



Article

Finite-Length Analysis for Spatially Coupled LDPC Codes Based on Base Matrix

Yang Liu ^{*}, Sha Sun, Yuzhi Zhang  and Bin Wang

School of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China

* Correspondence: lyang@xust.edu.cn

Abstract: Spatially coupled low density parity check (SC-LDPC) are prominent candidates for future communication standards due to their “threshold saturation” properties. To evaluate the finite-length performance of SC-LDPC codes, a general and efficient finite-length analysis from the perspective of the base matrix is proposed. We analyze the evolution of the residual graphs resulting at each iteration during the decoding process based on the base matrix and then derive the expression for the error probability. To verify the effectiveness of the proposed finite-length analysis, we consider the SC-LDPC code ensembles constructed by parallelly connecting multiple chains (PC-MS-SC-LDPC). The analysis results show that the predicted error probabilities obtained by using the derived expression for the error probability match the simulated error probabilities. The proposed finite-length analysis provides a useful engineering tool for practical SC-LDPC code design and for analyzing the effects of the code parameters on the performances.

Keywords: spatially coupled LDPC codes; finite-length performance analysis; peeling decoder



Citation: Liu, Y.; Sun, S.; Zhang, Y.; Wang, B. Finite-Length Analysis for Spatially Coupled LDPC Codes Based on Base Matrix. *Entropy* **2023**, *25*, 1041. <https://doi.org/10.3390/e25071041>

Academic Editor: Mehul Motani

Received: 18 April 2023

Revised: 3 July 2023

Accepted: 8 July 2023

Published: 11 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Spatially coupled low-density parity check (SC-LDPC) codes have been proven to improve the belief propagation (BP) thresholds up to the maximum a posteriori (MAP) thresholds of the underlying LDPC block codes for the binary erasure channel (BEC) [1]. Afterwards, many new structures were proposed to achieve better thresholds or low-complexity/delay decoding, including designing the coupling pattern, eliminating small absorbing/trapping sets, introducing slight irregularities and so on [2–4]. In addition, most of the literature focused on applying the concept of spatial coupling on other error correction codes to improve the decoding thresholds, such as spatially coupled repeat-accumulate (SC-RA) codes, spatially coupled turbo codes (SC-TCs), spatially coupled precoded rateless codes and so on [5–7]. Moreover, spatial coupling need not be limited to forming a single chain, and more general structures formed by connecting multiple coupled chains were presented to improve the decoding thresholds [8–10]. Different from connecting multiple identical coupled chains, the SC-LDPC codes constructed by parallelly connecting multiple different chains (PC-MS-SC-LDPC) were proposed in [11], which showed that the thresholds of the PC-MS-SC-LDPC code ensembles with flexible rates are very close to Shannon limits over the BEC.

The studies stated above mainly focused on the asymptotic performance analysis of the SC-LDPC ensembles. To analyze the finite-length performances, a scaling law for predicting the error probability of the SC-LDPC codes over the BEC using the peeling decoder (PD) was proposed in [12], which extended the finite-length analysis for LDPC codes in [13]. At each iteration, if the variable node is not erased through the channel or connected to a degree-one check node, it will be recovered successfully and then removed from the decoding graph along with all its attached edges. This decoding process gives rise to a sequence of residual graphs. Therefore, the analysis of the PD process is equivalent to analyzing the evolution of the residual graphs, which can be transformed to analyze the

evolution of the degree distributions (DDs) on the residual graphs. In [13], it was pointed out that the DDs on the residual graphs at any time converge to a multivariate Gaussian. As a result, by computing the mean and the variance of the DDs evolution during the PD process, the error probability can be estimated. Following this principle, a finite-length analysis for the loop ensemble constructed by connecting two identical coupled chains was presented in [14], which showed that connecting two coupled chains can result in better thresholds and improved finite-length performances.

These finite-length analyses focused on particular code structures, either the single SC-LDPC code ensemble or the loop ensemble. When the code structure is changed, these analyses will be either very complicated or invalid. According to the working principle of PD, it can be seen that the core of the finite-length analysis over the BEC is analyzing the DDs evolution of variable nodes and check nodes on the residual graph. The essence is to determine whether one variable/check node on the residual graph is removed or not according to the erasure probabilities computed by the received messages from the connected edges, which depends on the interconnections between the variable nodes and check nodes. Inspired by this, we proposed a general finite-length performance analysis from the perspective of the base matrix for SC-LDPC codes. We considered embedding the base matrix into analyzing the residual graph evolution during the decoding process. In particular, based on the base matrix, we derived the mean graph evolution and estimated the variance of the DDs on the residual graphs at each iteration to predict the error probabilities. To verify the effectiveness of the proposed finite-length analysis, we considered the PC-MS-C-LDPC codes because they have special connection structures that cannot be analyzed by the existing analyses. Using this expression, we plot the predicted error probabilities for different PC-MS-C-LDPC codes and also show the error probabilities by simulations. The comparison results show that the predicted error probabilities can fit the simulated ones well. Since this analysis is performed on the base matrix, it can be applied to other spatially coupled code ensembles defined by the base matrix, including the single SC-LDPC code ensemble and the loop ensemble. The analysis results of the conventional SC-LDPC codes demonstrate this statement.

This paper is organized as follows. In Section 2, we describe the SC-LDPC and PC-MS-C-LDPC code ensembles using the base matrix. In Section 3, we analyze the graph evolution based on the base matrix, including the mean graph evolution and the variance estimation. In Section 4, we show and compare the results for different PC-MS-C-LDPC codes and the conventional SC-LDPC codes. In Section 5, we conclude our work.

2. Construction of PC-MS-C-LDPC Codes

2.1. SC-LDPC Codes

A (J, K, L) SC-LDPC coupled chain was constructed by coupling L disjoint and small (J, K) -regular LDPC protographs. Each protograph was placed at one position in order and each position was denoted by u , $u = 1, 2, \dots, L$. Here, we considered a conventional fully connected coupling pattern to couple these L protographs. Specifically, let $w = \gcd(J, K)$, which denotes the greatest common divisor of J and K . Then, there are J' check nodes and K' variable nodes at each position with $J' = J/w$ and $K' = K/w$. To couple these L protographs, we spread J edges of each variable node at position u to all adjacent check nodes at position $u + i$, $i = 0, 1, \dots, w - 1$. In turn, for each check node at position u , K edges will be connected to all nearby variable nodes at position $u - i$, $i = 0, 1, \dots, w - 1$. To terminate the coupled chain, $w - 1$ extra positions only including additional check nodes will be added at the end.

A (J, K, L) SC-LDPC coupled chain can be viewed as a protograph and its associated incidence matrix \mathbf{B}_{sc} , called a base matrix, is defined in (1), where the submatrices \mathbf{B}_i , $i = 0, \dots, w - 1$ are identical $J' \times K'$ all-one matrices. A (J, K, L, M) SC-LDPC code can be obtained by taking an “ M -lifting” of the (J, K, L) coupled chain [15]. Specifically, the

parity check matrix can be generated by replacing each one entry by one $M \times M$ random permutation matrix and each zero entry by one $M \times M$ all-zero matrix.

$$\mathbf{B}_{sc} = \begin{bmatrix} \mathbf{B}_0 & & & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & & & \\ \vdots & \mathbf{B}_1 & \ddots & & & & \\ \mathbf{B}_{w-1} & \vdots & & & \mathbf{B}_0 & & \\ & \mathbf{B}_{w-1} & \ddots & & \mathbf{B}_1 & & \\ & & & & \vdots & & \\ & & & & \mathbf{B}_{w-1} & & \end{bmatrix}_{J'(L+w-1) \times K'L} \quad (1)$$

2.2. PC-MSCLDPC Codes

Consider C independent and unconnected coupled chains with the same coupling length L , where each chain has a different rate. The k th chain is denoted as $\mathcal{B}(J_k, K_k, L)$ and there are J'_k check nodes and K'_k variable nodes at each position, where $J'_k = J_k/w_k$, $K'_k = K_k/w_k$, $w_k = \text{gcd}(J_k, K_k)$ and $k = 1, 2, \dots, C$. The base matrix is denoted as $\mathbf{B}_{sc,k}$ and the size is $J'_k(L + w_k - 1) \times K'_k L$. Let $a = \min\{K'_1, K'_2, \dots, K'_C\}$ and $b = \min\{J'_1, J'_2, \dots, J'_C\}$.

Then, connect these C chains parallelly by edge exchanges. Specifically, a variable nodes and b check nodes are randomly selected for each position u of every chain $\mathcal{B}(J_k, K_k, L)$, where $u = 1, 2, \dots, L$ and $k = 1, 2, \dots, C$. Next, break the edges between these selected a variable nodes and b check nodes at each position u of $\mathcal{B}(J_k, K_k, L)$ and simultaneously connect these broken edges to b check nodes at each position u of $\mathcal{B}(J_z, K_z, L)$ with $z = (k \bmod C) + 1$. The construction process starts from $k = 1$ and stops until $k = C$. Take the u th position to illustrate this process in Figure 1, where the blue blank squares and red blank circles are the selected check nodes and variable nodes, respectively, and the red dash lines represent the exchange edges between two adjacent chains. The base matrix is

$$\mathbf{B}_{pc} = \begin{bmatrix} \mathbf{B}'_{sc,1} & & & & \mathbf{L}_C \\ \mathbf{L}_1 & \mathbf{B}'_{sc,2} & & & \\ & \mathbf{L}_2 & \ddots & & \\ & & \ddots & \mathbf{B}'_{sc,C-1} & \\ & & & \mathbf{L}_{C-1} & \mathbf{B}'_{sc,C} \end{bmatrix}_{m \times n} \quad (2)$$

where $\mathbf{B}'_{sc,k}$ denotes the remaining matrix after removing the exchange edges from the base matrix $\mathbf{B}_{sc,k}$ and \mathbf{L}_k represents the interconnections between the chain $\mathcal{B}(J_k, K_k, L)$ and $\mathcal{B}(J_z, K_z, L)$, $k = 1, 2, \dots, C$ and $z = (k \bmod C) + 1$. The size of \mathbf{L}_k is $J'_z(L + w_z - 1) \times K'_k L$. $m = \sum_{k=1}^C J'_k(L + w_k - 1)$ and $n = \sum_{k=1}^C K'_k L$. Denote the PC-MSCLDPC code ensemble defined by this base matrix as $\mathcal{P}(J_1, K_1, \dots, J_C, K_C, L)$. Take an “ M -lifting” operation on this base matrix and obtain the parity check matrix of a PC-MSCLDPC code. Denote the PC-MSCLDPC code as $\mathcal{C}(J_1, K_1, \dots, J_C, K_C, L, M)$. More details about the asymptotic performance analysis can be found in [11].

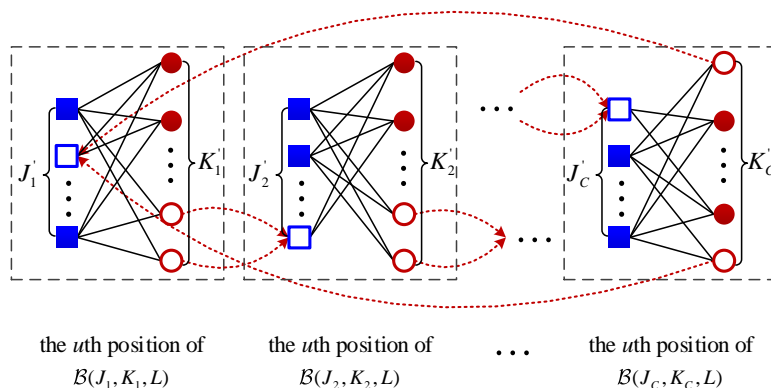


Figure 1. The connection structure at the u th position of the code ensemble $\mathcal{P}(J_1, K_1, \dots, J_C, K_C, L)$.

Example: Consider two coupled chains: $\mathcal{B}(3, 6, 8)$ and $\mathcal{B}(4, 6, 8)$. Since $K'_1 = 2, J'_1 = 1$ for $\mathcal{B}(3, 6, 8)$ and $K'_2 = 3, J'_2 = 2$ for $\mathcal{B}(4, 6, 8)$, we have $a = 2$ and $b = 1$. Thus, we need to select two variable nodes and one check node at each position for $\mathcal{B}(4, 6, 8)$. The connection structure of these two chains in parallel is shown in Figure 2, where the blue blank squares denote the selected one check node and the red blank circles are the selected two variable nodes. Specifically, at each position, we break the two edges between two variable nodes and one check node of $\mathcal{B}(3, 6, 8)$ and connect them to one selected check node of $\mathcal{B}(4, 6, 8)$ at the same position (shown in dash lines). Then, we break the two edges between these two selected variable nodes and one selected check node of $\mathcal{B}(4, 6, 8)$, and connect them to one check node of $\mathcal{B}(3, 6, 8)$ (shown in red dash lines). The base matrix can be obtained by (2).

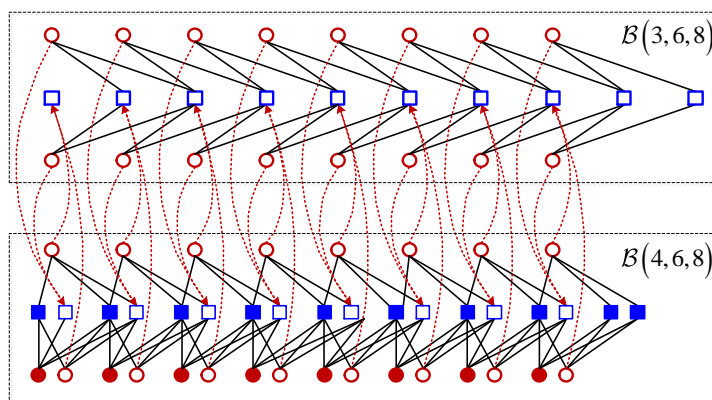


Figure 2. The connection structure of two chains $\mathcal{B}(3, 6, 8)$ and $\mathcal{B}(4, 6, 8)$ in parallel.

3. Graph Evolution under Peeling Decoder

Following the PD working principle, we proposed a general finite-length analysis based on the base matrix to predict the error probabilities. Without a loss of generality, we considered the PC-MS-C-LDPC codes transmitted over the BEC with erasure probability ϵ under PD.

3.1. Denotations of DDs

As described in Section 2, denote the check node at the i th row of the base matrix \mathbf{B} as Type- i check node and the variable node at j th column as Type- j variable node, where $1 \leq i \leq m$ and $1 \leq j \leq n$. The number of each type of check/variable node is M . The degrees of Type- i check node and Type- j variable node are $dc_i = \sum_{j=1}^n B(i, j)$ and $dv_j = \sum_{i=1}^m B(i, j)$, respectively, where $B(i, j)$ is the entry at the i th row and j th column of \mathbf{B} .

To describe the graph evolution under PD, let l denote time and let it be normalized by $\tau = l/M$. Since the PD peels off one variable node at each iteration from the decoding

graph and there are ϵMn variable nodes in total at the start of the decoding process, ϵMn iterations are required on average in order to reach the empty graph and realize successful decoding, i.e., $\tau \in [0, \epsilon n]$.

Let $V_j(l)$ denote the number of the remaining Type- j variable nodes and $R_{s,i}(l)$ denote the number of edges connected to Type- i check nodes with degree s at time l , where $1 \leq j \leq n, 1 \leq s \leq dc_i, 1 \leq i \leq m$. $V_j(l)$ and $R_{s,i}(l)$ are defined as the DDs at time l .

Denote $v_j(\tau)$ and $r_{s,i}(\tau)$ as the normalized versions, where they can be obtained by normalizing $V_j(l)$ and $R_{s,i}(l)$ with M at normalized time τ .

$$v_j(\tau) = \frac{V_j(l)}{M}, \quad r_{s,i}(\tau) = \frac{R_{s,i}(l)}{M}. \tag{3}$$

Since the expected values are required during the graph evolution, denote the expected values of $v_j(\tau)$ and $r_{s,i}(\tau)$ as $\hat{v}_j(\tau) = \mathbb{E}[v_j(\tau)]$ and $\hat{r}_{s,i} = \mathbb{E}[r_{s,i}(\tau)]$.

3.2. Mean Graph Evolution

Initialization Step: The number of the correctly received Type- j variable nodes is $(1 - \epsilon)M$ after passing the BEC with erasure probability ϵ . At $l = 0$, PD removes all these correctly received variable nodes along with their attached edges from the decoding graph. The expected number of Type- j variable nodes is given by

$$\mathbb{E}[V_j(0)] = \epsilon M, 1 \leq j \leq n, \tag{4}$$

and the normalized version is $\hat{v}_j(0) = \mathbb{E}[v_j(0)] = \mathbb{E}[V_j(0)]/M$.

At $l = 0$, since PD removed all correctly received variable nodes along with their attached edges from the decoding graph, the check nodes on the residual graph will lose edges and the degree will be decreased. If the degree of a Type- i check node with degree dc_i is decreased to s , it means that a total of $dc_i - s$ edges of this check node are connected to the correctly received variable nodes. Therefore, the expected value of $R_{s,i}(0)$ is

$$\mathbb{E}[R_{s,i}(0)] = sM \binom{dc_i}{s} \epsilon^s (1 - \epsilon)^{dc_i - s}, 1 \leq i \leq m. \tag{5}$$

At the right side of Equation (5), the former part sM represents the total number of edges connected to Type- i check nodes with degree s and the latter part is the probability that the degree of a Type- i check node is s . The normalized version is $\hat{r}_{s,i}(0) = \mathbb{E}[r_{s,i}(0)] = \mathbb{E}[R_{s,i}(0)]/M$.

Evolution Step: At time l , one degree-one check node is randomly selected and then removed along with its connected variable node and all connected edges. A new residual graph is produced.

The mean graph evolution is determined by the expected values of $r_{r,i}(\tau)$ and $v_j(\tau)$, which can be obtained by solving the following differential equations:

$$\frac{\partial \hat{v}_j(\tau)}{\partial \tau} = \frac{\partial \mathbb{E}[v_j(\tau)]}{\partial \tau} = \frac{\partial \mathbb{E}[V_j(l)]}{\partial l} = \mathbb{E}[V_j(l+1) - V_j(l) | R_{q,p}(l), V_t(l), \forall q, p, t], \tag{6}$$

$$\frac{\partial \hat{r}_{s,i}(\tau)}{\partial \tau} = \frac{\partial \mathbb{E}[r_{s,i}(\tau)]}{\partial \tau} = \frac{\partial \mathbb{E}[R_{s,i}(l)]}{\partial l} = \mathbb{E}[R_{s,i}(l+1) - R_{s,i}(l) | R_{q,p}(l), V_t(l), \forall q, p, t], \tag{7}$$

where $1 \leq i \leq m, 1 \leq s \leq dc_i, 1 \leq j \leq n$, and they have unique solutions. As pointed out in [13], when $M \rightarrow \infty$, any samples of $v_j(\tau)$ and $r_{s,i}(\tau)$ follow $\hat{v}_j(\tau)$ and $\hat{r}_{s,i}(\tau)$ closely. The solutions of Equations (6) and (7) are given as follows.

Solution of Equation (6): At time l , assume the removed degree-one check node to be a Type- c check node, which is chosen randomly from all degree-one check nodes on the residual graph with uniform probability $p_c(l)$.

$$p_c(l) = \frac{R_{1,c}(l)}{\sum_{i=1}^m R_{1,i}(l)}. \tag{8}$$

Then, the variable node connected to this removed Type- c degree-one check will be removed. Denote the probability that this variable node is a Type- j variable node as $\lambda_{c,j}(l)$.

$$\lambda_{c,j}(l) = \frac{V_j(l)B_{pc}(c,j)}{\sum_{u=1}^n V_u(l)B_{pc}(c,u)}, \tag{9}$$

where the denominator represents the total number of variable nodes connected to Type- c check nodes.

Since a Type- j variable node with probability $\lambda_{c,j}(l)$ is removed, the variation for the variable nodes is

$$\mathbb{E}[V_j(l+1) - V_j(l)|p_c(l)] = -\lambda_{c,j}(l). \tag{10}$$

Solution of Equation (7): At time l , if this Type- j variable node connected to the removed Type- c degree-one check node is removed, all the attached edges will be deleted, which results in every connected check node losing one edge. Then, denote the probability that a Type- i check node loses one edge as $\zeta_{c,i}(l)$ and, specifically, $\zeta_{c,c}(l) = 1$.

$$\zeta_{c,i}(l) = \sum_{j=1}^n B_{pc}(i,j)\lambda_{c,j}(l). \tag{11}$$

Since a Type- c check node with degree one and an edge connected to it are removed, the variation in the check nodes for the case $i = c$ can be calculated as

$$\mathbb{E}[R_{s,c}(l+1) - R_{s,c}(l)|p_c(l)] = \begin{cases} -1, & s = 1 \\ 0, & \text{others} \end{cases}. \tag{12}$$

For the case $i \neq c$, the graph loses one edge with probability $\zeta_{c,i}(l)$. This lost edge is connected to a degree- s check node with probability

$$\frac{R_{s,i}(l)}{\sum_{q=1}^{dc_i} R_{q,i}(l)}. \tag{13}$$

As a result, the graph will lose s edges of Type- i check nodes with degree s and gain $s - 1$ edges of the Type- i check nodes with degree $s - 1$. The expected graph evolution is

$$\mathbb{E}[R_{s,i}(l+1) - R_{s,i}(l)|p_c(l)] = s\zeta_{c,i}(l) \frac{R_{s+1,i}(l) - R_{s,i}(l)}{\sum_{q=1}^{dc_i} R_{q,i}(l)}, \tag{14}$$

where $R_{s+1,i}(l) = 0$ for $s = dc_i$.

Since the fraction of degree-one check nodes on the graph determines the successful decoding, we only consider the variation in the degree-one check nodes. In conclusion, by using Equations (10) and (14), the expected graph evolutions can be derived.

On the variable node side, we can obtain

$$\mathbb{E}[V_j(l+1) - V_j(l)] = -\sum_{p=1}^m \lambda_{c,j}(l)p_c(l). \tag{15}$$

On the degree-one check node side, we can obtain

$$\mathbb{E}[R_{1,i}(l+1) - R_{1,i}(l)] = -p_c(l) + \left(\frac{R_{2,i}(l) - R_{1,i}(l)}{\sum_{q=1}^{d_{c_1}} R_{q,i}(l)}\right)(\mathbf{p}(l)\boldsymbol{\xi}_i^T(l) - p_i(l)), \quad (16)$$

where $\mathbf{p}(l) = [p_1(l) p_2(l) \cdots p_m(l)]$ and $\boldsymbol{\xi}_i(l) = [\xi_{1,i}(l) \xi_{2,i}(l) \cdots \xi_{m,i}(l)]$.

Decoding Criteria: To ensure the successful decoding, the total number of degree-one check nodes must be kept positive until the whole graph is peeled off. Therefore, the BP threshold can be defined as the maximum ϵ to ensure that the mean fraction of degree-one check nodes $\hat{r}_1(\tau)$ is strictly positive for any $\tau \in [0, \epsilon n]$.

$$\hat{r}_1(\tau) = \sum_{i=1}^m \hat{r}_{1,i}(\tau) = \sum_{i=1}^m \mathbb{E}[r_{1,i}(\tau)]. \quad (17)$$

3.3. Variance Estimation

After describing the expected evolution of the random process $r_1(\tau)$, we need to compute the variance of $r_1(\tau)$ for estimating the error probability of the PC-MSC-LDPC codes. As pointed out in [12], for sufficiently large M , the distribution of $r_1(\tau)$ can converge to a Gaussian distribution with mean $\hat{r}_1(\tau)$ in Equation (17) and variance $\delta_1(\tau)$ in Equation (18). Therefore, we can estimate the variance empirically around the mean value using a large set of samples of $r_1(\tau)$ at time τ .

$$\text{Var}[r_1(\tau)] = \mathbb{E}[(r_1(\tau) - \hat{r}_1(\tau))^2] = \frac{1}{M} \sum_{i=1}^m \sum_{t=1}^m \delta_{1i,1t} = \frac{\delta_1(\tau)}{M}. \quad (18)$$

4. Performance Analysis and Results

We first show the mean value $\hat{r}_1(\tau)$ for the PC-MSC-LDPC code ensemble $\mathcal{P}(3, 6, 3, 9, 8)$ with different ϵ and $M = 500$ in Figure 3a. For $\epsilon = 0.37$, we also plot a set of 10 simulated decoding trajectories to confirm that they indeed concentrate around the predicted mean evolution. From Figure 3a, we can clearly observe that the local minimum decreases as ϵ increases and will be close to zero at the BP threshold $\epsilon^* = 0.4064$ computed by the density evolution in [11], which also coincides with the definition of the BP threshold in Section 3.2. The time at this local minimum was defined as the *critical point* and denoted as τ^* .

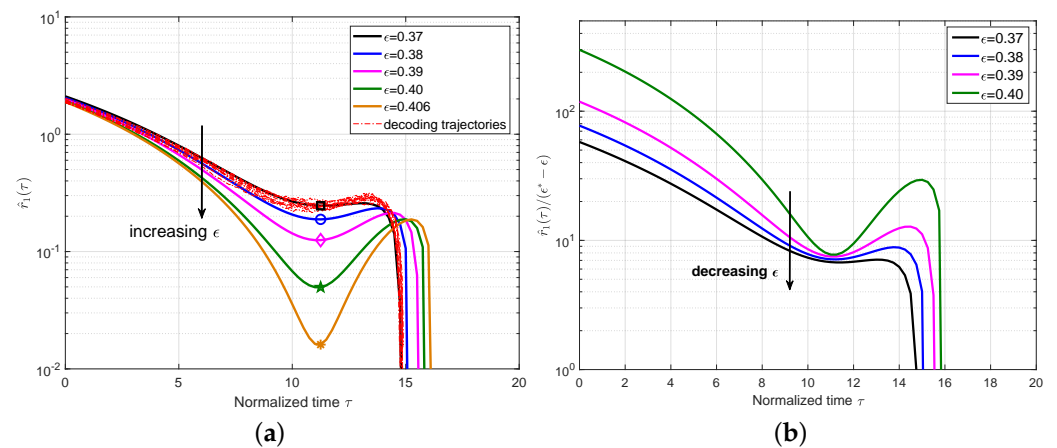


Figure 3. (a) Plot $\hat{r}_1(\tau)$ for the ensemble $\mathcal{P}(3, 6, 3, 9, 8)$ with $M = 500$ and different ϵ . The decoding trajectories are included for $\epsilon = 0.37$. The symbols in each line correspond to the critical points for each ϵ . (b) Plot $\hat{r}_1(\tau)/(\epsilon^* - \epsilon)$ with the threshold $\epsilon^* = 0.4064$.

In order to associate the error probability with ϵ , we considered a first-order Taylor expansion around the threshold ϵ^* for $\hat{r}_1(\tau^*)$ at τ^* .

$$\hat{r}_1(\tau^*) \approx \hat{r}_1(\tau^*)|_{\epsilon^*} + \gamma(\epsilon^* - \epsilon) + \mathcal{O}((\epsilon^* - \epsilon)^2). \tag{19}$$

We plot $\hat{r}_1(\tau)/(\epsilon^* - \epsilon)$ for the ensemble $\mathcal{P}(3, 6, 3, 9, 8)$ in Figure 3b. The approximately constant values can be observed for different ϵ at τ^* , which indicates that it is reasonable to remove the high-order components in Equation (19). Using $\hat{r}_1(\tau^*)|_{\epsilon^* = 0}$, we can obtain $\gamma \approx \hat{r}_1(\tau^*)/(\epsilon^* - \epsilon)$, which can characterize the expected number of degree-one check nodes for a given ϵ at the critical point τ^* .

Then, we extended the analysis to the case of $L = 50$ and plot the mean value $\hat{r}_1(\tau)$ of the ensemble $\mathcal{P}(3, 6, 3, 9, 50)$ with $M = 500$ in Figure 4a, which also includes a set of 10 simulated decoding trajectories for $\epsilon = 0.35$ to verify the accuracy of the mean evolution. Different from the ensemble $\mathcal{P}(3, 6, 3, 9, 8)$, we can observe that the local minimum appears not only at one critical point but also for a period of time. This phase is named as a *steady phase* and the critical point τ^* can be any time during this phase. At this steady phase, the expected number of degree-one check nodes is almost constant, which confirms the conclusion in [1] that the decoding waves travel away from the boundaries toward the center of the coupled chain at a constant speed. We also plot $\hat{r}_1(\tau)/(\epsilon^* - \epsilon)$ for the ensemble $\mathcal{P}(3, 6, 3, 9, 50)$ in Figure 4b. The approximately constant values can be observed for different ϵ at τ^* .

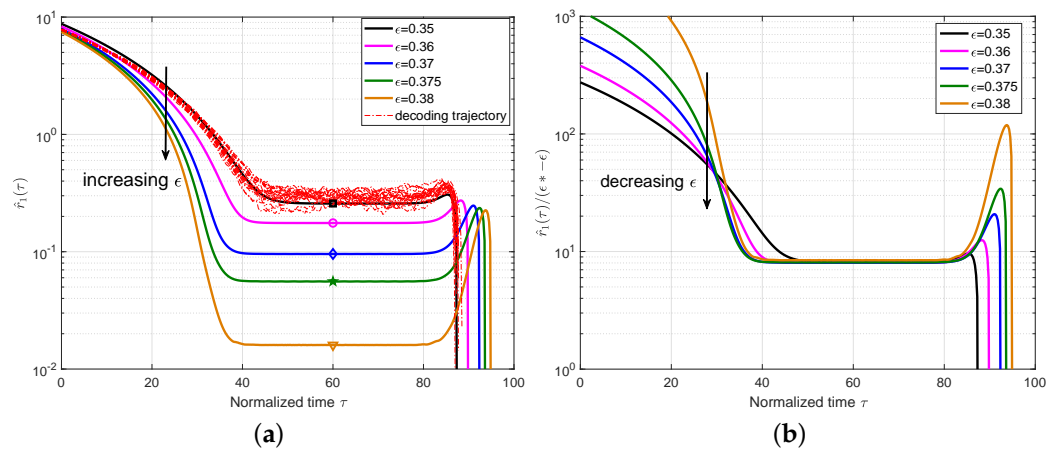


Figure 4. (a) Plot $\hat{r}_1(\tau)$ for the ensemble $\mathcal{P}(3, 6, 3, 9, 50)$ with $M = 500$ and different ϵ . The decoding trajectories are included for $\epsilon = 0.35$. The symbols in each line correspond to the critical points for each ϵ . (b) Plot $\hat{r}_1(\tau)/(\epsilon^* - \epsilon)$ with the threshold $\epsilon^* = 0.3819$.

As pointed out in [12], the fraction of degree-one check nodes at τ^* dominates the code performance, so we only need to estimate the variance $\text{Var}[r_1(\tau)]$ at τ^* . Specifically, we produced a set of 10^2 samples of $r_1(\tau)$ for each ϵ by using one randomly generated code from the PC-MSC-LDPC ensemble under the PD. For the ensemble $\mathcal{P}(3, 6, 3, 9, 8)$, the estimated $\delta_1(\tau^*)$ for different ϵ and M is listed in Table 1.

Table 1. The values of $\delta_1(\tau^*)$ for the ensemble $\mathcal{P}(3, 6, 3, 9, 8)$.

$\delta_1(\tau^*)$	$\epsilon = 0.385$	$\epsilon = 0.39$	$\epsilon = 0.395$	$\epsilon = 0.40$
$M = 200$	0.3526	0.3222	0.2548	0.1509
$M = 500$	0.5527	0.4796	0.4258	0.3031
$\delta_1(\tau^*)$	$\epsilon = 0.393$	$\epsilon = 0.394$	$\epsilon = 0.395$	$\epsilon = 0.40$
$M = 1000$	0.6174	0.6019	0.5978	0.4991

Since $r_1(\tau)$ converges to a Gaussian distribution [13], the error probability at τ^* can be obtained.

$$P = Q\left(\frac{\hat{r}_1(\tau^*)}{\sqrt{\text{Var}[\hat{r}_1(\tau^*)]}}\right) = Q\left(\frac{\gamma(\epsilon^* - \epsilon)}{\sqrt{\delta_1(\tau^*)/M}}\right). \tag{20}$$

Using Equation (20), we show the predicted error probabilities (dash lines) for different PC-MSC-LDPC codes in Figure 5 and also plot the simulated ones (solid lines) for comparisons. The results show that the predicted error probabilities are consistent with the simulated error probabilities but small gaps can be observed for relatively small M . They are caused due to two reasons. One is the deviation between the mean value $\hat{r}_1(\tau)$ and the true mean value of the process $r_1(\tau)$, but it deviates from the true mean value of the process $r_1(\tau)$ by less than $M^{-1/6}$. As $M \rightarrow \infty$, any sample of $r_1(\tau)$ follows $\hat{r}_1(\tau)$ closely. The other is the negligence of the decoding failure at $\tau \neq \tau^*$ caused by the small cycles or stopping sets in the graph. This effect is more severe for smaller values of M . However, since the SC-LDPC code ensemble has a linear growth of minimum distance with block length nM , the codes with small cycles or low-weight stopping sets can hardly be found for sufficiently large M . Therefore, when M increases to a few thousands, the effects on the prediction accuracy will be small enough to be ignored.

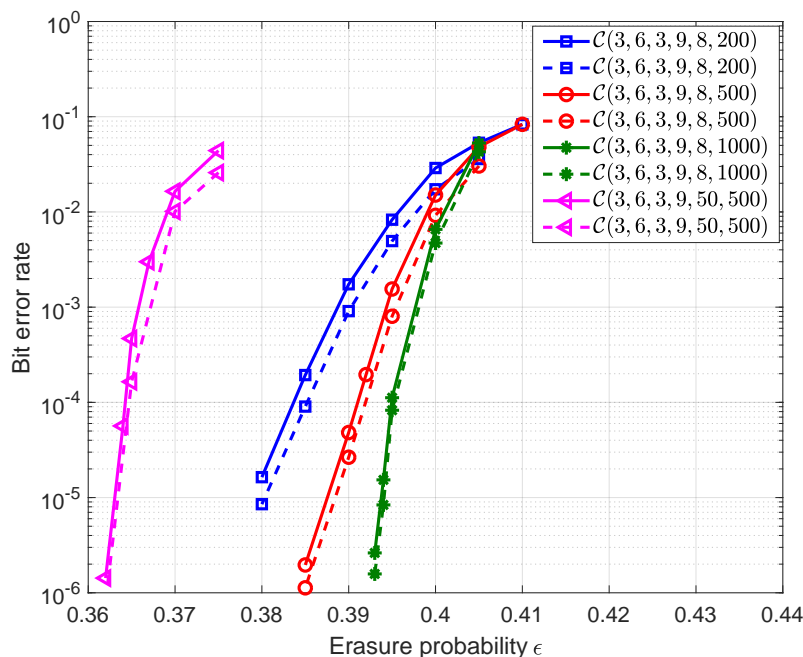


Figure 5. Simulated error probabilities (solid lines) and predicted error probabilities using the expression in Equation (20) (dash lines) for the different PC-MSC-LDPC codes. The rate and the codelength of $\mathcal{C}(3, 6, 3, 9, 50, 500)$ are 0.584 and 125,000. The rate of the other three codes is 0.5 and the codelengths are 8000, 20,000 and 40,000 respectively.

Next, we extended the analysis to the case of connecting three different chains and considered the ensemble $\mathcal{P}(3, 6, 3, 9, 3, 12, 15)$. The mean value $\hat{r}_1(\tau)$ with $M = 200$ and different ϵ is plotted in Figure 6. Similar results can be observed that, when approaching the threshold $\epsilon^* = 0.3114$, the $\hat{r}_1(\tau^*)$ values gradually decrease to approximately zero. In Table 2, the $\delta_1(\tau^*)$ values are calculated for the codes generated from the ensemble $\mathcal{P}(3, 6, 3, 9, 3, 12, 15)$. The predicted error probabilities and the simulated ones for these codes are shown in Figure 7. The comparison results show that these two curves can match well and that the accuracy of the prediction gets better as M becomes larger. For comparison, the finite length performance bounds obtained by Equation (290) in [16] for different codelengths along with the BP threshold and Shannon limit are also plotted in

Figure 7, from which we can observe that the gaps between the error probability curves and performance bounds are almost equal to the gap between the BP threshold and Shannon limit. It is known that the finite-length performance is consistent with the BP threshold. By increasing the coupling length, the BP threshold can be improved to be close to the Shannon limit, which can result in the error probabilities approaching the finite-length performance bounds.

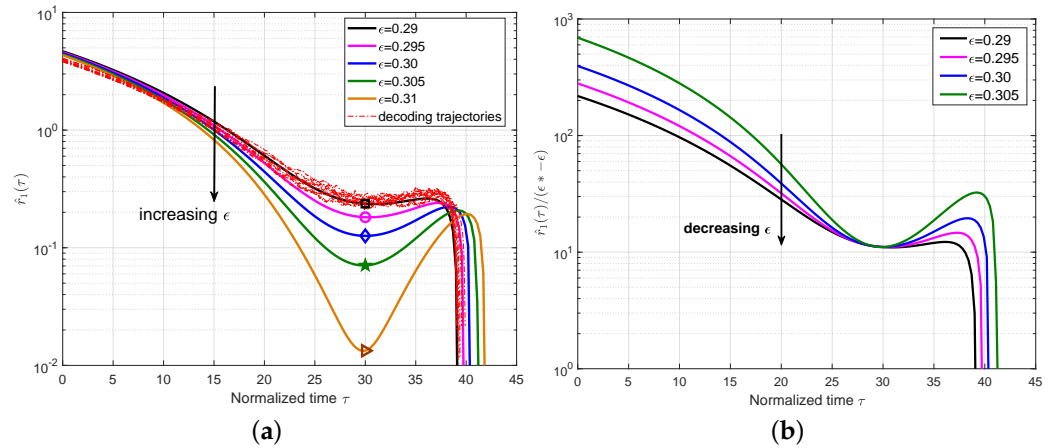


Figure 6. (a) Plot $\hat{r}_1(\tau)$ for the ensemble $\mathcal{P}(3, 6, 3, 9, 3, 12, 15)$ with $M = 200$ and ϵ from 0.29 to 0.31. A set of 10 empirical trajectories for $\epsilon = 0.29$ are included. The symbols in each line correspond to the critical points for each ϵ . (b) Plot $\hat{r}_1(\tau)/(\epsilon^* - \epsilon)$ with the threshold $\epsilon^* = 0.3114$.

Table 2. The values of $\delta_1(\tau^*)$ for the ensemble $\mathcal{P}(3, 6, 3, 9, 3, 12, 15)$.

$\delta_1(\tau^*)$	$\epsilon = 0.295$	$\epsilon = 0.30$	$\epsilon = 0.305$	$\epsilon = 0.31$
$M = 200$	0.4777	0.4146	0.1988	0.0094
$M = 500$	0.7972	0.7711	0.5394	0.0275

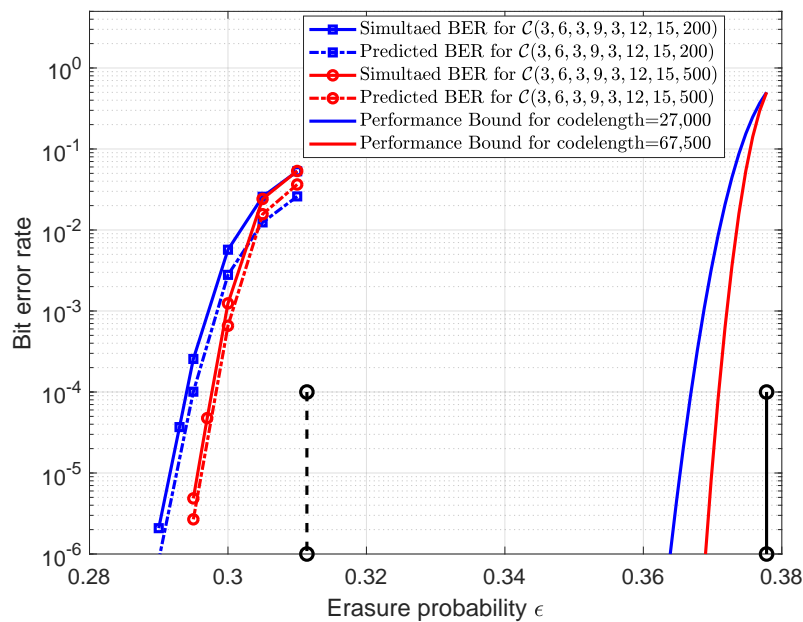


Figure 7. Simulated error probabilities (solid lines) and predicted error probabilities (dash lines) for the codes $\mathcal{C}(3, 6, 3, 9, 3, 12, 15, 200)$ and $\mathcal{C}(3, 6, 3, 9, 3, 12, 15, 500)$. The rate is 0.6222 and the code lengths are 27,000 and 67,500 respectively. The solid lines from left to right are the performance bounds for code length = 27,000 and code length = 67,500 in order. BP threshold (vertical dash line) and Shannon limit (vertical solid line) are also included.

Finally, we applied the analysis to the conventional SC-LDPC code $\mathcal{C}(3, 6, 8, 700)$. The mean value $\hat{r}_1(\tau)$ with different ϵ and plot $\hat{r}_1(\tau)/(\epsilon^* - \epsilon)$ is shown in Figure 8. It can be observed that the local minimum values decrease when increasing ϵ to the BP threshold $\epsilon^* = 0.5212$ and that they are small enough to be close to zero at $\epsilon = 0.52$. In addition, the approximately constant γ can be observed at the critical point as expected. Following similar steps, we computed and list the $\delta_1(\tau^*)$ values for different ϵ in Table 3. Using Equation (20), the predicted error probability for $\mathcal{C}(3, 6, 8, 700)$ can be plotted in Figure 9. It was shown that the predicted performance using this error probability expression can fit well with the simulated performance, which can demonstrate the effectiveness of the proposed finite-length analysis for other SC-LDPC code ensembles.

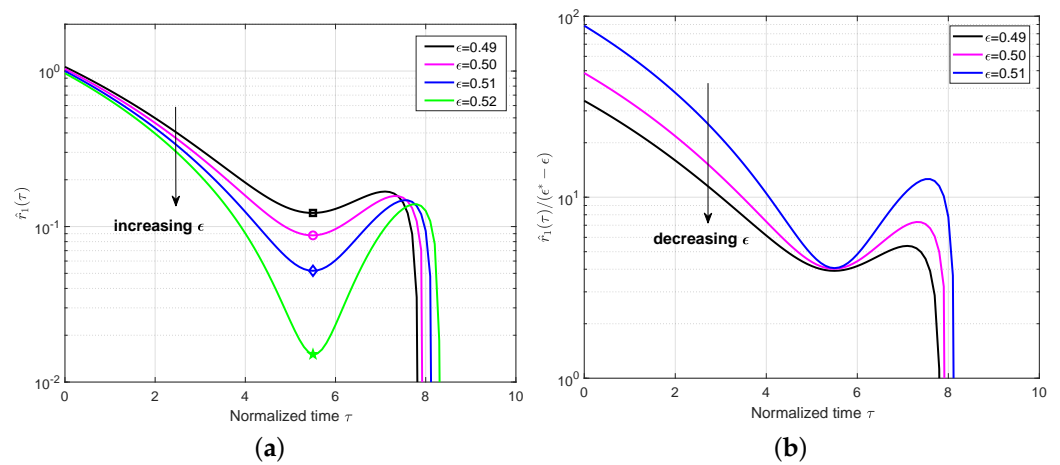


Figure 8. (a) Plot $\hat{r}_1(\tau)$ for the code $\mathcal{C}(3, 6, 8, 700)$, where ϵ varies from 0.49 to 0.52. The symbols in each line correspond to the critical points for each ϵ . (b) Plot $\hat{r}_1(\tau)/(\epsilon^* - \epsilon)$ with the BP threshold $\epsilon^* = 0.5212$.

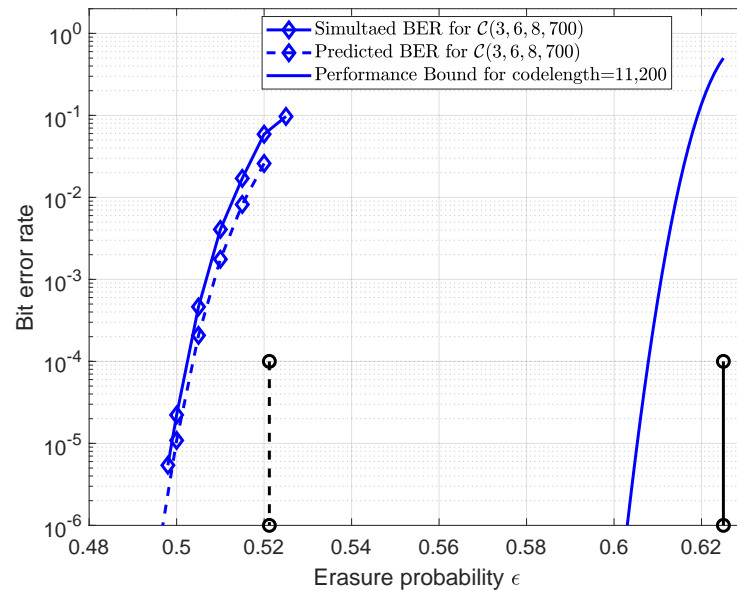


Figure 9. Simulated error probability and predicted error probability for the code $\mathcal{C}(3, 6, 8, 700)$. The rate is 0.375 and the codelength is 11,200. The blue solid line is the performance bound for codelength=11,200. BP threshold (vertical dash line) and Shannon limit (vertical solid line) are also included.

Table 3. The values of $\delta_1(\tau^*)$ for the code $\mathcal{C}(3, 6, 8, 700)$.

$\delta_1(\tau^*)$	$\epsilon = 0.52$	$\epsilon = 0.515$	$\epsilon = 0.51$	$\epsilon = 0.505$	$\epsilon = 0.50$	$\epsilon = 0.495$
$M = 700$	0.0422	0.1372	0.1694	0.2432	0.2825	0.3069

5. Conclusions

This paper proposed a general finite-length analysis from the perspective of the base matrix over the BEC and applied it to the PC-MSC-LDPC code ensembles to verify the effectiveness. The results show that the predicted error probabilities obtained by using the derived error probability expression are consistent with the simulated error probabilities and that the accuracy of the prediction will be further improved when M increases. Since the proposed analysis is performed on the base matrix, it can be generalized to any spatially coupled ensembles defined by the base matrix, such as SC-RA codes and spatially coupled generalized LDPC codes. Finite-length performance analysis provides a useful engineering tool for practical code design and analyzing the effects of the code parameters on the performances.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, S.S.; validation, S.S.; formal analysis, Y.L. and S.S.; writing—original draft preparation, Y.L. and S.S.; writing—review and editing, Y.L. and Y.Z.; funding acquisition, B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. U19B2015, No. 62271386, No. 61801371).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SC-LDPC	Spatially coupled low-density parity check
BP	Belief propagation
MAP	Maximum a posterior
BEC	Binary erasure channel
SC-RA	spatially coupled repeat-accumulate
SC-TCs	spatially coupled turbo codes
PC-MSC-LDPC	Parallely connecting multiple SC-LDPC
PD	Peeling decoder
DDs	Degree distributions

References

- Kudekar, S.; Richardson, T.J.; Urbanke, R.L. Threshold saturation via spatial coupling: why convolutional LDPC ensembles perform so well over the BEC. *IEEE Trans. Inf. Theory* **2011**, *57*, 803–834. [[CrossRef](#)]
- Schmalen, L.; Aref, V. Spatially Coupled LDPC Codes with Non-uniform Coupling for Improved Decoding Speed. In Proceedings of the IEEE Information Theory Workshop (ITW), Visby, Sweden, 25–28 August 2019; pp. 1–5.
- Naseri, S.; Banihashemi, A.H. Construction of Time Invariant Spatially Coupled LDPC Codes Free of Small Trapping Sets. *IEEE Trans. Commun.* **2021**, *69*, 3485–3501. [[CrossRef](#)]
- Amiri, B.; Reiszadeh, A.; Kliewer, J.; Dolecek, L. Optimized array-based spatially-coupled LDPC Codes: An absorbing set approach. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Hong Kong, China, 14–19 June 2015; pp. 51–55.
- Johnson, S.; Lechner, G. Spatially Coupled Repeat-Accumulate Codes. *IEEE Commun. Lett.* **2013**, *17*, 373–376. [[CrossRef](#)]
- Yang, C.; Zhao, S.; Ma, X. Hybrid Coupled Serially Concatenated Codes. *IEEE Trans. Commun.* **2022**, *70*, 4301–4315. [[CrossRef](#)]
- Sakata, K.; Kasai, K.; Sakaniwa, K. Spatially-coupled precoded rateless codes with bounded degree achieve the capacity of BEC under BP decoding. In Proceedings of the IEEE International Symposium on Information Theory, Honolulu, HI, USA, 29 June–4 July 2014; pp. 521–525.

8. Truhachev, D.; Mitchell, D.G.M.; Lentmaier, M.; Costello, D.J. New codes on graphs constructed by connecting spatially coupled chains. In Proceedings of the Information Theory and Applications Workshop, San Diego, CA, USA, 5–10 February 2012; pp. 392–397.
9. Olmos, P.M.; Mitchell, D.G.M.; Truhachev, D.; Costello, D.J. Continuous transmission of spatially coupled LDPC code chains. *IEEE Trans. Commun.* **2017**, *65*, 5097–5109. [[CrossRef](#)]
10. Truhachev, D.; Mitchell, D.G.M.; Lentmaier, M. Code design based on connecting spatially coupled graph chains. *IEEE Trans. Inf. Theory* **2019**, *65*, 5604–5617. [[CrossRef](#)]
11. Liu, Y.; Li, Y.; Chi, Y. Spatially coupled LDPC codes constructed by parallelly connecting multiple chains. *IEEE Commun. Lett.* **2015**, *19*, 1472–1475. [[CrossRef](#)]
12. Olmos, P.M.; Urbanke, R. A scaling law to predict the finite-length performance of spatially-coupled LDPC codes. *IEEE Trans. Inf. Theory* **2015**, *61*, 3164–3184. [[CrossRef](#)]
13. Amraoui, A.; Montanari, A.; Richardson, T.; Urbanke, R. Finite-length scaling for iteratively decoded LDPC ensembles. *IEEE Trans. Inf. Theory* **2009**, *55*, 473–498. [[CrossRef](#)]
14. Olmos, P.M.; Mitchell, D.G.M.; Truhachev, D.; Costello, D.J. A finite length performance analysis of LDPC codes constructed by connecting spatially coupled chains. In Proceedings of the IEEE Information Theory Workshop (ITW), Seville, Spain, 9–13 September 2013; pp. 1–5.
15. Mitchell, D.G.M.; Lentmaier, M.; Costello, D.J. Spatially coupled LDPC codes constructed from protographs. *IEEE Trans. Inf. Theory* **2015**, *62*, 4866–4889. [[CrossRef](#)]
16. Polyanskiy, Y.; Vincent, P.; Sergio, V. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory* **2010**, *56*, 2307–2350. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.