

## Article

# A Novel Flip-List-Enabled Belief Propagation Decoder for Polar Codes

Qasim Jan <sup>1,2</sup>, Shahid Hussain <sup>3</sup> , Muhammad Furqan <sup>1</sup>, Zhiwen Pan <sup>1,2,\*</sup>, Nan Liu <sup>1</sup> and Xiaohu You <sup>1,2</sup>

<sup>1</sup> National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China; qasim@seu.edu.cn (Q.J.); furqan@seu.edu.cn (M.F.); nanliu@seu.edu.cn (N.L.); xhyu@seu.edu.cn (X.Y.)

<sup>2</sup> Purple Mountain Laboratories, Nanjing 211100, China

<sup>3</sup> Data Science Institute, National University of Ireland Galway, Galway H91 TK33, Ireland; shahid.hussain@nuigalway.ie

\* Correspondence: pzw@seu.edu.cn

**Abstract:** Due to the design principle of parallel processing, belief propagation (BP) decoding is attractive, and it provides good error-correction performance compared with successive cancellation (SC) decoding. However, its error-correction performance is still inferior to that of successive cancellation list (SCL) decoding. Consequently, this paper proposes a novel flip-list- (FL)-enabled belief propagation (BP) method to improve the error-correction performance of BP decoding for polar codes with low computational complexity. The proposed technique identifies the vulnerable channel log-likelihood ratio (LLR) that deteriorates the BP decoding result. The FL is utilized to efficiently identify the erroneous channel LLRs and correct them for the next BP decoding attempt. The preprocessed channel LLR through FL improves the error-correction performance with minimal flipping attempts and reduces the computational complexity. The proposed technique was compared with the state-of-the-art BP, i.e., BP bit-flip (BP-BF), generalized BP-flip (GBPF), cyclic redundancy check (CRC)-aided (CA-SCL) decoding, and ordered statistic decoding (OSD), algorithms. Simulation results showed that the FL-BP had an excellent block error rate (BLER) performance gain up to 0.7 dB compared with BP, BP-BF, and GBPF decoder. Besides, the computational complexity was reduced considerably in the high signal-to-noise ratio (SNR) regime compared with the BP-BF and GBPF decoding methods.



check for updates

**Citation:** Jan, Q.; Hussain, S.; Furqan, M.; Pan, Z.; Liu, N.; You, X. A Novel Flip-List-Enabled Belief Propagation Decoder for Polar Codes. *Electronics* **2021**, *10*, 2302. <https://doi.org/10.3390/electronics10182302>

Academic Editors: Francisco Garcia-Herrero, Shahid Mumtaz and Carmen G. Almudever

**Keywords:** belief propagation; channel log-likelihood ratio; error-correcting codes; erroneous channel LLR; preprocessing genie-aided BP decoder

Received: 16 August 2021

Accepted: 15 September 2021

Published: 18 September 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

Erdal Arıkan proved that polar codes have the potential to achieve the Shannon capacity limit [1] of a binary-input discrete memoryless channel with infinite blocklength ( $N$ ) using an SC decoder with computational complexity  $O(N \log N)$  [2]. However, the BLER performance of polar codes under SC is not competitive with the low-density parity-check (LDPC) and Turbo codes with finite blocklengths [3,4]. Successive cancellation list (SCL) [5] and successive cancellation stack (SCS) [6] decoding algorithms are proposed to improve the BLER performance of SC decoding with computational complexities of  $O(LN \log N)$  and  $O(DN \log N)$ , where  $L$  is the list size in the SCL decoder and  $D$  is the maximal depth of the stack in the SCS decoder. In the literature, multiple variants of SCL and SCS have been introduced to improve the performance of the SCL and SCS decoders, albeit that the CA-SCL decoding algorithm [7] significantly improves the BLER performance of SC decoding, yet it suffers high decoding latency and implementation complexity compared with the conventional SC decoder. To alleviate such issues, a successive cancellation flip (SC-flip) decoding algorithm was proposed in [8]. SC-flip identifies the first incorrectly estimated bit that occurs during SC decoding and flips it to restrict the error propagation in the next SC decoding attempt. Several supplements of SC-flip with the average computational

complexity close to the standard SC decoder in the moderate to high SNR regime have been proposed [9–15]. However, they still rely on the SC decoding mechanism, which results in high decoding latency and low throughput due to the sequential processing nature.

Belief propagation (BP) is an alternative decoding mechanism with the intrinsic advantage of the parallel structure design and is more attractive due to its parallel processing [16]. It has low latency and considerably outperforms SC decoding in terms of BLER performance, but is unable to compete with the CA-SCL decoding [3,4]. This implies the inadequacy of the BP decoding and that there is a need for improvement to achieve the BLER performance of CA-SCL. Zhang et al. in [17] proposed a modified belief propagation method, in which the node's messages are rectified by multiplying them against the specified check nodes' messages to improve the reliability of these propagated messages. A hybrid BP-SC coupling the SC and BP decoders was introduced by the authors in [18,19]. In the hybrid BP-SC method, the BP and SC decoders are executed sequentially such that either BP or SC may be able to determine the valid output. Firstly, it executes BP to estimate the valid output, and if it fails, then the valid codeword is estimated by assigning the denoised LLRs to the SC decoder. A postprocessing algorithm classifying errors into converged, false converged, and oscillation errors was introduced in [20]. Elkelesh et al. in [21] proposed BP-list decoding based on the permuted factor introduced in [22] to improve BLER performance. In their method, BP-list uses the permuted version of the standard factor graph for the correction of the result presuming that the standard graph of BP-list decoding is unable to estimate the valid decoding result. Wang et al. in [23] proposed a bit-strengthening algorithm, which strengthens the prior message of stable information by a hard decision.

Recently, a bit-flipping strategy for the BP decoder that improves the BLER performance of BP significantly was proposed in [24]. Yu et al. in [24] utilized the critical set (CS) approach of [25] to identify the error-prone bit and flip the a priori knowledge of it when BP-decoding fails to estimate valid results. The bit-flip method for the BP decoder was initially introduced by [24] to improve the BLER performance of the BP decoder. Inspired by the BP bit-flip method, Yuyu et al. in [26] merged the BP-list decoding with bit-flip and artificial noise. The artificial noise was added to the CRC-aided belief propagation list (CA-BP-list) as a postdecoding process, and the flip bit set (FBS) was constructed if all the BP-list decoders failed. The work in [27] used multiple stopping criteria to determine undetected errors that contributed to the error floor and proposed multiple bit-flipping sets (BFSs) to improve the error-correction performance of the BP decoder. Most recently, GBPF and enhanced BP-flip (EBPF) decoders were proposed to better utilize the concept of BP bit-flip [28]. The GBPF and EBPF decoders construct the flip list through the smallest absolute decision LLRs and the reliability sequence of unfrozen bit indices, respectively. Initially, the smallest absolute decision LLRs are used as a metric for identifying the first error that occurs during the SC decoding process caused by channel noise, which may help to slow down or stop the error propagation phenomena in SC decoding [8]. However, later, it was proven by [9] that this metric is suboptimal for identifying the error-prone bit with a higher sorting complexity. Moreover, the BP decoding is free of error propagation phenomena, and all errors that occur are due to the channel noise. Thus, this implies that flipping of the a priori knowledge of the indices of the low absolute decision LLR and least reliable nonfrozen bit channel is not able to ensure the improvement for the finite blocklength. Besides, all of the aforementioned flipping algorithms focus on nonfrozen bit channels and their soft decision LLRs. On the contrary, the BP decoding algorithm initializes the messages by the channel LLRs. However, there exist erroneous channel LLRs that lead to an invalid BP decoding output. Consequently, the BP decoding messages should be initialized at a high accuracy to achieve a high BLER performance of BP decoding [29].

To the best of our knowledge, none of the previous works utilized channel LLRs for BP decoder performance improvement; despite that, the channel LLR plays an important role and has the potential to improve the BLER performance of the BP decoder. In the sequel, this work classifies the channel LLRs into the corresponding frozen and nonfrozen bit

indices by Monte Carlo simulation and considers the frozen bit indices in conjunction with the distribution of erroneous channel LLRs to improve the BLER performance of the BP decoder. Therefore, a novel flip-list-enabled belief propagation (FL-BP) decoder for polar codes is proposed. The FL-BP technique utilizes the corresponding frozen bit indices for the correction of erroneous channel LLRs. Simulation results show high BLER gains with the FL-BP compared with BP, BP bit-flip, GBPF decoding, and CA-SCL while retaining the average computational complexity close to the standard BP decoding in medium-to-high SNR regions.

Our contribution in this work is threefold:

- The functional behavior of the channel LLRs was analyzed by designing the preprocessing genie-aided BP decoder with the enabled (foreknowledge of transmitted information block ( $u_1^n$ ) and codeword ( $x_1^n$ )) feature. This analysis enabled us to identify the erroneous channel LLRs causing the invalid BP decoding results. Consequently, the correction of a single-channel LLR improved the BLER performance significantly;
- We classified the channel LLRs into frozen and nonfrozen bit indices by Monte Carlo simulation. Accordingly, it was observed that about half of the erroneous channel LLRs corresponded to the frozen bit indices and are called FL. Furthermore, we developed a method for identifying and correcting the erroneous channel LLR using FL to improve the BLER performance of the BP decoder for the polar code;
- We propose a novel FL-BP decoding algorithm that incorporates the erring channels from the FL list and performs correction through the preprocessed erroneous channel LLRs, in the event where the initial BP decoding fails to produce the correct output. The simulation results showed a significant SNR gain on conventional BP, BP bit-flip, and GBPF, and it considerably outperformed CA-SCL-8 with the average complexity close to the conventional BP decoding in medium-to-high SNR regions.

The remainder of this paper is organized as follows: Section 2 briefly explains the preliminaries of the polar code, BP decoder, and BP-BF decoder. Section 3 provides a comprehensive analysis of the erroneous channel LLRs and the flipping behavior of the erroneous channel LLR for incorrect BP decoding results. In Section 4, we present the details of our proposed FL-BP algorithm and describe the pseudocode. The simulation results and discussion are presented in Section 5, and finally, the paper is concluded in Section 6 with the possible future direction.

## 2. Preliminaries

### 2.1. Polar Codes

The polar codes split the  $N$  channels into extremely reliable ( $K$ ) channels for carrying the information and unreliable ( $N - K$ ) channels for the frozen bit. A polar code for the blocklength  $N = 2^n$  is a function  $P(N, K, A)$  having dimension  $K$  with rate  $R = K/N$ ,  $A$  being a subset of the indices for the information bits such that  $A \subset \{1, \dots, N\}$ . The term  $A^c$  is the complement of  $A$ , which represents the set of indices for the frozen bits such that  $A \cup A^c = \{1, \dots, N\}$ , with  $A^c$  known to both the transmitter and receiver. The encoding process of polar codes is represented by Equation (1):

$$x_1^N = u_1^N G^{\otimes n} \quad (1)$$

where  $x_1^N$  is an encoded codeword,  $u_1^N$  an input message vector, and  $G^{\otimes n}$  the  $n$ -th Kronecker product ( $\otimes$ ) of  $G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ , known as the Arıkan kernel.

In this work, the polar codes were constructed through the GA method by assuming that the all-zero codeword was transmitted over the AWGN channel [30]. The error probability for each of the polarized channels is thus computed according to Equation (2):

$$P_e(u_i) \approx Q(\sqrt{E[L_N^{(i)}]}/2), 1 \leq i \leq N \quad (2)$$

where the term  $P_e(u_i)$  is the error probability of each subchannel,  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{-\frac{\beta^2}{2}} d\beta$ , and  $E[L_N^{(i)}]$  is the expectation of the LLR of each subchannel.

### 2.2. Belief Propagation Decoding

Belief propagation is an iterative decoding mechanism proposed for the decoding of polar codes in [16]. Due to its parallel structured graph, it performs all the processing concurrently. A polar code  $P(N, K)$  can be represented through a factor graph with  $n$  total stages such that each stage consists of  $N/2$  processing elements (PEs) having two input and two output nodes, where  $n \in \{1, 2, 3, \dots\}$  and  $N = 2^n$ .

Following the polar codes' standard definition, Figure 1a illustrates the factor graph for polar codes  $P(8, 4)$ , where pair  $(i, j)$  denotes the row and column number of the factor graph. Moreover, as depicted in Figure 1b, a single PE consists of four nodes such that each of the nodes has two types of LLR messages (i.e. right-to-left and left-to-right) denoted by  $(L_{i,j})$  and  $(R_{i,j})$ , respectively. These messages are represented by  $\mathbf{L}$  and  $\mathbf{R}$  matrices such that each one has  $N \times (n + 1)$  dimensions. Prior to the BP decoding process, the two matrices  $\mathbf{L}_{N \times (n+1)}$  and  $\mathbf{R}_{N \times (n+1)}$  are initialized according to Equations (3) and (4):

$$L_{i,j} = \begin{cases} L(y_i), & j = n + 1, \\ 0, & j < n + 1. \end{cases} \tag{3}$$

$$R_{i,j} = \begin{cases} +\infty, & j = 1 \text{ and } i \in A^c, \\ 0, & j = 1 \text{ and } i \in A, \\ 0, & j \neq 1, \quad \forall i \in N. \end{cases} \tag{4}$$

$$L(y_i) = \frac{2y_i}{\sigma^2} \tag{5}$$

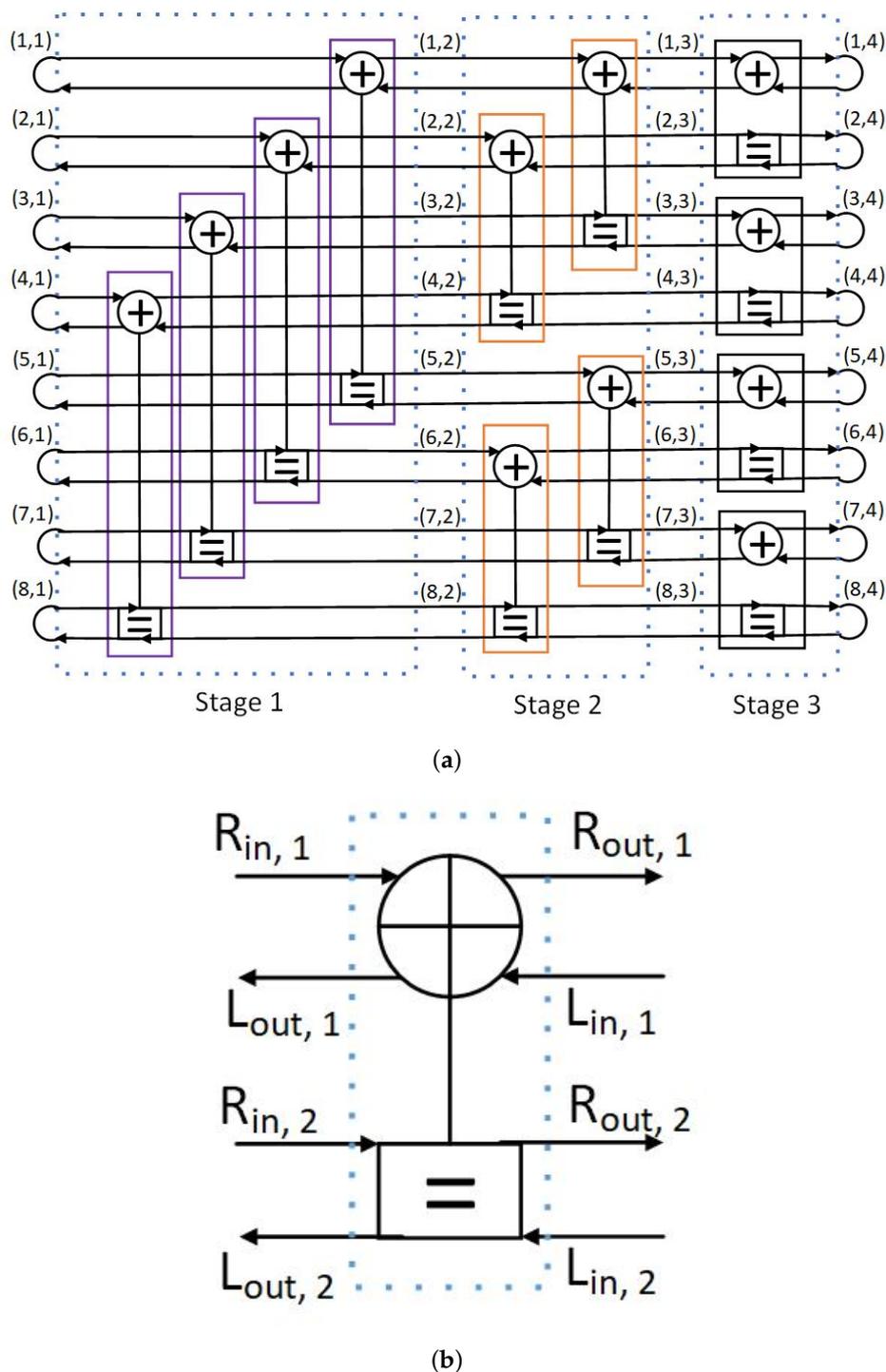
where  $L(y_i)$  is the LLR of the  $i$ -th received bit channel computed using Equation (5) and assigned to the right-most column of the  $\mathbf{L}$  matrix. The left-most column of the  $\mathbf{R}$  matrix represents the a priori knowledge of the information sequence with  $+\infty$  being the probability of frozen bits. In each iteration, the LLR messages of the  $\mathbf{L}$  and  $\mathbf{R}$  matrices are updated and propagated according to Equation (6):

$$\begin{aligned} L_{i,j} &= g(L_{i,j+1}, R_{i+2^j,j} + L_{i+2^j,j+1}) \\ L_{i+2^j,j} &= g(L_{i,j+1}, R_{i,j}) + L_{i+2^j,j+1} \\ R_{i,j+1} &= g(R_{i,j}, L_{i+2^j,j+1} + R_{i+2^j,j}) \\ R_{i+2^j,j+1} &= g(R_{i,j}, L_{i,j+1}) + R_{i+2^j,j} \end{aligned} \tag{6}$$

The function  $g(a, b) = 0.9375\text{sign}(a)\text{sign}(b)\min(|a|, |b|)$  is the *min-sum* approximation proposed for the polar coding in [31], where 0.9375 is the scaling factor suggested in [32] for balancing the approximation error. The decisions for the estimated information bit  $\hat{u}_i$  and the codeword  $\hat{x}_i$  are made upon the completion of each iteration, according to Equations (7) and (8), respectively:

$$\hat{u}_i = \begin{cases} 0, & L_{i,1} + R_{i,1} \geq 0, \\ 1, & L_{i,1} + R_{i,1} < 0, \end{cases} \tag{7}$$

$$\hat{x}_i = \begin{cases} 0, & L_{i,n+1} + R_{i,n+1} \geq 0, \\ 1, & L_{i,n+1} + R_{i,n+1} < 0, \end{cases} \tag{8}$$



**Figure 1.** (a) Factor graph of  $P(8,4)$  polar codes. (b) A single processing element (PE) of belief propagation (BP) decoding highlighting the left-to-right and right-to-left messages.

### 2.3. Belief Propagation Bit Flip Decoding

Recently, Yu et al. in [24] proposed a more advanced BP method with the introduction of a bit-flip mechanism having a maximum flipping order of length  $\omega$ . They incorporated the idea of the critical set (CS) originally defined for the SC [25] to identify the error-prone bit while retaining the underlying mechanism of the conventional BP decoder. In their algorithm, the standard BP decoding was used first to estimate the valid output  $u_1^N$ , then performing an  $m$  bit CRC over the obtained output to verify the correctness of the received information block. In the case where the CRC verifies that the estimated information block

has at least an error, the BP-BF decoder obtains the error-prone bit ( $u_i$ ) from the CS and sets the a priori knowledge for it according to Equation (9) for the next BP decoding attempt:

$$R_{i,1} = \begin{cases} +\infty, & \hat{u}_i = 1, \\ -\infty, & \hat{u}_i = 0. \end{cases} \tag{9}$$

The BP-BF sets the a priori knowledge of  $\omega$  indices to  $+\infty$  and  $-\infty$  consecutively for  $\omega > 1$  and thus requires additional decoding attempts to obtain the desired BP decoding results. Furthermore, additional attempts were performed for all the Rate-1 nodes until the correct results were obtained, which led to the growth in the size of CS and, hence, resulted in high computational complexity and decoding latency.

### 3. Analysis of the Channel LLRs and the Flipping Behavior for the Erroneous LLR

To investigate the erroneous behavior of the channel LLRs, we designed a preprocessing genie-aided decoder for the channel LLRs ( $L(y_1^N)$ ). An adaptive observation method was applied, and the erroneous channel LLRs were classified into the corresponding frozen and nonfrozen bit indices. Then, through the preprocessing genie-aided decoder, the improvement of the BLER performance by flipping the erroneous channel LLRs was analyzed. Details of the overall mechanism are presented in the subsequent sections.

#### 3.1. Analysis of the Channel LLRs

The designed preprocessing genie-aided BP decoder maintains the foreknowledge for the transmitted information block ( $u_1^N$ ) and codeword ( $x_1^N$ ). The preprocessing genie-aided decoder corrects the single erroneous channel LLRs, once it detects the failure of the BP decoder in estimating the correct information block. In our adaptive method, we analyzed the behavior of channel LLRs by running the Monte Carlo simulations for the  $10^7$  blocks over the additive white Gaussian noise (AWGN) channel with binary phase-shift keying (BPSK) modulation for different SNRs with a code rate of  $R = 0.5$ . The parameters for our Monte Carlo simulation are given in Table 1.

**Table 1.** Configuration for the Monte Carlo simulation with  $10^7$  blocks.

Parameter	Value
Modulation	Binary phase shift key (BPSK)
Channel	Additive white Gaussian noise (AWGN)
Code rate (R)	$R = 0.5$
Early stopping	By genie-aided decoder
Maximum iterations ( $I_{max}$ )	100

Before starting BP decoding, the preprocessing genie-aided BP decoder is able to obtain the received codeword ( $z_1^N$ ) from the hard decision according to Equation (10) using the channel LLRs and records the indices for the unmatched transmitted ( $x_1^N$ ) and received ( $z_1^N$ ) ( $x_i \neq z_i$ ) codeword following Equation (11). This enables the preprocessing genie-aided BP decoder to segregate the frozen bit indices  $I^{ef}$  and nonfrozen bit indices  $I^{enf}$  according to Equations (12) and (13):

$$z_i = \begin{cases} 1, & \text{if } L(y_i) < 0, \\ 0, & \text{if } L(y_i) \geq 0. \end{cases} \tag{10}$$

$$I^{et} = \{i|x_i \neq z_i, i \in N\} \tag{11}$$

$$I^{ef} = \{i|sgn(y_i) \neq sgn(x_i), i \in A^c\} \tag{12}$$

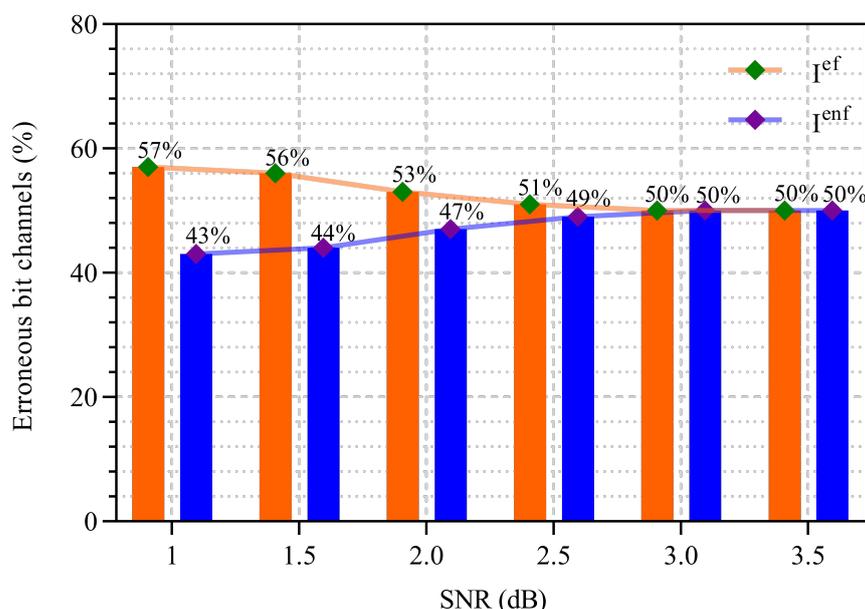
$$I^{enf} = \{i|sgn(y_i) \neq sgn(x_i), i \in A\} \tag{13}$$

$$I^{et} = I^{ef} \cup I^{enf} \tag{14}$$

where  $I^{et}$  represents the total number of erroneous bit indices. The output of the simulation concerning the expected value of the cardinality of the set ( $I^{et}$ ) known as the event gamma ( $\gamma$ ) against the different SNR is summarized in Table 2. It can be observed that the event gamma ( $\gamma$ ) increases linearly with the block size while decreasing with the increase of the SNR. For instance, given SNR = 3.5 dB, the resultant event is 17, 30, 63, and 133, for the polar codes  $P$  with block sizes (256, 128), (512, 256), (1024, 512), and (2048, 1024), respectively. We further analyzed the behavior of channel LLRs by taking a percentage measurement for the erroneous bit channels caused by the noise over the AWGN and plot a histogram for the polar codes  $P(256, 128)$  in Figure 2. It can be observed that in low SNR regions, the percentage of  $I^{ef}$  is higher compared with the  $I^{enf}$  and becomes closer with the increasing SNR values. Such behavior is due to the inverse relationship between the SNR and the erroneous channel LLRs, i.e., the number of erroneous LLR decreases when the SNR increases.

**Table 2.** Average frequency of event  $\gamma$  at different SNRs and code-length results for  $10^7$  blocks.

Code-Lengths	SNR (dB)				
	1.5	2	2.5	3	3.5
P(256, 128)	30	26	23	20	17
P(512, 256)	57	50	44	37	30
P(1024, 512)	116	101	89	77	63
P(2048, 1024)	236	207	181	155	133



**Figure 2.** Histogram of erroneous channels with  $N = 256$  and  $R = 1/2$  at different signal-to-noise ratios (SNRs) for  $10^7$  blocks.

### 3.2. Flipping Behavior of the Erroneous Channel LLR

Considering the foregoing analysis in Section 3.1, one pending question is for the given erroneously channel LLRs: What is the impact of correcting a single erroneous channel LLR, and would such rectification result in correct BP decoding information bits ( $u_1^N$ )? We answer this question as follows: whenever the BP decoder fails to estimate the valid output, the preprocessing genie-aided BP decoder flips the single erroneous channel LLR corresponding to the index from the list ( $i^{et}$ ) sequentially, before starting the next BP decoding process, as given in Equation (15):

$$L_{i^e, n+1} = (-1) \times L(y_{i^e}) \tag{15}$$

where  $L_{i^e, n+1}$  represents the erroneous channel LLR of the  $i^e$ -th index discussed in Equation (3). Once the LLR is flipped, the rest of the decoding follows the standard BP decoding procedure, and the decoding process is terminated, with the valid BP decoding results. The BLER performance of such a preprocessing genie-aided BP decoder is presented in Figure 3. It can be noticed that with the correction of a single erroneous channel LLR, the BP decoding converges to the correct output while improving the average BLER performance by 1.5 dB throughout the simulated SNR region.

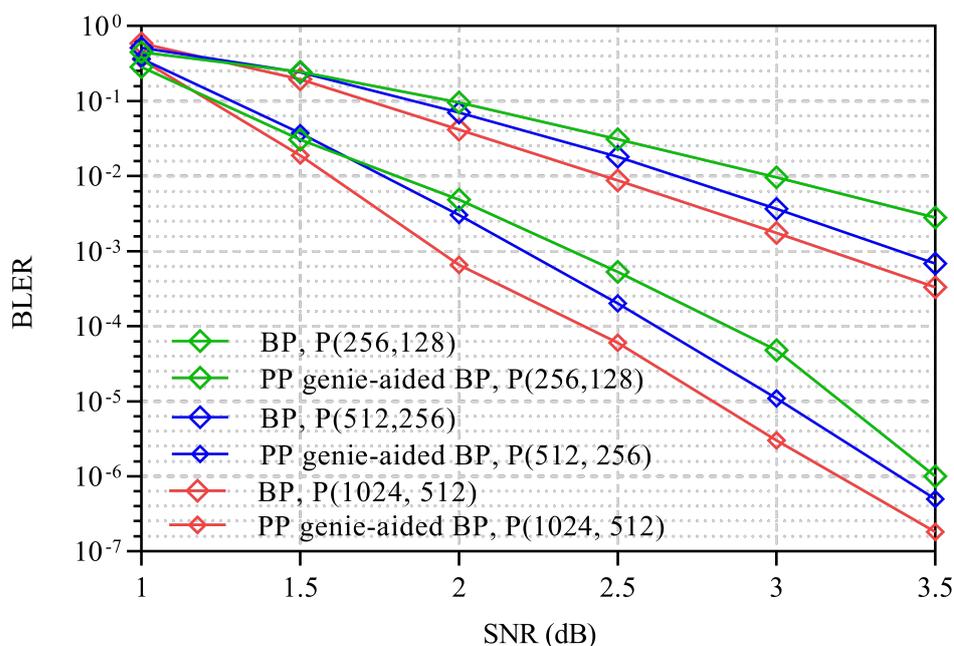


Figure 3. Block error rate (BLER) performance of the BP decoder and the preprocessing (PP) genie-aided BP decoder for  $P(256, 128)$ ,  $P(512, 256)$ , and  $P(1024, 512)$ .

We analyzed the simulation results of the  $10^7$  blocks through the frequency of event alpha ( $\alpha$ ) by considering the different code-lengths under the various SNR regimes, and the results are summarized in Table 3 where event ( $\alpha$ ) is the maximum number of flipping attempts to produce a valid result. We recorded the event  $\alpha$  by setting a maximum limit of 16 attempts, and it is clear from Table 3 that within such a limit, the results' accuracy for  $P(256, 128)$  was about 90% under all the SNR values and remained the same for  $P(512, 256)$  at SNR  $3.5 E_b / N_0$ . Due to the proportionality between the block size and erroneous channel LLRs, we can see a decreasing frequency of event  $\alpha$  for the large blocklength  $P(1024, 512)$ . The frequency of an event  $\alpha$  is trivial for the large blocks while, it is adequate for the small blocklength, which implies the suitability of bit flipping for the medium blocklength codes. Consequently, the correction of a single erroneous channel LLR improves the BLER performance significantly.

Table 3. Output (i.e., frequency and event  $\alpha$ ) of the preprocessing genie-aided BP flip decoder for  $10^7$  blocks.

Polar Codes	SNR (dB)				
	1.5	2	2.5	3	3.5
P(256, 128)	91%	94%	96%	98%	99%
P(512, 256)	75%	77%	81%	87%	89%
P(1024, 512)	59%	62%	63%	65%	66%

#### 4. The Proposed FL-Enabled BP Algorithm

In this section, we first present the design mechanism for the flip list to determine the erroneously received LLRs and then propose the BP flip algorithm based on the FL.

##### 4.1. Construction of the Flip List

Given the preceding analysis (Section 3.1), about half of the erroneous channel LLRs were observed to be from the  $I^{ef}$  at all SNR points along with the higher frequency of  $\alpha$  (i.e., 90% in Table 3) for  $P(256, 128)$ . In the sequel, it was believed that the  $I^{ef}$  represent an adequate parameter for the flip list and suitable for identifying erroneous channel LLR.

Suppose a codeword ( $x_1^N$ ) generated through Equation (1) is transmitted with BPSK modulation over the AWGN channel. The BPSK modulation maps the  $x_1^N$  into transmission symbol  $r_1^N = \{r_1, r_2, \dots, r_N\}$  with  $r_i = (-1)^{x_i} \in \{\pm 1\}$ . At the receiver end, the received signal  $y_1^N = \{y_1, y_2, \dots, y_N\}$  with  $y_i = r_i + n_i$  is the output of the sampler in the demodulator, where  $1 \leq i \leq N$  and  $n_i$  is the independent Gaussian noise with zero mean and variance  $N_0/2$ . The received vector ( $y_1^N$ ) is further employed to compute the channel LLR vector ( $L(y_1^N)$ ) using Equation (5). If a hard decision is performed on each  $i$ -th channel LLR  $L(y_i)$  independently, the natural choice for the measure of reliability is  $abs(L(y_i))$  [33]. Consequently, the proposed method sorts the channel LLRs in ascending order such that  $|L(y_1)| < |L(y_2)| < \dots < |L(y_N)|$ . In our work, each time an error occurred, the flip list was used for the identification of the erroneous channel LLR, as their indices were arranged according to their smallest absolute channel LLRs, as presented in Equation (16):

$$FL = \{k_1, k_2, \dots, k_n\} \quad \forall k = 1, 2, \dots, n \in FL \quad (16)$$

where  $k$  represents the index of the channel LLR.

##### 4.2. Pseudocode of the Proposed FL-BP Algorithm

The underlying mechanism of our proposed FL-BP is to identify the erroneously channel LLR by using the FL. FL-BP initializes the decoding process of the received codeword ( $x_1^N$ ) by exploiting the service of the standard BP decoder. The BP decoding requires early termination (ET) criteria to stop the decoding process on a correct output, which helps to eliminate the unnecessary iterations. This work considers the CRC termination mechanism due to its better performance in error identification. The decoding process stops by either meeting the maximum number of iterations or the CRC traces out the correct output. Once the BP decoder fails to obtain a sound output, the proposed FL-BP activates the erroneous channel LLR flip event and records the  $j$ -th index from the FL to identify the erroneous channel LLR using Equation (17):

$$j = \min_{i \in FL} \{L(y_i)\} \quad (17)$$

The algorithm flips the erroneous channel LLR ( $L(y_{j^e})$ ), which corresponds to the  $j$ -th index by using Equation (15), and initializes the  $L$  matrix with the new channel LLRs in the upcoming BP decoding, as discussed in Equation (3). The process continues until the end of the FL, provided that the CRC does not detect a valid result.

The pseudocodes of the main algorithm and subalgorithms for the proposed FL-BP are provided in Algorithms 1 and 2. The main algorithm takes several inputs and utilizes the services of the subalgorithms to identify and correct the received codewords. A detailed description of the main steps is presented as follows:

1. Initialize the local and global parameters such as loop control variables  $i$ ,  $j$ , and  $flag$  and the maximum number of iterations ( $I_{max}$ );
2. Arrange the left-to-right propagation matrices ( $L$ ) and right-to-left propagation matrices ( $R$ ) by iterating through the size of the codeword ( $N$ ) in Lines 2 to 9;
3. Obtain the estimated information block ( $u_1^k$ ) by utilizing the conventional BP decoding method and pass it through the CRC procedure. Check the remainder ( $r$ ) to validate

- whether the BP has succeeded in achieving the correct output or not. If the BP succeeded, set the *flag* value and break the loop or continue with the next iteration. Repeat this process from Lines 10 to 18;
4. Check the *flag* value; if the BP leads to a valid output, return the results, otherwise call Algorithm 2;
  5. Initialize the local variables such as the number of maximum flipping attempts (*T*), the erroneous channel LLR (*e*), and the loop control variables in Line 1 (Algorithm 2). Pick the index for the error-prone channel LLR from the FL and perform the flipping in Lines 3 and 4. Call the BP decoding method with rectified preprocessing LLRs ( $Pp\text{ll}r_1^N$ ) and verify the validity of the result through the CRC procedure in Lines 5–9. Finally, return the result to the main calling algorithm.

---

**Algorithm 1** Main algorithm of the proposed FL-BP decoding.
 

---

**Input:**  $\text{ll}r_1^N, A, I_{max}, FL$   
**Output:**  $\hat{u}_1^k$

- 1: Initialize system local and global parameters
- 2: **for**  $i \leftarrow 1$  to  $|N|$  **do**
- 3:   **if**  $(N[i] == f)$  **then** ▷ *f* is a frozen bit set
- 4:      $R_{i,1} \leftarrow \infty$
- 5:   **else**
- 6:      $R_{i,1} \leftarrow 0$
- 7:   **end if**
- 8:    $L_{i,n+1} \leftarrow \text{ll}r_i$
- 9: **end for**
- 10: **while**  $(j \leq I_{max})$  **do**
- 11:    $\hat{u}_1^k \leftarrow BP(\text{ll}r_1^N, A, L, R)$
- 12:    $r \leftarrow CRC(\hat{u}_1^k)$  ▷ Get the CRC for the codeword
- 13:   **if**  $(r == 0)$  **then**
- 14:      $flag \leftarrow 0$
- 15:     **break;**
- 16:   **end if**
- 17:    $j \leftarrow j + 1$
- 18: **end while**
- 19: **if**  $(flag \neq 0)$  **then**
- 20:    $\hat{u}_1^k \leftarrow \text{FL-BP}(\text{ll}r_1^N, A, L, R, I_{max}, FL)$
- 21: **end if**
- 22: **Return**  $\hat{u}_1^k$ ;

---



---

**Algorithm 2** FL-BP ( $\text{ll}r_1^N, A, L, R, I_{max}, FL$ ).
 

---

- 1: Initialize local parameters, i.e., *T*, *e*
- 2: **for**  $j \leftarrow 1$  to *T* **do**
- 3:    $e \leftarrow FL[j]$
- 4:    $Pp\text{ll}r_e \leftarrow (-1) \times \text{ll}r_e$
- 5:    $\hat{u}_1^k \leftarrow BP(Pp\text{ll}r_1^N, A, L, R, I_{max})$
- 6:    $r \leftarrow CRC(\hat{u}_1^k)$  ▷ Get the CRC for the codeword
- 7:   **if**  $(r == 0)$  **then**
- 8:     **break;**
- 9:   **end if**
- 10: **end for**
- 11: **Return**  $\hat{u}_1^k$ ;

---

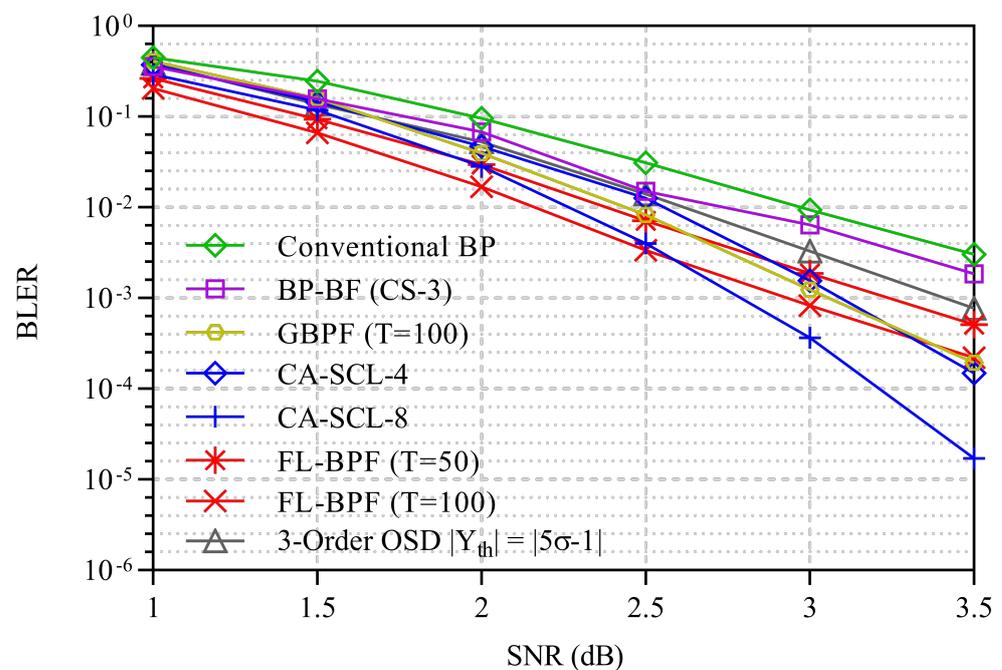
## 5. Simulation Results

The simulation was conducted under the GA construction with SNR = 2.5 dB and BPSK modulation over the AWGN channel. The performance of FL-BP was validated in terms of the BLER and average computational complexity and compared against the BP, BP-BF, GBPF, and CA-SCL decoders for  $P(256, 128)$  and  $P(512, 256)$ .

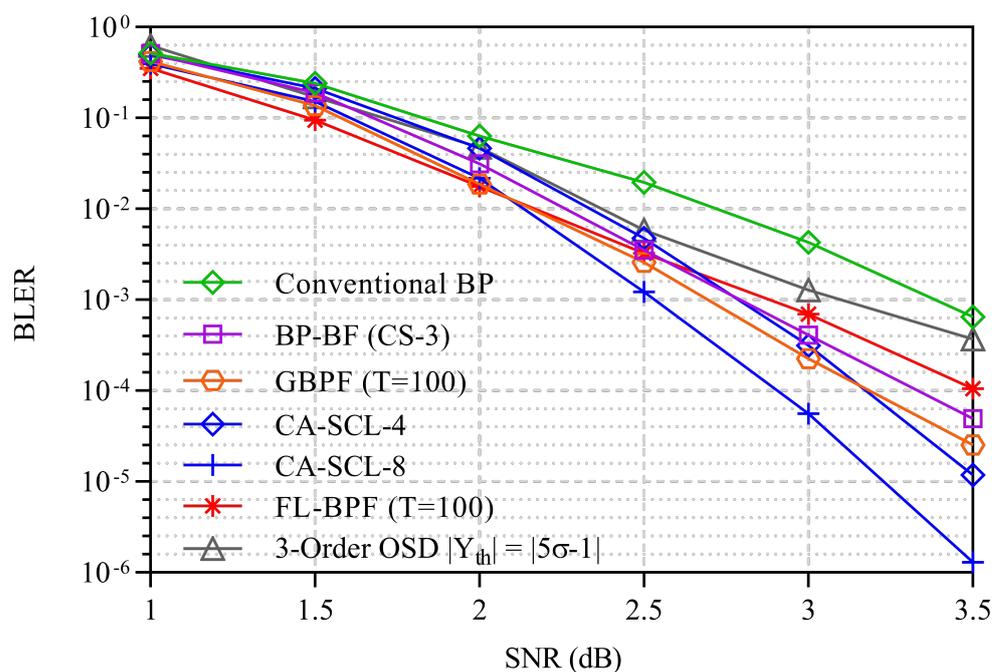
### 5.1. BLER Performance

The BLER performance of the proposed FL-BP for the polar code  $P(256, 128)$  with  $m$  bit CRC ( $m = 24$ ) is demonstrated in Figure 4. It compares the BLER of the proposed FL-BP with the conventional BP, BP-BF(CS-3) [24], and GBPF [28] decoders for  $P(256, 128 + 24)$ , the CA-SCL [7] decoder for  $P(256, 128 + 8)$ , and OSD [34] for  $P(256, 128 + 8)$ , where  $Y_{th}$  represents the threshold and  $\sigma$  is the variance of AWGN. It can be observed that the FL-BP had a remarkable BLER performance under  $P(256, 128 + 12)$  compared with the state-of-the-art flip decoders. The results reveal that the BLER performance of FL-BP had a higher gain at the low-to-moderate SNR regions compared with the conventional BP decoder. For instance, at SNR = 2 dB, it had up to a 0.7 dB and 0.4 dB gain over BP and CA-SCL-4 for FL-BP ( $T = 100$ ). Nevertheless, due to the error floor of BP decoding, its performance gradually deteriorated in a high SNR regime, but still considerably competed with BP-BF(CS-3) and GBPF. The BLER performance improvement rendered the importance of an early correction of a single LLR before initiating the decoding process.

A comparison of the BLER performance concerning the FL-BP, conventional BP [16], BP-BF(CS-3) [24], and GBPF [28] decoders for  $P(512, 256 + 24)$ , the CA-SCL [7] decoder for  $P(512, 256 + 16)$ , and OSD [34] for  $P(512, 256 + 8)$  is visualized in Figure 5. The results indicated a higher gain with the proposed FL-BP for  $P(512, 256)$  against the conventional BP for the given SNR  $\leq 3$  dB. In more detail, at SNR = 2.5 dB, it had up to a 0.7 dB and 0.2 dB gain over BP and CA-SCL-4 for FL-BP ( $T = 100$ ). As discussed earlier (Section 3), the FL size is proportional to the code-length and requires a higher number of flipping attempts to improve the BLER performance for the BP decoder. Consequently, the proposed FL-BP algorithms showed a more feasible solution for the short-to-medium code-length.



**Figure 4.** BLER performance of BP, BP bit-flip (BP-BF(CS-3)), generalized BP-flip (GBPF), CRC-aided successive cancellation list (CA-SCL), and the proposed technique for  $P(256, 128)$ .



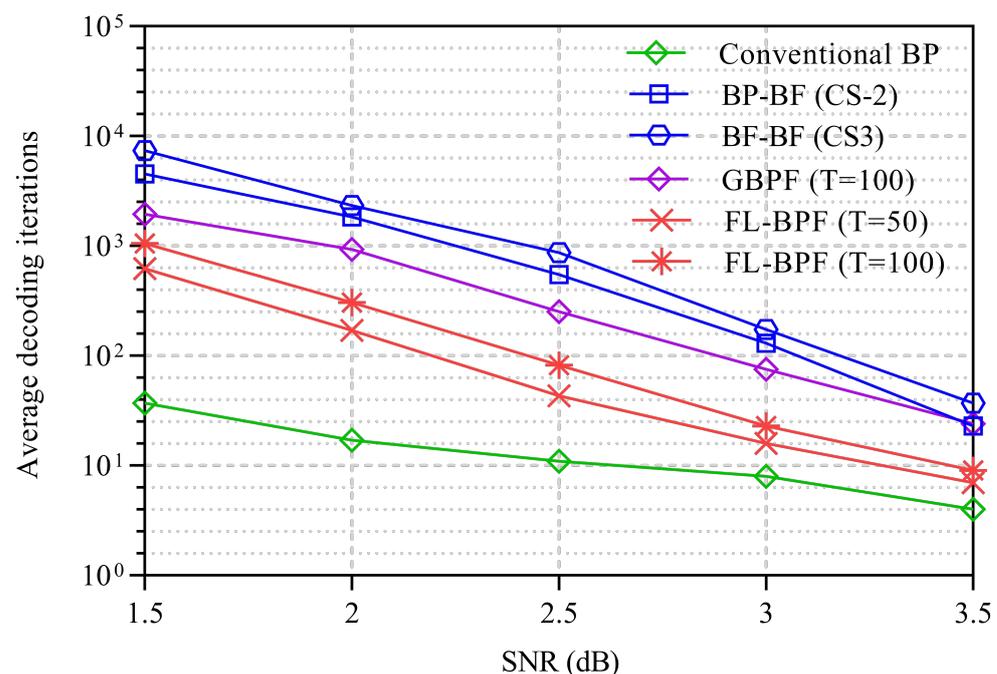
**Figure 5.** BLER performance of BP, BP-BF(CS-3), GBPF, CA-SCL, and the proposed technique for  $P(512, 256)$ .

### 5.2. Average Computational Complexity

The computational complexity for the BP-BF decoding of the polar code was measured in terms of the average iterations  $I_{avg}$  at SNR =  $\delta$  (i.e., a given SNR value), in conjunction with the overall processing of the error-prone list. Consequently, the same criteria were adopted to evaluate the computational complexity of the proposed FL-BP. The proposed FL-BP eased the computation burden of BP decoding with the introduction of the preprocessed FL and thereby lowered the computational complexity. The FL-BP had the worst-case number of flipping attempts  $T_{max} = |FL|$  and had the average-case flipping attempts  $I_{avg} \leq T_{max} \times I$ , regarding the computational complexity, where  $I$  represents BP decoding iterations such that  $I = 100$ . FL-BP's performance was measured in terms of the average computational complexity and compared with the BP-BF and GBPF decoders.

The average computational complexity of BP, BP-BF, GBPF, and FL-BP is visualized in Figure 6. It shows a rational number of iterations in the moderate SNR range (i.e., 1–2 dBs) with FL-BP compared with the underlying BP flip decoding methods while reaching close to the conventional BP decoders at higher SNRs.

In addition, the computational complexity of FL-BP was further analyzed in terms of decoding clock cycles and evaluated for  $thP(256, 128 + 24)$  concerning BP-BF, GBPF, and the proposed FL-BP, as tabulated in Table 4. It can be observed that due to the preprocessed FL, FL-BP resulted in an early correction of the codeword and, thus, had substantially low decoding cycles compared with the other BP-BF methods. The decoding clock cycles showed an inverse relationship to the SNR and decreased with the increase of the SNRs for all the BP flip algorithms. In contrast to others BP flip algorithms, FL-BP was more feasible for a medium blocklength of the polar codes, while compared to the conventional BP algorithm at a high service.



**Figure 6.** Average decoding iterations concerning BP, BP-BF, generalized BP flip, and FL-BP for  $P(256, 128)$  with  $I_{max}=100$ .

**Table 4.** Comparison of decoding clock cycles with respect to the FL-BP, BP-BF, and GBPF techniques under different SNRs for  $P(256, 128 + 24)$ .

Methods	SNR (dB)				
	1.5	2	2.5	3	3.5
BP-BF (CS-2)	72,240	29,376	8784	2080	368
BP-BF (CS-3)	118,016	37,296	13,984	2768	592
GBPF (T = 100)	31,216	14,832	4048	1200	384
FL-BP (T = 50)	9936	2720	688	256	112
FL-BP (T = 100)	16,944	4880	1312	368	144

## 6. Conclusions

In this paper, we highlighted the consequences of single-channel LLR correction over BP performance and presented a novel FL-BP algorithm based on the FL mechanism. The proposed FL-BP identified the erroneous channel LLRs and analyzed their functional behavior through the preprocessing genie-aided BP decoder to improve the BLER performance. Based on Monte Carlo simulation analysis, we considered a list of the bit indices (i.e., FL) that were more susceptible to errors. FL-BP employed the FL to identify the erroneous channel LLRs and corrected them for the next BP decoding attempt. The preprocessed FL improved the error-correction performance with a minimal flipping attempt and thus lowered the decoding clock cycles. The simulation results showed a better BLER performance with minimal computational complexity concerning the proposed FL-BP compared with the BP flipping algorithms. In more detail, FL-BP had a higher gain of 0.7 dB and 0.4 dB at low-to-moderate SNRs (i.e., 1 to 2 dB) compared to the conventional BP and CA-SCL-4 algorithms. The computational complexity of FL-BP was competitive with the conventional BP decoding at high SNRs and more sensible at SNR = 1 and 2 dB compared with the rest of the BP flip decoding methods. In the future, this work can be expanded for the identification of multiple-erroneous-channel LLRs.

**Author Contributions:** Q.J.: development of the idea and conceptualization, development of the preprocessing genie-aided decoder, channel LLR analyses, mathematical modeling, algorithm development, simulation coding, and writing—original draft, review, and editing. S.H.: technical support, erroneous channel LLR analyses, checking and validation of the simulations, and support in the original writing, review, and editing. M.F.: meetings and discussion. Z.P.: supervision and technical support in the conceptualization, review of and feedback on the draft, checking and validation of the simulation, final revision and approval, and funding accusation. N.L.: support and discussion in the draft preparation. X.Y.: Draft checking and feedback and final validation checking and approval. All authors have read and agreed to the published version of the manuscript

**Funding:** This work was partially supported by the National Key Research and Development Project under Grant 2018YFB1802402.

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

## References

1. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [[CrossRef](#)]
2. 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](#)]
3. Balatsoukas-Stimming, A.; Giard, P.; Burg, A. Comparison of Polar Decoders with Existing Low-Density Parity-Check and Turbo Decoders. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
4. Niu, K.; Chen, K.; Lin, J.; Zhang, Q.T. Polar codes: Primary Concepts and Practical Decoding algorithms. *IEEE Commun. Mag.* **2014**, *52*, 192–203. [[CrossRef](#)]
5. Tal, I.; Vardy, A. List Decoding of Polar codes. In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings, Saint Petersburg, Russia, 31 July–5 August 2011; pp. 1–5.
6. Niu, K.; Chen, K. Stack Decoding of Polar codes. *Electron. Lett.* **2012**, *48*, 695–697. [[CrossRef](#)]
7. Niu, K.; Chen, K. CRC-Aided Decoding of Polar Codes. *IEEE Commun. Lett.* **2012**, *16*, 1668–1671. [[CrossRef](#)]
8. Afisiadis, O.; Balatsoukas-Stimming, A.; Burg, A. A Low-Complexity Improved Successive Cancellation Decoder for Polar codes. In Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2–5 November 2014; pp. 2116–2120.
9. Chandesris, L.; Savin, V.; Declercq, D. An Improved SCFlip Decoder for Polar Codes. In Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 4–8 December 2016; pp. 1–6. [[CrossRef](#)]
10. Giard, P.; Burg, A. Fast-SSC-Flip Decoding of Polar codes. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; pp. 73–77. [[CrossRef](#)]
11. Chandesris, L.; Savin, V.; Declercq, D. Dynamic-SCFlip Decoding of Polar Codes. *IEEE Trans. Commun.* **2018**, *66*, 2333–2345. [[CrossRef](#)]
12. Guo, R.; Chen, K.; Liu, H. Multi-CRC Polar Codes and M-SCFlip-Based Decoding. *IEEE Access* **2019**, *7*, 98366–98373. [[CrossRef](#)]
13. Wang, X.; Wang, T.; Li, J.; Zhang, Y. Improved Multiple Bit-Flipping Fast-SSC Decoding of Polar Codes. *IEEE Access* **2020**, *8*, 27851–27860. [[CrossRef](#)]
14. Yang, D.; Yang, K. Error-Aware SCFlip Decoding of Polar Codes. *IEEE Access* **2020**, *8*, 163758–163768. [[CrossRef](#)]
15. Ercan, F.; Tonnellier, T.; Doan, N.; Gross, W.J. Practical Dynamic SC-Flip Polar Decoders: Algorithm and Implementation. *IEEE Trans. Signal Process.* **2020**, *68*, 5441–5456. [[CrossRef](#)]
16. Arikan, E. Polar codes: A Pipelined Implementation. In Proceedings of the 4th International Symposium on Broadband Communication (ISBC 2010), Melaka, Malaysia, 11–14 July 2010.
17. Zhang, Y.; Liu, A.; Pan, X.; Ye, Z.; Gong, C. A Modified Belief Propagation Polar Decoder. *IEEE Commun. Lett.* **2014**, *18*, 1091–1094. [[CrossRef](#)]
18. Yuan, B.; Parhi, K.K. Algorithm and Architecture for Hybrid Decoding of Polar Codes. In Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2–5 November 2014; pp. 2050–2053. [[CrossRef](#)]
19. Cammerer, S.; Leible, B.; Stahl, M.; Hoydis, J.; ten Brink, S. Combining Belief Propagation and Successive Cancellation List Decoding of Polar Codes on a GPU Platform. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 3664–3668. [[CrossRef](#)]
20. Sun, S.; Cho, S.; Zhang, Z. Post-Processing Methods for Improving Coding Gain in Belief Propagation Decoding of Polar Codes. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6. [[CrossRef](#)]
21. Elkelesh, A.; Ebada, M.; Cammerer, S.; ten Brink, S. Belief Propagation List Decoding of Polar Codes. *IEEE Commun. Lett.* **2018**, *22*, 1536–1539. [[CrossRef](#)]
22. Elkelesh, A.; Ebada, M.; Cammerer, S.; ten Brink, S. Belief Propagation Decoding of Polar Codes on Permuted Factor Graphs. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6. [[CrossRef](#)]

23. Wang, X.; Zheng, Z.; Li, J.; Shan, L.; Li, Z. Belief Propagation Bit-Strengthening Decoder for Polar Codes. *IEEE Commun. Lett.* **2019**, *23*, 1958–1961. [[CrossRef](#)]
24. Yu, Y.; Pan, Z.; Liu, N.; You, X. Belief Propagation Bit-Flip Decoder for Polar Codes. *IEEE Access* **2019**, *7*, 10937–10946. [[CrossRef](#)]
25. Zhang, Z.; Qin, K.; Zhang, L.; Zhang, H.; Chen, G.T. Progressive Bit-Flipping Decoding of Polar Codes over Layered Critical Sets. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6. [[CrossRef](#)]
26. Yuyu, Y.; Chao, Y.; Qasim, J.; Yaoyue, H.; Zhiwen, P.; Nan, L.; Xiaohu, Y. Noise-Aided Belief Propagation List Bit-Flip Decoder for Polar Codes. In Proceedings of the 2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 21–23 October 2020; pp. 807–810. [[CrossRef](#)]
27. Zhang, J.; Wang, M. Belief Propagation Decoder With Multiple Bit-Flipping Sets and Stopping Criteria for Polar Codes. *IEEE Access* **2020**, *8*, 83710–83717. [[CrossRef](#)]
28. Shen, Y.; Song, W.; Ren, Y.; Ji, H.; You, X.; Zhang, C. Enhanced Belief Propagation Decoder for 5G Polar Codes with Bit-Flipping. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 901–905. [[CrossRef](#)]
29. Proakisand, J.G.; Salehi, M. *Digital Communications*, 5th ed.; McGraw-Hill: New York, NY, USA, 2005.
30. Trifonov, P. Efficient Design and Decoding of Polar Codes. *IEEE Trans. Commun.* **2012**, *60*, 3221–3227. [[CrossRef](#)]
31. Leroux, C.; Tal, I.; Vardy, A.; Gross, W.J. Hardware Architectures for Successive Cancellation Decoding of Polar Codes. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 1665–1668. [[CrossRef](#)]
32. Yuan, B.; Parhi, K.K. Early Stopping Criteria for Energy-Efficient Low-Latency Belief-Propagation Polar Code Decoders. *IEEE Trans. Signal Process.* **2014**, *62*, 6496–6506. [[CrossRef](#)]
33. Fossorier, M.; Lin, S. Soft-Decision Decoding of Linear Block Codes Based on Ordered Statistics. *IEEE Trans. Inf. Theory* **1995**, *41*, 1379–1396. [[CrossRef](#)]
34. Wu, D.; Li, Y.; Guo, X.; Sun, Y. Ordered Statistic Decoding for Short Polar Codes. *IEEE Commun. Lett.* **2016**, *20*, 1064–1067. [[CrossRef](#)]