IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 9, 2021, accepted August 21, 2021, date of publication August 24, 2021, date of current version September 1, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3107584*

# Optimizing Code Parameters of Finite-Length SC-LDPC Codes Using the Scaling Law

**HEE-YOUL KWAK[1], JAE-WON KIM[2], (Member, IEEE), AND JONG-SEON NO[1], (Fellow, IEEE)**
[1]Department of Electrical and Computer Engineering, INMC, Seoul National University, Seoul 08826, South Korea
[2]Department of Electronic Engineering, Engineering Research Institute (ERI), Gyeongsang National University, Jinju 52828, South Korea

Corresponding author: Jae-Won Kim (jaewon07.kim@gnu.ac.kr)

This work was supported by Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) under Grant 2021-0-00400, Development of Highly Efficient PQC Security and Performance Verification for Constrained Devices.

**ABSTRACT** In this paper, we optimize code parameters of finite-length spatially coupled low-density parity-check (SC-LDPC) codes, represented by a set of code parameters $(l, r, w, L, M)$. Although the finite-length scaling behavior of SC-LDPC codes was studied in the existing literature, the previous works impose a constraint such that the coupling width $w$ is equal to the variable node degree $l$ and they do not focus on optimizing the code parameters for given code and decoder specifications such as the code rate, frame size, and decoding complexity. In order to optimize the code parameters with the target specifications, we first extend the scaling law of SC-LDPC codes without the constraint $w = l$. Using the scaling law formulated with a new variable $w$, we show that the coupling width $w$ directly affects the slope of the performance curve and performance comparisons are given to investigate trade-offs inherent in the code parameters. It is shown that there are trade-offs for the code parameters in the perspective of the asymptotic performance limit, code rate, and scaling behaviors. In addition, the scaling law allows us to find the optimal code parameter set showing the best finite-length performance. Interestingly, the optimal code parameter set $(l, r, w)$ varies depending on the coupling length $L$ and uncoupled code length $M$ that determine the code and decoder specifications, which means there is no specific code parameter set prevailing over different kinds of applications. Finally, we illustrate this result using the investigated trade-offs on the code parameters, which gives us useful insight on how to choose the code parameters.

**INDEX TERMS** Coupling width, finite-length performance, low-density parity-check (LDPC) code, scaling law, spatially coupled LDPC (SC-LDPC) code.

## I. INTRODUCTION

Spatially coupled low-density parity-check (SC-LDPC) codes are a special class of low-density parity-check (LDPC) codes, where $L$ multiple $(l, r)$ regular LDPC codes of variable node degree $l$ and check node degree $r$ are coupled in a chain form [1]–[3]. SC-LDPC codes have attracted significant attention due to their appealing properties as follows. First, SC-LDPC codes are known to show capacity approaching decoding performance under low-complexity belief propagation (BP) decoding over the binary erasure channel (BEC) as the degrees $(l, r)$ and coupling length $L$ increase [4], [5]. Second, this capacity approaching property is preserved universally over general binary memoryless channels [6]. While conventional block LDPC codes require a troublesome optimization for the target channel to achieve capacity-approaching performance [7], a single SC-LDPC ensemble can be applicable to various channels with superior BP decoding performance [8]–[11]. Third, the minimum distance growth property of uncoupled codes still maintains after coupling [12], which means the SC-LDPC codes show low error floors [13].

In this paper, we consider $(l, r, w, L, M)$ finite-length SC-LDPC codes, where $w$ is the coupling width that determines the number of consecutive positions to be coupled and $M$ is the number of variable nodes at each position [4]. Based on comprehensive asymptotic analysis, researches on the code design of $(l, r, w, L, M)$ SC-LDPC codes having superior finite-length performance have been followed [13]–[21]. It is desirable to construct practically good codes showing superior finite-length performances for a given code rate and code length. The code rate and BP threshold are important criteria to design capacity approaching codes in terms of asymptotic perspective. Thus, SC-LDPC codes with sufficiently large $L$

The associate editor coordinating the review of this manuscript and approving it for publication was Xueqin Jiang.

and degrees $(l, r)$ are known to be the best choice because the inherent rate-loss of SC-LDPC codes approaches to zero as $L$ grows and the BP threshold approaches the Shannon limit as the degrees $(l, r)$ increase. For example, the threshold gap to the Shannon limit of SC-LDPC codes is almost zero with sufficiently large $L$ and degrees $(l, r) = (5, 10)$.

However, the scaling law, which is an analytical expression of the frame error rate (FER) of an ensemble, revealed two weak points of such code parameters with large $L$ and high degrees $(l, r)$ [14]–[16]; i) the finite-length performance is degraded linearly as $L$ grows and ii) high degrees $(l, r)$ result in the slow falling slope of the FER curve. The scaling law of SC-LDPC codes is a function of $L$, $M$, BP threshold, and so-called scaling parameters $(\gamma, \delta, \theta)$ that depend on $(l, r, w)$. Especially, it was shown in [14] that the ratio $\alpha \triangleq \gamma/\sqrt{\delta}$ is an important parameter directly affecting the slope of FER curve. In [14], the authors showed that for the fixed $l/r$ ratio, the value of $\alpha$ tends to decrease as the degrees $(l, r)$ increase. In other words, when determining code parameters, there is a trade-off between the asymptotic property and scaling property. Both properties affect the resulting finite-length performance but which one has a stronger effect on the finite-length performance is not well known up to now. For example, between the $(3, 6, w, L, M)$ SC-LDPC codes and the $(4, 8, w, L, M)$ SC-LDPC codes with the same code rate, which code shows better finite-length performances has not been investigated. Two codes have their own strong point; $(3, 6, w, L, M)$ codes have the better scaling parameter $\alpha$ while $(4, 8, w, L, M)$ codes have the better BP threshold. In addition, the effect of the coupling width $w$ on scaling parameters and resulting finite-length performances is still an open problem. It has been pointed out that the coupling width $w$ is related to the girth property [17], [18], trapping sets [21], and convergence speed [22]. They share an intuitive conclusion that large $w$ has a positive impact on the finite-length performance. However, a research on finding the direct relationship between the scaling behavior and the coupling width $w$ has not been carried out.

As the first contribution of this paper, we first focus on investigating how the parameter $w$ affects the scaling parameters. For the $(l, r, w, L, M)$ SC-LDPC codes, the scaling parameters are given in [14] but they assume $w = l$ for simplifying the scaling analysis. By generalizing the approach in [14], we obtain scaling parameters for various code parameters not restricted on the assumption $w = l$. This scaling behavior analysis shows that the scaling parameter $\alpha$ increases as $w$ grows, resulting in a sharp slope of the FER curve. However, at the cost of the performance improvement, large $w$ values cause a more severe rate-loss. It means that the coupling width $w$ provides a trade-off between the code rate and scaling property. Further, we observe that the scaling parameter $\alpha$ of $(l, r) = (4, 8)$ is noticeably worse than that of $(l, r) = (3, 6)$ for a given coupling width $w$ while the BP threshold of $(l, r) = (4, 8)$ is better than that of $(l, r) = (3, 6)$. We note again that, in the previous work [14], the scaling parameters of the $(3, 6, 3, L, M)$ and $(4, 8, 4, L, M)$ codes are compared with different $w$ values, which means the effect of the coupling width $w$ is not considered independently.

Considering the conflicting behaviors of code parameters $(l, r, w)$ affecting the finite-length performance, it is natural to have a question what is the best code parameter set for given code and decoder specifications such as the code rate, frame size, and decoding complexity. The scaling parameters over various ranges of $(l, r, w)$ without the constraint $w = l$, computed in this paper, allow simple optimization for the code parameters using the scaling law without time-consuming simulations. The second contribution of this paper is to search the best code parameter set $(l, r, w)$ for given $L$ and $M$ using the scaling law, where the difference in the code rates due to the rate-loss is compensated by the random puncturing technique [23]. We assume that the values of $L$ and $M$ are given because they are related to the code rate, frame size, and decoding complexity [24], [25] that are usually determined by external decoder specifications and a target application. For example, some candidate applications of SC-LDPC codes such as optical communications [26] and NAND flash memory [27] can support a long frame size (large $L$) and heavy decoding complexity (large $M$) to show the capacity approaching performance while commercial communication systems with limited hardware resources cannot support those heavy implementation costs. Thus, it is practically important to suggest the optimal family of SC-LDPC codes for given available hardware resources (i.e., for given $L$ and $M$).

The optimization result shows that there is no specific parameter set $(l, r, w)$ exhibiting the best performance for the entire ranges of $L$ and $M$. Instead, we observe the optimal parameter set varies according to the range of $L$ and $M$. As the third contribution of this paper, we provide discussions to illustrate what properties determine the optimal parameter set in a particular region. For example, for small $M$, SC-LDPC codes with the better scaling parameter $\alpha$ such as $(l, r) = (3, 6)$ outperform SC-LDPC codes with better BP thresholds such as $(l, r) = (4, 8)$ while the opposite phenomenon happens for large $M$. From our analysis, it is observed that the BP threshold is an effective measure to predict the performance for large $M$ but the effect of the scaling parameter $\alpha$ is more dominant when predicting the performance for moderate and small $M$. Regarding the value of $L$, for small $L$, SC-LDPC codes with large $w$ take a higher rate-loss compared to SC-LDPC codes with small $w$ and thus the performance is significantly degraded after puncturing to match the code rate. Thus, SC-LDPC codes with small $w$ tend to outperform those with large $w$ in this region for small $L$ while the opposite phenomenon is observed for large $L$. Consequently, we show that the regions of $L$ and $M$ are split into several partitions and that the best code parameters of each partition differ from each other. Finally, we note again that the objective of this paper is not to propose a new class of SC-LDPC codes but to optimize the code parameters of existing SC-LDPC codes. In addition, although the scope of this paper is limited to conventional regular SC-LDPC codes under the BEC, the proposed framework can be extended to other channels such as the additive white Gaussian noise (AWGN) channel and other code structures such as time-invariant SC-LDPC codes [17] and irregular SC-LDPC codes [20] as long as their corresponding scaling law is provided.

This paper is organized as follows. Section II introduces the construction method of the $(l, r, w, L, M)$ SC-LDPC ensemble and the analysis tools for analyzing the decoding performance of SC-LDPC ensembles. In Section III, we provide the generalized expected graph evolution to compute the scaling parameters of various values of $(l, r, w)$ and discuss trade-offs for code parameters between the code rate, BP threshold, and scaling behavior. Section IV presents a series of comparisons to capture the effect of the code parameters and finds the optimal code parameters $(l, r, w)$ for given $L$ and $M$. Finally, the work is concluded in Section V.

## II. CONSTRUCTION AND ANALYSIS OF SC-LDPC CODE ENSEMBLES

### A. CONSTRUCTION OF THE SC-LDPC ENSEMBLE

The $(l, r, w, L, M)$ SC-LDPC ensemble was first introduced in [4] and it was slightly modified in [14]. $(l, r, w, L, M)$ SC-LDPC codes are composed of $M$ variable nodes of degree $l$ located at each of $L$ positions along with $Ml/r \in \mathbb{N}$ check nodes of degree $r$ located at each of $L + w - 1$ positions. Let $L$ and $w$ denote the coupling length and coupling width, respectively. From the definition in [4], each variable node of degree $l$ at position $u$ is connected to randomly chosen $l$ check nodes at position in the range of $\{u, \ldots, u + w - 1\}$. On the contrary, in [14], they consider the more structured edge distribution with the assumption $w = l$ to simplify the analysis of the scaling behavior such that each variable node at position $u$ has exactly one connection to a check node at position $u + i, i = 0, \ldots, l - 1$. For example, for the $(3, 6, 3, L, M)$ SC-LDPC ensemble, the definition in [4] permits that a variable node at position 5 is connected two check nodes at position 5 and one check node at position 6. However, according to the definition in [14], a variable node at position 5 is connected to three check nodes from each of different positions $\{5, 6, 7\}$. Since this paper also utilizes the scaling law, we follow the definition in [14] but generalize the code design method without the constraint $w = l$.

To generate a code from the $(l, r, w, L, M)$ ensemble, we first define the edge spreading type of variable nodes at position $u$ using vector $\underline{x} = (x_1, \ldots, x_w)$, where $x_t$ is the number of edges connected to check nodes at position $u + t - 1$. There are $w$ edge spreading types at each position represented by $\underline{S}_1, \ldots, \underline{S}_w$, where we define $\underline{S}_1$ as

$$\underline{S}_1 = (\overbrace{\lceil l/w\rceil, \ldots, \lceil l/w\rceil}^{l \bmod w}, \overbrace{\lfloor l/w\rfloor, \ldots, \lfloor l/w\rfloor}^{w-(l \bmod w)}).$$

In addition, $\underline{S}_k$ for $2 \leq k \leq w$ is obtained by right circular shifting $\underline{S}_1$ by $k$ times. Then, each variable node in a given position exploits one of $w$ edge spreading types, that is, each group of randomly selected $M/w \in \mathbb{N}$ variable nodes corresponds to $\underline{x} = \underline{S}_k$ for some $k \in \{1, \ldots, w\}$. For example, for $l = 4$ and $w = 3$, there are $w$ edge spreading types determining the edge connection and they are represented by $\underline{S}_1 = (2, 1, 1)$, $\underline{S}_2 = (1, 2, 1)$, and $\underline{S}_3 = (1, 1, 2)$. It means that each of $M/3$ variable nodes with $\underline{S}_1 = (2, 1, 1)$ is connected to two check nodes at position $u$, one check node at position $u + 1$, and one check node at position $u + 2$. The detailed construction method is described in [14]. Fig. 1 shows a Tanner graph of the $(4, 8, 3, L, M)$ SC-LDPC ensemble, where the first variable node at position $u$ has the edge spreading type $\underline{S}_1 = (2, 1, 1)$ and the second node has the edge spreading type $\underline{S}_2 = (1, 2, 1)$.

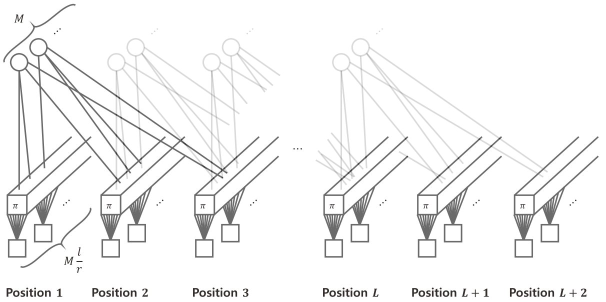


**FIGURE 1. A Tanner graph of $(4, 8, 3, L, M)$ SC-LDPC ensembles. The first and second variable nodes at position $u$ are connected to four check nodes at positions $\{u, u + 1, u + 2\}$ while they follow different edge spreading types. The edge spreading type of the first one is $(2, 1, 1)$ and that of the second one is $(1, 2, 1)$.**

The total number of variable nodes and check nodes in the graph of $(l, r, w, L, M)$ SC-LDPC codes are $LM$ and $(L + w - 1)Ml/r$, respectively. Thus, the code length (frame size) is $LM$ and the code rate of the $(l, r, w, L, M)$ SC-LDPC ensemble is expressed as

$$R_{(l,r,w,L,M)} = \left(1 - \frac{l}{r}\right) - \frac{l}{r}\frac{w-1}{L}, \tag{1}$$

where the first term is the code rate of uncoupled $(l, r)$ regular LDPC codes and the second term corresponds to the rate-loss. We can see that larger $w$ and smaller $L$ induce the higher rate-loss. Note that some check nodes at position $v$ for $v \in \{1, \ldots, w - 1\} \cup \{L + 1, \ldots, L + w - 1\}$ cannot be connected to any variable nodes in the graph and the actual code rate is slightly higher than that in (1) [14]. However, the effect is generally much smaller than other terms in (1) and thus we use (1) for calculating the code rate.

### B. SCALING LAW OF SC-LDPC ENSEMBLES

In this paper, the channel is assumed to be the BEC with erasure probability $\epsilon$. The decoding performance of a code ensemble can be analyzed by the density evolution and the scaling law. The density evolution is used to derive the asymptotic performance limit called the BP threshold $\epsilon^{\mathrm{BP}}$. The density evolution of SC-LDPC codes is described in [4]. While the BP threshold is the performance limit as $M$ goes to infinity, the scaling law predicts the FER of finite-length codes with finite values of $M$. The FER of the $(l, r, w, L, M)$ SC-LDPC ensemble, denoted by $\mathbb{P}_{(l,r,w,L,M)}$, can be estimated by the scaling law as [14]

$$\mathbb{P}_{(l,r,w,L,M)} \approx 1 - \exp\left(-\frac{\epsilon L - \tau^*}{\mu_0(\gamma,\delta,\theta')}\right), \tag{2}$$

where

$$\mu_0(\gamma,\delta,\theta') = \frac{\sqrt{2\pi}}{\theta'}\int_0^{\gamma\sqrt{M/\delta}(\epsilon^{\mathrm{BP}}-\epsilon)} \Phi(z)e^{\frac{1}{2}z^2}dz$$

and $\Phi(z)$ is the cumulative distribution function of the Gaussian distribution, $\tau^*$ is the start point of the steady state, and $(\gamma,\delta,\theta')$ are the scaling parameters. From (2), it is noted that the finite-length performance is a function of $M, L, \epsilon^{\mathrm{BP}}$, and a set of scaling parameters consisting of the mean parameter $\gamma$, the variance parameter $\delta$, and the correlation parameter $\theta'$. The scaling law is derived based on the property that the peeling decoding process of SC-LDPC codes is modeled by the Ornstein-Uhlenbeck process and the scaling parameters $(\gamma,\delta,\theta')$ represent their stochastic quantities [14]. Especially, it is shown in [14] that the FER decays exponentially fast with the ratio $\alpha \triangleq \gamma/\sqrt{\delta}$. Thus, it is practically important to improve the quantity of $\alpha$ to design superior finite-length SC-LDPC codes with a steep slope in the FER curve.

As an extension of the conventional scaling law [14], a refined scaling law was proposed in [16]. The refined scaling law predicts the performance more accurately by replacing the stochastic model with two independent Ornstein-Uhlenbeck processes. They showed that the refined scaling law provides a much better FER prediction compared to the conventional scaling law. We also adopt the refined scaling law for predicting the decoding performance in this paper. The refined scaling law is expressed as

$$\mathbb{P}_{(l,r,w,L,M)} \approx 1 - \left(1 + \frac{\epsilon L - \tau^*}{\mu_0(\frac{\gamma}{2},\frac{\delta}{2},\theta)}\right)\exp\left(-\frac{\epsilon L - \tau^*}{\mu_0(\frac{\gamma}{2},\frac{\delta}{2},\theta)}\right), \tag{3}$$

where correlation parameter $\theta$ is derived from truncated SC-LDPC ensembles for which check nodes at positions in the range $\{L+1,\ldots,L+w-1\}$ are removed [16].

Note that the scaling parameters $(\gamma,\delta,\theta)$ are dependent on $(l,r,w)$ and independent of $(L,M)$. Thus, we use a simple representation, $(l,r,w)$ SC-LDPC codes, when dealing with the scaling parameters. In [14] and [16], they compute the scaling parameters for $(l,r,w)$ SC-LDPC codes with the assumption $w=l$. Our goal is to compute the scaling parameters $(\gamma,\delta,\theta)$ for the $(l,r,w)$ SC-LDPC ensemble without the assumption, which is described in Section III. Since the $(l,r,w)$ SC-LDPC ensemble without any constraints on $w$ obviously follows the same stochastic model as the one proposed in  [16], the remaining issue for deriving the scaling law is to compute their scaling parameters.

## III. SCALING PARAMETERS $(\gamma,\delta,\theta)$ OF SC-LDPC ENSEMBLES

In this section, we obtain the scaling parameters $(\gamma,\delta,\theta)$ of the $(l,r,w)$ SC-LDPC ensemble without the constraint $w=l$. The scaling parameters are obtained by analyzing statistical properties of the number of degree-one check nodes in the remaining graph under peeling decoding [14]. The peeling decoding algorithm sequentially decodes an unknown variable node along with its connected degree-one check node at each iteration and removes the variable and check nodes from the graph. Successful decoding is achieved if at least one degree-one check node remains alive in the graph until all unknown variable nodes are recovered. For iteration $\ell$ of the peeling decoding, let $\tau$ be the normalized iteration by $M$, i.e., $\tau=\ell/M$. Since only one erased variable node is recovered per iteration, we require $\epsilon LM$ iterations on average to recover all erasures, which implies $0\le\tau\le\epsilon L$. Let $r_1(\tau)$ be the number of degree-one check nodes normalized by $M$ at time $\tau$. The scaling parameters $(\gamma,\delta,\theta)$ are related to the stochastic properties of the evolution of $r_1(\tau)$ as [14], [16]

$$\begin{aligned}\mathbb{E}[r_1(\tau)] &\approx \gamma(\epsilon^{\mathrm{BP}}-\epsilon)\\ \mathtt{Var}[r_1(\tau)] &\approx \frac{\delta}{M}\\ \mathbb{E}[r_1(\tau)r_1(\zeta)]-\mathbb{E}[r_1(\tau)]\,\mathbb{E}[r_1(\zeta)] &\approx \frac{\delta}{M}e^{-\theta|\zeta-\tau|},\end{aligned}$$

where these approximations hold on the steady state phase for which $\mathbb{E}[r_1(\tau)]$ remains essentially constant. In [14], it is shown that the evolution of $\overline{r}_1(\tau)\triangleq\mathbb{E}[r_1(\tau)]$ can be obtained by solving a set of differential equations. We now describe an extension of the differential equations for $\overline{r}_1(\tau)$ to the $(l,r,w)$ SC-LDPC ensemble without the constraint $w=l$.

### A. EXPECTED GRAPH EVOLUTION OF THE SC-LDPC ENSEMBLE

A system of coupled differential equations for computing the expected number of degree-one check nodes is obtained as follows. Consider the check nodes at position $v$. Let $\rho_{m,v}$ be the probability that a check node chosen at random from position $v$ is of degree $m$, which is given as

$$\rho_{m,v} = \begin{cases}\binom{r}{m}\left(\frac{v}{w}\right)^m\left(1-\frac{v}{w}\right)^{r-m}, & \mathtt{if}\ \ v\in\{1,\ldots,w-1\}\\ 1, \ \mathtt{if}\ \ m=r, v\in\{w,\ldots,L\}\\ 0, \ \mathtt{if}\ \ m<r, v\in\{w,\ldots,L\}\\ \rho_{m,L+w-v}, \ \ \mathtt{if}\ \ v\in\{L+1,\ldots,L+w-1\}.\end{cases}$$

At time $\tau$, let $R_{j,v}(\tau)$ be the number of edges connected to the check nodes of degree $j$, $j=1,\ldots,r$, at position $v$, $v\in\{1,\ldots,L+w-1\}$. Likewise, let $U_{\underline{x},u}(\tau)$ be the number of edges that are connected to variable nodes with the edge spreading type $\underline{x}$ at position $u$, $u\in\{1,\ldots,L\}$. After the initialization of the peeling decoder, the expected value of $R_{j,v}(0)$ is expressed as

$$\mathbb{E}[R_{j,v}(0)] = jMl/r\sum_{m\ge j}^{r}\rho_{m,v}\binom{m}{j}\epsilon^j(1-\epsilon)^{m-j}.$$

In addition, the initial values of the expected value of $U_{\underline{x},u}(\ell)$ can be computed as

$$\mathbb{E}[U_{\underline{x},u}(0)] = \begin{cases}\epsilon lM/w, & u\in\{1,\ldots,L\}, \underline{x}=\underline{S}_k\\ 0, & \mathtt{otherwise}.\end{cases}$$

Let $\mathbb{E}[\Delta R_{j,v}(\tau)] = \mathbb{E}[R_{j,v}(\tau + 1/M) - R_{j,v}(\tau)]$ and $\mathbb{E}[\Delta U_{\underline{x},u}(\tau)] = \mathbb{E}[U_{\underline{x},u}(\tau + 1/M) - U_{\underline{x},u}(\tau)]$, where the expectation is determined given the degree distributions in the remaining graph at time $\tau$. In order to compute the expectation of $R_{j,v}(\tau)$ and $U_{\underline{x},u}(\tau)$ from the initial values, we need to solve the system of differential equations described as

$$\frac{\partial R_{j,v}(\tau)}{\partial \tau} = \frac{\mathbb{E}\left[\Delta R_{j,v}(\tau)\right]}{1/M}, \quad \frac{\partial U_{\underline{x},u}(\tau)}{\partial \tau} = \frac{\mathbb{E}\left[\Delta U_{\underline{x},u}(\tau)\right]}{1/M}.$$

The procedure used to obtain $\mathbb{E}[\Delta R_{j,v}(\tau)]$ and $\mathbb{E}[\Delta U_{\underline{x},u}(\tau)]$ is described as follows. Let $\phi_{m,\underline{x},u}(\tau)$ be the probability that a variable node of type $\underline{x}$ connected to a degree-one check node at position $m$ belongs to position $u$. Then, we have

$$\phi_{m,\underline{x},u}(\tau) = \begin{cases} \dfrac{\frac{x_{m-u+1}}{|\underline{x}|} U_{\underline{x},u}(\tau)}{\sum\limits_{i \in S(m)} \left( \sum_{\underline{x}'} \frac{x'_{m-i+1}}{|\underline{x}'|} U_{\underline{x}',i}(\tau) \right)}, & \texttt{if} \;\; u \in S(m) \\ 0, & \texttt{otherwise}, \end{cases}$$

where $S(m) = \{j | \min(m - (w-1), 1) \leq j \leq m\}$. When a degree-one check node from position $m$ and the variable node connected to it are removed, we define $\xi_{m,v,t}(\tau)$ as the probability that $t$ edges of the removed variable node are connected to the check nodes other than the removed check node at position $v$. Then, we have

$$\xi_{m,v,t}(\tau) = \begin{cases} \sum\limits_{i \in S(v)} \left( \sum\limits_{\underline{x}: x_{v-i+1} = t} \phi_{m,\underline{x},i}(\tau) \right), & \texttt{if} \;\; m \neq v \\ \sum\limits_{i \in S(v)} \left( \sum\limits_{\underline{x}: x_{v-i+1} = t+1} \phi_{m,\underline{x},i}(\tau) \right), & \texttt{if} \;\; m = v \end{cases}$$

for $t \leq l - 1$. The average number of degree $j$ check nodes losing one edge when $t$ edges are randomly removed from check nodes at position $v$ is given as

$$F_{j,v,t}(\tau) = \sum_{k=1}^{t} k \binom{t}{k} \delta_{j,v}^{k}(\tau) \left(1 - \delta_{j,v}(\tau)\right)^{t-k},$$

where $\delta_{j,v}(\tau) = R_{j,v}(\tau) / \sum\limits_{q=1}^{r} R_{q,v}(\tau)$ for $j \leq r$ and $\delta_{r+1,v}(\tau) = 0$. Then, we have

$$\begin{aligned} &\mathbb{E}[\Delta U_{\underline{x},u}(\tau) | \texttt{pos}(\tau) = m] \\ &= -|\underline{x}| \phi_{m,\underline{x},u}(\tau) \\ &\mathbb{E}[\Delta R_{j,v}(\tau) | \texttt{pos}(\tau) = m] \\ &= \begin{cases} j \sum\limits_{t=1}^{l-1} \xi_{m,v,t}(\tau) \Big( F_{j+1,v,t}(\tau) - F_{j,v,t}(\tau) \Big) - 1, & \\ & \texttt{if}\; v{=}m, j{=}1 \\ j \sum\limits_{t=1}^{l-1} \xi_{m,v,t}(\tau) \Big( F_{j+1,v,t}(\tau) - F_{j,v,t}(\tau) \Big), & \texttt{otherwise}, \end{cases} \end{aligned}$$

where $\texttt{pos}(\tau)$ is the position at which a degree-one check node is removed at time $\tau$. Finally, the expectations of $\Delta R_{j,v}(\tau)$ and $\Delta U_{\underline{x},u}(\tau)$ are described as

$$\mathbb{E}[\Delta R_{j,v}(\tau)] = \sum_{m=1}^{L+w-1} \mathbb{E}[\Delta R_{j,v}(\tau) | \texttt{pos}(\tau) = m] p_m(\tau)$$

$$\mathbb{E}[\Delta U_{\underline{x},u}(\tau)] = \sum_{m=1}^{L+w-1} \mathbb{E}[\Delta U_{\underline{x},u}(\tau) | \texttt{pos}(\tau) = m] p_m(\tau),$$

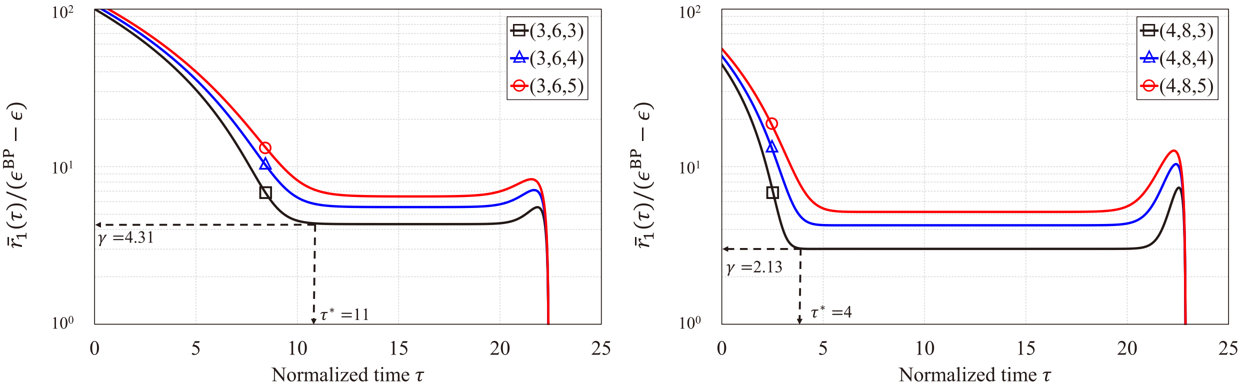
where $p_m(\tau) = R_{1,m}(\tau) / \sum\limits_{v=1}^{L+w-1} R_{1,v}(\tau)$. Then, the expectation of the total number of degree-one check nodes $\mathbb{E}[R_1(\tau)]$ in the remaining graph at $\tau$ becomes $\sum_v \mathbb{E}[R_{1,v}(\tau)]$ and the expected and normalized number of degree-one check nodes $\bar{r}_1(\tau)$ is computed as

$$\bar{r}_1(\tau) = \mathbb{E}[R_1(\tau)]/M. \tag{4}$$

### B. COMPUTING SCALING PARAMETERS OF SC-LDPC ENSEMBLES

With the differential equations in the previous subsection, we can now obtain the expected graph evolution for the $(l, r, w)$ SC-LDPC ensemble. In Fig. 2(a), we plot $\bar{r}_1(\tau)/(\epsilon^{\mathrm{BP}} - \epsilon)$ of the $(3, 6, w)$ ensembles for $w = 3, 4, 5$ and $L = 50$, where $\bar{r}_1(\tau)$ is obtained by (4). The value of $\bar{r}_1(\tau)/(\epsilon^{\mathrm{BP}} - \epsilon)$ in the steady state is the mean parameter $\gamma$. Fig. 2(a) shows that the value of $\gamma$ increases as $w$ grows. Further, the starting point of the steady state $\tau^*$ depends on $w$ as well and the length of the steady state decreases as $w$ grows. In other words, the probability of decoding failure in the steady state tends to decrease as $w$ grows. The same observation is noticed for the $(4, 8, w)$ ensembles as shown in Fig. 2(b). Comparing Fig. 2(a) with Fig. 2(b), we can see that the values of $\gamma$ and $\tau^*$ of the $(3, 6, w)$ SC-LDPC ensemble are larger than those of the $(4, 8, w)$ SC-LDPC ensemble for the same $w$. For example, the $(3, 6, 3)$ SC-LDPC ensemble has $\gamma = 4.31$ and $\tau^* = 11$ while the $(4, 8, 3)$ SC-LDPC ensemble has $\gamma = 2.13$ and $\tau^* = 4$, which indicates the scaling behavior of the $(3, 6, w)$ ensemble is superior compared to the $(4, 8, w)$ ensemble for given $w$.

We summarize the BP threshold, the scaling parameters, and $\tau^*/L$ of the $(l, r, w)$ SC-LDPC ensembles in Table 1. The variance and correlation parameters $\delta$ and $\theta$ are estimated via Monte Carlo simulations at which $M$ is set to 4,000 with 10,000 samples. The scaling parameters in Table 1 are estimated for a channel parameter $\epsilon = \epsilon^{\mathrm{BP}} - 0.04$. Note that the range of code parameters is limited to $(l, r) \in \{(3, 6), (4, 8)\}$ and $w \in \{3, 4, 5\}$ because this limited range is enough to figure out the overall trend. For given $(l, r)$, Table 1 shows that the value of $\alpha$ increases as the coupling width $w$ increases while the BP threshold is not changed. In other words, the coupling width $w$ affects the slope of finite-length performance but the asymptotic performance with infinite code length remains the same. This behavior will be confirmed by simulation in Section IV. In addition, we also observe from Table 1 that $\alpha$ value of the $(3, 6, w)$ ensemble is significantly better than that of the $(4, 8, w)$ ensemble for given $w$. Although the case for higher degrees $(l, r) = (5, 10), (6, 12)$ is not covered in this paper to limit the search space, it is observed that the scaling parameter is degraded as we further increase the density of the graph.



(a) $(3, 6, w)$

(b) $(4, 8, w)$

**FIGURE 2.** The evolution of $r_1(\tau)/(\epsilon^{\mathrm{BP}}-\epsilon)$ of the $(l, r, w)$ SC-LDPC ensembles for $(l, r) = (3, 6), (4, 8)$, $w = 3, 4, 5$, and $L = 50$.

**TABLE 1.** Scaling parameters of $(l, r, w)$ SC-LDPC ensembles.

| $(l, r)$ | $w$ | $\epsilon^{\mathrm{BP}}$ | $\gamma$ | $\delta$ | $\alpha$ | $\theta$ | $\tau^*/L$ |
|---|---|---|---|---|---|---|---|
| (3,6) | 3 | 0.4881 | 4.31 | 0.67 | 5.27 | 1.28 | 0.20 |
| | 4 | 0.4881 | 5.56 | 0.86 | 6.00 | 0.97 | 0.22 |
| | 5 | 0.4881 | 6.48 | 1.02 | 6.42 | 0.79 | 0.24 |
| (4,8) | 3 | 0.4977 | 3.02 | 0.58 | 3.97 | 1.89 | 0.08 |
| | 4 | 0.4977 | 4.24 | 0.82 | 4.68 | 1.29 | 0.10 |
| | 5 | 0.4977 | 5.16 | 1.00 | 5.16 | 1.05 | 0.12 |

**TABLE 2.** Effects of code parameters $(l, r, w, L, M)$ on design rate, BP threshold, and scaling behavior.

| | Design rate | BP threshold | Scaling behavior |
|---|---|---|---|
| As $L$ grows | Positive effect | Unchanged for large $L$ [4] | Negative effect [14] |
| As $(l, r)$ grows | Unchanged | Positive effect [4] | Negative effect [14] [This work] |
| As $w$ grows | Negative effect | Unchanged [4] | Positive effect [This work] |

### C. EFFECTS OF CODE PARAMETERS ON THE FINITE-LENGTH PERFORMANCE

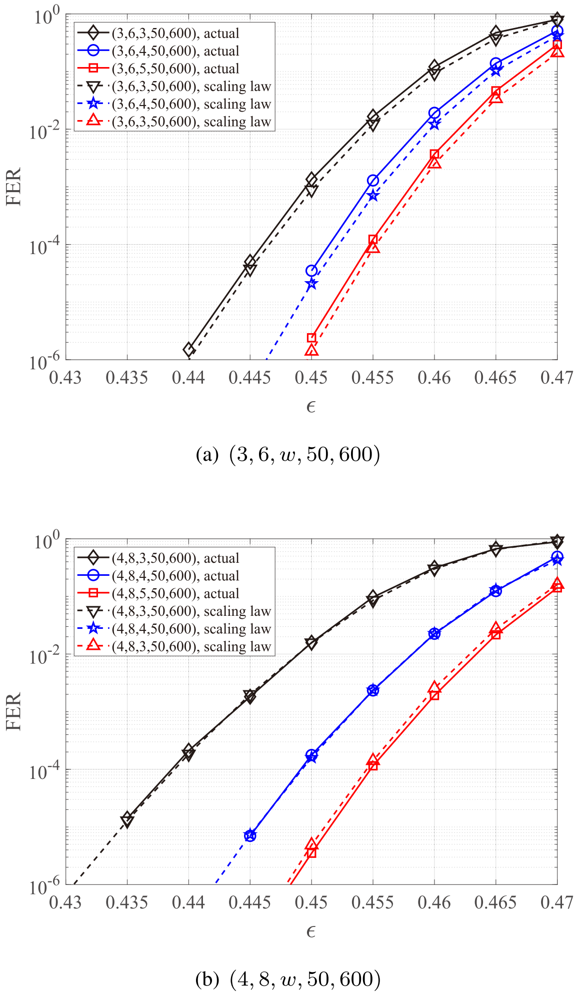
In Table 2, we summarize how to behave the finite-length performance as the code parameters $(l, r, w, L, M)$ vary. First, the effect of $L$ is well known in the literature. As the coupling length $L$ increases, the rate-loss approaches to zero while the BP threshold remains unchanged for large $L$ [4]. Since the long code length can be addressed by windowed decoding [24], the large coupling length $L$ is preferable in the perspective of the asymptotic performance. However, the scaling analysis shows that large $L$ induces the finite-length performance degradation as a linear function of $L$ [14]. Besides, the minimum distance growth rate converges to zero as $L$ increases, which implies the poor error floor performance [3]. Second, consider the degrees $(l, r)$ with the fixed $l/r$ ratio. One can simply improve the BP threshold by increasing the degrees $(l, r)$ [4]. However, the higher degrees $(l, r)$ result in the worse scaling parameter $\alpha$ as shown in Table 1. Third, increasing the coupling width $w$ can lead to the improved scaling behavior while this improved property comes at the cost of the increased rate-loss. To sum up, we can see that all operating code parameters $(l, r, w, L)$ have conflicting effects between the asymptotic performance, code rate, and scaling property. Clearly, the problem to find the optimal code parameters is not trivial. In the next section, we will find the best code parameters for given constraints.

## IV. FINITE-LENGTH PERFORMANCE COMPARISONS

The objective of this section is to find the best code parameters of $(l, r, w, L, M)$ SC-LDPC codes for a given code rate and length. As noted in the introduction, we focus on comparisons among various parameters $(l, r, w)$ while the values of $L$ and $M$ are fixed for each comparison because the values of $L$ and $M$ are commonly determined by external constraints on the frame size, decoding complexity, and latency. Since the code parameters $L$ and $M$ are changed based on different applications and their available hardware resources, we have to investigate the best code parameters $(l, r, w)$ for the target application. We first perform three comparisons to show the effect of $(l, r)$ and $w$ on the performance one by one. While introducing various simulation results, we carry out corresponding discussions and analysis to study the finite-length performance properties of $(l, r, w, L, M)$ SC-LDPC codes. Finally, we find the best parameters $(l, r, w)$ for given $L$ and $M$ and discuss the result with the revealed properties, which provides us an intuition on how to choose the code parameters.

### A. COMPARISON 1: VARIABLE COUPLING WIDTH WITH FIXED OTHER PARAMETERS

In Fig. 3, we compare the FER of the $(3, 6, w, 50, 600)$ and $(4, 8, w, 50, 600)$ SC-LDPC codes with $w = 3, 4, 5$. We obtain the actual performance (solid line) by Monte Carlo simulation over the BEC. The result for the $(3, 6, w, 50, 600)$ SC-LDPC codes shown in Fig. 3(a) demonstrates that $w$ values directly influence the slope of the curve as expected by the scaling behavior analysis. These three codes have the same



(a) $(3, 6, w, 50, 600)$

(b) $(4, 8, w, 50, 600)$

**FIGURE 3. Finite-length performances of $(3, 6, w, 50, 600)$ and $(4, 8, w, 50, 600)$ SC-LDPC codes for $w = 3, 4, 5$. Solid line corresponds to the actual performance and dashed line corresponds to the estimated performance by the scaling law.**

BP threshold 0.4881 but the finite-length performance significantly differs from each other because of their difference in $\alpha$ values as shown in Table 1. This result gives us a lesson that the finite-length performance highly depends on the scaling parameters as well as the BP threshold. A similar result is shown in Fig. 3(b) for the $(4, 8, w, 50, 600)$ SC-LDPC codes. In addition, we add the estimated performance obtained by the refined scaling law in (3), where the estimate is very accurate. Note that there is a slight gap between the actual simulated curve and the estimated curve for $(l, r) = (3, 6)$, which is also remarked in [16]. However, the negligible gap is not large enough to reverse the relative superiority of compared codes and thus we keep using the scaling law in the following comparisons instead of performing actual simulations.

Note that, although using large $w$ results in significantly improved performances, the comparison is not fair because the code rates of the compared codes in Fig. 3 are not equivalent i.e., $R_{(3,6,3,50,600)} = 0.48$, $R_{(3,6,4,50,600)} = 0.47$, and $R_{(3,6,5,50,600)} = 0.46$. A fair comparison is required with the fixed code rate, which is introduced in Subsection IV-D.

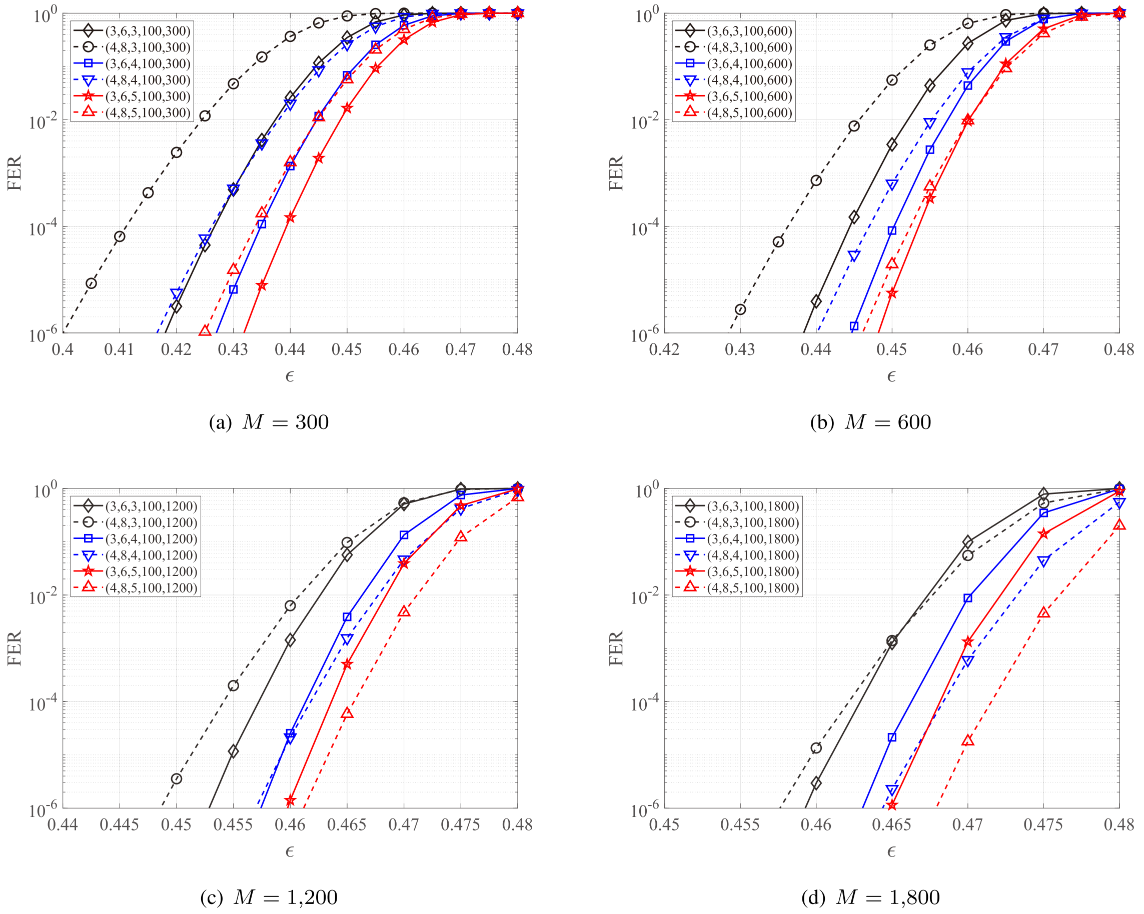
### B. COMPARISON 2: COMPARING DEGREES $(3, 6)$ AND $(4, 8)$

In this subsection, we compare two SC-LDPC codes of $(3, 6, w, L, M)$ and $(4, 8, w, L, M)$ with fixed $(w, L, M)$. It is generally known that the code performance of the $(4, 8, w, L, M)$ SC-LDPC ensemble is superior to that of the $(3, 6, w, L, M)$ SC-LDPC ensemble because of the gap between their BP thresholds, i.e., $\epsilon^{\mathrm{BP}}(4, 8, w, L, M) > \epsilon^{\mathrm{BP}}(3, 6, w, L, M)$. However, the fact that $\alpha$ of the $(3, 6, w, L, M)$ ensemble is higher than $\alpha$ of the $(4, 8, w, L, M)$ ensemble implies the possibility that the $(3, 6, w, L, M)$ ensemble outperforms the $(4, 8, w, L, M)$ ensemble in terms of the finite-length performance.

The FER curves of $(l, r, w, 100, M)$ finite-length SC-LDPC codes for $(l, r) = (3, 6), (4, 8)$, $w = 3, 4, 5$, and $M = 300, 600, 1{,}200, 1{,}800$ are depicted in Fig. 4. The codes in each sub-graph have the same $M$ value. The solid curves correspond to $(l, r) = (3, 6)$ whereas the dashed curves correspond to $(l, r) = (4, 8)$. In order to clarify the comparison, we use the same color for the same $w$. Fig. 4(a) shows that the $(3, 6, w, 100, 300)$ codes outperform their counterpart codes with $(l, r) = (4, 8)$ for all $w$. It can be interpreted that the effect of $\alpha$ is stronger than the difference in the BP thresholds for small $M = 300$. On the contrary, as $M$ value increases, the finite-length performance gets close to the asymptotic performance, which means the BP threshold plays a critical role for determining the finite-length performance. Fig. 4(d) shows $(4, 8, w, 100, 1800)$ codes have better performance than $(3, 6, w, 100, 1800)$ for all $w$ up to FER $10^{-3}$. It is also shown that the $(3, 6, 3, 100, 1800)$ code still exhibits a notably fast falling slope compared to the $(4, 8, 3, 100, 1800)$ code, and consequently surpasses the $(4, 8, 3, 100, 1800)$ code in FER lower than $10^{-3}$. From Fig. 4(a) to Fig. 4(d), we can confirm that the relative superiority between codes with $(3, 6)$ and codes with $(4, 8)$ varies over the range of $M$.

### C. RANDOM PUNCTURING FOR MATCHING THE CODE RATE

Since the code rate varies depending on $w$ and $L$ as represented in (1), comparing two codes with different $w$ and $L$ values is not fair. In this section, we match the code rate to the target rate using the random puncturing technique in [23]. This allows us to compare various codes without considering differences in their code rates. Let $\rho$ be the puncturing fraction and $\rho M$ variable nodes at each position are punctured. Then, the total number of punctured bits becomes $p = \rho LM = \rho n$ and the code rate of punctured codes becomes $R(\rho) = 1 - \frac{k}{n-p} = \frac{R}{1-\rho}$. To match the code rate to target rate $R_{\mathtt{target}}$, the puncturing fraction should be $\rho = 1 - \frac{R}{R_{\mathtt{target}}}$. Since punctured bits can be considered as erasure of channel, the punctured codes under the BEC with $\epsilon$ is identical to transmit across the equivalent BEC channel



(a) $M = 300$

(b) $M = 600$

(c) $M = 1{,}200$

(d) $M = 1{,}800$

**FIGURE 4.** **Finite-length performances of the $(l, r, w, 100, M)$ SC-LDPC codes for $(l, r) = (3, 6), (4, 8)$, $w = 3, 4, 5$, and $M = 300, 500, 1{,}200, 1{,}800$.**

with erasure probability $\epsilon'$, where $(1-\epsilon')n = (1-\epsilon)(n-p)$, i.e., $\epsilon' = \epsilon + (1-\epsilon)\rho$. Consequently, the FER of punctured codes $P_{\mathrm{B},\mathtt{Punct}}(\epsilon)$ is equal to $P_{\mathrm{B}}(\epsilon + (1-\epsilon)\rho)$, which is equivalent to the shifted FER of unpunctured codes. In other words, punctured codes suffer the puncturing loss as much as $(1-\epsilon)\rho$.

### D. COMPARISON 3: VARIABLE COUPLING WIDTH WITH FIXED OTHER PARAMETERS AND THE SAME CODE RATE

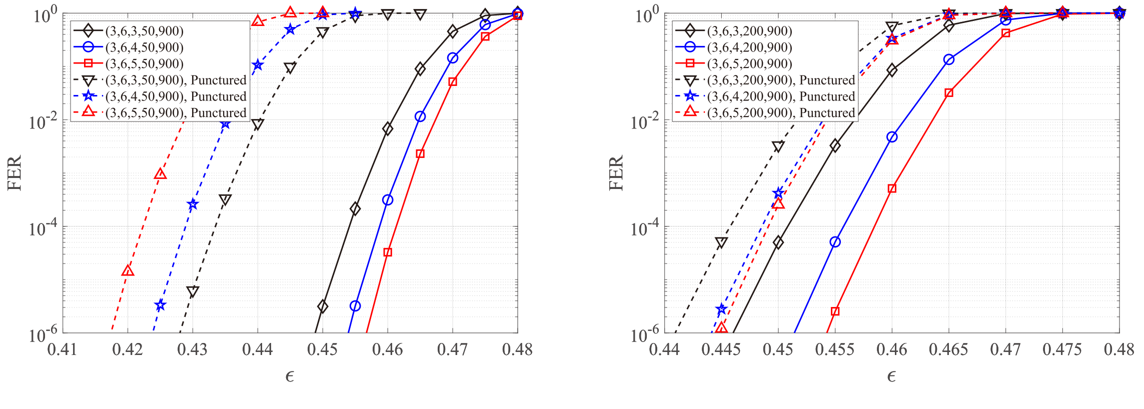
We reconsider the performance comparison performed in Subsection IV-A, where $w$ is a variable and $l, r, L, M$ are fixed but we include punctured codes achieving target code rate 0.5. Fig. 5(a) shows the performances of the $(3, 6, w, 50, 900)$ ensembles for $w = 3, 4, 5$ together with their punctured codes with code rate 0.5. As we can see, larger $w$ suffers more severe puncturing loss and the relative superiority between the codes with different $w$ is reversed after puncturing, i.e., the punctured code with $w = 3$ shows the best performance among the punctured codes while the unpunctured code with $w = 5$ is the best among the unpunctured codes. On the other hand, for large $L$ as shown in Fig. 5(b), the rate-loss is negligible due to sufficiently large $L$, and accordingly the puncturing loss is not strong enough to change the relative superiority, i.e., the SC-LDPC codes with $w = 5$ are the best for both punctured and unpunctured cases.

### E. OPTIMAL CODE PARAMETERS OF SC-LDPC CODES

From the previous comparisons 1, 2, 3, three properties on the finite-length performance of $(l, r, w, L, M)$ SC-LDPC codes are confirmed as follows.

- Large $w$ corresponds to a higher value of $\alpha$ and a fast decaying slope of the FER curve.
- For large $M$, the finite-length performance is dominated by the BP threshold rather than the scaling parameter $\alpha$, whereas the value of $\alpha$ has more influence on the performance for small $M$.
- For large $L$, the puncturing loss to match the target code rate is relatively low, thus codes with large $w$ mostly outperform those with small $w$. For small $L$, the trend is opposite due to the considerable puncturing loss.

Considering the above properties on the finite-length performance, we can expect which code parameters will show the best performance for given $L$ and $M$. For example, for large $L$ and small $M$, the puncturing loss is negligible and the value of $\alpha$ has more effect on the performance than the BP threshold and thus we can expect that codes with larger $w$



(a) $L = 50$

(b) $L = 200$

**FIGURE 5. Finite-length performances of the $(3, 6, w, L, 900)$ SC-LDPC codes for $w = 3, 4, 5$ and $L = 50, 200$ and their punctured codes with code rate 0.5.**

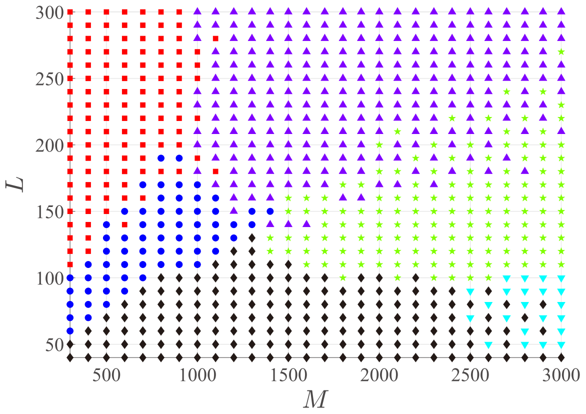
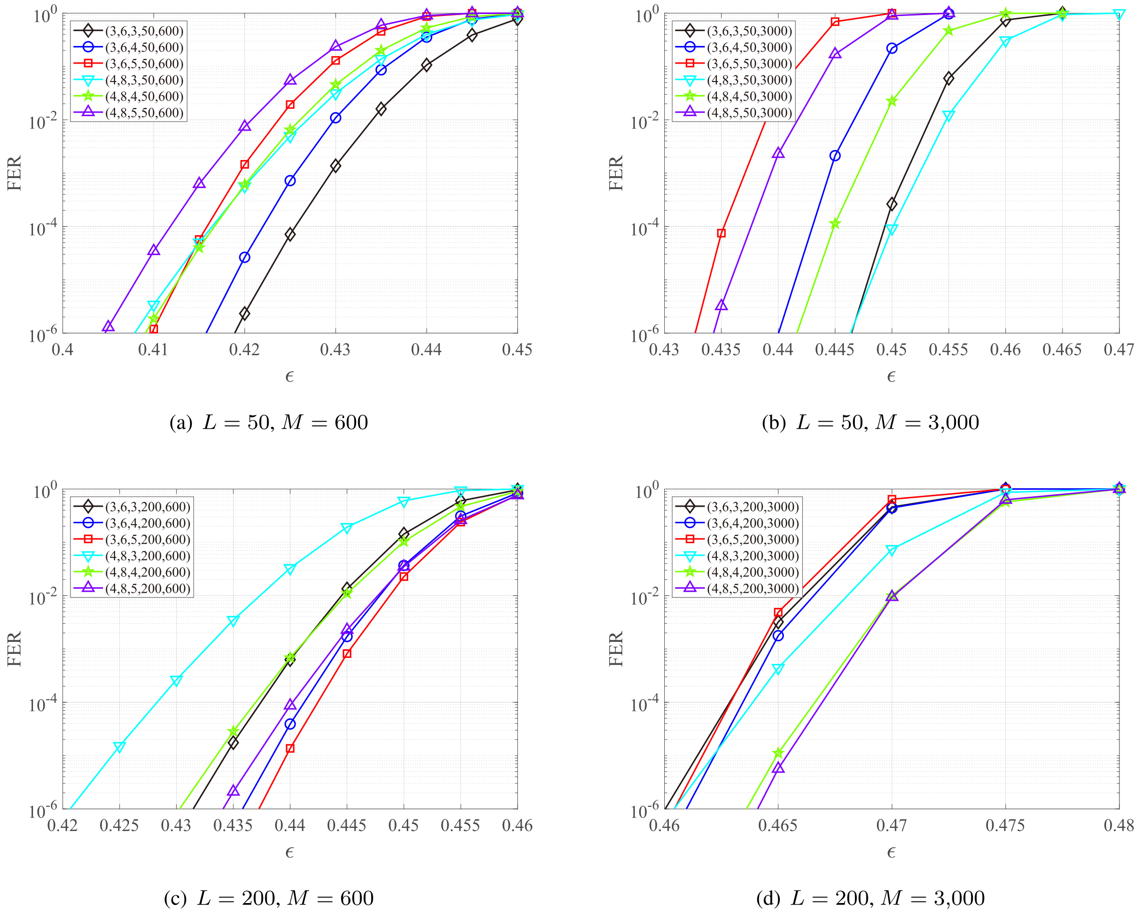


**FIGURE 6. 2-D plot showing the optimal code parameters $(l, r, w)$ showing the best performance with target FER $10^{-4}$ for a given $L$ and $M$. All code rates are set to 0.5. Black diamond: $(3, 6, 3)$, Blue circle: $(3, 6, 4)$, Red square: $(3, 6, 5)$, Cyan reverse-triangle: $(4, 8, 3)$, Green star: $(4, 8, 4)$, Purple triangle: $(4, 8, 5)$.**

and higher $\alpha$ value such as $(3, 6, 3, L, M)$ codes show the best performance. Using the scaling law in (3), it is possible to find the optimal code parameters $(l, r, w)$ for given $L$ and $M$ that show the best finite-length performance. More specifically, we set a target FER and search for the erasure probability at which the estimated FER by the scaling law matches the target FER for all possible code parameters $(l, r, w)$. Finally, choose the optimal parameters showing the highest value of the erasure probability achieving the target FER.

Fig. 6 shows 2-D plot of the optimal code parameters $(l, r, w)$ for given $L$ and $M$, where each point indicates the best code parameters $(l, r, w)$ with target FER $10^{-4}$. Each symbol represents as follows: black diamond $(3, 6, 3)$, blue circle $(3, 6, 4)$, red square $(3, 6, 5)$, cyan reverse-triangle $(4, 8, 3)$, green star $(4, 8, 4)$, and purple triangle $(4, 8, 5)$. The code rates of all codes are set to 0.5 by using puncturing. As we can see, there is no specific code parameter set that shows the best performance for all regions of $L$ and $M$. That is, the best SC-LDPC code parameters are changed based on different applications. Fig. 6 shows each code parameter set has its own dominant region. We can interpret the result as follows.

- In the region for small $L$ and small $M$ such as $L = 50$ and $M = 600$, the puncturing loss is relatively significant due to small $L$ and the value of $\alpha$ is more important than the BP threshold for small $M$. The puncturing loss of $w = 3$ is the least and the code with $(l, r) = (3, 6)$ has the higher $\alpha$ compared to the code with $(l, r) = (4, 8)$. Thus, the code parameters $(3, 6, 3)$ (black diamond) exhibit the best performance. Fig. 7(a) shows the performance comparison for $L = 50$ and $M = 600$.
- In the region for small $L$ and large $M$ such as $L = 50$ and $M = 3{,}000$, the puncturing loss is still relatively large and it enters the region where the BP threshold is more important than the value of $\alpha$. Thus, the degrees $(l, r) = (4, 8)$ showing the higher BP threshold and the small coupling width $w = 3$ are expected to show the best performance. Fig. 7(b) shows that the code parameter $(4, 8, 3)$ (cyan reverse-triangle) exhibits the best performance for $L = 50$ and $M = 3{,}000$.
- In the region for large $L$ and small $M$ such as $L = 200$ and $M = 600$, the puncturing loss is not significant and the value of $\alpha$ is more important measure than the BP threshold. According to Table 1, the code parameter set $(3, 6, 5)$ has the largest $\alpha$ and the puncturing loss can be negligible even for large $w = 5$ due to large $L$. Thus, the code parameter set $(3, 6, 5)$ (red square) shows the best performance as shown in Fig. 7(c).
- In the region for large $L$ and large $M$ such as $L = 200$ and $M = 3{,}000$, the puncturing loss is not significant and the BP threshold has more dominant effect on the performance than the value of $\alpha$. Thus, the degrees $(4, 8)$ with high BP threshold have an advantage compared to the degrees $(3, 6)$. Among the degrees $(4, 8)$, large $w$ shows the better performance with their negligible puncturing loss, which means the code parameter set $(4, 8, 5)$ (purple triangle) shows the best performance in



(a) $L = 50$, $M = 600$

(b) $L = 50$, $M = 3{,}000$

(c) $L = 200$, $M = 600$

(d) $L = 200$, $M = 3{,}000$

**FIGURE 7. Finite-length performances of ($l, r, w, L, M$) SC-LDPC codes with code rate 0.5.**

this region. Fig. 7(d) shows the performance comparison for $L = 200$ and $M = 3{,}000$.

Comparing the operating ranges of the channel parameter $\epsilon$ for the results in Fig. 7(a) ($0.4 \leq \epsilon \leq 0.45$) and Fig. 7(d) ($0.46 \leq \epsilon \leq 0.48$), the finite-length performances of SC-LDPC codes for large $L$ and $M$ are considerably superior to those for small $L$ and $M$. Thus, it is desirable to choose large values of $L$ and $M$ if a target application can support relatively heavy implementation costs. In addition, the result in Fig. 7(d) recommends choosing code parameters (4, 8, 5) that shows the best performance for large values of $L$ and $M$. However, Fig. 7(a) shows the opposite phenomenon for small $L$ and $M$ such that the code parameters (4, 8, 5) exhibit the worst performance while the counterpart parameters (3, 6, 3) shows the best performance. It means that (3, 6, 3) is preferable for hardware resource-limited scenarios and the best code parameters depend heavily on the target application.

## V. CONCLUSION

In this paper, a study on the finite-length performance of SC-LDPC codes was carried out to reveal the existing trade-offs for code parameters and find the optimal code parameters. Based on the scaling behavior analysis focusing on the coupling width $w$, it was shown that the larger coupling width $w$ leads to the higher scaling parameter $\alpha$ and a steep falling slope of the finite-length performance at the cost of the increased rate-loss. Moreover, by including the parameter $w$ in the scaling law as a variable parameter, we could perform comparisons to understand the behavior of the finite-length performance affected by code parameters. It was shown that the performance measure of $\alpha$ is more critical than the BP threshold for small $M$, and smaller $w$ could be a better choice for small $L$ because of its relatively low puncturing loss when codes are compared at the same code rate using puncturing. Consequently, we obtained the optimal code parameter sets for given $L$ and $M$ and provided sufficient interpretation for the optimization results. Since the appropriate values of $L$ and $M$ depend on available hardware resources and target applications, we believe that this optimization result over wide range of $L$ and $M$ helps to select the code parameters of SC-LDPC codes showing superior finite-length performances for given constraints.

## REFERENCES

[1] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.

[2] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[3] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866–4889, Sep. 2015.

[4] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.

[5] A. Yedla, Y. Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of Maxwell saturation for coupled scalar recursions," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6943–6965, Nov. 2014.

[6] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[7] T. J. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, U.K.: Cambridege Univ. Press, 2008.

[8] S. Kudekar and K. Kasai, "Spatially coupled codes over the multiple access channel," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Aug. 2011, pp. 2816–2820.

[9] D. Truhachev, "Achieving AWGN multiple access channel capacity with spatial graph coupling," *IEEE Commun. Lett.*, vol. 16, no. 5, pp. 585–588, May 2012.

[10] K. Takeuchi, T. Tanaka, and T. Kawabata, "Performance improvement of iterative multiuser detection for large sparsely spread CDMA systems by spatial coupling," *IEEE Trans. Inf. Theory*, vol. 61, no. 4, pp. 1768–1794, Apr. 2015.

[11] Y. Zhang, K. Peng, J. Song, and Y. Wu, "Quasi-cyclic spatially coupled LDPC code for broadcasting," *IEEE Trans. Broadcast.*, vol. 66, no. 1, pp. 187–194, Mar. 2020.

[12] A. Sridharan, D. Truhachev, M. Lentmaier, D. J. Costello, and K. S. Zigangirov, "Distance bounds for an ensemble of LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4537–4555, Dec. 2007.

[13] D. J. Costello, Jr., L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell, and R. Smarandache, "Spatially coupled sparse codes on graphs: Theory and practice," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 168–176, Jul. 2014.

[14] P. M. Olmos and R. Urbanke, "A scaling law to predict the finite-length performance of spatially coupled LDPC codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3164–3184, Jun. 2015.

[15] M. Stinner and P. M. Olmos, "On the waterfall performance of finite-length SC-LDPC codes constructed from protographs," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 345–361, Feb. 2016.

[16] R. Sokolovskii, A. G. I. Amat, and F. Brannstrom, "Finite-length scaling of spatially coupled LDPC codes under window decoding over the BEC," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 5988–5998, Oct. 2020.

[17] M. Battaglioni, A. Tasdighi, G. Cancellieri, F. Chiaraluce, and M. Baldi, "Design and analysis of time-invariant SC-LDPC convolutional codes with small constraint length," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 918–931, Mar. 2018.

[18] M. H. Tadayon, A. Tasdighi, M. Battaglioni, M. Baldi, and F. Chiaraluce, "Efficient search of compact QC-LDPC and SC-LDPC convolutional codes with large girth," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1156–1159, Jun. 2018.

[19] H. Esfahanizadeh, A. Hareedy, and L. Dolecek, "Finite-length construction of high performance spatially-coupled codes via optimized partitioning and lifting," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 3–16, Jan. 2019.

[20] H.-Y. Kwak, J.-S. No, and H. Park, "Design of irregular SC-LDPC codes with non-uniform degree distributions by linear programming," *IEEE Trans. Commun.*, vol. 67, no. 4, pp. 2632–2646, Apr. 2019.

[21] S. Naseri and A. H. Banihashemi, "Construction of time invariant spatially coupled LDPC codes free of small trapping sets," *IEEE Trans. Commun.*, vol. 69, no. 6, pp. 3485–3501, Jun. 2021.

[22] V. Aref, L. Schmalen, and S. T. Brink, "On the convergence speed of spatially coupled LDPC ensembles," in *Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Oct. 2013, pp. 342–349.

[23] D. G. M. Mitchell, M. Lentmaier, A. E. Pusane, and D. J. Costello, "Randomly punctured LDPC codes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 408–421, Feb. 2016.

[24] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.

[25] A. R. Iyengar, P. H. Siegel, R. L. Urbanke, and J. K. Wolf, "Windowed decoding of spatially coupled codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2277–2292, Apr. 2013.

[26] L. Schmalen, V. Aref, J. Cho, D. Suikat, D. Rösener, and A. Leven, "Spatially coupled soft-decision error correction for future lightwave systems," *J. Lightw. Technol.*, vol. 33, no. 5, pp. 1109–1116, Mar. 1, 2015.

[27] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 429–439, Feb. 2011.

**HEE-YOUL KWAK** received the B.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2013 and 2019, respectively. Since 2019, he has been a Staff Engineer with Samsung Electronics, South Korea. He joined the Faculty of Institute of New Media and Communications, Seoul National University, in 2021, where he is currently a Senior Researcher. His research interests include error-correcting codes, coding theory, and coding for memory.

**JAE-WON KIM** (Member, IEEE) received the B.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2014 and 2020, respectively. From 2020 to 2021, he was a Staff Engineer with Samsung Electronics, South Korea. He joined Gyeongsang National University, Jinju, South Korea, in 2021, where he is currently an Assistant Professor with the Department of Electronic Engineering. His research interests include error-correcting codes, coding theory, index coding, and DNA storage.

**JONG-SEON NO** (Fellow, IEEE) received the B.S. and M.S.E.E. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1981 and 1984, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1988. From 1988 to 1990, he was a Senior MTS with Hughes Network Systems. From 1990 to 1999, he was an Associate Professor with the Department of Electronic Engineering, Konkuk University, Seoul. He joined the Faculty of the Department of Electrical and Computer Engineering, Seoul National University, in 1999, where he is currently a Professor. His current research interests include error-correcting codes, cryptography, sequences, LDPC codes, interference alignment, and wireless communication systems. He became a fellow of the IEEE through the IEEE Information Theory Society, in 2012. He became a member of the National Academy of Engineering of Korea (NAEK), in 2015, where he is also the Division Chair of electrical, electronic, and information engineering. He was a recipient of the IEEE Information Theory Society Chapter of the Year Award, in 2007. From 1996 to 2008, he served as the Founding Chair for the Seoul Chapter of the IEEE Information Theory Society. He was the General Chair of Sequence and Their Applications 2004 (SETA2004), Seoul. He also served as the General Co-Chair for the International Symposium on Information Theory and Its Applications 2006 (ISITA2006) and the International Symposium on Information Theory 2009 (ISIT2009), Seoul. From 2012 to 2013, he served as the Co-Editor-in-Chief for the *Journal of Communications and Networks*.