An Efficient Puncturing Scheme for Polar Codes Based on Fixed Information Set

Zijian Qin[1], Yangcan Zhou[1], Jingjing Zhu[1], and Zhongfeng Wang[1,2]
[1]School of Electronic Science and Engineering, Nanjing University, China
[2]School of Integrated Circuits, Sun Yat-sen University, China
{qingjian, yczhou, jingjingzhu}@smail.nju.edu.cn, zfwang@nju.edu.cn

***Abstract*—For traditional polar codes, the code length is limited to a power of two, due to the use of the 2×2 kernel matrix. To obtain flexible code lengths, puncturing is widely considered for polar codes. In this paper, we propose an efficient puncturing scheme based on fixed information set (FIS) of the mother code. First, we estimate the qualities of bit channels using the Gaussian approximation after puncturing. In addition, the search scope of punctured bits in the codeword is significantly reduced to the number of frozen bits. Furthermore, a computationally efficient method for selecting the most promising puncturing pattern is employed. Moreover, a low-complexity recursive method is leveraged to construct the puncturing pattern. Simulation results demonstrate that the proposed puncturing scheme outperforms the state-of-the-art FIS-based puncturing schemes remarkably in terms of error-correction performance.**

***Index Terms*—Polar codes, puncturing, rate-compatible, Gaussian approximation.**

I. Introduction

Polar codes, proposed by Arıkan [1], have been proved to achieve the capacity of arbitrary symmetric binary-input discrete memoryless channels (B-DMCs) under the successive cancellation (SC) decoding with infinite code length. To mitigate the performance loss of the SC decoder at short to moderate code lengths, the SC list (SCL) and the cyclic-redundancy-check (CRC) aided SCL (CA-SCL) decoding algorithms have been proposed [2] [3], making polar codes surpass the low-density parity-check codes and turbo codes in error-correction performance.

Despite the excellent performance of polar codes, the code length is limited to a power of 2 due to the use of the 2×2 kernel matrix. To make the code length flexible, punctured polar codes have been investigated, for which certain bits are removed from the original codeword and not transmitted through the channel. However, puncturing leads to the degradation of bit channels, with some even turning to zero-capacity channels. The set containing the indices of punctured bits is referred to as the puncturing pattern. It is necessary to cautiously select reasonable puncturing patterns to avoid catastrophic loss in error-correction performance.

The puncturing methods proposed in [4] [5] select the information set after determining the puncturing pattern, making the information set vary between puncturing and non-puncturing modes. However, for certain communication systems, the ability to switch between puncturing and non-puncturing modes, as well as reusing the existing encoder and decoder for punctured polar codes, is highly needed [6]. Thus, puncturing schemes based on fixed information set (FIS) are of great value. In view of this, several FIS-based puncturing schemes were proposed. The worst quality puncturing (WQP) [6] designs the puncturing pattern based on the original bit channel error probabilities, which minimizes overall quality loss of bit channels by choosing bits with the largest error probabilities. The information set approximation puncturing (ISAP) [7] sets some guard bits to protect information bits from the impact of the puncturing, and bits within the frozen set except those designated as guard bits can be punctured. The puncturing scheme in [8] calculates reliability scores of candidate puncturing patterns and selects the best one accordingly. Additionally, the method in [8] leverages a recursive method to generate the puncturing pattern.

To further improve the performance of punctured polar codes, an efficient FIS-based puncturing scheme is proposed in this paper. Punctured bits can be treated as if they were transmitted through completely noisy channels [9], allowing us to utilize the Gaussian approximation (GA) [10] to estimate the qualities of bit channels after puncturing. In the previous work [9], the search scope for punctured bits is reduced from the number of source bits to the number of frozen bits. However, the search in [9] is performed on source vector bits. For each bit being examined in the source vector, a complicated mapping is needed to locate the corresponding punctured bit in the codeword. On the contrary, in this paper, the search is directly conducted on codeword bits. Moreover, by indirectly comparing the union bound of error-correction performance to circumvent complex mathematical operations, and recursively constructing the puncturing pattern to further reduce the complexity, an efficient puncturing scheme is finally developed. Simulation results show that the proposed method substantially outperforms the state-of-the-art FIS-based puncturing schemes in terms of error-correction performance.

II. Preliminaries

A. Punctured Polar Codes

A polar code is defined by a three-tuple (N, K, $\mathcal{I}$), where $N = 2^n$ ($n \in \mathbb{N}$) is the code length, K is the information length, and $\mathcal{I}$ denotes the set of indices of information bits. Let the binary vector $\boldsymbol{u} = (u_0, u_1, ..., u_{N-1})$ and $\boldsymbol{x} = (x_0, x_1, ..., x_{N-1})$ represent the source vector and the codeword, respectively. Then $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}_N$, where $\boldsymbol{G}_N = \boldsymbol{F}^{\otimes n}$

is the generator matrix, and $\boldsymbol{F}^{\otimes n}$ denotes the nth Kronecker product of $\boldsymbol{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. The source vector $\boldsymbol{u}$ can be divided into the information part and the frozen part. For $i \in \mathcal{I}$, u_i is an information bit; for $i \in \mathcal{F} = \mathcal{I}^c$ (the complement set of $\mathcal{I}$ with respect to the set $\{0, 1, ..., N-1\}$), u_i is a frozen bit. The value of the frozen bit is commonly set to zero, and the frozen set $\mathcal{F}$ is known by both the encoder and the decoder.

To get a non power of two code length, suppose m bits are punctured from the codeword. Consequently, the transmitted length is $M = N - m$. Let $\mathcal{P}$ represent the puncturing pattern. For $i \in \mathcal{P}$, x_i is punctured from the codeword and not transmitted. Let y be a received symbol. Denote $L(y)$ as the Log-likelihood ratio (LLR) of y and $(L(y_0), L(y_1), ..., L(y_{N-1}))$ as the LLR vector input into the decoder. Since the punctured bit x_i $(i \in \mathcal{P})$ is not transmitted through the channel, the decoder lacks information about the corresponding y_i, thus $L(y_i)$ is set to zero.

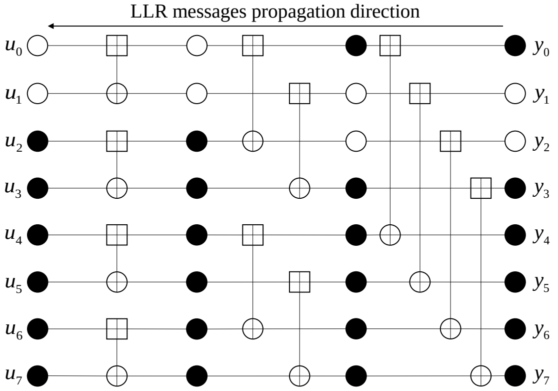


Fig. 1. Decoding structure for an $(8, 4)$ polar code. White circles represent nodes of which the LLR messages become zero due to puncturing.

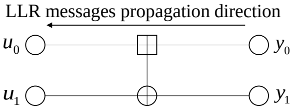


Fig. 2. Basic decoding structure.

Fig. 1 shows the decoding structure of an $(8, 4)$ polar code, which is recursively constructed using the basic structure as shown in Fig. 2. Note that $\boxplus$ and $\oplus$ represent f function and g function, respectively. Without loss of generality, assume an all-zero vector is transmitted. Consider the basic structure in Fig. 2, f and g functions are given as follows:

$$f: \; L(u_0) = 2\tanh^{-1}(\tanh\frac{L(y_0)}{2}\tanh\frac{L(y_1)}{2}), \quad (1)$$

$$g: \; L(u_1) = L(y_0) + L(y_1). \quad (2)$$

If either $L(y_0) = 0$ or $L(y_1) = 0$, then $L(u_0) = 0$ but $L(u_1) \neq 0$; if $L(y_0) = L(y_1) = 0$, then $L(u_0) = L(u_1) = 0$. Puncturing results in all bit channels deteriorating to some extent [11]. As LLR messages are computed and propagated from right to left through the basic structure, some bit channels will receive zero-LLR messages, turning to zero-capacity channels. This phenomenon contributes to the degradation in error-correction performance of punctured polar codes. *Lemma* 1 shows the relationship between the number of punctured bits and the number of zero-capacity bit channels, which was introduced and proved in [9].

Lemma 1: If m bits are punctured, there will be exactly m bit channels degrading to zero-capacity.

B. Gaussian Approximation

Due to the phenomenon of the channel polarization [1], certain bit channels are more reliable than others. To determine the information set $\mathcal{I}$ which contains K most reliable bit channels, the GA [10] is often used when the additive white Gaussian noise (AWGN) channel is adopted. Considering the binary phase shift keying (BPSK) modulation, for the AWGN channel with noise variance σ^2, the LLR of the received symbol y is calculated as $L(y) = \frac{2y}{\sigma^2}$. Assume an all-zero codeword is transmitted, then $L(y)$ follows a Gaussian distribution with mean $\frac{2}{\sigma^2}$ and variance $\frac{4}{\sigma^2}$, denoted as $L(y) \sim \mathcal{N}(\frac{2}{\sigma^2}, \frac{4}{\sigma^2})$. Punctured bits are never actually transmitted, but they can be viewed as if they were sent through a completely noisy channel with $\sigma \to \infty$. Therefore, for $i \in \mathcal{P}$, $L(y_i) \sim \mathcal{N}(0, 0)$.

Let $E[x]$ represent the expectation of the random variable x. Consider the basic structure in Fig. 2, LLR expectations are calculated as follows:

$$E[L(u_0)] = \phi^{-1}(1 - (1 - \phi(E[L(y_0)])) \cdot (1 - \phi(E[L(y_1)]))), \quad (3)$$

$$E[L(u_1)] = E[L(y_0)] + E[L(y_1)], \quad (4)$$

where the function $\phi(\cdot)$ is very complicated and can be approximated [12] as

$$\phi(t) = \begin{cases} e^{-0.4527t^{0.86}+0.0218} & 0 < t < 10, \\ \sqrt{\frac{\pi}{t}}e^{-\frac{t}{4}}(1 - \frac{10}{7t}) & t \geqslant 10. \end{cases} \quad (5)$$

Then, the error probability P_e^i of the ith bit channel is estimated [10] as

$$P_e^i = Q(\sqrt{\frac{E[L(u_i)]}{2}}), \quad (6)$$

where $Q(\zeta) = \frac{1}{\sqrt{2\pi}}\int_{\zeta}^{+\infty} e^{-\frac{z^2}{2}}\,dz$ is the Gaussian Q-function, which involves very complex integration operation. Finally, K bit channels with the lowest P_e^i values are selected to transmit information bits. Denote the union bound of block error rate (BLER) of the SC decoding as P_{eb}, which is calculated [1] as

$$P_{eb} = \sum_{i \in \mathcal{I}} P_e^i = \sum_{i \in \mathcal{I}} Q(\sqrt{\frac{E[L(u_i)]}{2}}). \quad (7)$$

C. Exhaustive Puncturing

In [9], an exhaustive puncturing scheme was proposed as outlined in Alg. 1. All bits in the codeword can be punctured, leading to $\binom{N}{m}$ candidate puncturing patterns in total. Then, the puncturing pattern with the lowest P_{eb} is selected. Clearly, the search complexity of Alg. 1 is $\binom{N}{m}$, which becomes infeasible as the code length N increases. Furthermore, to estimate the union bound P_{eb} for each candidate puncturing pattern, the Q-function must be calculated numerous times, involving complex integration operations. Given these challenges, more efficient and scalable algorithms to find the promising puncturing pattern are highly desired. In [9], a heuristic method was further proposed to reduce the complexity, which was subsequently optimized based on the FIS as the WQP algorithm in [6].

```
Algorithm 1: Exhaustive puncturing
Input: $m$, $\mathcal{I}$, $\mathcal{F}$.
Output: $\mathcal{P}$.
Step. 1 Generate all possible $\binom{N}{m}$ candidate puncturing
    patterns $\mathcal{C}_t$ $(1 \leq t \leq \binom{N}{m})$.
Step. 2 For each candidate puncturing pattern $\mathcal{C}_t$, set
    $E\left[L(y_j)\right] = 0$ for $j \in \mathcal{C}_t$. Then utilize (3), (4), and
    (6) to estimate $P_e^i$ for $i \in \mathcal{I}$.
Step. 3 Calculate the union bound $P_{eb}^t$ using (7) for
    each puncturing pattern $\mathcal{C}_t$.
Step. 4 $\mathcal{P} = \mathcal{C}_p$, where $p = \underset{t}{\operatorname{argmin}}(P_{eb}^t)$
    $(1 \leq t \leq \binom{N}{m})$.
```

III. Proposed Puncturing Scheme

A. Reduce the Search Scope

In this subsection, our aim is to narrow down the search scope for puncturing patterns. As mentioned earlier, puncturing bits from the codeword leads to reduced capacities for all bit channels, with some even becoming zero-capacity. Hence, when reducing the search scope, it is essential to prevent any information bit channel from becoming zero-capacity, or it will cause severe error-correction performance loss. Refer that $\mathcal{P}$ is the puncturing pattern, and denote $\mathcal{U}_p$ as the set of indices of bit channels that become zero-capacity due to the puncturing. Then we have *Lemma* 2.

Lemma 2: If $\mathcal{P} \subset \mathcal{F}$, then $\mathcal{U}_p \subset \mathcal{F}$.

The proof of *Lemma* 2 can be found in [6].

Lemma 2 provides a valuable insight: if we exclusively select punctured bits with indices in $\mathcal{F}$, we can ensure that none of the bit channels corresponding to information bits will become zero-capacity. By doing so, we effectively narrow down the search scope from N to $N-K$, which greatly reduces the search complexity. It is worth noting that, in Alg. 3 of [9], the search scope is also reduced from N to $N-K$. However, the search in [9] is conducted on source vector bits. For each bit being examined in the source vector, a complicated mapping is required to locate the corresponding bit in the codeword that should be punctured. In contrast, in this paper, the search is directly performed on codeword bits, thus the complicated mapping procedure is avoided, resulting in a substantial reduction in search complexity.

B. Circumvention of Complex Calculation

In Alg. 1, to choose the most promising puncturing pattern, P_{eb} need to be calculated for each candidate puncturing pattern using (7). This process involves repeatedly calculating the Q-function, which includes complex integration operations. To reduce the computational complexity, we propose an efficient method to indirectly and qualitatively compare union bounds of BLER of candidate puncturing patterns, then choose the most promising one.

Refer that the error probability of the ith bit channel is $P_e^i = Q(\sqrt{\frac{E[L(u_i)]}{2}})$. Note that Q-function is monotone decreasing. Considering all bit channels of information bits, we suppose that the one with the largest P_e^i (in other word, the smallest $E[L(u_i)]$) significantly impacts the union bound P_{eb}. This insight prompts us to investigate the relationship between the minimum $E\left[L(u_i)\right]$ and P_{eb} of each candidate puncturing pattern. To this end, an experiment is conducted using a $(1024, 540)$ polar code as mother code. To puncture the code to length $M = 900$, we randomly select 4×10^5 puncturing patterns. As depicted in Fig. 3, in general, a larger minimum LLR expectation of information bit channels of a puncturing pattern corresponds to a lower union bound of BLER.

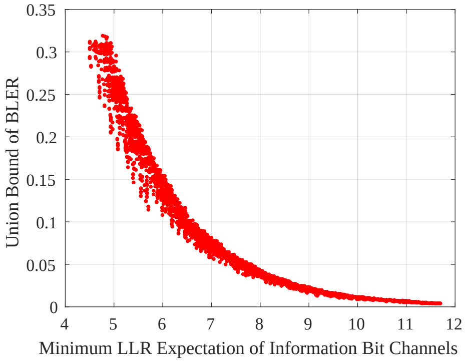


Fig. 3. Relationship between minimum LLR expectations of information bit channels and the union bound of BLER for each puncturing pattern.

C. Low-Complexity Recursive Puncturing

In this subsection, to further reduce the search complexity, the low-complexity recursive puncturing is proposed as shown in Alg. 2. Note that the approach to generate the puncturing pattern recursively was also explored in [8] and [13]. However, the criterion we use to determine each punctured bit is different from the prior works. Let $\mathcal{P}_i$ $(1 \leq i \leq m)$ be a puncturing pattern containing i punctured indices. Note that $\mathcal{P}_{i+1}$ is constructed recursively based on $\mathcal{P}_i$, while $\mathcal{P}_i$ is constructed recursively based on $\mathcal{P}_{i-1}$, and so on. When $\mathcal{P}_{i-1}$ is obtained, to derive $\mathcal{P}_i$, one more punctured bit from the codeword should be selected from the set $\mathcal{B} = \mathcal{F} \backslash \mathcal{P}_{i-1}$. Each candidate bit j gives rise to a potential puncturing pattern, denoted as $\mathcal{C}_j$. For $\mathcal{C}_j$, a vector $\boldsymbol{r}^j$ consisting of LLR expectations of information bit channels is generated using the GA. Each vector $\boldsymbol{r}^j$ is sorted in ascending order, deriving the sorted vector $\hat{\boldsymbol{r}}^j$. A comparison procedure for LLR expectations is then performed.

Algorithm 2: Low-complexity recursive puncturing

```
input : $m$, $\mathcal{I}$, $\mathcal{F}$.
output: $\mathcal{P}_i$ $(1 \leq i \leq m)$.
Initialize $\mathcal{P}_0 = \emptyset$.
for $i = 1$ to $m$ do
    $\mathcal{B} = \mathcal{F} \backslash \mathcal{P}_{i-1}$.
    for $j \in \mathcal{B}$ do
        Create $\mathcal{C}_j = \mathcal{P}_{i-1} \cup j$.
        Set $E\left[L_j(y_l)\right] = 0$ for each $l \in \mathcal{C}_j$. Calculate
          LLR expectations of information bit channels
          using (3) (4), then generate the vector $\boldsymbol{r}^j =$
          $\left(E\left[L_j(u_{l_1})\right], E\left[L_j(u_{l_2})\right], ..., E\left[L_j(u_{l_K})\right]\right)$
          $(l_1, l_2, ..., l_K \in \mathcal{I})$.
        $\hat{\boldsymbol{r}}^j \leftarrow \mathbf{sort}(\boldsymbol{r}^j)$ in ascending order.
    $\mathbb{C} = \mathcal{B}$.
    for $l = 0$ to $K - 1$ do
        $\theta = \max(\hat{r}_l^j)$ for $j \in \mathbb{C}$.
        $\mathcal{P}_i = \mathcal{C}_p$, where $p = \underset{j}{\operatorname{argmax}}(\hat{r}_l^j)$ for $j \in \mathbb{C}$.
        $\mathbb{C}_{old} = \mathbb{C}$.
        $\mathbb{C} = \{j | \hat{r}_l^j = \theta$, for $j \in \mathbb{C}_{old}\}$.
        if $|\mathbb{C}| == 1$ then
            break.
```

The puncturing pattern having the largest minimum LLR expectation of information bit channels is chosen as the most promising one. In case where more than one puncturing patterns share the same largest minimum LLR expectation of information bit channels, we proceed to compare the second minimum LLR expectation of information bit channels among these patterns, then the one having the largest second minimum LLR expectation is chosen. This comparison process continues until the most promising pattern is determined or all K elements in vectors $\hat{\boldsymbol{r}}$ have been compared. The set $\mathbb{C}$ contains indices of candidate puncturing patterns that are potential to be the most promising one. Note that the cardinality of $\mathbb{C}$ gets smaller and smaller during the comparison process.

As shown in Alg. 2, $\mathcal{P}_{m-1} \subset \mathcal{P}_m$. $\mathcal{P}_m$ relies on $\mathcal{P}_{m-1}$, and $\mathcal{P}_{m-1}$ relies on $\mathcal{P}_{m-2}$, creating a recursive relationship. Hence, the search complexity of Alg. 2 is $\sum_{i=0}^{m-1}(N-K-i) = \frac{m(2N-2K-m+1)}{2}$, while that of Alg. 1 is $\binom{N}{m} = \frac{N!}{(N-m)!m!}$. This recursive nature of Alg. 2 effectively reduces the complexity compared with the direct enumeration in Alg. 1, making Alg. 2 feasible for practical use. The comparison of search complexities with [6]–[9] is shown in Table. I.

TABLE I
COMPARISON OF SEARCH COMPLEXITY.

Method	Complexity
WQP [6]	$(N-K)\log(N-K)$
ISAP [7]	$K(M-K-\lfloor M/K \rfloor+1)$
method in [8]	$\sum_{i=0}^{m-1}(N-K-i)$
Alg. 1 [9]	$\binom{N}{m}$
Alg. 2	$\sum_{i=0}^{m-1}(N-K-i)$

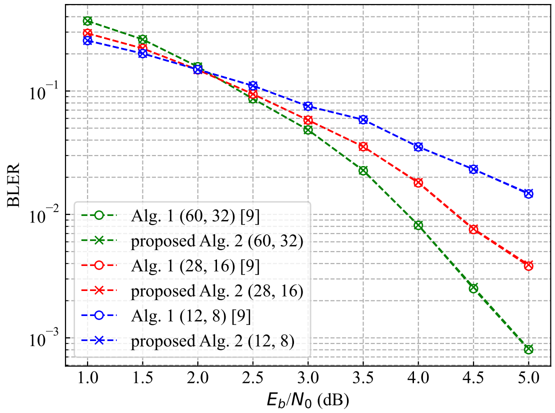


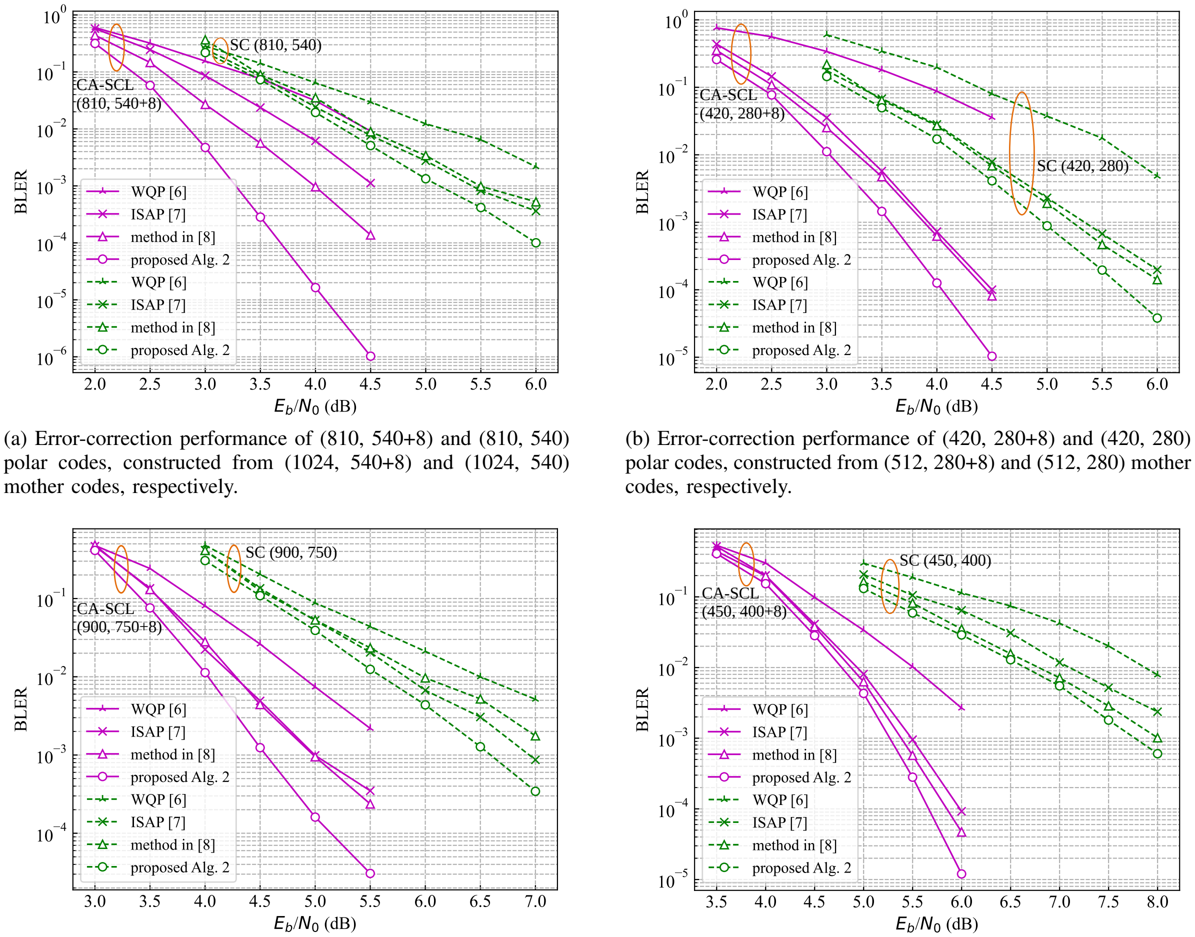
Fig. 4. Error-correction performance of $(60, 32)$, $(28, 16)$, and $(12, 8)$ polar codes, constructed from $(64, 32)$, $(32, 16)$, and $(16, 8)$ mother codes, respectively.

IV. Simulation Results

In this section, we compare the error-correction performance of Alg. 2 with the state-of-the-art FIS-based puncturing schemes, i.e., Alg. 1 [9], WQP [6], ISAP [7], and the method in [8]. The simulation is carried out on the AWGN channel and the BPSK modulation. At each simulation point, at least 100 erroneous blocks are collected.

Since Alg. 1 [9] performs the exhaustive search, the punctured code constructed using Alg. 1 ashieves the optimal performance. To prove the effectiveness of the proposed Alg. 2, first, we compare the error-correction performance of punctured polar codes constructed using Alg. 1 and Alg. 2. As depicted in Fig. 4, Alg. 2 exhibits the same error-correction performance as Alg. 1, while the search complexity of Alg. 2 is much lower than that of Alg. 1. Here, $(60, 32)$, $(28, 16)$, and $(12, 8)$ polar codes are used, constructed from $(64, 32)$, $(32, 16)$, and $(16, 8)$ mother codes, respectively. It is worth noting that Alg. 1 is infeasible to implement when the code length becomes larger. This is why these short codes are used for comparison.

When compared with WQP [6], ISAP [7], and the method in [8], we employ both the SC decoder and the CA-SCL decoder, the latter is implemented with a list size of 8 and an 8-bit CRC of polynomial $x^8 + x^4 + x^3 + x^2 + 1$. As shown in Fig. 5(a), utilizing the SC and CA-SCL decoders, the proposed Alg. 2 outperforms the state-of-the-art work in [8] by 0.4 dB at BLER $= 10^{-3}$ and 0.8 dB at BLER $= 10^{-4}$, respectively. In Fig. 5(b), using the SC and CA-SCL decoder, Alg. 2 achieves a gain of 0.3 dB and 0.4 dB at BLER $= 10^{-4}$, respectively. As depicted in Fig. 5(c), using the SC and CA-SCL decoder, Alg. 2 shows an improvement of 0.3 dB at BLER $= 10^{-3}$ and 0.6 dB at BLER $= 2 \times 10^{-4}$, respectively. And as demonstrated in Fig. 5(d), using the SC and CA-SCL decoder, Alg. 2 exhibits a gain of 0.25 dB at BLER $= 10^{-3}$ and at BLER $= 10^{-4}$, respectively. These simulation results reveal that while the analysis in this paper is based on the SC decoding, the proposed Alg. 2 also achieves excellent error-



(a) Error-correction performance of (810, 540+8) and (810, 540) polar codes, constructed from (1024, 540+8) and (1024, 540) mother codes, respectively.

(b) Error-correction performance of (420, 280+8) and (420, 280) polar codes, constructed from (512, 280+8) and (512, 280) mother codes, respectively.

(c) Error-correction performance of (900, 750+8) and (900, 750) polar codes, constructed from (1024, 750+8) and (1024, 750) mother codes, respectively.

(d) Error-correction performance of (450, 400+8) and (450, 400) polar codes, constructed from (512, 400+8) and (512, 400) mother codes, respectively.

Fig. 5. Simulation results of error-correction performance.

correction performance when the CA-SCL decoder is used.

V. Conclusion

In this paper, a low-complexity FIS-based puncturing scheme for polar codes is proposed. The Gaussian approximation is utilized to estimate the qualities of bit channels of information bits after puncturing. Moreover, we restrict the search scope to indices within the frozen set, followed by introducing an efficient method to choose the most promising puncturing pattern with low computational complexity. Additionally, a recursive puncturing scheme remarkably mitigating the overall search complexity is employed. Simulation results indicate that the proposed method significantly surpasses the state-of-the-art FIS-based puncturing schemes in terms of error-correction performance.

References

[1] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] I. Tal and A. Vardy, “List decoding of polar codes,” *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[3] K. Niu and K. Chen, “Crc-aided decoding of polar codes,” *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1668–1671, 2012.

[4] K. Niu, K. Chen, and J.-R. Lin, “Beyond turbo codes: Rate-compatible punctured polar codes,” in *IEEE Int. Conf. Commun. (ICC)*, 2013, pp. 3423–3427.

[5] R. Wang and R. Liu, “A novel puncturing scheme for polar codes,” *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2081–2084, 2014.

[6] L. Li, W. Song, and K. Niu, “Optimal puncturing of polar codes with a fixed information set,” *IEEE Access*, vol. 7, pp. 65 965–65 972, 2019.

[7] J. Zhao, W. Zhang, and Y. Liu, “A novel puncturing scheme of low rate polar codes based on fixed information set,” *IEEE Commun. Lett.*, vol. 25, no. 7, pp. 2104–2108, 2021.

[8] S. Han, B. Kim, and J. Ha, “Rate-compatible punctured polar codes,” *IEEE Commun. Lett.*, vol. 26, no. 4, pp. 753–757, 2022.

[9] L. Zhang, Z. Zhang, X. Wang, Q. Yu, and Y. Chen, “On the puncturing patterns for punctured polar codes,” in *IEEE Int. Symp. Inf. Theory*, 2014, pp. 121–125.

[10] P. Trifonov, “Efficient design and decoding of polar codes,” *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3221–3227, 2012.

[11] I. Tal and A. Vardy, “How to construct polar codes,” *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.

[12] S.-Y. Chung, T. Richardson, and R. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, 2001.

[13] M. El-Khamy, H.-P. Lin, J. Lee, and I. Kang, “Circular buffer rate-matched polar codes,” *IEEE Trans. Commun.*, vol. 66, no. 2, pp. 493–506, 2018.