasure floor expression is in focusing on the number of *minimal* stopping sets. These are stopping set that are not the union of smaller stopping sets. The *asymptotic* distribution of the number of *minimal* stopping sets contained in an LDPC graph was already studied in [1]. We recall that the distribution of the number of minimal stopping sets tends to a Poisson distribution with independent components as the length tends to infinity. Because of this independence one can relate the number of minimal stopping sets to the number of stopping sets – any combination of minimal stopping sets gives rise to a stopping set. In the limit of infinity blocklenghts the minimal stopping sets are non-overlapping with probability one so that the weight of the resulting stopping set is just the sum of the weights of the individual stopping sets. For example, the number of stopping sets of size two is equal to the number of minimal stopping sets of size two plus the number of stopping sets we get by taking all pairs of (minimal) stopping sets of size one.

Therefore, define $\tilde{A}(x) = \sum_{s\geq 1} \tilde{A}_s x^s$, with $\tilde{A}_s$, the expected number of minimal stopping sets of size s in the graph. Define further $A(x) = \sum_{s\geq 0} A_s x^s$, with A_s the expected number of stopping sets of size s in the graph (not necessarily minimal). We then have

$$A(x) = e^{\tilde{A}(x)} = 1 + \tilde{A}(x) + \frac{\tilde{A}(x)^2}{2!} + \frac{\tilde{A}(x)^3}{3!} + \cdots,$$

so that conversely $\tilde{A}(x) = \log(A(x))$.

It remains to determine the number of stopping sets. As remarked right after the statement of the lemma, any expression which converges in the limit of large blocklength to the asymptotic value would satisfy the statement of the lemma but we get the best empirical agreement for short lengths if we use the exact finite-length averages. These average were already compute in [1] and are given as in (16).

Consider now e.g. the bit erasure probability. We first compute $A(x)$ using (16) and then $\tilde{A}(x)$ by means of $\tilde{A}(x) = \log(A(x))$. Consider one minimal stopping set of size s. The probability that its s associated bits are all erased is equal to ϵ^s and if this is the case this stopping set causes s erasures. Since there are in expectation $\bar{A}_s$ minimal stopping sets of size s and minimal stopping sets are non-overlapping with increasing probability as the blocklength increases a simple union bound is asymptotically tight. The expression for the block erasure probability is derived in a similar way. Now we are interested in the probability that a particular graph and noise realization results in no (small) stopping set. Using the fact that the distribution of minimal stopping sets follows a Poisson distribution we get equation (15). ∎

D. Complete Approximation

In Section II-B, we have studied the erasure probability stemming from failures of size bigger than $n\gamma\nu^*$ where $\gamma \in (0,1)$ and $\nu^* = \epsilon^* L(y^*)$, i.e., ν^* is the asymptotic fractional number of erasures remaining after the decoding at the threshold. In Section II-C, we have studied the probability of erasures resulting from stopping sets of size between $s_{\min}$ and $n\gamma\nu^*$. Combining the results in the two previous sections, we get

$$\begin{aligned} \mathrm{P}_B(n,\lambda,\rho,\epsilon) =& \mathrm{P}_B^W(n,\lambda,\rho,\epsilon) + \mathrm{P}_{B,s_{\min}}^E(n,\lambda,\rho,\epsilon) \\ =& Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right) \\ &+ 1 - e^{-\sum_{s\geq s_{\min}} \tilde{A}_s \epsilon^s}, \end{aligned} \tag{17}$$

$$\begin{aligned} \mathrm{P}_b(n,\lambda,\rho,\epsilon) =& \mathrm{P}_b^W(n,\lambda,\rho,\epsilon) + \mathrm{P}_{b,s_{\min}}^E(n,\lambda,\rho,\epsilon) \\ & \nu^* Q\left(\frac{\sqrt{n}(\epsilon^* - \beta n^{-\frac{2}{3}} - \epsilon)}{\alpha}\right) \\ &+ \sum_{s\geq s_{\min}} s\tilde{A}_s \epsilon^s. \end{aligned} \tag{18}$$

Here we assume that there is a single critical point. If the degree distribution has several critical points (at different values of the channel parameter ϵ_1^*, ϵ_2^*,...) then we simply take a sum of terms $\mathrm{P}_B^W(n,\lambda,\rho,\epsilon)$, one for each critical point.

Let us finally notice that summing the probabilities of different error types provides in principle only an upper bound on the overall error probability. However, for each given channel parameter ϵ, only one of the terms in Eqs. (17), (18) dominates. As a consequence the bound is actually tight.

III. Analytic Determination of α

Let us now show how the scaling parameter α can be determined analytically. We accomplish this in two steps. We first compute the variance of the number of erasure messages. Then we show in a second step how to relate this variance to the scaling parameter α.

A. Variance of the Messages

Consider the ensemble LDPC(n,λ,ρ) and assume that transmission takes place over a BEC of parameter ϵ. Perform ℓ iterations of BP decoding. Set $\mu_i^{(\ell)}$ equal to 1 if the message sent out along edge i from variable to check node is an erasure and 0, otherwise. Consider the variance of these messages in the limit of large blocklengths. More precisely, consider

$$\mathcal{V}^{(\ell)} \equiv \lim_{n\to\infty} \frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2] - \mathbb{E}[(\sum_i \mu_i^{(\ell)})]^2}{nL'(1)}.$$

Lemma 3 in Section IV contains an analytic expression for this quantity as a function of the degree distribution pair (λ,ρ), the channel parameter ϵ, and the number of iterations ℓ. Let us consider this variance as a function of the parameter ϵ and the number of iterations ℓ. Figure 5 shows the result of this evaluation for the case $(L(x) = \frac{2}{5}x^2 + \frac{3}{5}x^3; R(x) = \frac{3}{10}x^2 + \frac{7}{10}x^3)$. The threshold for this example is $\epsilon^* \approx 0.8495897455$.

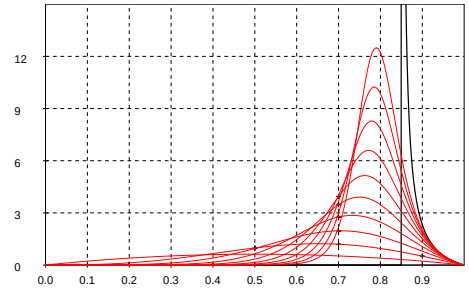

Fig. 5: The variance as a function of ϵ and $\ell = 0,\cdots,9$ for $(L(x) = \frac{2}{5}x^2 + \frac{3}{5}x^3; R(x) = \frac{3}{10}x^2 + \frac{7}{10}x^3)$.

This value is indicated as a vertical line in the figure. As we can see from this figure, the variance is a unimodal function of the channel parameter. It is zero for the extremal values of ϵ (either all messages are known or all are erased) and it takes on a maximum value for a parameter of ϵ which approaches the critical value ϵ^* as ℓ increases. Further, for increasing ℓ the maximum value of the variance increases. The limit of these curves as ℓ tends to infinity $\mathcal{V} = \lim_{\ell\to\infty} \mathcal{V}^{(\ell)}$ is also shown (bold curve): the variance is zero below threshold; above threshold it is positive and diverges as the threshold is approached. In Section IV we state the exact form of the limiting curve. We show that for ϵ approaching ϵ^* from above

$$\mathcal{V} = \frac{\gamma}{(1-\epsilon\lambda'(y)\rho'(\bar{x}))^2} + O((1-\epsilon\lambda'(y)\rho'(\bar{x}))^{-1}), \tag{19}$$

where

$$\gamma = \epsilon^{*2}\lambda'(y^*)^2 \left\{ [\rho(\bar{x}^*)^2 - \rho(\bar{x}^{*2}) + \rho'(\bar{x}^*)(1 - 2x^*\rho(\bar{x}^*)) - \bar{x}^{*2}\rho'(\bar{x}^{*2})] + \epsilon^{*2}\rho'(\bar{x}^*)^2[\lambda(y^*)^2 - \lambda(y^{*2}) - y^{*2}\lambda'(y^{*2})] \right\} .$$

Here y^* is the unique critical point, $x^* = \epsilon^*\lambda(y^*)$, and $\bar{x}^* = 1 - x^*$. Since $(1 - \epsilon\lambda'(y)\rho'(\bar{x})) = \Theta(\sqrt{\epsilon - \epsilon^*})$, Eq. (19) implies a divergence at ϵ^*.

B. Relation Between γ and α

Now that we know the asymptotic variance of the edges messages, let us discuss how this quantity can be related to the scaling parameter α. Think of a decoder operating above the threshold of the code. Then, for large blocklengths, it will get stuck with high probability before correcting all nodes. In Fig 6 we show R_1, the number of degree-one check nodes, as a function of the number of erasure messages for a few decoding runs. Let $\mathcal{V}_*$ represent the normalized variance of

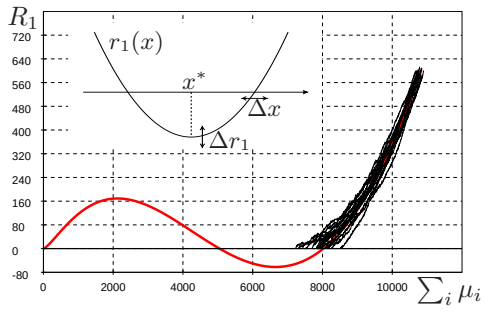

Fig. 6: Number of degree-one check nodes as a function of the number of erasure messages in the corresponding BP decoder for LDPC$(n = 8192, \lambda(x) = x^2, \rho(x) = x^5)$. The thin lines represent the decoding trajectories that stop when $r_1 = 0$ and the thick line is the mean curve predicted by density evolution.

the number of erased messages in the decoder after an infinite number of iterations

$$\mathcal{V}_* \equiv \lim_{n\to\infty} \lim_{\ell\to\infty} \frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2] - \mathbb{E}[(\sum_i \mu_i^{(\ell)})]^2}{nL'(1)}.$$

In other words, $\mathcal{V}_*$ is the variance of the point at which the decoding trajectories hit the $R_1 = 0$ axis.

This quantity can be related to the variance of the number of degree-one check nodes through the slope of the density evolution curve. Normalize all the quantities by $nL'(1)$, the number of edges in the graph. Consider the curve $r_1(\epsilon, x)$ given by density evolution, and representing the fraction of degree-one check nodes in the residual graph, around the critical point for an erasure probability above the threshold (see Fig.6). The real decoding process stops when hitting the $r_1 = 0$ axis. Think of a virtual process identical to the decoding for $r_1 > 0$ but that continues below the $r_1 = 0$ axis (for a proper definition, see [3]). A simple calculation shows that if the point at which the curve hits the x-axis varies by Δx while keeping the minimum at x^*, it results in a variation of the height of the curve by

$$\Delta r_1 = \frac{\partial^2 r_1(\epsilon, x)}{\partial x^2}\Big|_* (x - x^*)\Delta x + o(x - x^*)$$

Taking the expectation of the square on both side and letting ϵ tend to ϵ^*, we obtain the normalized variance of R_1 at threshold

$$\begin{aligned}\delta^{r_1,r_1}|_* &= \lim_{\epsilon\to\epsilon^*} \left(\left(\frac{\partial^2 r_1(\epsilon, x)}{\partial x^2}\Big|_* \right)^2 (x - x^*)^2\mathcal{V} + o((x - x^*)^2) \right) \\ &= \left(\frac{x^*}{\epsilon^*\lambda'(y^*)} \right)^2 \lim_{\epsilon\to\epsilon^*} (1 - \epsilon\lambda'(y)\rho'(\bar{x}))^2\mathcal{V}_* .\end{aligned}$$

The transition between the first and the second line comes the relationship between the ϵ and x, with $r_1(\epsilon, x) = 0$ when ϵ tends to ϵ^*.

The quantity $\mathcal{V}_*$ differs from $\mathcal{V}$ computed in the previous paragraphs because of the different order of the limits $n \to \infty$ and $\ell \to \infty$. However it can be proved that the order does not matter and $\mathcal{V} = \mathcal{V}_*$. Using the result (19), we finally get

$$\delta^{r_1,r_1}|_* = \left(\frac{x^*}{\epsilon^*\lambda'(y^*)} \right)^2 \gamma .$$

We conclude that the scaling parameter α can be obtained as

$$\alpha = \sqrt{\frac{\delta^{r_1 r_1}|_*}{L'(1)\left(\frac{\partial r_1}{\partial \epsilon}\right)^2}} = \sqrt{\frac{\gamma}{L'(1)x^{*2}\lambda'(y^*)^2\rho'(\bar{x}^*)^2}}$$

The last expression is equal to the one in Lemma 1.

IV. Message Variance

Consider the ensemble LDPC(n, λ, ρ) and transmission over the BEC of erasure probability ϵ. As pointed out in the previous section, the scaling parameter α can be related to the (normalized) variance, with respect to the choice of the graph and the channel realization, of the number of erased edge messages sent from the variable nodes. Although what really matters is the limit of this quantity as the blocklength and the number of iterations tend to infinity (in this order) we start by providing an exact expression for finite number of iterations ℓ (at infinite blocklength). At the end of this section, we shall take the limit $\ell \to \infty$.

To be definite, we initialize the iterative decoder by setting all check-to-variable messages to be erased at time 0. We let x_i (respectively y_i) be the fraction of erased messages sent from variable to check nodes (from check to variable nodes), at iteration i, in the infinite blocklength limit. These values are determined by the density evolution [1] recursions $y_{i+1} = 1 - \rho(\bar{x}_i)$, with $x_i = \epsilon\lambda(y_i)$ (where we used the notation $\bar{x}_i = 1 - x_i$). The above initialization implies $y_0 = 1$. For future convenience we also set $x_i = y_i = 1$ for $i < 0$.

Using these variables, we have the following characterization of $\mathcal{V}^{(\ell)}$, the (normalized) variance after ℓ iterations.

Lemma 3: Let $\mathtt{G}$ be chosen uniformly at random from LDPC(n, λ, ρ) and consider transmission over the BEC of erasure probability ϵ. Label the $nL'(1)$ edges of $\mathtt{G}$ in some fixed order by the elements of $\{1, \cdots, nL'(1)\}$. Assume that the receiver performs ℓ rounds of Belief Propagation decoding and let $\mu_i^{(\ell)}$ be equal to one if the message sent at the end of the ℓ-th iteration along edge i (from a variable node to a check

node) is an erasure, and zero otherwise. Then

$$\begin{aligned}\mathcal{V}^{(\ell)} &\equiv \lim_{n\to\infty} \frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2] - \mathbb{E}[(\sum_i \mu_i^{(\ell)})]^2}{nL'(1)} \qquad (20)\\
&= x_\ell + x_\ell(1,0)\Big(\sum_{j=0}^{\ell} \mathtt{V}(\ell)\cdots\mathtt{C}(\ell-j)\Big)(1,0)^T \\
&\qquad\qquad \text{edges in } \mathtt{T}_1\\
&\quad + x_\ell^2\rho'(1)\sum_{i=0}^{\ell-1}\lambda'(1)^i\rho'(1)^i \qquad \text{edges in } \mathtt{T}_2\\
&\quad + x_\ell(1,0)\Big(\sum_{j=1}^{2\ell} \mathtt{V}(\ell)\cdots\mathtt{C}(\ell-j)\Big)(1,0)^T \\
&\qquad\qquad \text{edges in } \mathtt{T}_3\\
&\quad + (1,0)\Bigg(\sum_{j=0}^{\ell}\big(y_{\ell-j}U^\star(j,j) + (1-y_{\ell-j})U^0(j,j)\big)\\
&\qquad + \sum_{j=\ell+1}^{2\ell} \mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)\\
&\qquad\qquad \cdot\big(y_{\ell-j}U^\star(j,\ell) + (1-y_{\ell-j})U^0(j,\ell)\big)\Bigg) - \\
&\qquad\qquad \text{edges in } \mathtt{T}_4\\
&\quad - x_\ell W(\ell,1)\\
&\quad + \sum_{i=1}^{\ell} F_i\left(x_i W(\ell,1) - \epsilon W(\ell,y_i)\right)\\
&\quad - \sum_{i=1}^{\ell} F_i\epsilon\lambda'(y_i)\big(D(\ell,1)\rho(\bar{x}_{i-1}) - D(\ell,\bar{x}_{i-1})\big)\\
&\quad + \sum_{i=1}^{\ell-1} F_i\left(x_\ell + (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)\\
&\qquad\qquad \cdot(1,0)\mathtt{V}(i)\cdots\mathtt{C}(i-\ell)(1,0)^T\\
&\quad - \sum_{i=1}^{\ell-1} F_i x_i\left(x_\ell + (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right),\\
&\qquad\qquad \cdot\left(\lambda'(1)\rho'(1)\right)^\ell\end{aligned}$$

where we introduced the shorthand

$$\mathtt{V}(i)\cdots\mathtt{C}(i-j) \equiv \prod_{k=0}^{j-1}\mathtt{V}(i-k)\mathtt{C}(i-k-1). \qquad (21)$$

We We define the matrices

$$\mathtt{V}(i) = \begin{pmatrix} \epsilon\lambda'(y_i) & 0\\ \lambda'(1)-\epsilon\lambda'(y_i) & \lambda'(1)\end{pmatrix}, \qquad (22)$$

$$\mathtt{C}(i) = \begin{pmatrix} \rho'(1) & \rho'(1)-\rho'(\bar{x}_i)\\ 0 & \rho'(\bar{x}_i)\end{pmatrix}, \qquad i\geq 0, \qquad (23)$$

$$\mathtt{V}(i) = \begin{pmatrix} \lambda'(1) & 0\\ 0 & \lambda'(1)\end{pmatrix}, \qquad (24)$$

$$\mathtt{C}(i) = \begin{pmatrix} \rho'(1) & 0\\ 0 & \rho'(1)\end{pmatrix}, \qquad i<0. \qquad (25)$$

Further, $U^\star(j,j)$, $U^\star(j,\ell)$, $U^0(j,j)$ and $U^0(j,\ell)$ are computed through the following recursion. For $j\leq\ell$, set

$$\begin{aligned}U^\star(j,0) &= (y_{\ell-j}\epsilon\lambda'(y_\ell), (1-y_{\ell-j})\epsilon\lambda'(y_\ell))^T,\\ U^0(j,0) &= (0,0)^T,\end{aligned}$$

whereas for $j>\ell$, initialize by

$$\begin{aligned}U^\star(j,j-\ell) =& (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)(1,0)^T\begin{pmatrix}\epsilon\lambda'(y_{2\ell-j})\\ 0\end{pmatrix}\\ &+ (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)(0,1)^T\begin{pmatrix}\epsilon(\lambda'(1)-\lambda'(y_{2\ell-j}))\\ \lambda'(1)(1-\epsilon)\end{pmatrix},\\ U^0(j,j-\ell) =& (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)(0,1)^T\begin{pmatrix}\epsilon\lambda'(1)\\ (1-\epsilon)\lambda'(1)\end{pmatrix}.\end{aligned}$$

The recursion is

$$\begin{aligned}U^\star(j,k) =& M_1(j,k)\mathtt{C}(\ell-j+k-1)U^\star(j,k-1) \qquad (26)\\ &+ M_2(j,k)[N_1(j,k-1)U^\star(j,k-1)\\ &\qquad + N_2(j,k-1)U^0(j,k-1)],\\ U^0(j,k) =& \mathtt{V}(\ell-j+k)[N_1(j,k-1)U^\star(j,k-1) \qquad (27)\\ &\qquad + N_2(j,k-1)U^0(j,k-1)],\end{aligned}$$

with

$$M_1(j,k) = \begin{pmatrix} \epsilon\lambda'(y_{\max\{\ell-k,\ell-j+k\}}) & 0\\ \mathbf{1}_{\{j<2k\}}\epsilon(\lambda'(y_{\ell-k})-\lambda'(y_{\ell-j+k})) & \epsilon\lambda'(y_{\ell-k})\end{pmatrix},$$

$$M_2(j,k) = \begin{pmatrix} \mathbf{1}_{\{j>2k\}}\epsilon(\lambda'(y_{\ell-j+k})-\lambda'(y_{\ell-k})) & 0\\ \lambda'(1)-\epsilon\lambda'(y_{\min\{\ell-k,\ell-j+k\}}) & \lambda'(1)-\epsilon\lambda'(y_{\ell-k})\end{pmatrix},$$

$$N_1(j,k) = \begin{pmatrix} \rho'(1)-\rho'(\bar{x}_{\ell-k-1}) & \rho'(1)-\rho'(\bar{x}_{\max\{\ell-k-1,\ell-j+k\}})\\ 0 & \mathbf{1}_{\{j\leq 2k\}}(\rho'(\bar{x}_{\ell-j+k})-\rho'(\bar{x}_{\ell-k-1}))\end{pmatrix},$$

$$N_2(j,k) = \begin{pmatrix} \rho'(\bar{x}_{\ell-k-1}) & \mathbf{1}_{\{j>2k\}}(\rho'(\bar{x}_{\ell-k-1})-\rho'(\bar{x}_{\ell-j+k}))\\ 0 & \rho'(\bar{x}_{\min\{\ell-k-1,\ell-j+k\}})\end{pmatrix}.$$

The coefficients F_i are given by

$$F_i = \prod_{k=i+1}^{\ell}\epsilon\lambda'(y_k)\rho'(\bar{x}_{k-1}), \qquad (28)$$

and finally

$$\begin{aligned}W(\ell,\alpha) =& \sum_{k=0}^{2\ell}(1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(\ell-k)A(\ell,k,\alpha)\\ &+ x_\ell(\alpha\lambda'(\alpha)+\lambda(\alpha))\rho'(1)\sum_{i=0}^{\ell-1}\rho'(1)^i\lambda'(1)^i,\end{aligned}$$

with $A(\ell,k,\alpha)$ equal to

$$\begin{cases}\begin{pmatrix}\epsilon\alpha y_{\ell-k}\lambda'(\alpha y_{\ell-k})+\epsilon\lambda(\alpha y_{\ell-k})\\ \alpha\lambda'(\alpha)+\lambda(\alpha)-\epsilon\alpha y_{\ell-k}\lambda'(\alpha y_{\ell-k})-\epsilon\lambda(\alpha y_{\ell-k})\end{pmatrix}, & k\leq\ell\\ \begin{pmatrix}\alpha\lambda'(\alpha)+\lambda(\alpha)\\ 0\end{pmatrix}, & k>\ell\end{cases}$$

and

$$D(\ell,\alpha) = \sum_{k=1}^{2\ell}(1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(\ell-k+1)\mathtt{V}(\ell-k+1) \cdot \begin{pmatrix} \alpha\rho'(\alpha)+\rho(\alpha)-\alpha(\bar{x}_{\ell-k})\rho'(\alpha\bar{x}_{\ell-k})-\rho(\alpha\bar{x}_{\ell-k}) \\ \alpha\bar{x}_{\ell-k}\rho'(\alpha\bar{x}_{\ell-k})+\rho(\alpha\bar{x}_{\ell-k}) \end{pmatrix} + x_\ell(\alpha\rho'(\alpha)+\rho(\alpha))\sum_{i=0}^{\ell-1}\rho'(1)^i\lambda'(1)^i.$$

Proof: Expand $\mathcal{V}^{(\ell)}$ in (20) as

$$\begin{aligned}\mathcal{V}^{(\ell)} &= \lim_{n\to\infty}\frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2]-\mathbb{E}[\sum_i \mu_i^{(\ell)}]^2}{nL'(1)}, \\ &= \lim_{n\to\infty}\frac{\sum_j\left(\mathbb{E}[\mu_j^{(\ell)}\sum_i\mu_i^{(\ell)}]-\mathbb{E}[\mu_j^{(\ell)}]\mathbb{E}[\sum_i\mu_i^{(\ell)}]\right)}{nL'(1)}, \\ &= \lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_i\mu_i^{(\ell)}]-\mathbb{E}[\mu_1^{(\ell)}]\mathbb{E}[\sum_i\mu_i^{(\ell)}], \\ &= \lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_i\mu_i^{(\ell)}]-nL'(1)x_\ell^2. \end{aligned} \tag{29}$$

In the last step, we have used the fact that $x_\ell = \mathbb{E}[\mu_i^{(\ell)}]$ for any $i \in \{1,\cdots,\Lambda'(1)\}$. Let us look more carefully at the first term of (29). After a finite number of iterations, each message $\mu_i^{(\ell)}$ depends upon the received symbols of a *subset* of the variable nodes. Since ℓ is kept finite, this subset remains finite in the large blocklength limit, and by standard arguments is a tree with high probability. As usual, we refer to the subgraph containing all such variable nodes, as well as the check nodes connecting them as to the *computation tree* for $\mu_i^{(\ell)}$.

It is useful to split the sum in the first term of Eq. (29) into two contributions: the first contribution stems from edges i so that the computation trees of $\mu_1^{(\ell)}$ and $\mu_i^{(\ell)}$ *intersect*, and the second one stems from the remaining edges. More precisely, we write

$$\lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)}\sum_i\mu_i^{(\ell)}]-nL'(1)x_\ell^2\right) = \lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}}\mu_i^{(\ell)}] + \lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}^c}\mu_i^{(\ell)}]-nL'(1)x_\ell^2\right). \tag{30}$$

We define $\mathtt{T}$ to be that subset of the variable-to-check edge indices so that if $i \in \mathtt{T}$ then the computation trees $\mu_i^{(\ell)}$ and $\mu_1^{(\ell)}$ intersect. This means that $\mathtt{T}$ includes all the edges whose messages depend on some of the received values that are used in the computation of $\mu_1^{(\ell)}$. For convenience, we complete $\mathtt{T}$ by including all edges that are connected to the same variable nodes as edges that are already in $\mathtt{T}$. $\mathtt{T}^c$ is the complement in $\{1,\cdots,nL'(1)\}$ of the set of indices $\mathtt{T}$.

The set of indices $\mathtt{T}$ depends on the number of iterations performed and on the graph realization. For any fixed ℓ, $\mathtt{T}$ is a tree with high probability in the large blocklength limit, and admits a simple characterization. It contains two sets of edges: the ones 'above' and the ones 'below' edge 1 (we call this the 'root' edge and the variable node it is connected to, the 'root' variable node). Edges *above* the root are the ones departing from a variable node that can be reached by a non reversing path starting with the root edge and involving at most ℓ variable nodes (not including the root one). Edges *below* the root are the ones departing from a variable node that can be reached by a non reversing path starting with the opposite of the root edge and involving at most 2ℓ variable nodes (not including the root one). Edges departing from the root variable node are considered below the root (apart from the root edge itself).

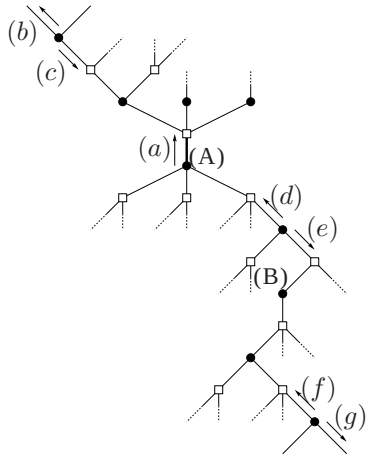

Fig. 7: Graph representing all edges contained in $\mathtt{T}$, for the case of $\ell = 2$. The small letters represent messages sent along the edges from a variable node to a check node and the capital letters represent variable nodes. The message $\mu_1^{(\ell)}$ is represented by (a).

We have depicted in Fig. 7 an example for the case of an irregular graph with $\ell = 2$. In the middle of the figure the edge $(a) \equiv 1$ carries the message $\mu_1^{(\ell)}$. We will call $\mu_1^{(\ell)}$ the root message. We expand the graph starting from this root node. We consider ℓ variable node levels above the root. As an example, notice that the channel output on node (A) affects $\mu_1^{(\ell)}$ as well as the message sent on (b) at the ℓ-th iteration. Therefore the corresponding computation trees intersect and, according to our definition $(b) \in \mathtt{T}$. On the other hand, the computation tree of (c) does not intersect the one of (a), but $(c) \in \mathtt{T}$ because it shares a variable node with (b). We also expand 2ℓ levels below the root. For instance, the value received on node (B) affects both $\mu_1^{(\ell)}$ and the message sent on (g) at the ℓ-th iteration.

We compute the two terms in (30) separately. Define $S = \lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}}\mu_i^{(\ell)}]$ and $S^c = \lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}^c}\mu_i^{(\ell)}]-nL'(1)x_\ell^2\right)$.

1) Computation of S: Having defined $\mathtt{T}$, we can further identify four different types of terms appearing in S and write

$$\begin{aligned} S &= \lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}}\mu_i^{(\ell)}] \\ &= \lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}_1}\mu_i^{(\ell)}]+\lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}_2}\mu_i^{(\ell)}]+ \\ &\quad \lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}_3}\mu_i^{(\ell)}]+\lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}_4}\mu_i^{(\ell)}] \end{aligned}$$

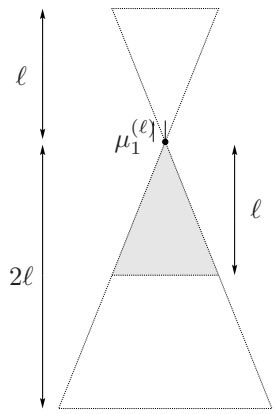

Fig. 8: Size of $\mathtt{T}$. It contains ℓ layers of variable nodes above the root edge and 2ℓ layer of variable nodes below the root variable node. The gray area represent the computation tree of the message $\mu_1^{(\ell)}$. It contains ℓ layers of variable nodes below the root variable node.

The subset $\mathtt{T}_1 \subset \mathtt{T}$ contains the edges above the root variable node that carry messages that point upwards (we include the root edge in $\mathtt{T}_1$). In Fig. 7, the message sent on edge (b) is of this type. $\mathtt{T}_2$ contains all edges above the root but point downwards, such as (c) in Fig. 7. $\mathtt{T}_3$ contains the edges below the root that carry an upward messages, like (d) and (f). Finally, $\mathtt{T}_4$ contains the edges below the root variable node that point downwards, like (e) and (g).

Let us start with the simplest term, involving the messages in $\mathtt{T}_2$. If $i \in \mathtt{T}_2$, then the computation trees of $\mu_1^{(\ell)}$, and $\mu_i^{(\ell)}$ are with high probability disjoint in the large blocklength limit. In this case, the messages $\mu_1^{(\ell)}$ and $\mu_i^{(\ell)}$ do not depend on any common channel observation. The messages are nevertheless correlated: conditioned on the computation graph of the root edge the degree distribution of the computation graph of edge i is biased (assume that the computation graph of the root edge contains an unusual number of high degree check nodes; then the computation graph of edge i must contain in expectation an unusual low number of high degree check nodes). This correlation is however of order $O(1/n)$ and since $\mathtt{T}$ only contains a finite number of edges the contribution of this correlation vanishes as $n \to \infty$. We obtain therefore

$$\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_2} \mu_i^{(\ell)}] = x_\ell^2 \rho'(1) \sum_{i=0}^{\ell-1} \rho'(1)^i \lambda'(1)^i,$$

where we used $\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \mu_i^{(\ell)}] = x_\ell^2$, and the fact that the expected number of edges in $\mathtt{T}_2$ is $\rho'(1) \sum_{i=0}^{\ell-1} \lambda'(1)^i \rho'(1)^i$.

For the edges in $\mathtt{T}_1$ we obtain

$$\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)}] = x_\ell + \tag{31}$$
$$x_\ell (1,0) \left(\sum_{j=1}^{\ell} \mathtt{V}(\ell)\mathtt{C}(\ell-1)\cdots\mathtt{V}(\ell-j+1)\mathtt{C}(\ell-j) \right) (1,0)^T,$$

with the matrices $\mathtt{V}(i)$ and $\mathtt{C}(i)$ defined in Eqs. (22) and (23). In order to understand this expression, consider the following case (cf. Fig. 9 for an illustration). We are at the i-th iteration of BP decoding and we pick an edge at random in the graph. It is connected to a check node of degree j with probability ρ_j. Assume further that the message carried by this edge from the variable node to the check node (incoming message) is erased with probability p and known with probability $\bar{p}$. We want to compute the expected numbers of erased and known messages sent out by the check node on its other edges (outgoing messages). If the incoming message is erased, then the number of erased outgoing messages is exactly $(j-1)$. Averaging over the check node degrees gives us $\rho'(1)$. If the incoming message is known, then the expected number of erased outgoing messages is $(j-1)(1-(1-x_i)^{j-2})$. Averaging over the check node degrees gives us $\rho'(1) - \rho'(1-x_i)$. The expected number of erased outgoing messages is therefore, $p\rho'(1) + \bar{p}(\rho'(1) - \rho'(1-x_i))$. Analogously, the expected number of known outgoing messages is $\bar{p}\rho'(x_i)$. This result can be written using a matrix notation: the expected number of erased (respectively, known) outgoing messages is the first (respectively, second) component of the vector $\mathtt{C}(i)(p,\bar{p})^T$, with $\mathtt{C}(i)$ being defined in Eq. (23).

The situation is similar if we consider a variable node instead of the check node with the matrix the matrix $\mathtt{V}(i)$ replacing $\mathtt{C}(i)$. The result is generalized to several layers of check and variable nodes, by taking the product of the corresponding matrices, cf. Fig. 9.

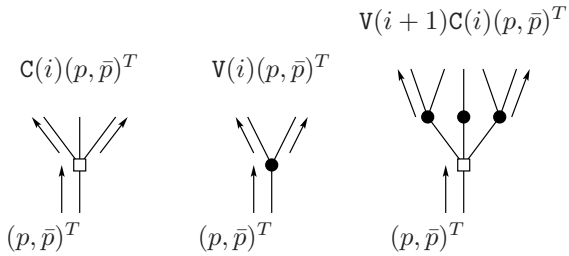

Fig. 9: Number of outgoing erased messages as a function of the probability of erasure of the incoming message.

The contribution of the edges in $\mathtt{T}_1$ to S is obtained by writing

$$\lim_{n\to\infty} \mathbb{E}[\mu_1^{(\ell)} \sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)}]$$
$$= \lim_{n\to\infty} \mathrm{P}\{\mu_1^{(\ell)} = 1\} \mathbb{E}[\sum_{i\in\mathtt{T}_1} \mu_i^{(\ell)} \mid \mu_1^{(\ell)} = 1]. \tag{32}$$

The conditional expectation on the right hand side is given by

$$1 + \sum_{j=1}^{\ell} (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(\ell-j)\begin{pmatrix}1\\0\end{pmatrix}. \tag{33}$$

where the 1 is due to the fact that $\mathbb{E}[\mu_1^{(\ell)} \mid \mu_1^{(\ell)} = 1] = 1$, and each summand $(1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(\ell-j)(1,0)^T$, is the expected number of erased messages in the j-th layer of edges in $\mathtt{T}_1$, conditioned on the fact that the root edge is erased at iteration

ℓ (notice that $\mu_1^{(\ell)}=1$ implies $\mu_1^{(i)}=1$ for all $i\le\ell$). Now multiplying (33) by $\mathrm{P}\{\mu_1^{(\ell)}=1\}=x_\ell$ gives us (31).

The computation is similar for the edges in $\mathtt{T}_3$ and results in

$$\lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}_3}\mu_i^{(\ell)}]= \quad x_\ell\sum_{j=1}^{2\ell}(1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(\ell-j)\begin{pmatrix}1\\0\end{pmatrix}.$$

In this sum, when $j>\ell$, we have to evaluate the matrices $\mathtt{V}(i)$ and $\mathtt{C}(i)$ for negative indices using the definitions given in (24) and (25). The meaning of this case is simple: if $j>\ell$ then the observations in these layers do not influence the message $\mu_1^{(\ell)}$. Therefore, for these steps we only need to *count* the expected number of edges.

In order to obtain S, it remains to compute the contribution of the edges in $\mathtt{T}_4$. This case is slightly more involved than the previous ones. Recall that $\mathtt{T}_4$ includes all the edges that are below the root node and point downwards. In Fig. 7, edges (e) and (g) are elements of $\mathtt{T}_4$. We claim that

$$\begin{aligned}&\lim_{n\to\infty}\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}_4}\mu_i^{(\ell)}]\\ =&(1,0)\sum_{j=0}^{\ell}\left(y_{\ell-j}U^\star(j,j)+(1-y_{\ell-j})U^0(j,j)\right)\qquad(34)\\ &+(1,0)\sum_{j=\ell+1}^{2\ell}\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)\\ &\qquad\cdot\left(y_{\ell-j}U^\star(j,\ell)+(1-y_{\ell-j})U^0(j,\ell)\right).\end{aligned}$$

The first sum on the right hand side corresponds to messages $\mu_i^{(\ell)}$, $i\in\mathtt{T}_4$ whose computation tree contains the root variable node. In the case of Fig. 7, where $\ell=2$, the contribution of edge (e), would be counted in this first sum. The second term in (34) corresponds to edges $i\in\mathtt{T}_4$, that are separated from the root edge by more than $\ell+1$ variable nodes. In Fig. 7, edge (g) is of this type.

In order to understand the first sum in (34), consider the root edge and an edge $i\in\mathtt{T}_4$ separated from the root edge by $j+1$ variable node with $j\in\{0,\cdots,\ell\}$. For this edge in $\mathtt{T}_4$, consider two messages it carries: the message that is sent from the variable node to the check node at the ℓ-th iteration (this 'outgoing' message participates in our second moment calculation), and the one sent from the check node to the variable node at the $(\ell-j)$-th iteration ('incoming'). Define the two-components vector $U^\star(j,j)$ as follows. Its first component is the joint probability that both the root and the outgoing messages are erased conditioned on the fact that the incoming message is erased, multiplied by the expected number of edges in $\mathtt{T}_4$ whose distance from the root is the same as for edge i. Its second component is the joint probability that the root message is erased and that the outgoing message is known, again conditioned on the incoming message being erased, and multiplied by the expected number of edges in $\mathtt{T}_4$ at the same distance from the root. The vector $U^0(j,j)$ is defined in exactly the same manner except that in this case we condition on the incoming message being *known*. The superscript $\star$ or 0 accounts respectively for the cases where the incoming message is erased or known.

From these definitions, it is clear that the contribution to S of the edges that are in $\mathtt{T}_4$ and separated from the root edge by $j+1$ variable nodes with $j\in\{0,\cdots,\ell\}$, is $(1,0)\left(y_{\ell-j}U^\star(j,j)+(1-y_{\ell-j})U^0(j,j)\right)$. We still have to evaluate $U^\star(j,j)$ and $U^0(j,j)$. In order to do this, we define the vectors $U^\star(j,k)$ and $U^0(j,k)$ with $k\le j$, analogously to the case $k=j$, except that this time we consider the root edge and an edge in $i\in\mathtt{T}_4$ separated from the root edge by $k+1$ variable nodes. The outgoing message we consider is the one at the $(\ell-j+k)$-th iteration and the incoming message we condition on, is the one at the $(\ell-k)$-th iteration. It is easy to check that $U^\star(j,j)$ and $U^0(j,j)$ can be computed in a recursive manner using $U^\star(j,k)$ and $U^0(j,k)$. The initial conditions are

$$U^\star(j,0)=\begin{pmatrix}y_{\ell-j}\epsilon\lambda'(y_\ell)\\(1-y_{\ell-j})\epsilon\lambda'(y_\ell)\end{pmatrix},\quad U^0(j,0)=\quad\begin{pmatrix}0\\0\end{pmatrix},$$

and the recursion is for $k\in\{1,\cdots,j\}$ is the one given in Lemma 3, cf. Eqs. (26) and (27). Notice that any received value which is on the path between the root edge and the edge in $\mathtt{T}_4$ affects both the messages $\mu_1^{(\ell)}$ and $\mu_i^{(\ell)}$ on the corresponding edges. This is why this recursion is slightly more involved than the one for $\mathtt{T}_1$. The situation is depicted in the left side of Fig. 10.

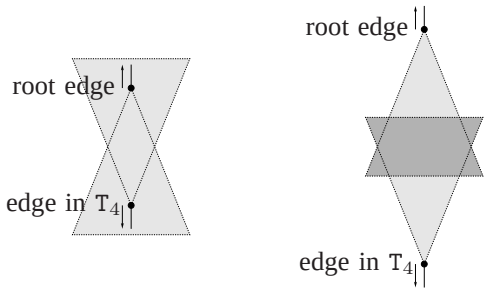

Fig. 10: The two situations that arise when computing the contribution of $\mathtt{T}_4$. In the left side we show the case where the two edges are separated by at most $\ell+1$ variable nodes and in the right side, the case where they are separated by more than $\ell+1$ variable nodes.

Consider now the case of edges in $\mathtt{T}_4$ that are separated from the root edge by more than $\ell+1$ variable nodes, cf. right picture in Fig. 10. In this case, not all of the received values along the path connecting the two edges, do affect both messages. We therefore have to adapt the previous recursion. We start from the root edge and compute the effect of the received values that only affect this message resulting in a expression similar to the one we used to compute the contribution of $\mathtt{T}_1$. This gives us the following initial condition

$$\begin{aligned}U^\star(j,j-\ell)=&(1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)(1,0)^T\begin{pmatrix}\epsilon\lambda'(y_{2\ell-j})\\0\end{pmatrix}\\&+(1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)(0,1)^T\begin{pmatrix}\epsilon(\lambda'(1)-\lambda'(y_{2\ell-j}))\\\lambda'(1)(1-\epsilon)\end{pmatrix},\\U^0(j,j-\ell)=&(1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)(0,1)^T\begin{pmatrix}\epsilon\lambda'(1)\\(1-\epsilon)\lambda'(1)\end{pmatrix}.\end{aligned}$$

We then apply the recursion given in Lemma 3 to the intersection of the computation trees. We have to stop the recursion at $k=\ell$ (end of the intersection of the computation trees). It remains to account for the received values that only affect the messages on the edge in $\mathtt{T}_4$. This is done by writing

$$(1,0)\sum_{j=\ell+1}^{2\ell}\mathtt{V}(\ell)\cdots\mathtt{C}(2\ell-j)\cdot\left(y_{\ell-j}U^{\star}(j,\ell)+(1-y_{\ell-j})U^{0}(j,\ell)\right),$$

which is the second term on the right hand side of Eq. (34).

2) Computation of S^c: We still need to compute $S^c=\lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}^{\mathtt{c}}}\mu_i^{(\ell)}]-nL'(1)x_\ell^2\right)$. Recall that by definition, all the messages that are carried by edges in $\mathtt{T}^c$ at the ℓ-th iteration are functions of a set of received values distinct from the ones $\mu_1^{(\ell)}$ depends on. At first sight, one might think that such messages are independent from $\mu_1^{(\ell)}$. This is indeed the case when the Tanner graph is regular, i.e. for the degree distributions $\lambda(x)=x^{\mathtt{l}-1}$ and $\rho(x)=x^{\mathtt{r}-1}$. We then have

$$\begin{aligned}S^c&=\lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}^{\mathtt{c}}}\mu_i^{(\ell)}]-nL'(1)x_\ell^2\right)\\&=\lim_{n\to\infty}\left(|\mathtt{T}^{\mathtt{c}}|x_\ell^2-\Lambda'(1)x_\ell^2\right)\\&=\lim_{n\to\infty}\left((\Lambda'(1)-|\mathtt{T}|)x_\ell^2-\Lambda'(1)x_\ell^2\right)\\&=-|\mathtt{T}|x_\ell^2\end{aligned}$$

with the cardinality of $\mathtt{T}$ being $|\mathtt{T}|=\sum_{i=0}^{2\ell}(\mathtt{l}-1)^i(\mathtt{r}-1)^i\mathtt{l}+\sum_{i=1}^{\ell}(\mathtt{l}-1)^{i-1}(\mathtt{r}-1)^i\mathtt{l}$.

Consider now an irregular ensemble and let $\mathtt{G}_\mathtt{T}$ be the graph composed by the edges in $\mathtt{T}$ and by the variable and check nodes connecting them. Unlike in the regular case, $\mathtt{G}_\mathtt{T}$ is not fixed anymore and depends (in its size as well as in its structure) on the graph realization. It is clear that the root message $\mu_1^{(\ell)}$ depends on the realization of $\mathtt{G}_\mathtt{T}$. We will see that the messages carried by the edges in $\mathtt{T}^{\mathtt{c}}$ also depend on the realization of $\mathtt{G}_\mathtt{T}$. On the other hand they are clearly conditionally independent given $\mathtt{G}_\mathtt{T}$ (because, conditioned on $\mathtt{G}_\mathtt{T}$, $\mu_1^{(\ell)}$ is just a deterministic function of the received symbols in its computation tree). If we let j denote a generic edge in $\mathtt{T}^{\mathtt{c}}$ (for instance, the one with the lowest index), we can therefore write

$$\begin{aligned}S^c&=\lim_{n\to\infty}\left(\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}^{\mathtt{c}}}\mu_i^{(\ell)}]-nL'(1)x_\ell^2\right)\\&=\lim_{n\to\infty}\left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[\mathbb{E}[\mu_1^{(\ell)}\sum_{i\in\mathtt{T}^{\mathtt{c}}}\mu_i^{(\ell)}\mid\mathtt{G}_\mathtt{T}]]-nL'(1)x_\ell^2\right)\\&=\lim_{n\to\infty}\left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[|\mathtt{T}^{\mathtt{c}}|\mathbb{E}[\mu_1^{(\ell)}\mid\mathtt{G}_\mathtt{T}]\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]]-nL'(1)x_\ell^2\right)\\&=\lim_{n\to\infty}\Big(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[(nL'(1)-|\mathtt{T}|)\,\mathbb{E}[\mu_1^{(\ell)}\mid\mathtt{G}_\mathtt{T}]\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]]\\&\qquad-nL'(1)x_\ell^2\Big)\\&=\lim_{n\to\infty}nL'(1)\left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[\mathbb{E}[\mu_1^{(\ell)}\mid\mathtt{G}_\mathtt{T}]\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]]-nL'(1)x_\ell^2\right)\\&\quad-\lim_{n\to\infty}\mathbb{E}_{\mathtt{G}_\mathtt{T}}[|\mathtt{T}|\mathbb{E}[\mu_1^{(\ell)}\mid\mathtt{G}_\mathtt{T}]\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]].\end{aligned}\tag{35}$$

We need to compute $\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]$ for a fixed realization of $\mathtt{G}_\mathtt{T}$ and an arbitrary edge j taken from $\mathtt{T}^{\mathtt{c}}$ (the expectation does not depend on $j\in\mathtt{T}^{\mathtt{c}}$: we can therefore consider it as a random edge as well). This value differs slightly from x_ℓ for two reasons. The first one is that we are dealing with a fixed-size Tanner graph (although taking later the limit $n\to\infty$) and therefore the degrees of the nodes in $\mathtt{G}_\mathtt{T}$ are correlated with the degrees of nodes in its complement $\mathtt{G}\backslash\mathtt{G}_\mathtt{T}$. Intuitively, if $\mathtt{G}_\mathtt{T}$ contains an unusually large number of high degree variable nodes, the rest of the graph will contain an unusually small number of high degree variable nodes affecting the average $\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]$. The second reason why $\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]$ differs from x_ℓ, is that certain messages carried by edges in $\mathtt{T}^{\mathtt{c}}$ which are close to $\mathtt{G}_\mathtt{T}$ are affected by messages that flow out of $\mathtt{G}_\mathtt{T}$.

The first effect can be characterized by computing the degree distribution on $\mathtt{G}\backslash\mathtt{G}_\mathtt{T}$ as a function of $\mathtt{G}_\mathtt{T}$. Define $V_i(\mathtt{G}_\mathtt{T})$ (respectively $C_i(\mathtt{G}_\mathtt{T})$) to be the number of variable nodes (check nodes) of degree i in $\mathtt{G}_\mathtt{T}$, and let $V(x;\mathtt{G}_\mathtt{T})=\sum_i V_i(\mathtt{G}_\mathtt{T})x^i$ and $C(x;\mathtt{G}_\mathtt{T})=\sum_i C_i(\mathtt{G}_\mathtt{T})x^i$. We shall also need the derivatives of these polynomials: $V'(x;\mathtt{G}_\mathtt{T})=\sum_i iV_i(\mathtt{G}_\mathtt{T})x^{i-1}$ and $C'(x;\mathtt{G}_\mathtt{T})=\sum_i iC_i(\mathtt{G}_\mathtt{T})x^{i-1}$. It is easy to check that if we take a bipartite graph having a variable degree distributions $\lambda(x)$ and remove a variable node of degree i, the variable degree distribution changes by

$$\delta_i\lambda(x)=\frac{i\lambda(x)-ix^{i-1}}{nL'(1)}+O(1/n^2).$$

Therefore, if we remove $\mathtt{G}_\mathtt{T}$ from the bipartite graph, the remaining graph will have a variable perspective degree distribution that differ from the original by

$$\delta\lambda(x)=\frac{V'(1;\mathtt{G}_\mathtt{T})\lambda(x)-V'(x;\mathtt{G}_\mathtt{T})}{nL'(1)}+O(1/n^2).$$

In the same way, the check degree distribution when we remove $\mathtt{G}_\mathtt{T}$ changes by

$$\delta\rho(x)=\frac{C'(1;\mathtt{G}_\mathtt{T})\rho(x)-C'(x;\mathtt{G}_\mathtt{T})}{nL'(1)}+O(1/n^2).$$

If the degree distributions change by $\delta\lambda(x)$ and $\delta\rho(x)$, the fraction x_ℓ of erased variable-to-check messages changes by δx_ℓ. To the linear order we get

$$\begin{aligned}\delta x_\ell=&\sum_{i=1}^{\ell}\prod_{k=i+1}^{\ell}\epsilon\lambda'(y_k)\rho'(\bar{x}_{k-1})\left[\epsilon\delta\lambda(y_i)-\epsilon\lambda'(y_i)\delta\rho(\bar{x}_{i-1})\right],\\=&\frac{1}{\Lambda'(1)}\sum_{i=1}^{\ell}F_i\left[\epsilon(V'(1;\mathtt{G}_\mathtt{T})\lambda(y_i)-V'(y_i;\mathtt{G}_\mathtt{T}))\right.\\&\left.-\epsilon\lambda'(y_i)(C'(1;\mathtt{G}_\mathtt{T})\rho(\bar{x}_{i-1})-C'(\bar{x}_{i-1};\mathtt{G}_\mathtt{T}))\right]+O(1/n^2),\end{aligned}$$

with F_i defined as in Eq. (28).

Imagine now that we ix the degree distribution of $\mathtt{G}\backslash\mathtt{G}_\mathtt{T}$. The conditional expectation $\mathbb{E}[\mu_j^{(\ell)}\mid\mathtt{G}_\mathtt{T}]$ still depends on the detailed structure of $\mathtt{G}_\mathtt{T}$. The reason is that the messages that flow out of the boundary of $\mathtt{G}_\mathtt{T}$ (both their number and value) depend on $\mathtt{G}_\mathtt{T}$, and these message affect messages in $\mathtt{G}\backslash\mathtt{G}_\mathtt{T}$. Since the fraction of such (boundary) messages is $O(1/n)$, their effect can be evaluated again perturbatively.

Call $\mathcal{B}$ the number of edges forming the boundary of $\mathtt{G}_\mathtt{T}$ (edges emanating upwards from the variable nodes that are ℓ

levels above the root edge and emanating downwards from the variable nodes that are 2ℓ levels below the root variable node) and let $\mathcal{B}_i^\star$ be the number of erased messages carried at the i-th iteration by these edges. Let $\tilde{x}_i$ be the fraction of erased messages, incoming to check nodes in $\mathtt{G}\backslash\mathtt{G}_\mathtt{T}$ from variable nodes in $\mathtt{G}\backslash\mathtt{G}_\mathtt{T}$, at the i-th iteration. Taking into account the messages coming from variable nodes in $\mathtt{G}_\mathtt{T}$ (i.e. corresponding to boundary edge), the overall fraction will be $\tilde{x}_i + \delta\tilde{x}_i$, where

$$\delta\tilde{x}_i = \frac{\mathcal{B}_i^\star - \mathcal{B}\tilde{x}_i}{nL'(1)} + O(1/n^2).$$

This expression simply comes from the fact that at the i-th iteration, we have $(n\Lambda'(1) - \mathtt{T}) = n\Lambda'(1)(1 + O(1/n))$ messages in the complement of $\mathtt{G}_\mathtt{T}$ of which a fraction $\tilde{x}_i$ is erased. Further $\mathcal{B}$ messages incoming from the boundaries of which $\mathcal{B}_i^\star$ are erasures.

Combining the two above effects, we have for an edge $j \in \mathtt{T}^\mathtt{c}$

$$\begin{aligned}\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}] = x_\ell + \frac{1}{\Lambda'(1)} \sum_{i=1}^{\ell} F_i \,[x_i V'(1;\mathtt{G}_\mathtt{T}) - \epsilon V'(y_i;\mathtt{G}_\mathtt{T}) \\ -\epsilon\lambda'(y_i)(C'(1;\mathtt{G}_\mathtt{T})\rho(\bar{x}_{i-1}) - C'(\bar{x}_{i-1};\mathtt{G}_\mathtt{T}))] \\ + \frac{1}{\Lambda'(1)} \sum_{i=1}^{\ell-1} F_i \,(\mathcal{B}_i^\star - \mathcal{B}x_i) + O(1/n^2).\end{aligned}$$

We can now use this expression (35) to obtain

$$\begin{aligned}S^c = &\lim_{n\to\infty} \Lambda'(1) \left(\mathbb{E}_{\mathtt{G}_\mathtt{T}}[\mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G}_\mathtt{T}]\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]] - nL'(1)x_\ell^2\right) \\ &- \lim_{n\to\infty} \mathbb{E}_{\mathtt{G}_\mathtt{T}}[|\mathtt{T}|\mathbb{E}[\mu_1^{(\ell)} \mid \mathtt{G}_\mathtt{T}]\mathbb{E}[\mu_j^{(\ell)} \mid \mathtt{G}_\mathtt{T}]] \\ &= \sum_{i=1}^{\ell} F_i \left(x_i\mathbb{E}[\mu_1^{(\ell)}V'(1;\mathtt{G}_\mathtt{T})] - \epsilon\mathbb{E}[\mu_1^{(\ell)}V'(y_i;\mathtt{G}_\mathtt{T})]\right) \\ &- \sum_{i=1}^{\ell} F_i\epsilon\lambda'(y_i) \Big(\mathbb{E}[\mu_1^{(\ell)}C'(1;\mathtt{G}_\mathtt{T})]\rho(\bar{x}_{i-1}) \\ &\qquad\qquad -\mathbb{E}[\mu_1^{(\ell)}C'(\bar{x}_{i-1};\mathtt{G}_\mathtt{T})]\Big) \\ &+ \sum_{i=1}^{\ell-1} F_i\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star] - \sum_{i=1}^{\ell-1} F_i x_i \mathbb{E}[\mu_1^{(\ell)}\mathcal{B}] - x_\ell\mathbb{E}[\mu_1^{(\ell)}V^{\mathtt{G}_\mathtt{T}'}(1)],\end{aligned}$$

where we took the limit $n \to \infty$ and replaced $|\mathtt{T}|$ by $V'(1;\mathtt{G}_\mathtt{T})$.

It is clear what each of these values represent. For example, $\mathbb{E}[\mu_1^{(\ell)}V'(1;\mathtt{G}_\mathtt{T})]$ is the expectation of $\mu_1^{(\ell)}$ times the number of edges that are in $\mathtt{G}_\mathtt{T}$. Each of these terms can be computed through recursions that are similar in spirit to the ones used to compute S. These recursions are provided in the body of Lemma 3. We will just explain in further detail how the terms $\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}]$ and $\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star]$ are computed.

We claim that

$$\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}] = \left(x_\ell + (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)\left(\lambda'(1)\rho'(1)\right)^\ell.$$

The reason is that $\mu_1^{(\ell)}$ depends only on the realization of its computation tree and not on the whole $\mathtt{G}_\mathtt{T}$. From the definitions of $\mathtt{G}_\mathtt{T}$, the boundary of $\mathtt{G}_\mathtt{T}$ is in average $(\lambda'(1)\rho'(1))^\ell$ larger than the boundary of the computation tree. Finally, the expectation of $\mu_1^{(\ell)}$ times the number of edges in the boundary of its computation tree is computed analogously to what has been done for the contribution of S. The result is $\left(x_\ell + (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right)$ (the term x_ℓ accounts for the root edge, and the other one of the lower boundary of the computation tree). Multiplying this by $(\lambda'(1)\rho'(1))^\ell$, we obtain the above expression.

The calculation of $\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star]$ is similar. We start by computing the expectation of $\mu_1^{(\ell)}$ multiplied by the number of edges in the boundary of its computation tree. This number has to be multiplied by $(1,0)\mathtt{V}(i)\mathtt{C}(i-1)\cdots\mathtt{V}(i-\ell+1)\mathtt{C}(i-\ell)(1,0)^T$ to account for what happens between the boundary of the computation tree and the boundary of $\mathtt{G}_\mathtt{T}$. We therefore obtain

$$\begin{aligned}\mathbb{E}[\mu_1^{(\ell)}\mathcal{B}_i^\star] = &\left(x_\ell + (1,0)\mathtt{V}(\ell)\cdots\mathtt{C}(0)\mathtt{V}(0)(1,0)^T\right) \\ &\cdot (1,0)\mathtt{V}(i)\cdots\mathtt{C}(i-\ell)(1,0)^T.\end{aligned}$$

■

The expression provided in the above lemma has been used to plot $\mathcal{V}^{(\ell)}$ for $\epsilon \in (0,1)$ and for several values of ℓ in the case of an irregular ensemble in Fig. 5.

It remains to determine the asymptotic behavior of this quantity as the number of iterations converges to infinity.

Lemma 4: Let $\mathtt{G}$ be chosen uniformly at random from LDPC(n,λ,ρ) and consider transmission over the BEC of erasure probability ϵ. Label the $nL'(1)$ edges of $\mathtt{G}$ in some fixed order by the elements of $\{1,\ldots,nL'(1)\}$. Set $\mu_i^{(\ell)}$ equal to one if the message along edge i from variable to check node, after ℓ iterations, is an erasure and equal to zero otherwise. Then

$$\begin{aligned}\lim_{\ell\to\infty}\lim_{n\to\infty} &\frac{\mathbb{E}[(\sum_i \mu_i^{(\ell)})^2] - \mathbb{E}[(\sum_i \mu_i^{(\ell)})]^2}{nL'(1)} = \\ &+ \frac{\epsilon^2\lambda'(y)^2(\rho(\bar{x})^2 - \rho(\bar{x}^2) + \rho'(\bar{x})(1-2x\rho(\bar{x})) - \bar{x}^2\rho'(\bar{x}^2))}{(1-\epsilon\lambda'(y)\rho'(\bar{x}))^2} \\ &+ \frac{\epsilon^2\lambda'(y)^2\rho'(\bar{x})^2(\epsilon^2\lambda(y)^2 - \epsilon^2\lambda(y^2) - y^2\epsilon^2\lambda'(y^2))}{(1-\epsilon\lambda'(y)\rho'(\bar{x}))^2} \\ &+ \frac{(x - \epsilon^2\lambda(y^2) - y^2\epsilon^2\lambda'(y^2))(1+\epsilon\lambda'(y)\rho'(\bar{x})) + \epsilon y^2\lambda'(y)}{1-\epsilon\lambda'(y)\rho'(\bar{x})}.\end{aligned}$$

The proof is a (particularly tedious) calculus exercise, and we omit it here for the sake of space.

REFERENCES

[1] T. Richardson and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2006. In preparation.

[2] A. Montanari. Finite-size scaling of good codes. In *Proc. 39th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2001.

[3] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke. Finite-length scaling for iteratively decoded ldpc ensembles. submitted to IEEE IT, June 2004.

[4] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke. Finite-length scaling and finite-length shift for low-density parity-check codes. In *Proc. 42th Annual Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2004.

[5] A. Amraoui, A. Montanari, and R. Urbanke. Finite-length scaling of irregular LDPC code ensembles. In *Proc. IEEE Information Theory Workshop*, Rotorua, New-Zealand, Aug-Sept 2005.

[6] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In *Proceedings of the* 29*th annual ACM Symposium on Theory of Computing*, pages 150–159, 1997.

[7] C. Méasson, A. Montanari, and R. Urbanke. Maxwell's construction: The hidden bridge between maximum-likelihood and iterative decoding. submitted to IEEE Transactions on Information Theory, 2005.