

Article

Multi-Kernel Polar Codes versus Classical Designs with Different Rate-Matching Approaches [†]

Souradip Saha * and Marc Adrat *

Fraunhofer Institute for Communication, Information Processing and Ergonomics, Fraunhoferstraße 20, 53343 Wachtberg, Germany

* Correspondence: souradip.saha@fkie.fraunhofer.de (S.S.); marc.adrat@fkie.fraunhofer.de (M.A.)

[†] This article is an extended version of the following paper: Saha, S.; Adrat, M. Versatile Polar Codes with Different Kernel Sizes and Rate Matching Approaches. In Proceedings of the 2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS), Adelaide, SA, Australia, 14–16 December 2020; pp. 1–7, doi:10.1109/ICSPCS50536.2020.9310019.

Abstract: Polar codes, which have been proposed as a family of linear block codes, has garnered a lot of attention from the scientific community, owing to their low-complexity implementation and provably capacity-achieving capability. Thus, they have been proposed to be used for encoding information on the control channels in the upcoming 5G wireless networks. The basic approach introduced by Arikan in his landmark paper to polarize bit channels of equal capacities to those of unequal capacities can be used to design only codewords of length $N = 2^n$, which is a major limitation when codewords of different lengths are required for the underlying applications. In the predecessor paper, this aspect was partially addressed by using a 3×3 kernel circuit (used to generate codewords of length $M = 3^m$), along with downsizing techniques such as puncturing and shortening to assess the optimal design and resizing techniques based on the underlying system parameters. In this article, we extend this research to include the assessment of multi-kernel rate-matched polar codes for applicability over a much wider range of codeword lengths.

Keywords: polar codes; puncturing; shortening; multi-kernel; 5G



Citation: Saha, S.; Adrat, M. Multi-Kernel Polar Codes versus Classical Designs with Different Rate-Matching Approaches. *Electronics* **2021**, *10*, 1717. <https://doi.org/10.3390/electronics10141717>

Academic Editor: Tadeusz A. Wysocki

Received: 31 May 2021
Accepted: 14 July 2021
Published: 17 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

As polar codes of length $N = 2^n$ might not always be suitable for an underlying system, developing techniques to generate more versatile codeword lengths is very important to realize their utilities in a wider range of practical systems. They are proposed to be used for encoding/decoding data over the 5G control channels with certain utilitarian criteria and challenges mentioned in [1]. In the latest version of 3GPP 5G NR specifications [2], three resizing techniques for polar codes have been used, namely, repetition (upsizing) or puncturing and shortening (downsizing). In this article, we only investigate the downsizing aspect. These techniques w.r.t 5G, however, have been proposed only over codewords of length $N = 2^n$. Additionally, we perform a comparative analysis with polar codes generated by a 3×3 polarization circuit (same as the one used in [3]), resulting in codewords of length $M = 3^m$, and use the 2×2 and 3×3 circuits to design multi-kernel polar codes, as shown in the series of articles [4–10]. By assessing the error rate performances and complexities of polar code designs generated by these different construction and downsizing techniques, we determine the effect of different parameters and how to optimally design polar codes for a given set of parameter values. This expands the scope of practical applications of polar codes to include scenarios when nonconventional (2^n) codeword lengths are desirable.

The rest of the article is organized as follows. In Section 2, we provide an overview of the polarization kernels of sizes 2 and 3, which correspond to the same circuits used in [3]. Additionally, the design technique for multi-kernel codes is also provided. In Section 3,

a brief description of the downsizing techniques used in this article, namely, puncturing and shortening is provided. In Section 4, a comparative analysis of downsized single kernel codes to multi-kernel codes is presented, based on bit error rate (BER) performance and complexity, to determine the conditions on system parameters for optimal design of polar codes of desired codeword lengths. The open questions that have arisen owing to the research conducted for this article are stated in Section 5. Finally, the concluding remarks are provided in Section 6.

Notations and Remarks: N and M denote codeword lengths (corresponding to the exponents n and m of 2 and 3, respectively, (i.e., $N = 2^n$ and $M = 3^m$) of codewords generated by 2×2 and 3×3 kernels, respectively, whereas K is used to denote codeword lengths of multi-kernel codes (i.e., $K = 2^n \times 3^m$) generated by 2×2 and 3×3 kernels. G denotes generator matrix, with a subscript denoting the codeword length generated by G . R_d denotes code rate. Z denotes the z -parameter (Bhattacharyya parameter) value of a given bit channel, with a subscript denoting the channel number. z -parameter indicates the inverse of the capacity of a bit channel (refer to [11] for details) and is used to determine the effect of channel polarization. All the BER simulations have been performed over additive white Gaussian noise channel (AWGNC) with binary phase shift keying (BPSK) modulation scheme and z -parameter value $Z = 0.5$ for channel polarization. Amongst all the available polar decoding options, successive cancellation (SC) decoder is used.

2. Polarization Circuits

2.1. 2×2 Kernel Circuit

In his landmark paper [11], Arikan proposed the method of channel polarization to design a family of block codes called polar codes. He proposed using a 2×2 circuit (as shown by the dashed red boxes in Figures 1 and 2) to polarize 2 bit channels of equal capacities to 2 bit channels of unequal capacities.

We denote the z -parameter value of the real channel (\vec{x} to \vec{y} in Figures 1 and 2) as Z . The virtual bit channels obtained from the transformation by the 2×2 polarization circuit (three sets from \vec{u} to \vec{x} in Figure 1 or three sets from \vec{x} to \vec{y} in Figure 2) are denoted as Z_1 and Z_2 and are given by (1) and (2) over a binary erasure channel (BEC).

$$Z_1 = 2Z - Z^2 \quad (1)$$

$$Z_2 = Z^2 \quad (2)$$

where $Z_1 + Z_2 = 2Z$, resulting in the conservation of channel capacity after polarization. Clearly as $Z_1 \geq Z$ and $Z_2 \leq Z$, one virtual channel has higher capacity than the real channel, while the other one has a lower capacity. Although the equality in Equations (1) and (2) are valid just for BECs, the concept of channel transformation to polarize their effective capacity holds true for any given channel model. Due to the recursive nature of constructing polar code circuits using the 2×2 kernel element, only codewords of length $N = 2^n$ can be obtained. For more details, the reader is recommended to refer to Section IIA of [3] or [11].

2.2. 3×3 Kernel Circuit

The idea of channel polarization to transform multiple bit channels with equal capacities to the same number of bit channels with unequal capacities can be expanded to any number of bit channels. In this subsection, we will look into polarizing 3 bit channels simultaneously, using a 3×3 circuit. There have been multiple proposals in the literature as to how to polarize the three channels (or how to design a 3×3 circuit) similar to the ones either in the set of articles [4–10] or, alternatively, in [12–14] or [15]. Within the scope of this article, we only use the circuit structure proposed in [12], owing to easier implementation and proximity of design to Arikan's original circuit, and it has also been used in [3]. It is shown by the dashed blue boxes in Figures 1 and 2.

Here, we denote the z-parameter value of the real channel (\vec{x} to \vec{y} in Figures 1 and 2) as Z and the corresponding virtual bit channels by a 3×3 polarization circuit (two sets from \vec{x} to \vec{x} in Figure 1 or two sets from \vec{u} to \vec{x} in Figure 2) as Z_1, Z_2 , and Z_3 . Equations (3)–(5) provide the corresponding z-parameter transformation over a BEC.

$$Z_1 = 3Z - 3Z^2 + Z^3 \tag{3}$$

$$Z_2 = 2Z^2 - Z^3 \tag{4}$$

$$Z_3 = Z^2 \tag{5}$$

Clearly, as $Z_1 + Z_2 + Z_3 = 3Z$, the channel capacity is conserved, and this results in valid channel polarization. Thus, the aforementioned circuit can be used recursively to design channel polarization circuits to generate polar codewords of length $M = 3^m$. For more details, the reader is recommended to refer to Section IIB of [3].

2.3. Multi-Kernel Circuit

For channel polarization, one need not be limited to the recursive usage of a single kernel circuit to polarize bit channels. The polarization effect can also be achieved by using multiple kernel sizes simultaneously within the same encoding/decoding circuits, called multi-kernel polar codes. The concept of channel polarization using different kernel designs within the same polar circuit and the theoretical analysis of channel polarization aspects of such multi-kernel designs have been investigated in a series of papers [4–10]. In this section and within the scope of this article, we investigate multi-kernel polar codes generated by the combination of circuit kernels of sizes 2 and 3 to obtain codeword lengths of the form $K = 2^n \times 3^m$. Using $n = 1$ and $m = 1$, we can obtain 6×6 polarization circuits. The generator matrices are denoted by G . In this section, the circuit design with 2×2 kernels in the first stage and 3×3 kernels in the second stage is denoted as the A version, whereas the one with 3×3 kernels in the first stage and 2×2 kernels in the second stage is denoted as the B version.

The circuit design for $G_{6A} = perm(G_2 \otimes G_3)$ (where $perm$ denotes the permutation of rows of G to establish the correct connection between stages of polarization; $perm = bit-reversal$ for G_{2^n}) is shown in Figure 1, with the dashed red and blue boxes encapsulating the 2×2 and 3×3 polarization circuits, respectively.

The generator matrix for the circuit in Figure 1 is given as

$$G_{6A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{6}$$

through which the codeword $\vec{x} = [x_1, x_2, x_3, x_4, x_5, x_6]$ is generated from the input bit vector $\vec{u} = [u_1, u_2, u_3, u_4, u_5, u_6]$ by (7).

$$\vec{x} = \vec{u} \cdot G_{6A} \tag{7}$$

Using the same convention as in Sections 2.1 and 2.2, Equations (8)–(13) provide the z-parameter transformation by the polarization circuit in Figure 1.

$$Z_1 = 6Z - 15Z^2 + 20Z^3 - 15Z^4 + 6Z^5 - Z^6 \tag{8}$$

$$Z_2 = 8Z^2 - 16Z^3 + 14Z^4 - 6Z^5 + Z^6 \tag{9}$$

$$Z_3 = 4Z^2 - 4Z^3 + Z^4 \tag{10}$$

$$Z_4 = 3Z^2 - 3Z^4 + Z^6 \tag{11}$$

$$Z_5 = 2Z^4 - Z^6 \tag{12}$$

$$Z_6 = Z^4 \tag{13}$$

Clearly, as, $Z_1 + Z_2 + Z_3 + Z_4 + Z_5 + Z_6 = 6Z$, the channel capacity is conserved, and this results in valid channel polarization. Thus, the aforementioned circuit can be used recursively to design channel polarization circuits to generate polar codewords of length $K = 6^k$.

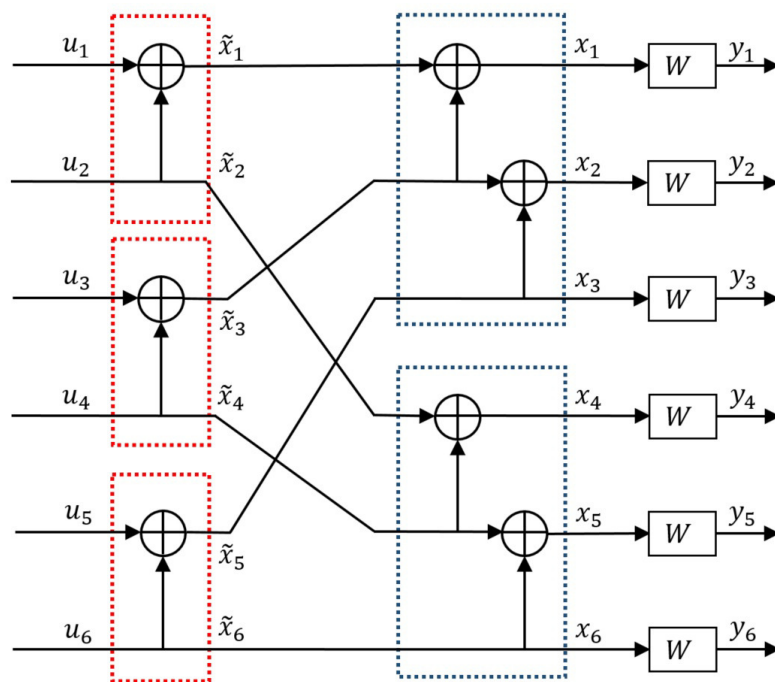


Figure 1. Version A of the 6×6 circuit for channel polarization.

The circuit design for $G_{6B} = perm(G_3 \otimes G_2)$ is shown in Figure 2. The generator matrix for the circuit in Figure 2 is given as

$$G_{6B} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{14}$$

through which the codeword $\vec{x} = [x_1, x_2, x_3, x_4, x_5, x_6]$ is generated from the input bit vector $\vec{u} = [u_1, u_2, u_3, u_4, u_5, u_6]$ by (15).

$$\vec{x} = \vec{u} \cdot G_{6B} \tag{15}$$

Using the same convention as before, Equations (16)–(21) provide the z-parameter transformation by the polarization circuit in Figure 2.

$$Z_1 = 6Z - 15Z^2 + 20Z^3 - 15Z^4 + 6Z^5 - Z^6 \tag{16}$$

$$Z_2 = 4Z^2 - 2Z^3 - 4Z^4 + 4Z^5 - Z^6 \tag{17}$$

$$Z_3 = 2Z^2 - Z^4 \tag{18}$$

$$Z_4 = 9Z^2 - 18Z^3 + 15Z^4 - 6Z^5 + Z^6 \tag{19}$$

$$Z_5 = 4Z^4 - 4Z^5 + Z^6 \tag{20}$$

$$Z_6 = Z^4 \tag{21}$$

Here too, $Z_1 + Z_2 + Z_3 + Z_4 + Z_5 + Z_6 = 6Z$, implying conservation of channel capacity resulting in valid channel polarization, and using it recursively can help design polarization circuits to generate codewords of length $K = 6^k$.

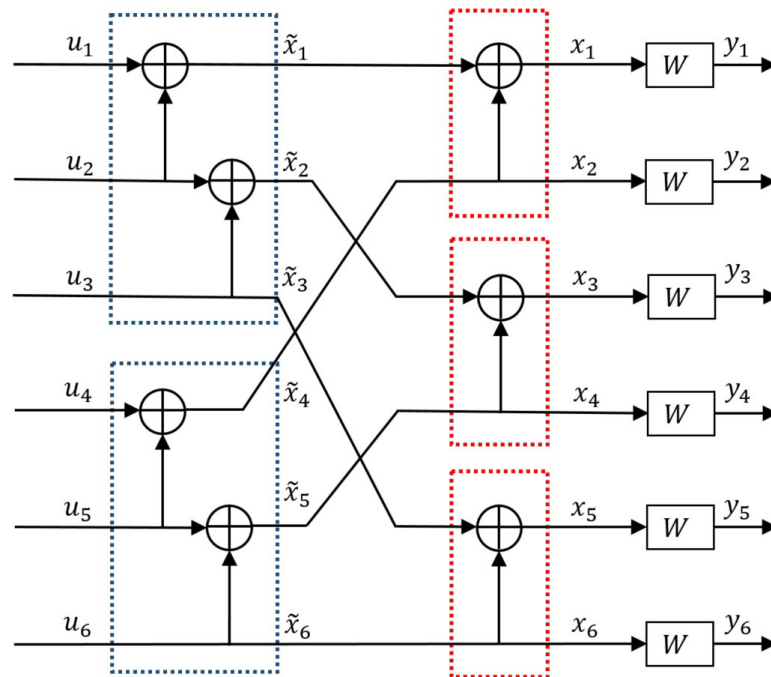


Figure 2. Version B of the 6×6 circuit for channel polarization.

Using a different number of circuit elements in each stage or different order of kernel elements in different stages, one can generate different polarization circuits for the same codeword length $K = 2^n \times 3^m$. Note from the set of Equations (8)–(13) and (16)–(21) that they are different. This signifies that, as the order of kernel elements within the polarization circuit differs, so do the z -parameter values of the virtual bit channels as well as the generator matrices. Therefore, based on the required system parameter values such as K , R_d and Z , there exists an optimal choice of designing the polarization circuit or ordering the kernel elements within a multi-kernel circuit design, such that the bit channels chosen to encode the information bits have the maximum possible capacity. Within the scope of this article, amongst the various algorithms available in the literature, we would use the density evolution (DE) technique for polar code construction, as provided in the set of articles [16–18] and is also used in [4]. Note that w.r.t. to Figures 1 and 2 the size of \vec{u} and \vec{y} is constant and is equal to the codeword length. Based on R_d , a subset of $R_d \cdot K$ bits in \vec{u} is used to encode the information bits, whereas the remaining $(1 - R_d) \cdot K$ bits are used to encode the frozen bits (generally 0 s). The choice of the set of bit indices for the information and frozen bits is also determined by the DE technique.

3. Resizing Polar Codes

In this section, we provide a brief description of two downsizing techniques for polar codes, namely, puncturing and shortening. These techniques are quite well known in the research community, and multiple variations/approaches exist in the literature.

3.1. Puncturing

Puncturing of channel codes is a well-known technique to reduce the number of codeword bits that are eventually transmitted over the channel. There exist some puncturing techniques for polar codes in the literature, such as those in [19–21]. For the implementation in this article, the simplest approach of puncturing scheme, as discussed in [19], has been used.

In puncturing, the punctured codeword bits are not transmitted over the channel, i.e., the bits are entirely avoided during an ongoing transmission. The decoder typically handles such punctured bits as bits erased by the channel. Thus, the choice of bits to be punctured should be such that only the frozen bit channels at the encoder input are affected by it, because decapitating any of the information bit channels would result in an unnecessary loss of performance and degradation of the error-correcting capability of the codes on top of introducing an error floor. Ideally, the bits with the lowest capacity (or highest z -parameter values) should be targeted to minimize the loss of total capacity while puncturing. For details regarding the encoding or decoding of punctured polar codes generated for analysis in this article, the reader is recommended to refer to Section IIIA of [3] or [19].

Puncturing some bits results in degradation of total the capacity of input bit channels. This fact is justified by the channel polarization method, in which the channel capacity is conserved from input to output of the polarizing circuit. Thus, the overall reduction in capacity corresponds to the total capacity of punctured bits. Therefore, it is justified to use the least capacity bit channels for puncturing to minimize the loss of total capacity. The effective code rate of the transmitted codewords is increased by puncturing, due to the utilization of some bit channels for puncturing. This leads to a reduction in channel capacity and consequently worse error rate performance.

3.2. Shortening

Similar to puncturing, shortening is another way of downsizing channel codes. Shortening is usually applied to codes of the systematic form; however, in the case of polar codes, shortening can be easily applied to the nonsystematic form as well. There exist some shortening techniques for polar codes in the literature, such as those in [19,22]. For the implementation in this article, we focus only on the shortening of nonsystematic codes, as shown in [19], for a fair comparison to downsizing by puncturing in Section 3.1.

Similar to puncturing, in the case of shortening, one or more codeword bits are not transmitted over the channel, i.e., the bits are entirely avoided during an ongoing communication. However, contrary to puncturing, the decoder handles these nontransmitted bits as bits known with complete confidence at the receiver instead of erased bits, i.e., they are reconstructed at the receiver as a priori bits. Hence, it is advantageous to utilize bit channels with the highest capacities for shortening to minimize error in the prediction/assignment of a priori bits at the receiver. Using low-capacity channels for shortening would result in unnecessary errors owing to falsely predicted bits at the receiver, thus resulting in an error floor and loss in error-correcting capability. For details regarding the encoding or decoding of shortened polar codes generated for analysis in this article, the reader is recommended to refer to Section IIIB of [3] or [19].

Shortening some bits results in degradation of total the capacity of input bit channels, due to the absence of highest-capacity bits being shortened; some lower-capacity bits need to be used for encoding the information bits, i.e., the overall capacity of information bit channels is reduced. This fact is justified by the channel polarization method, in which the channel capacity is conserved from input to output of the polarizing circuit. The effective

code rate of the transmitted codewords is increased by shortening. This leads to a reduction in channel capacity and consequently worse error rate performance.

4. Analysis of Optimal Design Techniques

In this section, we perform a comparative analysis of polar codes over different sets of parameter settings and assess them based on the error rate performances (Section 4.1) and complexities (Section 4.2). From the discussions in Sections 2 and 3, we can conclude that the following techniques for designing polar codes are at our disposal:

- A 2×2 circuit of kernel size 2;
- A 3×3 circuit of kernel size 3;
- Multi-kernel circuit composed of 2×2 and 3×3 circuits;
- Puncturing;
- Shortening.

In [3], we analyzed the performance of downsized (punctured/shortened) codewords generated by the 2×2 kernel circuit, compared to codewords generated by the 3×3 kernel circuit and vice versa. In this article, we analyze the performance of downsized (punctured/shortened) codewords generated by either 2×2 or 3×3 kernel circuits to a corresponding multi-kernel design. We use two scenarios with differing codeword lengths for such analysis. A summary of the corresponding parameter settings is provided in Table 1. We perform the analysis over AWGNC and $Z = 0.5$, which corresponds to $designSNR_{dB} \approx -1.6$ dB as per the metric for polar code construction shown in [23].

Table 1. Parameter settings for the simulation environment.

	Scenario 1			Scenario 2		
	N	M	K	N	M	K
Encoding/Decoding length	1024	729	648	512	729	432
Number of downsized bits	376	81	0	80	297	0
Z	0.5			0.5		
Modulation Scheme	BPSK			BPSK		
Coderates R_d	{1/4, 1/2, 3/4}			{1/4, 1/2, 3/4}		

From the results in [3] and per the specifications provided in [2] for using downsized polar codes for encoding over the 5G control channels, we observed that for low code rates, puncturing tends to be the preferable downsizing option, while for high code rates, shortening tends to be the preferable downsizing option. In [24], this aspect has been validated for multi-kernel polar codes as well, using a downsizing type selection (DTS) parameter, also based on DE, which provides a more accurate prediction of the optimality of choice of downsizing technique based on the desired system parameter settings.

4.1. Error Rate Performance

In this section, we perform the comparative analysis using error rate performance as the quantifier of optimality of code design and performance.

4.1.1. Scenario 1

In this subsection, we examine the error rate performance of multi-kernel polar codes of length $K = 648$, compared to polar codewords of length $N = 1024$ generated using only the 2×2 kernel circuit (as in Figure 1 from [3]) and downsized (punctured/shortened) by 376 bits, as well as polar codewords of length $M = 729$ generated using only the 3×3 kernel circuit (as in Figure 2 from [3]) and downsized (punctured/shortened) by 81 bits. Code rate values of $R_d = 1/4, 1/2$, and $3/4$, i.e., low, half, and high code rates are used for a comparative analysis of the error rate performance. Using the DE technique, mentioned in Section 2.3, we determine the optimal configuration for the arrangement of kernel designs

within the polar code circuits for a given code rate value, i.e., $G_{648}^{R_d} = perm(f(G))$. Details of the figures corresponding to respective R_d and $f(G)$ are provided in Table 2.

Table 2. Simulation cases for Scenario 1.

	Figure 3	Figure 4	Figure 5
Coderates R_d	1/4	1/2	3/4
Arrangement $G_{648}^{R_d}$	$G_3^{\otimes 4} \otimes G_2^{\otimes 3}$	$G_2^{\otimes 2} \otimes G_3 \otimes G_2 \otimes G_3^{\otimes 3}$	$G_2^{\otimes 2} \otimes G_3 \otimes G_2 \otimes G_3^{\otimes 3}$

From Figures 3–5, the following conclusions can be made:

1. For the low code rate, $R_d = 1/4$, punctured $N = 1024$ codewords outperform the shortened ones. The multi-kernel $K = 648$ codewords outperform all but punctured $N = 1024$ codewords, with a ≈ 0.5 dB difference. One interesting observation is w.r.t. downsized $M = 729$ codewords. Although at low R_d , one would expect the punctured ones to be better than the shortened ones, the performance difference is marginal with shortened ones narrowly outperforming the punctured ones. This corresponds to the observations made in [3], where shortened $M = 729$ codewords outperform the punctured ones even at low code rates.
2. For the half code rate, $R_d = 1/2$, shortened $N = 1024$ and $M = 729$ codewords outperform the respective punctured ones. The multi-kernel $K = 648$ codewords outperform nearly all but shortened $N = 1024$, with just a ≈ 0.25 dB difference.
3. For the high code rate, $R_d = 3/4$, shortened $N = 1024$ and $M = 729$ codewords outperform the respective punctured ones. The multi-kernel $K = 648$ codewords outperform all but shortened $N = 1024$ codewords, with just a ≈ 0.25 dB difference.

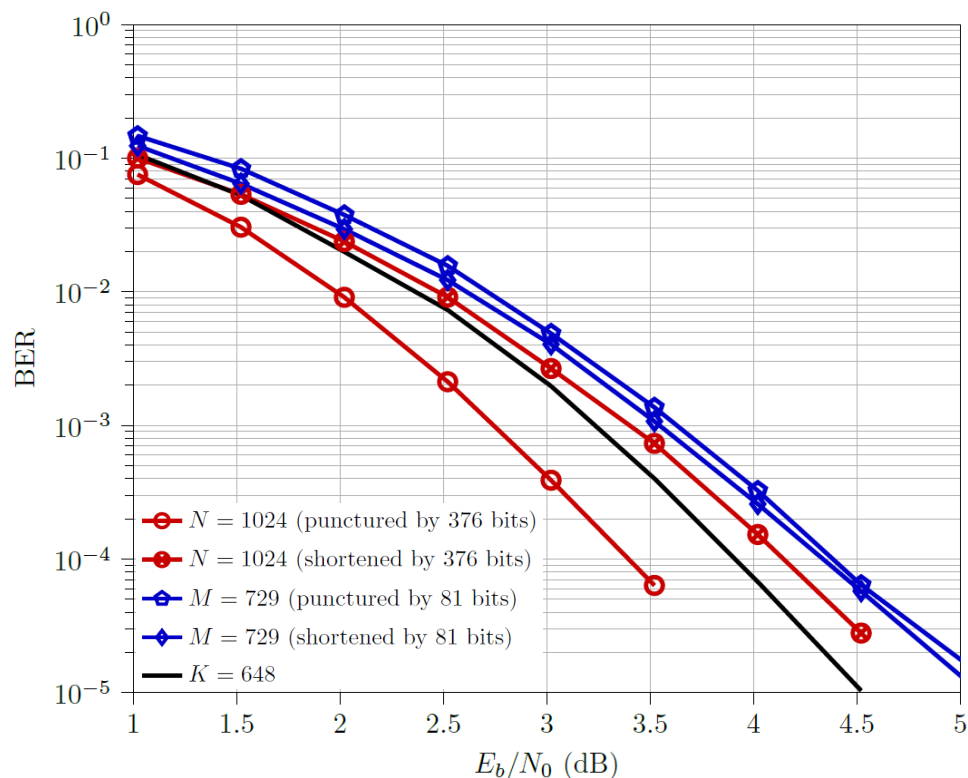


Figure 3. BER performance of Scenario 1 for $R_d = 1/4$.

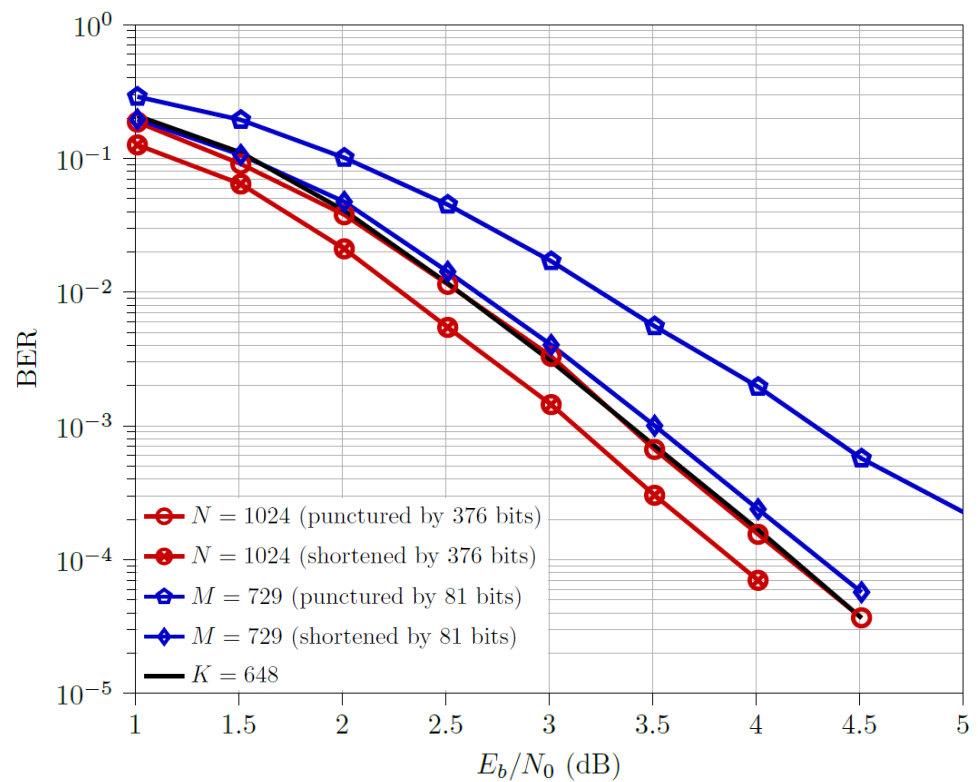


Figure 4. BER performance of Scenario 1 for $R_d = 1/2$.

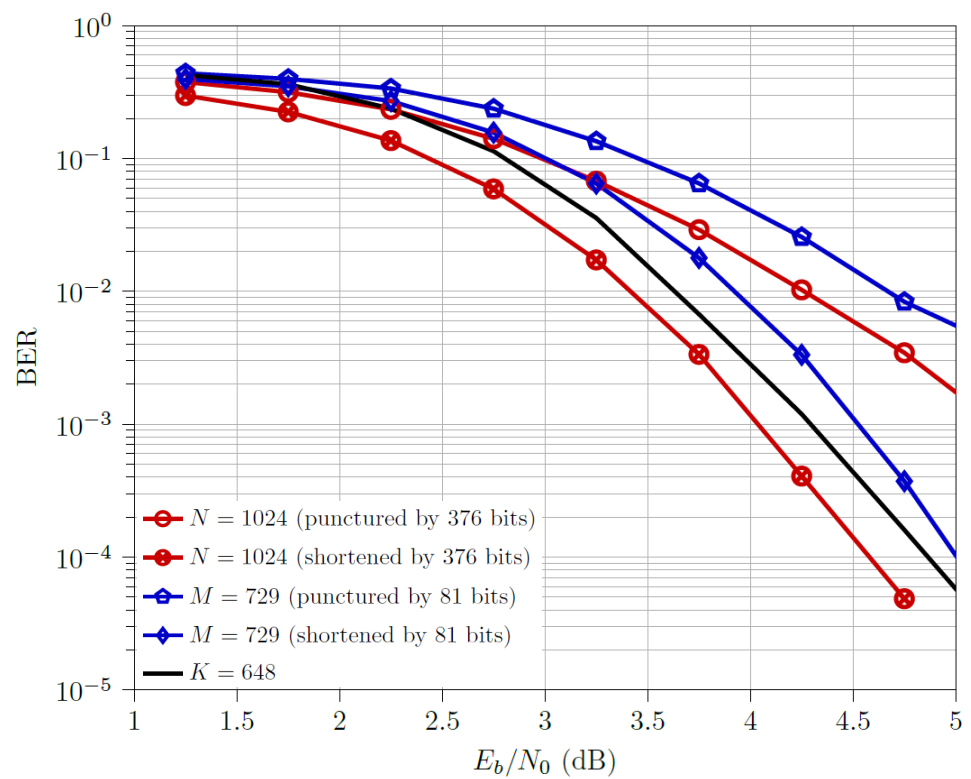


Figure 5. BER performance of Scenario 1 for $R_d = 3/4$.

From all three plots in Scenario 1, we observe that with an exception at the low code rate, the assumption that puncturing for low code rates and shortening for high code rates are better options for downsizing holds true. When the number of bits used for downsizing is relatively high, this assumption has higher validity, and the performance difference

between the corresponding punctured and shortened codewords is higher. The downsized N codewords are always better than their corresponding downsized M codewords. At low and half code rates, both the punctured and shortened N codewords are better than both the downsized M codewords. Although the number of downsized bits for N codewords is much higher than M codewords in this scenario, it follows the observations in [3] about 2×2 kernel being a better design choice than the 3×3 kernel. A multi-kernel code design is evidently a good design choice when the corresponding K codeword length is desirable.

4.1.2. Scenario 2

In this subsection, we investigate the error rate performance of multi-kernel polar codes of length $K = 432$, compared to polar codewords of length $N = 512$ generated using only the 2×2 kernel circuit (as in Figure 1 from [3]) and downsized (punctured/shortened) by 80 bits, as well as polar codewords of length $M = 729$ generated using only the 3×3 kernel circuit (as in Figure 2 from [3]) and downsized (punctured/shortened) by 297 bits. Code rate values as in Section 4.1.1, i.e., $1/4$, $1/2$, and $3/4$ are used. Furthermore, using the DE technique, the optimal configuration $f(G)$ for the multi-kernel design of $G_{432}^{R_d} = perm(f(G))$ is determined. Details of the figures corresponding to respective R_d and $f(G)$ are provided in Table 3.

Table 3. Simulation cases for Scenario 2.

	Figure 6	Figure 7	Figure 8
R_d	1/4	1/2	3/4
$G_{432}^{R_d}$	$G_3^{\otimes 2} \otimes G_2 \otimes G_3 \otimes G_2^{\otimes 3}$	$G_2^{\otimes 2} \otimes G_3 \otimes G_2 \otimes G_3 \otimes G_2 \otimes G_3$	$G_2^{\otimes 2} \otimes G_3 \otimes G_2^{\otimes 2} \otimes G_3^{\otimes 2}$

From Figures 6–8, the following conclusions can be made:

1. For the low code rate, $R_d = 1/4$, punctured $N = 1024$ and $M = 729$ codewords outperform the corresponding shortened ones. This phenomenon of punctured M being better than their shortened counterparts had not been observed in any plots of [3]. Note that here, downsizing of $M = 729$ codewords was carried out by 297 bits, whereas in [3], downsizing was performed by 217 bits, indicating that a higher number of downsized bits improves the prediction of using code rate value for determining optimal downsizing technique, as observed in Scenario 1 as well. The multi-kernel $K = 432$ codewords outperform both the downsized $M = 729$ codewords but are outperformed by both the downsized $N = 512$ codewords, with a performance gap of ≈ 0.5 dB from the optimal design (i.e., punctured $N = 512$ codewords).
2. For the half code rate, $R_d = 1/2$, shortened $N = 512$ and $M = 729$ codewords outperform the respective punctured ones. The multi-kernel $K = 432$ codewords outperform both the downsized $M = 729$ codewords but are outperformed by both the downsized $N = 512$ codewords, with a performance gap of ≈ 0.5 dB from the optimal design (i.e., shortened $N = 512$ codewords).
3. For the high code rate, $R_d = 3/4$, shortened $N = 512$ and $M = 729$ codewords outperform the respective punctured ones. The multi-kernel $K = 432$ codewords outperform all but the shortened $N = 512$ codewords, with a ≈ 0.3 dB difference.

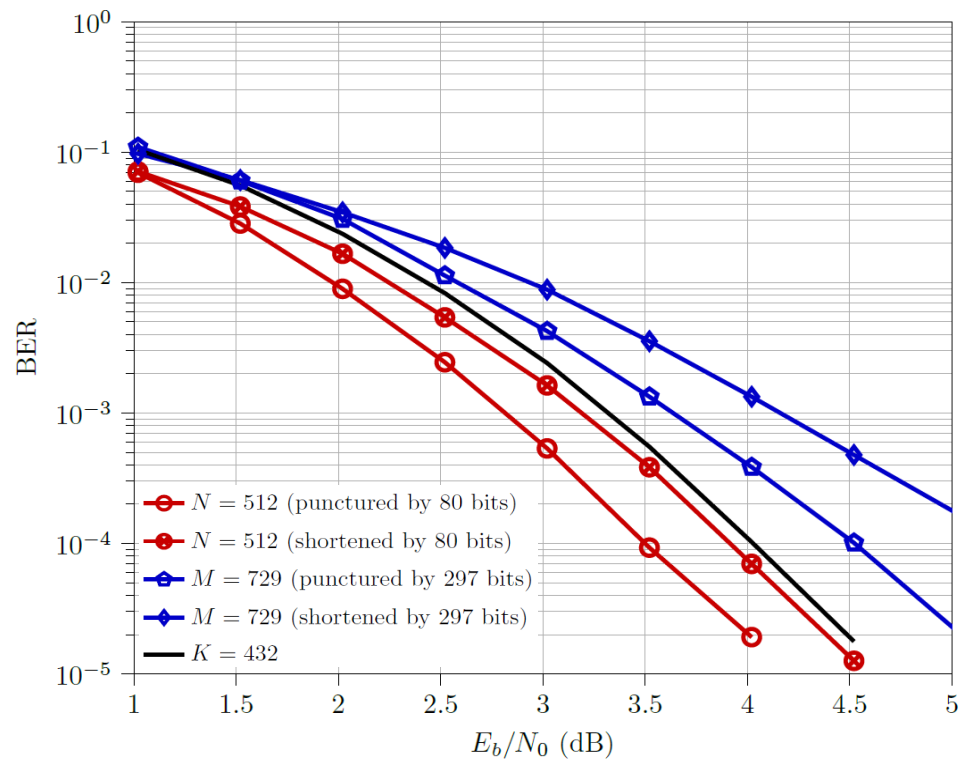


Figure 6. BER performance of Scenario 2 for $R_d = 1/4$.

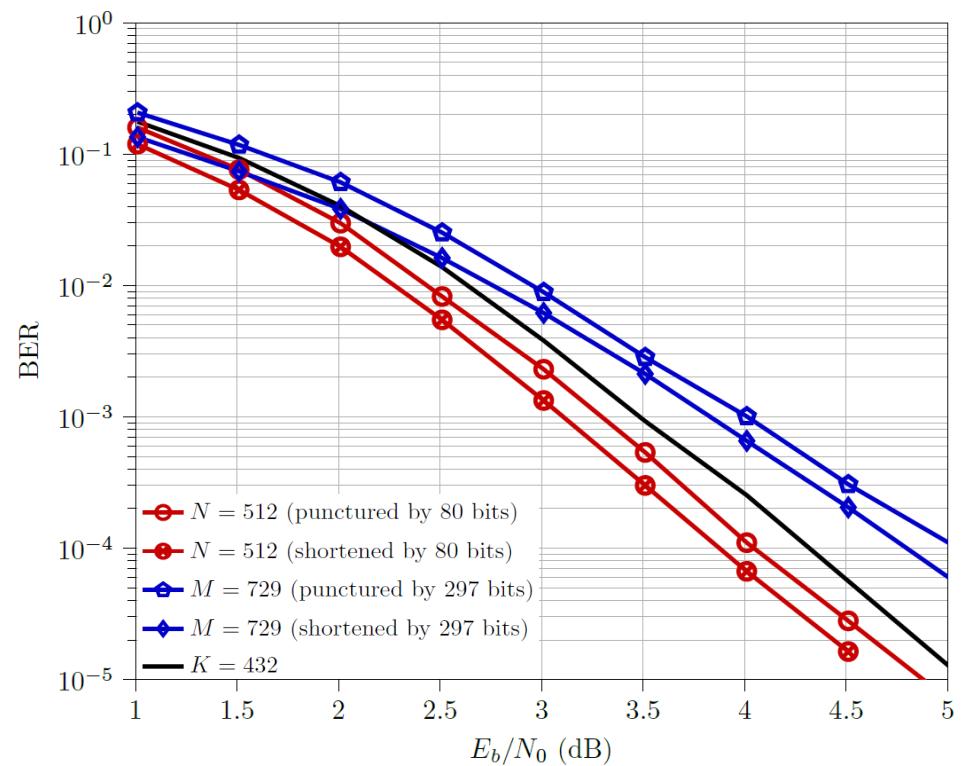


Figure 7. BER performance of Scenario 2 for $R_d = 1/2$.

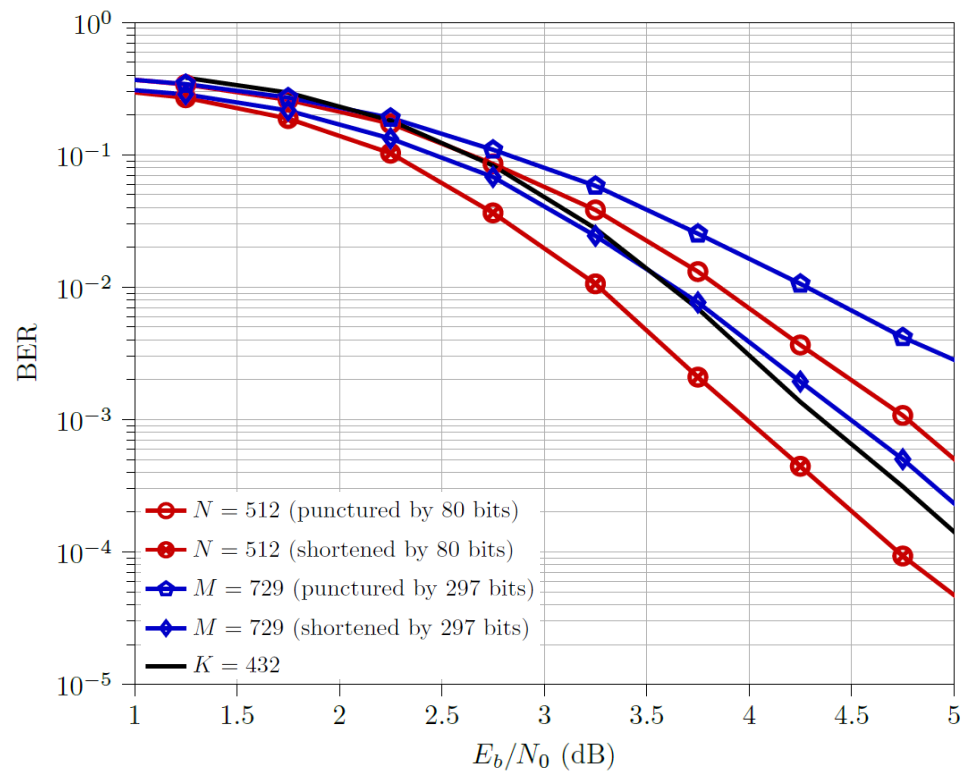


Figure 8. BER performance of Scenario 2 for $R_d = 3/4$.

From all three plots in Scenario 2, we observe that the assumption of downsizing by puncturing for low code rates and by shortening for high code rates being better options holds true. The downsized N codewords are always better than their corresponding downsized M codewords. At low and half code rates, both the punctured and shortened N codewords are better than both the downsized M codewords. In spite of the fact that, in this scenario, the number of downsized bits for $M = 729$ codewords is much higher than $N = 512$ codewords, the downsized $N = 512$ codewords perform better than the downsized $M = 729$ ones, thus following the observations in [3] about 2×2 kernel being a better design choice than the 3×3 kernel. A multi-kernel code design is evidently a good design choice when the corresponding K codeword length is desirable.

4.2. Computational Complexity

In this section, we discuss the complexity of polar code design for both scenarios 1 and 2. The encoding and decoding complexity of 2×2 polar codes is given as $\mathcal{O}(N \cdot \log_2 N)$, i.e., dependent on codeword length and number of stages within the polarization circuit [11]. Similarly, it can be quantified for M (for 3×3 circuit) and K (for multi-kernel circuit). Effectively, at the encoder, the XOR gate (as seen in Figures 1 and 2) generates a modulo 2 sum operation. At the decoder, this corresponds to one boxplus and one summation operation of the log-likelihood ratios (LLRs). For simplicity, we consider one XOR gate contributing one unit of complexity to the code design. Using this convention, we state that the complexity of polar circuits shown in Figures 1 and 2 is 7 units.

The complexities of polar codes in scenarios 1 and 2 are tabulated in Table 4. Note that higher values of complexity units indicate a more complex unit, i.e., worse design choice. We observe that the complexity is directly proportional to the number of downsized bits. This is because, although the downsized (punctured/shortened) bits are not transmitted/received across the channel, they still need to be encoded and decoded at the transmitter and receiver, respectively. Thus, if a corresponding codeword length is required, such that a multi-kernel circuit can directly generate it, then using a design based on downsizing from a single kernel circuit results in a more complex design. Thus, a less

complex alternative to downsizing single kernel polar codewords at a small performance degradation could be an acceptable compromise if it suffices the desired quality-of-service (QoS).

Table 4. Complexity comparison of the polar code designs.

	Scenario 1			Scenario 2		
	N	M	K	N	M	K
Encoding/Decoding length	1024	729	648	512	729	432
Number of downsized bits	376	81	0	80	297	0
Complexity units	5120	2916	2700	2304	2916	1728

4.3. Assessment

Based on the analysis performed in Sections 4.1 and 4.2, we can derive some conclusions regarding the comparisons of error rate performances and complexities of the polar code design techniques implemented for simulation scenarios 1 and 2. The corresponding summaries of observations are provided in Tables 5 and 6, respectively. We use the multi-kernel design as a reference (0), with ++, +, −, or -- indicating better to worse choices.

Table 5. Tabulated observations from Scenario 1.

	Punctured N	Shortened N	K	Punctured M	Shortened M
BER at $R_d = 1/4$	++	−	0	--	--
BER at $R_d = 1/2$	0	+	0	--	0
BER at $R_d = 3/4$	--	+	0	--	−
Complexity	--		0	−	

Table 6. Tabulated observations from Scenario 2

	Punctured N	Shortened N	K	Punctured M	Shortened M
BER at $R_d = 1/4$	++	+	0	--	--
BER at $R_d = 1/2$	+	++	0	--	−
BER at $R_d = 3/4$	−	+	0	--	0
Complexity		−	0	--	

Clearly, the desired code rate is a key parameter used to determine the optimal choice of polar code design. Nevertheless, if a codeword length that can be obtained by a multi-kernel design is required, then the multi-kernel design is a good approach, providing near-optimal BER performance with reduced complexity. Additionally, it extends the scope of polar codeword lengths that can be generated without the requirement of any downsizing mechanism.

For instance, for code rates $R_d = 1/2$ and $3/4$ in Scenario 1, multi-kernel design is able to reach the performance of the optimal (shortened) 2×2 kernel design, with a gap of 0.25 dB with 47.3% complexity reduction (refer to Figures 4 and 5). Simultaneously, it outperforms all other design techniques.

5. Future Work

Within the scope of research conducted for this article, we have polarization kernels of sizes 2 and 3 to generate single and multi-kernel polarization circuit designs. Nevertheless, the idea of channel polarization is not limited to only 2×2 and 3×3 kernel sizes. It would be interesting to study the comparative error rate performance with higher kernel sizes such as 5, 7, etc. or even non-prime standalone kernel sizes (i.e., not composed of smaller divisible kernel circuits), examples of which are available in [8]. Additionally, these

higher-ordered kernel sizes can be used to generate polar codes of an even wider range of codeword lengths without the need for resizing.

In this article, we only used the SC decoder to assess the comparative BER performance; however, better variations exist such as the SC List (shown in [25]) or Flip (shown in [26]) decoders. Soft decoding techniques such as belief propagation (BP) have also been developed for polar codes. Recently, some interesting polar code construction techniques have been proposed, such as using the information bottleneck method (shown in [27]) or deep learning (shown in [28]). Additionally, some efficient simulation methods to analyze polar codes performance via importance-sampling techniques exist, such as those provided in [29,30]. Analyzing the error rate performance over these different techniques for downsized and multi-kernel designs can help identify the effects of specific system parameter settings. This, in turn, would aid in the generalization of polar code design and standardized use for practical applications.

6. Conclusions

In this article, we have discussed downsizing techniques such as puncturing, shortening, and using a different kernel size (such as 3) to design single- or multi-kernel polar codes of length other than 2^n form. The disadvantage of a limited choice of codeword lengths is a major obstacle for utilizing polar codes over a wide range of applications. Thus, when resizing is necessary, depending on availability, one may use alternatives to the classical 2×2 polar circuit design such as higher-ordered kernel sizes, by themselves or within a multi-kernel framework, to generate polar codes without the need for resizing at all. Additionally, certain design aspects may prove to be optimal for given underlying system parameter settings. The threshold value of $R_d = 7/16$, mentioned in [2], used for determining the downsizing choice, is not a very accurate assumption (also observed in [24]), and one also needs to take into account other parameters such as codeword length, the number of downsized bits, channel conditions, etc. as well to determine optimal code design for the desired application. In the case of multi-kernel codes, the order of kernel elements amongst the stages within the polar circuit is also a key design factor to be accounted for optimal performance as the selection of sets of information/frozen bits would vary for a different order of kernels. Using a multi-kernel design provides a higher degree of freedom for choosing codeword lengths without the requirement for downsizing while providing near-optimal performance.

Author Contributions: Conceptualization, S.S. and M.A.; methodology, S.S. and M.A.; software, S.S.; validation, S.S.; formal analysis, S.S.; investigation, S.S.; resources, M.A.; data curation, S.S.; writing—original draft preparation, S.S.; writing—review and editing, S.S. and M.A.; visualization, S.S. and M.A.; supervision, M.A.; project administration, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Abbreviations

The following abbreviations are used in this manuscript:

AWGNC	Additive White Gaussian Noise Channel
BEC	Binary Erasure Channel
BER	Bit-Error Rate
BPSK	Binary Phase Shift Keying
DE	Density Evolution
DTS	Downsizing Type Selection
LLR	Log-Likelihood Ratio
QoS	Quality of Service
SC	Successive Cancellation

References

1. Bioglio, V.; Condo, C.; Land, I. Design of Polar Codes in 5G New Radio. *IEEE Commun. Surv. Tutor.* **2020**, *23*, 29–40. [CrossRef]
2. 3GPP TS 38.212 V16.5.0. Technical Specification, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; Multiplexing and Channel Coding (Release 16). 2021. Available online: https://www.3gpp.org/ftp/Specs/archive/\38_series/38.212/ (accessed on 28 May 2021).
3. Saha, S.; Adrat, M. Versatile Polar Codes with different Kernel Sizes and Rate Matching approaches. In Proceedings of the 14th International Conference on Signal Processing and Communication Systems (ICSPCS), Adelaide, SA, Australia, 14–16 December 2020; pp. 1–7.
4. Gabry, F.; Bioglio, V.; Land, I.; Belfiore, J. Multi-kernel construction of polar codes. In Proceedings of the IEEE International Conference on Communication (ICC), Paris, France, 21–25 May 2017; pp. 761–765.
5. Benammar, M.; Bioglio, V.; Gabry, F.; Land, I. Multi-Kernel Polar Codes: Proof of Polarization and Error Exponents. In Proceedings of the IEEE Information Theory Workshop (ITW), Kaohsiung, Taiwan, 6–10 November 2017; pp. 101–105.
6. Bioglio, V.; Gabry, F.; Land, I.; Belfiore, J. Minimum-Distance Based Construction of Multi-Kernel Polar Codes. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Singapore, 4–8 December 2017; pp. 1–6.
7. Bioglio, V.; Land, I.; Gabry, F.; Belfiore, J. Flexible design of Multi-Kernel Polar Codes by reliability and distance properties. In Proceedings of the IEEE 10th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Hong Kong, China, 3–7 December 2018; pp. 1–5.
8. Bioglio, V.; Land, I. On the Marginalization of Polarizing Kernels. In Proceedings of the IEEE 10th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), Hong Kong, China, 3–7 December 2018; pp. 1–5.
9. Bioglio, V.; Land, I.; Condo, C. Improved Hybrid Design of Polar Codes and Multi-Kernel Polar Codes. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019; pp. 1947–1951.
10. Bioglio, V.; Gabry, F.; Land, I.; Belfiore, J. Multi-Kernel Polar Codes: Concept and Design Principles. *IEEE Trans. Commun.* **2020**, *68*, 5350–5362. [CrossRef]
11. Arikan, E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 3051–3073. [CrossRef]
12. Mahdaviifar, H.; El-Khamy, M.; Lee, J.; Kang, I. Polar Coding for Bit-Interleaved Coded Modulation. *IEEE Trans. Veh. Technol.* **2016**, *65*, 3115–3127. [CrossRef]
13. Zhang, L.; Zhang, Z.; Wang, X. Polar code with block-length $N = 3^n$. In Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP), Huangshan, China, 25–27 October 2012; pp. 1–6.
14. Cheng, L.; Zhou, W.; Zhang, L. Hybrid Multi-Kernel Construction of Polar Codes. In Proceedings of the IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 28 April–1 May 2019; pp. 1–5.
15. Xia, C.; Tsui, C.-Y.; Fan, Y. Construction of Multi-Kernel Polar Codes With Kernel Substitution. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1879–1883. [CrossRef]
16. Mori, R.; Tanaka, T. Performance of Polar Codes with the Construction using Density Evolution. *IEEE Commun. Lett.* **2009**, *13*, 519–521. [CrossRef]
17. Mori, R.; Tanaka, T. Performance and construction of polar codes on symmetric binary-input memoryless channels. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Seoul, Korea, 28 June–4 July 2009; pp. 1496–1500.
18. Hassani, S.; Mori, R.; Tanaka, T.; Urbanke, R. Rate-Dependent Analysis of the Asymptotic Behavior of Channel Polarization. *IEEE Trans. Inf. Theory* **2013**, *59*, 2267–2276. [CrossRef]
19. Gabry, F.; Bioglio, V.; Land, I. Low-Complexity Puncturing and Shortening of Polar Codes. In Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
20. Wang, R.; Liu, R. A Novel Puncturing Scheme for Polar Codes. *IEEE Commun. Lett.* **2014**, *18*, 2081–2084. [CrossRef]
21. Zhang, L.; Zhang, Z.; Wang, X.; Yu, Q.; Chen, Y. On the puncturing patterns for punctured polar codes. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Honolulu, HI, USA, 30 June–5 July 2014; pp. 121–125.
22. Miloslavskaya, V. Shortened Polar Codes. *IEEE Trans. Inf. Theory* **2015**, *61*, 4852–4865. [CrossRef]
23. Vangala, H.; Viterbo, E.; Hong, Y. A Comparative Study of Polar Codes Constructions for the AWGN Channel. *arXiv* **2015**, arXiv:1501.02473v1.
24. Saha, S.; Adrat, M. Novel Multi-Parameter based Rate-Matching of Polar Codes. In Proceedings of the International Conference on Military Communications and Information Systems (ICMCIS), Oeiras, Portugal, 4–5 May 2021.
25. Tal, I.; Vardy, A. List Decoding of Polar Codes. *IEEE Trans. Inf. Theory* **2015**, *61*, 2213–2226. [CrossRef]
26. Afisiadis, O.; Balatsoukas-Stimming, A.; Burg, A. A low-complexity improved successive cancellation decoder for polar codes. In Proceedings of the 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2–5 November 2014; pp. 2116–2120.
27. Stark, M.; Shah, A.; Bauch, G. Polar code construction using the information bottleneck method. In Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; pp. 7–12.

28. Ebada, M.; Cammerer, S.; Elkelesh, A.; ten Brink, S. Deep Learning-based Polar Code Design. *arXiv* **2019**, arXiv:1909.12035v2.
29. Romano, G.; Ciuonzo, D. Minimum-Variance Importance-Sampling Bernoulli Estimator for Fast Simulation of Linear Block Codes over Binary Symmetric Channels. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 486–496. [[CrossRef](#)]
30. Romano, G.; Drago, A.; Ciuonzo, D. Sub-optimal importance sampling for fast simulation of linear block codes over BSC channels. In Proceedings of the 8th International Symposium on Wireless Communication Systems, Aachen, Germany, 6–9 November 2011; pp. 141–145.