

# Improving Decodability of Polar Codes by Adding Noise

Lingjun Kong <sup>1</sup>, Haiyang Liu <sup>2,\*</sup>, Wentao Hou <sup>3</sup> and Bin Dai <sup>4</sup>

<sup>1</sup> Faculty of Network and Telecommunication Engineering, Jinling Institute of Technology, Nanjing 211169, China; kong@jit.edu.cn

<sup>2</sup> Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China

<sup>3</sup> College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 1220013235@njupt.edu.cn

<sup>4</sup> School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; daibin@njupt.edu.cn

\* Correspondence: liuhaiyang@ime.ac.cn

**Abstract:** This paper presents an online perturbed and directed neural-evolutionary (Online-PDNE) decoding algorithm for polar codes, in which the perturbation noise and online directed neuro-evolutionary noise sequences are sequentially added to the received sequence for re-decoding if the standard polar decoding fails. The new decoding algorithm converts uncorrectable received sequences into error-correcting regions of their decoding space for correct decoding by adding specific noises. To reduce the decoding complexity and delay, the PDNE decoding algorithm and sole neural-evolutionary (SNE) decoding algorithm for polar codes are further proposed, which provide a considerable tradeoff between the decoding performance and complexity by acquiring the neural-evolutionary noise in an offline manner. Numerical results suggest that our proposed decoding algorithms outperform the other conventional decoding algorithms. At high signal-to-noise ratio (SNR) region, the Online-PDNE decoding algorithm improves bit error rate (BER) performance by more than four orders of magnitude compared with the conventional simplified successive cancellation (SSC) decoding algorithm. Furthermore, in the mid-high SNR region, the average normalized complexity of the proposed algorithm is almost the same as that of the SSC decoding algorithm, while preserving the decoding performance gain.

**Keywords:** fifth generation; channel coding; polar code; perturbation noise; neuro-evolution



**Citation:** Kong, L.; Liu, H.; Hou, W.; Dai, B. Improving Decodability of Polar Codes by Adding Noise. *Symmetry* **2022**, *14*, 1156. <https://doi.org/10.3390/sym14061156>

Academic Editors: Pingping Chen, Long Shi and Yi Fang

Received: 11 May 2022

Accepted: 1 June 2022

Published: 3 June 2022

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



**Copyright:** © 2022 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

The development of fifth generation (5G) communication technology is driven not only by the requirements of faster and higher-capacity extreme mobile broadband (eMBB) applications, but also by the rapidly evolving area of Internet of Things (IoT) that needs a massive connectivity of devices with ultra-reliable and ultra-low-latency connectivity over Internet Protocol [1–4]. Channel coding is an integral part of any communication system, which plays an important role in meeting the system reliability requirements [5–7]. A prominent feature of 5G new radio (NR) is the adoption of a new class of error correction codes, i.e., polar codes, for control channels [8,9]. Future wireless communication technologies are proliferating in the connection between people and things, and their scenarios will place new requirements on the channel coding performances [10].

Polar codes were proposed by Arikan [11] in 2009, who also originally indicated the symmetry of polar codes for binary-input discrete memoryless channels (B-DMCs). With the symmetry, the output vector can be divided into equivalence classes in terms of their transition probabilities. Based on the channel polarization theory that entails channel combining and channel splitting, this new coding scheme is capacity achieving as opposed to just capacity approaching in symmetric B-DMCs with efficient construction and low complexity [12,13]. Several decoding algorithms have recently been developed for polar

codes in the literature [14–21]. The successive cancellation (SC) algorithm proposed by Arikan [11] is an effective method for decoding polar codes. To reduce the decoding latency and algorithmic complexity of the SC decoder, a simplified successive cancellation (SSC) decoder was proposed in [14]. In [15], Tal and Vardy proposed the successive cancellation list (SCL) decoder to tackle the problem that only one decoding path was reserved for SC decoding, which may lead to the loss of the correct path. CRC (cyclic redundancy check)-aided decoding schemes were proposed to improve the performance of polar codes in [16]. In addition, the recent development of deep learning methods provides a new insight into the decoding of linear codes [22–27]. However, deep-learning-based channel decoding is doomed by the curse of dimensionality, in which the learning process is limited by the complexity as the number of information bits increases.

In the signal processing area, it can be shown that the performance of a suboptimal detector may be improved by adding noise to the received data under certain conditions [28,29]. Inspired by this method, some researchers have investigated how to improve the decoding performance of a (suboptimal) decoder by adding noise. In [30], a belief propagation list (BPL) decoding algorithm was proposed, in which adding a small amount of noise enables the decoder to handle non-convergent errors. A dynamic perturbation decoding method for Polar-CRC concatenation codes through dynamically controlling the interference noise was proposed in [31]. In [32], a generalized framework for multi-round BP decoding with input perturbation for short low-density parity-check (LDPC) codes was proposed, where the perturbation is done iteratively on a few symbols to widen the search space.

In [33], a CRC-assisted perturbation decoding algorithm for polar codes was proposed, which is called the PB-SSC decoding algorithm in this paper. When the CRC check of the SSC decoder fails, the PB-SSC decoding algorithm can provide multiple possible candidate vectors for re-decoding by adding disturbance noises. However, the performance gain achieved by the PB-SSC decoding algorithm is limited. A decoding algorithm for polar codes based on the perturbation with a convolution neural network (CNN) was proposed in [34]. In [35], a post-processing technique was proposed to improve the performance of the SSC polar decoder in the 2D intersymbol interference (ISI) data storage system, namely the post-processing SSC (PP-SSC) decoding algorithm, in which the perturbation algorithm and the genetic algorithm (GA) successively generate perturbation vectors that accelerate the convergence of the decoder. Unfortunately, the secondary generation of perturbation noise by the GA is performed online, which greatly increases the delay and complexity of the decoder.

In order to address these issues, this paper proposes several decoding algorithms for polar codes by applying the idea of adding noise, which generalizes the methods in our previous work [36]. However, more effective operations in the GA training process are used in this work. We first propose an online perturbed and directed neural-evolutionary (Online-PDNE) decoding algorithm. Then, a simplified version of Online-PDNE decoding algorithm, called the PDNE decoding algorithm, is proposed by using the genetic process in an offline manner. Finally, to further reduce the decoding complexity, we further propose a sole neural-evolutionary (SNE) decoding algorithm, which only invokes the pre-trained offline directed neuro-evolutionary noise and provides a considerable balance between the decoding performance and complexity. Simulation results suggest that our proposed decoding algorithms outperform the other conventional algorithms for decoding polar codes. In addition, the algorithms in this paper have more generality compared with the algorithm in [36].

The main contributions of this paper are summarized as follows:

1. An online perturbed and directed neural-evolutionary (Online-PDNE) decoding algorithm is proposed, which makes polar codes have enhanced error correction ability.
2. To avoid the online training process, the PDNE decoding algorithm is proposed, in which perturbation noise and pre-trained offline directed neuro-evolutionary noise sequences are sequentially employed for re-decoding.
3. In order to further reduce the decoding complexity, the SNE decoding algorithm is further proposed, where only the pre-trained offline directed neuro-evolutionary noise by the GA algorithms is employed to improve the decoding performance.
4. The decoding algorithms proposed in this paper are more suitable for the scenarios where the channel quality gradually degrades, such as the storage channel. A good balance can be achieved in terms of the performance and complexity.

The rest of the paper is organized as follows. Section 2 reviews the related work on polar decoding algorithms. In Section 3, the proposed decoding algorithms of polar codes are presented. Simulation results are provided in Section 4. Finally, Section 5 concludes the paper.

## 2. Related Works

In this paper, the error correction performance of polar codes is improved by adding specific noises. In this section, we briefly review the related works.

### 2.1. Conventional Perturbation Based Decoding Algorithms

The concept of stochastic perturbation opens a new perspective where systems can benefit from adding artificial noise. In 1981, Benzi [37] found that the addition of suitable noise under certain conditions leads to an increase in a measure of the quality of signal transmission performance, which could be explained by the phenomenon of stochastic resonance [38,39]. It can be shown that the performance of certain suboptimal detector may be improved by adding some white Gaussian noise [28], where the loss of detectability caused by lowering the signal-to-noise ratio (SNR) is offset by the increased sensitivity of the new noise.

Based on the similar concept, a perturbed decoding algorithm (PA) was proposed for a concatenated CRC and convolutional code system [40]. The original signal is first decoded by the conventional Viterbi algorithm. If the CRC check fails, a perturbed received signal is created and then decoded by the inner Viterbi decoder. The perturbation by artificial noise injection is expected to increase the possibility that the transmitted codeword is obtained.

In [30], a theoretical analysis was presented to gain further insight into stochastic resonance phenomenon, where the performance of a stochastic resonance enhanced detector was derived in terms of the probability of detection and the probability of false alarm. The theory behind PA was elaborated in [41]. The distribution of the numbers of perturbed decoding was derived for independent Gaussian perturbations. The dominant terms of the distribution indicate that the complexity of PA is highly dependent on the geometric structure of the error control code. If an ML decoder is employed as the inner decoder, the distribution for the numbers of perturbed decoding can be well approximated by a function of SNR, signal-to-perturbation-noise ratio (SPNR), and the two-centroid code spectrum, which provides theoretical support for related works.

A BPL decoder that relied on artificial noise as a frame error rate (FER) or bit error rate (BER) performance booster in a subject of coding theory was presented in [30]. Artificially generated noises with different intensities are added to the received signal to avoid false convergence in a BP-based decoder. A dynamic perturbation decoding method for polar-CRC cascaded codes was proposed in [31]. Dynamic perturbation decoding can adjust the variance of the added perturbation noise according to the currently decoded codeword, so that the sequences obtained after each perturbation are as different as possible. Based on the analysis of FER and BER, a noise-assisted decoding algorithm for polar codes was proposed to improve the decoding performance [42]. The algorithm is realized by adding

the human-made noise, which is a post-compensation processing method for the existing algorithms. The disadvantage of the algorithm is that the noise power needs to be set artificially, and a large number of decoding attempts are performed without any direction, which greatly increases the decoding complexity.

## 2.2. GA Based Decoding Algorithms

Since the overall search strategy in the optimization process of the GA does not depend on gradient information or other auxiliary knowledge [43], it provides a general framework for solving complex system problems in various areas, such as combinatorial optimization, machine learning, signal processing, and adaptive control.

The GA has been used in LDPC decoding [43,44]. However, its application in polar decoding is still in the early stages. The authors in [33,35] employed the GA to generate perturbation noise, so as to perturb the received signal that failed to decode. Inspired by the GA of AI technology, the authors in [45] used the mean of the log-likelihood ratio (LLR) distribution as the fitness function of each individual to update the successive cancellation flip (SCF) decoding algorithm of the population. Simulation results verified that the improved SCF decoding algorithm based on new constructed candidate flipping positions sets can achieve competitive decoding performance compared with some state-of-the-art SCF decoding algorithms.

In addition, the GA has also achieved good results in polar code constructions [46–49]. In [46], the authors proposed a GA-based framework to construct polar codes using the BER as the fitness function of the GA, thereby reducing the size of the SCL list and the number of iterations. In [47], the authors proposed a scheme for constructing polar codes based on a hash table update population, which reduced the computational complexity of repeated candidates. Aiming at the joint detector and decoder over the resistive random-access memory (ReRAM) channel model, effective polar codes were constructed using the GA in [48].

## 3. Conventional Schemes

In this section, we first provide the perturbation decoding principle. Then, we briefly summarize the recent related works, including the PB-SSC decoding algorithm [33] and the PP-SSC decoding algorithm [35].

### 3.1. Perturbation Decoding Principle

The perturbation of the received signal can make each received signal have multiple decoding outputs, which can improve the reliability of the transmission process and avoid retransmission of erroneously decoded information [34]. The perturbation of the received signal can be interpreted as adding independent random noise to improve the performance of suboptimal decoders.

As shown in Figure 1, for each valid codeword  $c(c \in \{c_1, c_2, \dots, c_S\})$ , an error correction region  $a(a \in \{a_1, a_2, \dots, a_S\})$  in the decoding space is specified, where  $S$  is the total number of valid codewords. When the received signal  $y^N$  falls into the error correction region, the decoder succeeds in decoding, where  $N$  is the code length. However, if the received signal  $y^N$  falls outside the error correction region, the decoder fails in decoding. At this point, adding a random noise  $n$  to the received signal may convert the perturbed signal  $y^N + n$  into the error correction region, resulting in a successful decoding.

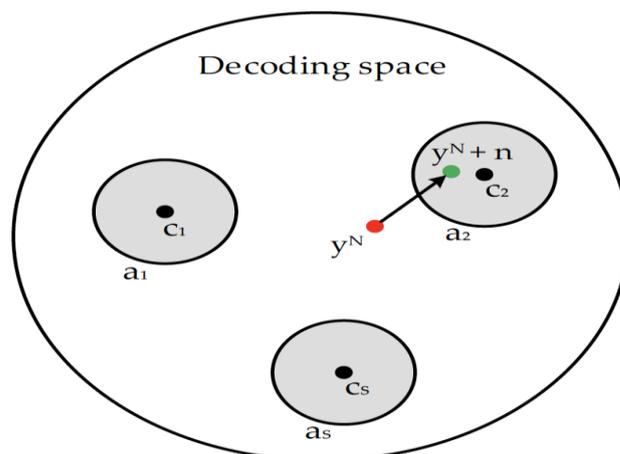


Figure 1. Schematic diagram of perturbation decoding principle.

### 3.2. PB-SSC Decoding Algorithm

According to the perturbation decoding principle in Section 3.1, the PB-SSC decoding algorithm has been proposed to improve the performance of the SSC decoder in [33].

When the SSC decoding fails, the PB-SSC decoding algorithm is activated. The perturbed noise is added to the received signal  $y^N$  to help the received signal to approach the error correction domain. The perturbed received signal  $y_1^N$  can be expressed by

$$y_1^N = y^N + Q_p \cdot \text{randn}(1, N), \tag{1}$$

where  $Q_p$  is the variance of the perturbed noise, and  $\text{randn}$  generates random numbers from the standard normal distribution. The decoding is an iterative process that continues until a valid codeword is obtained or the prescribed maximum number of iterations,  $T_1$ , is reached. We refer the readers to [33] for further details on the PB-SSC decoding algorithm.

### 3.3. PP-SSC Decoding Algorithm

The PP-SSC decoding algorithm [35] has been proposed to improve the performance of polar codes in data storage systems, where the GA procedure is employed to enhance the iteration process by producing perturbation vectors that are inherently better than the directed perturbed ones.

The fitness function of an individual is evaluated as

$$F_c = 1 / \sqrt{\sum_{k \in A} [d(k) - d_p(k)]^2 / |A|}, \tag{2}$$

where  $d$  and  $d_p$  are the decoded sequences of the received signal and the perturbed signal, respectively, and  $A$  is the information set of the polar code.

Selection operations in the GA are used to select the parents of the next offspring at each evolution stage, where fitter individuals are forwarded as parents for the upcoming offspring. Then, the surviving individuals will then encounter evolutionary transformations, namely, mutations and crossovers, to generate offspring which would represent the new population. The perturbed vector generated by the GA process in the PP-SSC decoding algorithm will be added to the received signal  $y^N$  again for decoding, until a valid codeword is obtained or the prescribed maximum number of generations  $T_2$  is reached. We refer the readers to [31] for further details on the PP-SSC decoding algorithm.

## 4. Proposed Decoding Algorithms

In this section, three decoding algorithms for polar codes are proposed by adding noise based on the perturbation decoding principle. The computational complexities are then analyzed.

#### 4.1. Online-PDNE Decoding Algorithm

For the GA, the choice of the fitness function plays an important role in the quality of the final solution and the speed of convergence. The evolutionary search process of the GA is only based on the fitness of each individual in the population. Therefore, the selection of the fitness function directly affects the convergence speed of the GA and whether the optimal solution can be found.

In Equation (2),  $d$  is the failed decoding sequence output by the decoder, and  $d_p$  is the corresponding perturbed decoded sequence. There is no guarantee that their difference can converge effectively unless  $d$  is the correct decoding result. However, in the PP-SSC decoder, the GA-based decoding is performed only if the previous perturbed decoding fails. Therefore, we need better fitness function to carry out the genetic process to find the optimal solution.

Similar to the PP-SSC decoder, we propose an online perturbed and directed neural-evolutionary (Online-PDNE) decoding algorithm for polar codes, which adopts the same decoding structure, but with a new fitness function:

$$F_c = 1 / (1 + \sum \text{CRC\_Calc}(d_p)), \quad (3)$$

where  $\text{CRC\_Calc}(\cdot)$  is the CRC check operation [50].

Fitness function is used to measure whether an individual is the optimal solution in the GA process, which requires the individual's fitness value to be as high as possible. Depending on the fitness function and the selection algorithm, the population can continuously evolve towards the local optimal solution. In this paper, we take the sum of the CRC check remainders as the fitness function. When using Equation (3), it is ensured that the more number of zeros in the CRC remainder, the larger the value of  $F_c$ , which meets the requirements of the fitness function. To ensure the evolution direction of the GA population, the roulette wheel selection strategy is used to select the offspring, which is selected according to the cumulative probability. The fitness score of an individual is calculated by

$$F_s = F_c(j) / \sum_{j=1}^{T_1} F_c(j). \quad (4)$$

As shown in Figure 2, when the maximum number of decoding attempts  $T_1$  in the second round is exceeded, the directed neural-evolutionary noise (NE)  $n_{ne}^{(k)}$  will be generated online by the NE noise generator and added to the received signal  $y^N$  as

$$y_2^N = y^N + n_{ne}^{(k)}. \quad (5)$$

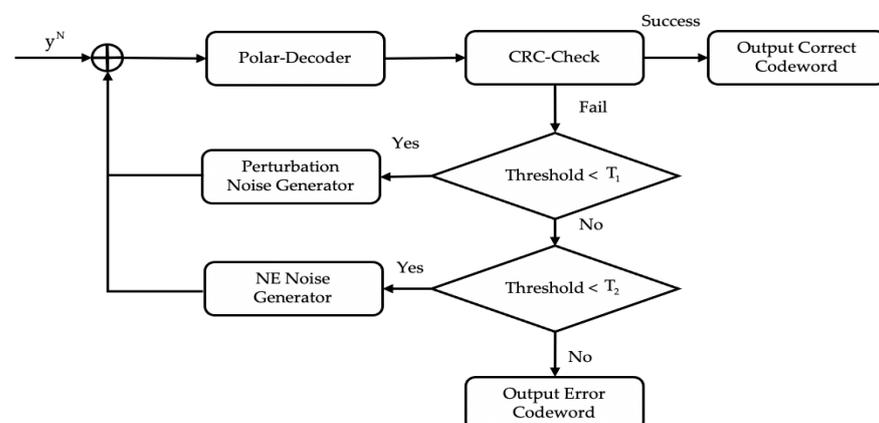


Figure 2. Block diagram of the Online-PDNE decoding algorithm.

In this paper, the GA is employed to realize neuro-evolution to obtain the required directed neural-evolutionary noise. The genetic process  $NE(\cdot)$  is as follows:

1. Initialization: Unlike the initialization method in [31], which adopts the perturbation noise when the second round of decoding fails as the initial value. In this paper, the initial population  $S_n^{(0)} = \{n_{ne}^{(i)} | i = 1, 2, \dots, T\}$  is randomly generated, where  $T$  is the number of individuals in the population.
2. Selection: According to Equations (3) and (4), the individuals which have high fitness scores with roulette wheel selection from the initial population are forwarded as parents for the upcoming offspring.
3. Population reproduction: This step includes the crossover and mutation. The evolution of GA towards the (sub)optimal solution is mainly due to the crossover operation. The mutation operation guarantees more diversity and reduces the occurrence of a famous phenomenon called premature convergence.
4. Termination criterion: The new directed neural-evolutionary noise generated in the above steps is added to the input of the decoder, and if the CRC checking is successful, the decoding result is output. Otherwise, the GA is continued until the decoding is successful or the maximum number of generations  $T_2$  is reached. The details of the Online-PDNE decoding algorithm are given in Algorithm 1.

---

**Algorithm 1:** Online-PDNE Decoding algorithm

---

```

Input:  $y^N$  // Received signal
        $Q_p$  // Variance of the perturbed noise
        $T_1$  // Maximum number of the perturbed attempts
        $T_2$  // Maximum number of generations
        $T$  // Number of individuals in the population
Output:  $\hat{u}_1^N$  // Estimated codeword
1: Initialization:  $\hat{u}_1^N \leftarrow 0, i \leftarrow 1, k \leftarrow 1.$ 
2:  $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y^N)$ 
3: if  $\text{CRC}(\hat{u}_1^N) == \text{success}$ 
4:   break
5: else
6:   while  $i \leq T_1$  do
7:      $y_1^N = y^N + Q_p \cdot \text{randn}(1, N)$ 
8:      $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y_1^N)$ 
9:      $i \leftarrow i + 1$ 
10:    if  $\text{CRC}(\hat{u}_1^N) == \text{success}$ 
11:      break
12:    end if
13:  end while
14: Initial population  $S_n^{(0)} = \{n_{ne}^{(i)} | i = 1, 2, \dots, T\}$ 
15: for  $j = 1, \dots, T_2$  do
16:    $S_n^{(j)} = NE(S_n^{(j-1)})$ 
17:   while  $k \leq T$  do
18:    Choose  $n_{ne}^{(k)} \in S_n^{(j)}$ 
19:     $y_2^N = y^N + n_{ne}^{(k)}$ 
20:     $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y_2^N)$ 
21:     $k \leftarrow k + 1$ 
22:    if  $\text{CRC}(\hat{u}_1^N) == \text{success}$ 
23:      break
24:    end if
25:  end while
26: end for
27: end if
28: Return  $\hat{u}_1^N$ 

```

---

#### 4.2. PDNE Decoding Algorithm

On account of the iterative genetic evolution, the proposed online-PDNE decoding algorithm needs to seek the optimal solution in an online manner. Although the error correction performance of the polar code is greatly improved, it is achieved at the expense of increasing the complexity and delay of the decoder. To this end, a perturbed and directed neural-evolutionary (PDNE) decoding algorithm for polar codes is proposed, in which the perturbation noise and pre-trained offline directed neuro-evolutionary noise sequences are sequentially added to the received sequence for re-decoding, as given in Equations (1) and (5).

In the PDNE decoding algorithm,  $n_{ne}$  is chosen from a set  $S_n^*$  of noise patterns that is generated offline according to the method described in the following. This perturbed decoding process is performed until the cardinality  $T_s$  of the set  $S_n^*$  is reached.

In the offline training, population  $S_n^{(t)}$  at generation  $t$  ( $t = 0, 1, \dots, T_2 - 1$ ) are constructed in an iterative process, where the initial population  $S_n^{(0)}$  is the first generation randomly created. The new population noise produced by each generation is sequentially and independently added to the decoding failed channel output sequence. The fitness function is defined as the same as Equation (5) in the proposed online-PDNE decoding algorithm. As shown in Figure 3, if the decoding is successful, the current neural-evolutionary noise  $n_{ne}$  will be stored in a set  $S_n^*$ . Otherwise, the genetic process is continued on the population noise until the correct directed individual is obtained or the maximum number of generations  $T_2$  is reached to reinitialize the population for the next round of evolution. The details of the PDNE decoding algorithm are given in Algorithm 2.

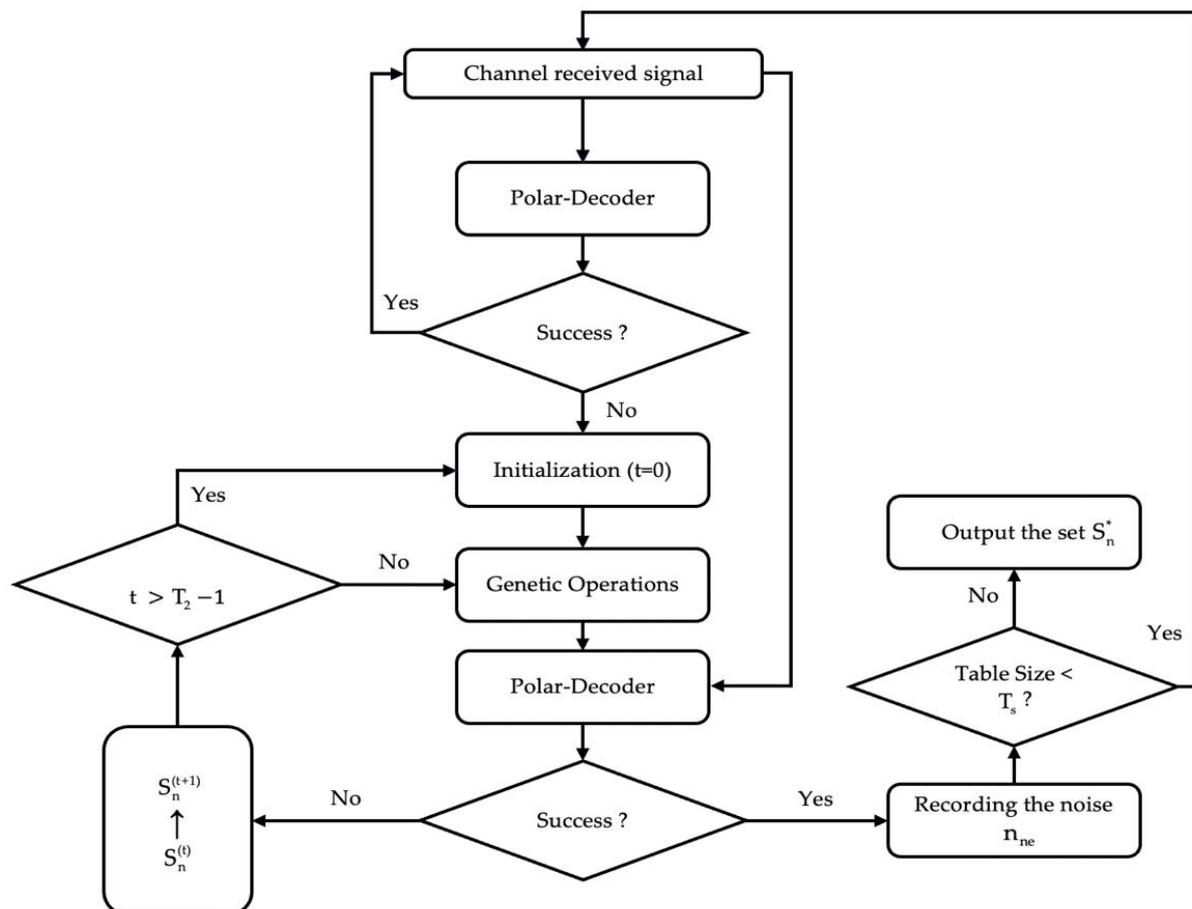


Figure 3. Block diagram of training a directed neural evolutionary noise in an offline manner.

Compared with the algorithm in [36], the proposed PDNE decoding algorithm in this paper has the following advantages:

1. The proposed PDNE decoding algorithm adopts the fitness function shown in Equation (3) instead of the bit error rate (BER)-based one in [36].
2. The mutation operations in this paper judge whether the individual needs to be mutated bit by bit, thereby increasing the variation diversity, while the operations in [36] only mutate one of the first  $\lfloor N \cdot p_m \rfloor$  positions of each individual, where  $p_m$  is the mutation probability.
3. Different from the optimization of the set  $S_n^*$  for each SNR value in [36], the proposed PDNE decoding algorithm in this paper only needs to optimize one set  $S_n^*$ , which greatly shortens the optimization process and reduces the spaces for storing the noises from the perspective of implementation. This indicates the algorithm in the paper has better generality.

---

**Algorithm 2:** PDNE Decoding algorithm

---

```

Input:  $y^N$  // Received signal
       $Q_p$  // Variance of the perturbed noise
       $T_1$  // Maximum number of the perturbed attempts
       $T_s$  // Cardinality of the set  $S_n^*$ 
       $S_n^*$  // Set of the noise patterns
Output:  $\hat{u}_1^N$  // Estimated codeword
1: Initialization:  $\hat{u}_1^N \leftarrow 0$ ,  $i \leftarrow 1$ ,  $j \leftarrow 1$ .
2:  $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y^N)$ 
3: if  $\text{CRC}(\hat{u}_1^N) == \text{success}$ 
4:   break
5: else
6:   while  $i \leq T_1$  do
7:      $y_1^N = y^N + Q_p \cdot \text{randn}(1, N)$ 
8:      $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y_1^N)$ 
9:      $i \leftarrow i + 1$ 
10:    if  $\text{CRC}(\hat{u}_1^N) == \text{success}$ 
11:      break
12:    end if
13:  end while
14:  while  $j \leq T_s$  do
15:    Choose  $n_{ne}^{(j)} \in S_n^*$ 
16:     $y_2^N = y^N + n_{ne}^{(j)}$ 
17:     $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y_2^N)$ 
18:     $j \leftarrow j + 1$ 
19:    if  $\text{CRC}(\hat{u}_1^N) == \text{success}$ 
20:      break
21:    end if
22:  end while
23: end if
24: Return  $\hat{u}_1^N$ 

```

---

#### 4.3. SNE Decoding Algorithm

In order to further simplify the PDNE decoder structure and reduce the processing delay of the decoder, a sole neural-evolutionary (SNE) decoding algorithm of polar codes is proposed, in which only pre-trained offline directed neuro-evolutionary noise sequences are added to the received sequence for re-decoding. When the SSC decoding fails, the pre-trained neural-evolutionary noise is directly called, which greatly reduces the decoding delay. The details of the SNE decoding algorithm are given in Algorithm 3.

**Algorithm 3:** SNE Decoding algorithm

---

```

Input:  $y^N$  // Received signal
       $T_s$  // Cardinality of the set  $S_n^*$ 
       $S_n^*$  // Set of the noise patterns
Output:  $\hat{u}_1^N$  // Estimated codeword
1: Initialization:  $\hat{u}_1^N \leftarrow 0, j \leftarrow 1$ .
2:  $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y_1^N)$ 
3: if CRC( $\hat{u}_1^N$ ) == success
4:   break
5: else
6:   while  $j \leq T_s$  do
7:     Choose  $n_{ne}^{(j)} \in S_n^*$ 
8:      $y_2^N = y^N + n_{ne}^{(j)}$ 
9:      $\hat{u}_1^N \leftarrow \text{Polar\_decoder}(y_2^N)$ 
10:     $j \leftarrow j + 1$ 
11:    if CRC( $\hat{u}_1^N$ ) == success
12:      break
13:    end if
14:  end while
15: end if
16: Return  $\hat{u}_1^N$ 

```

---

**4.4. Complexity Analysis**

The additional complexities required by the proposed three decoding algorithms for polar codes are discussed in this sub-section. Note that the complexity required for CRC check is ignored in the following analysis. Table 1 compares the computational complexity of the proposed algorithms and other decoding schemes, where  $C_p$  and  $C_g$  are the unit calculations required in the perturbation operation and genetic operation, as shown in Table 2.

**Table 1.** Computational complexity of different decoding algorithms.

Decoding Algorithms	Computational Complexity
SSC [14]	$O(N \log N)$
PB-SSC [33]	$O(C_p N \log N)$
PP-SSC [35]	$O(C_p N \log N) + O(C_g N \log N)$
Proposed Online-PDNE	$O(C_p N \log N) + O(C_g N \log N)$
Proposed PDNE	$O(C_p N \log N)$
Proposed SNE	$O(C_p^+ N \log N)$

**Table 2.** Unit calculations in GA process.

Perturbation Operation Calculations ( $C_p$ )	Genetic Operation Calculations ( $C_g$ )
Multiplication $C_p^\times$ (generation of perturbation noise)	Division, square root (fitness value evaluation)
Addition $C_p^+$ (perturbing the received signal)	Summation, division (fitness score evaluation)
	Comparison (selection)
	Comparison, addition (crossover)
	Comparison, addition (mutation)
	Addition (perturbing the received signal)

When the standard polar decoding fails, the Online-PDNE or PDNE decoding algorithm is activated, in which multiple perturbation noises are generated for the first  $T_1$

attempts. Given the code length  $N$ , the number of computations required to generate multiple candidate codewords by perturbing the output signal in the worst case is  $C_p T_1 N$ . The number of computations required for the worst-case genetic operation process can be estimated as  $C_g T_2 N$ . Therefore, the overall additional complexity brought by the proposed Online-PDNE decoder is  $C_p T_1 N + C_g T_2 N$ , which is the same as the PP-SSC decoder.

In the PDNE decoding algorithm, the genetic process runs in an offline manner, so the extra complexity required is only  $C_p T_1 N$ , which is due to perturbation operations.

As for the SNE decoding algorithm, the pre-trained noise set  $S_n^*$  is directly called for secondary decoding when the SSC decoding fails. Compared with the PDNE decoding algorithm, the SNE decoding algorithm only needs to perform the addition operation  $C_p^+$  in the perturbation operation.

## 5. Simulation Results

In this section, the performance of the proposed decoding algorithms is evaluated on a BPSK-modulated additive white Gaussian noise (AWGN) channel, in which the SSC decoder is employed for the standard polar decoding. However, it can easily be extended for other decoding algorithms. In the simulations, we use the same polar codes from [10] with code rates 1/2 and 3/4, and both codes have length  $N = 1024$ . The parameters and related values used in the training process are shown in Table 3.

**Table 3.** Parameters of the training process.

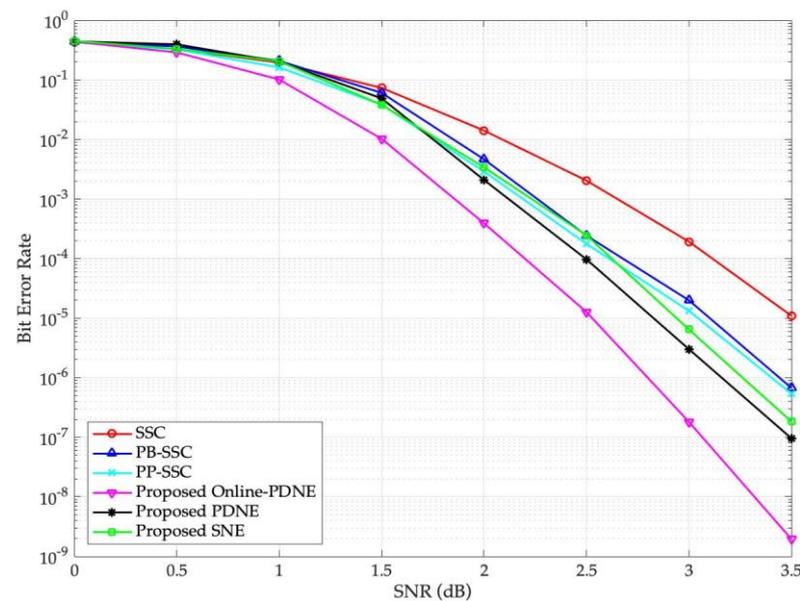
Parameters	Value
Maximum number of the perturbed attempts $T_1$	10
Maximum number of generations $T_2$	100
Number of individuals in the population $T$	10
Crossover probability $p_c$	0.8
Mutation probability $p_m$	0.1
Cardinality $T_s$ of the set $S_n^*$	10
Variance of the perturbed noise $Q_p$	0.25

Figures 4 and 5 illustrate the BER and the FER performances of the rate-1/2 polar code with the proposed decoding algorithms, the SSC algorithm [14], the PB-SSC algorithm [33], as well as the PP-SSC algorithm [35], respectively. As shown in Figure 4, the performance of the proposed three decoding algorithms is better than that of the other algorithms. The performance of the proposed Online-PDNE decoding algorithm is about 1.0 dB, 0.6 dB and 0.5 dB superior to that of the conventional SSC algorithm, the PB-SSC algorithm, and the PP-SSC algorithm at the BER of  $10^{-5}$ , respectively. When  $SNR = 3.5$  dB, we can see from the figure that our proposed Online-PDNE decoding algorithm can improve BER performance by nearly four orders of magnitude compared with the SSC algorithm, nearly three orders of magnitude compared with the PB-SSC algorithm, and more than two orders of magnitude compared with the PP-SSC algorithm.

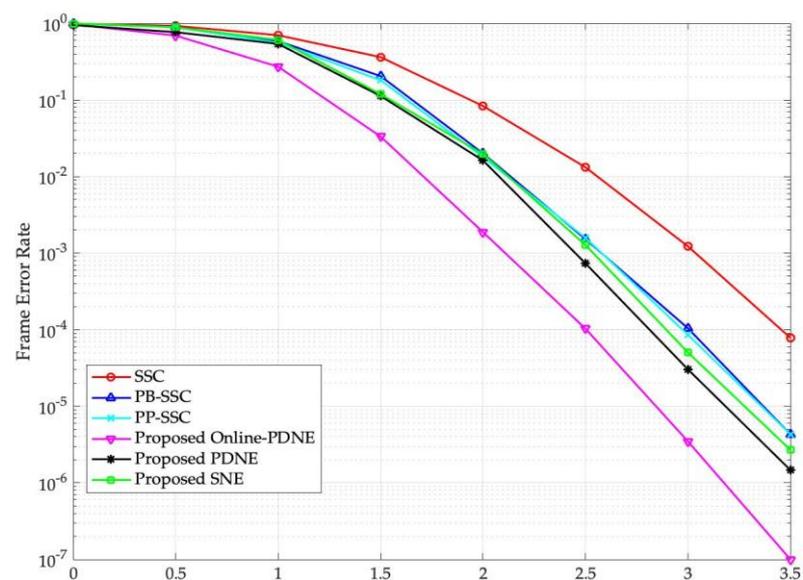
In addition, among the three proposed decoding algorithms, the Online-PDNE decoding algorithm has the best performance, followed by the PDNE decoding algorithm. The reason why the Online-PDNE decoding algorithm has the best performance is that the directed neural-evolutionary noise is generated online by the GA process for the uncorrectable sequence, which converts the received signal into the error correction region of its decoding space more accurately.

Since the PDNE decoding algorithm and the SNE decoding algorithm use the offline GA process to generate the directed neural-evolutionary noise, the decoding complexity and delay are greatly reduced compared with the Online-PDNE decoding algorithm. However, due to the limitation of the size of the directed noise set and uncorrectable error codewords not encountered during offline training, their performances suffer slightly. Compared with the PDNE decoding algorithm, the performance of the SNE decoding algorithm decreases

slightly, about 0.2 dB, but its decoding complexity and delay are the lowest, and it only depends on the directed neural-evolutionary noise of offline training to decode correctly.



**Figure 4.** BER performance of the (1024, 512) polar code with different decoding algorithms. The SSC algorithm is in [14], the PB-SSC algorithm is in [33], and the PP-SSC algorithm is in [35].

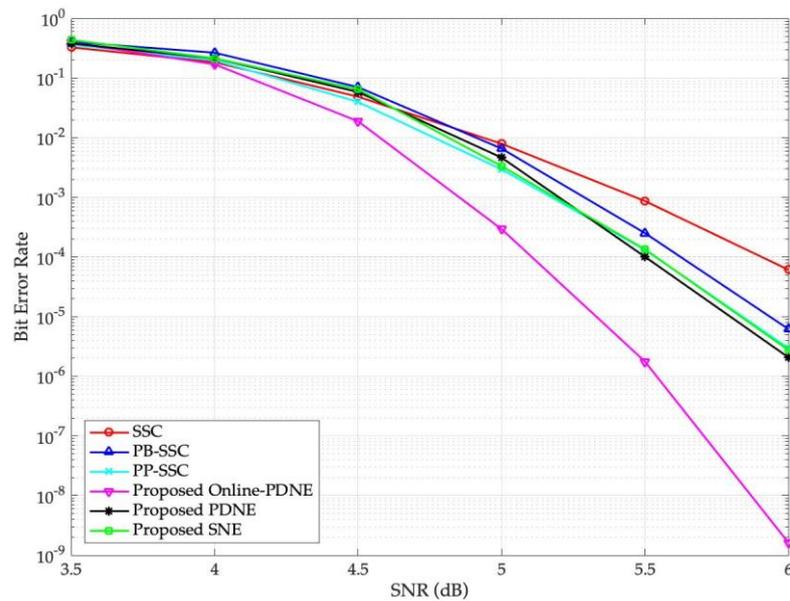


**Figure 5.** FER performance of the (1024, 512) polar code with different decoding algorithms. The SSC algorithm is in [14], the PB-SSC algorithm is in [33], and the PP-SSC algorithm is in [35].

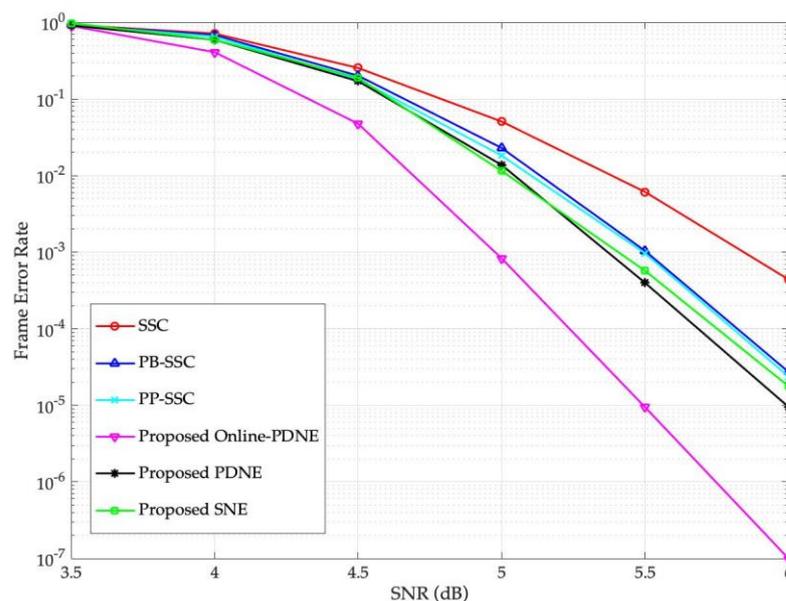
From Figure 5, it can be seen that our proposed three decoding algorithms achieve better performance than the others. This conclusion agrees with the results in Figure 4. When SNR = 3.5 dB, we can see from the figure that our proposed Online-PDNE decoding algorithm can improve FER performance by three orders of magnitude compared with the SSC algorithm, more than one order of magnitude compared with the PB-SSC algorithm and the PP-SSC algorithm.

To further evaluate the error correction performance of the proposed decoding algorithms as described in Section 3, we also simulate the rate-3/4 polar code. In Figures 6 and 7, we compare the BER and FER performances of the designed decoding algorithms

with other decoding algorithms for the rate-3/4 polar code, respectively. We also see that the proposed decoding algorithms have better error correction performance than other decoding algorithms when the code rate increases.



**Figure 6.** BER performance of the (1024, 768) polar code with different decoding algorithms. The SSC algorithm is in [14], the PB-SSC algorithm is in [33], and the PP-SSC algorithm is in [35].



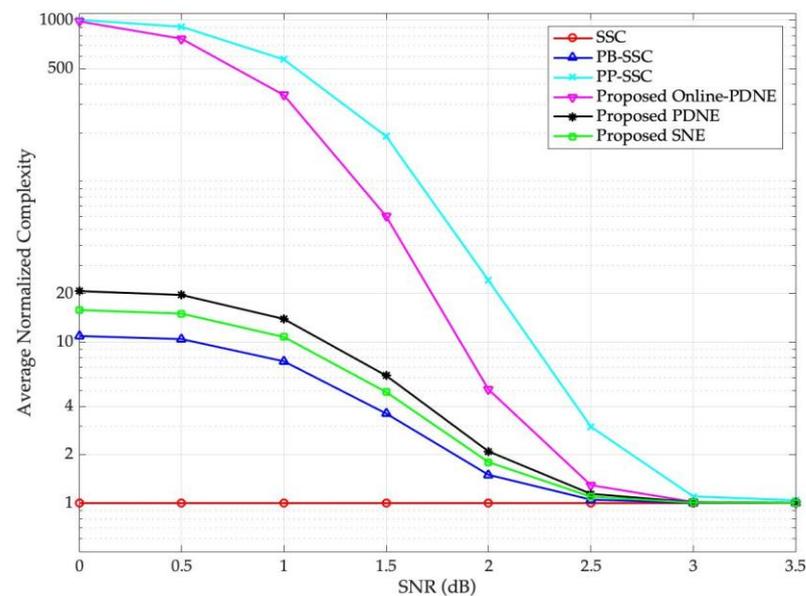
**Figure 7.** FER performance of the (1024, 768) polar code with different decoding algorithms. The SSC algorithm is in [14], the PB-SSC algorithm is in [33], and the PP-SSC algorithm is in [35].

In Figure 6, when SNR = 6.0 dB, we can see from the figure that our proposed Online-PDNE decoding algorithm can improve the BER performance by more than four orders of magnitude compared with the SSC algorithm, more than three orders of magnitude compared with the PB-SSC algorithm and the PP-SSC algorithm.

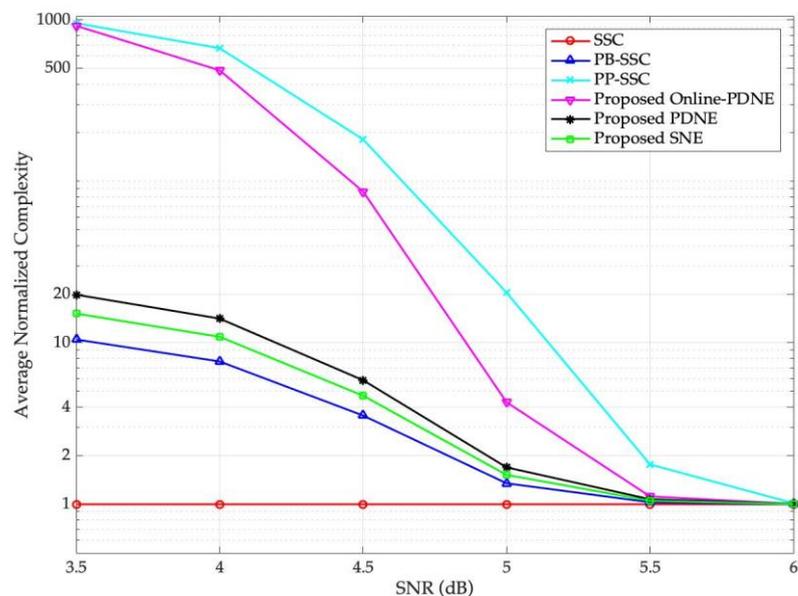
As can be seen from Figure 7 that when SNR = 6.0 dB, the proposed Online-PDNE decoding algorithm improves the FER performance by more than three orders of magnitude compared with the SSC algorithm and more than two orders of magnitude compared with the PB-SSC algorithm and PP-SSC algorithm. Note that compared to the proposed PDNE decoding algorithm, the SNE decoding algorithm significantly reduces the decoding

complexity with negligible performance degradation, which coincide with the results in Figures 4 and 5.

In Figures 8 and 9, we compare the average normalized complexities of the proposed Online-PDNE decoding algorithm, the proposed PDNE decoding algorithm, the SNE decoding algorithm and other decoding algorithms for polar codes with two rates, respectively. The average normalized complexity of these algorithms is normalized by the SSC decoding algorithm.



**Figure 8.** The average normalized complexities for the (1024, 512) polar code. The SSC algorithm is in [14], the PB-SSC algorithm is in [33], and the PP-SSC algorithm is in [35].



**Figure 9.** The average normalized complexities for the (1024, 768) polar code. The SSC algorithm is in [14], the PB-SSC algorithm is in [33], and the PP-SSC algorithm is in [35].

It can be observed that the average normalized complexity of the proposed PDNE decoding algorithm and SNE decoding algorithm is much lower than that of the PP-SSC decoding algorithm, which is due to the offline training. Among the proposed decoding algorithms, the online-PDNE decoding algorithm has the best error correction performance and the highest complexity, but it is still much less complicated than the PP-SSC decoding

algorithm. As the SNR increases, the complexity of the proposed decoding algorithms reduces to the same level as the SSC decoding algorithm. The proposed decoding algorithms can benefit from more reliable channel conditions and require smaller perturbation attempts to successfully decode, thereby reducing the normalized decoding delay.

It can also be seen from Figures 4 and 8 that the proposed PDNE and SNE decoding algorithms have slightly higher average normalized complexity compared with the PB-SSC algorithm at low SNR region. When  $\text{SNR} \geq 2.5$  dB, their complexities are almost the same, but the performance of the proposed PDNE and SNE algorithms is much better than that of the PB-SSC algorithm. In Figures 6 and 8, when  $\text{SNR} \geq 5.5$  dB, similar conclusions can be drawn, which coincides with the results in Figures 4 and 8.

Tables 4 and 5 provide the average normalized complexities corresponding to some fixed SNR values of polar codes with two rates under different decoding algorithms, respectively. Taking Table 4 as an example, when the SNR increases from 0.5 dB to 2.5 dB, the average normalized complexity of the proposed Online-PDNE decoding algorithm drops sharply from 767.300 to 1.293, which is the same order of magnitude as the SSC decoding algorithm.

**Table 4.** The average normalized complexities for the (1024, 512) polar code for certain SNR values.

Decoding Algorithms	SNR			
	0.5 dB	1.5 dB	2.5 dB	3.5 dB
SSC [14]	1	1	1	1
PB-SSC [33]	10.440	3.602	1.004	1.002
PP-SSC [35]	910.800	189.900	2.982	1.042
Proposed Online-PDNE	767.300	60.500	1.293	1.001
Proposed PDNE	19.640	6.195	1.142	1.007
Proposed SNE	15.040	4.902	1.124	1.005

**Table 5.** The average normalized complexities for the (1024, 768) polar code for certain SNR values.

Decoding Algorithms	SNR			
	3.5 dB	4.5 dB	5.5 dB	6.0 dB
SSC [14]	1	1	1	1
PB-SSC [33]	10.475	3.544	1.031	1.002
PP-SSC [35]	951.757	181.610	1.767	1.016
Proposed Online-PDNE	917.032	86.376	1.119	1.007
Proposed PDNE	19.842	5.865	1.075	1.005
Proposed SNE	15.168	4.704	1.052	1.003

## 6. Conclusions

In this paper, three decoding algorithms were proposed for polar codes by exploiting the perturbed and directed neural-evolutionary noise, in which uncorrectable received sequences can be transformed into error-corrected regions of their decoding space. In addition to the SSC decoding algorithm, the proposed algorithms are also applicable to other standard polar code decoding algorithms. Simulation results verified that our proposed Online-PDNE decoding algorithm can achieve better performance than other algorithms and obtain up to four orders of magnitude compared with the SSC algorithm, and no error floor is observed down to a BER of  $10^{-9}$ . The performance of the proposed Online-PDNE decoding algorithm is about 1.0 dB, 0.6 dB and 0.5 dB superior to that of the conventional SSC algorithm, the PB-SSC algorithm, and the PP-SSC algorithm at the BER of  $10^{-5}$ , respectively. This is due to the directed neural-evolutionary noise is generated online by the GA process for the uncorrectable sequences, which converts the received signal into the error correction region of its decoding space more accurately. To further reduce the decoding complexity and simplify the decoding structure, the PDNE and SNE

decoding algorithms were proposed. While ensuring the error correction performance, the complexity is reduced by employing the offline neuro-evolution. In addition, it is worth mentioning that the proposed decoding algorithms can be extended to other channel codes in a straightforward manner. As a future work, we plan to apply the proposed decoding algorithms to the polar codes in the 5G standard.

**Author Contributions:** Conceptualization, L.K. and H.L.; methodology, L.K. and H.L.; software, W.H. and L.K.; validation, W.H. and L.K.; formal analysis, W.H.; investigation, W.H. and B.D.; resources, W.H.; data curation, W.H.; writing—original draft preparation, L.K.; writing—review and editing, H.L.; visualization, W.H. and B.D.; supervision, B.D.; project administration, L.K. and H.L.; funding acquisition, L.K. and H.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the JITSF (grant No. jit-b-202110) and the NSFC (grant No. 61871376).

**Data Availability Statement:** Not applicable.

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

## References

- Vaezi, M.; Azari, A.; Khosravirad, S.R.; Shirvanimoghaddam, M.; Azari, M.M.; Chasaki, D.; Popovski, P. Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road towards 6G. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1117–1174. [[CrossRef](#)]
- Fang, Y.; Bu, Y.; Chen, P.; Lau, F.C.; Al Otaibi, S. Irregular-mapped protograph LDPC-coded modulation: A bandwidth-efficient solution for 6G-enabled mobile networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, 1–14. [[CrossRef](#)]
- Miah, M.S.; Schukat, M.; Barrett, E. Sensing and throughput analysis of a MU-MIMO based cognitive radio scheme for the internet of things. *Comput. Commun.* **2020**, *154*, 442–454. [[CrossRef](#)]
- Chen, L.; Chen, P.; Lin, Z. Artificial intelligence in education: A review. *IEEE Access* **2020**, *8*, 75264–75278. [[CrossRef](#)]
- Hui, D.; Sandberg, S.; Blankenship, Y.; Andersson, M.; Grosjean, L. Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE. *IEEE Veh. Technol. Mag.* **2018**, *13*, 60–69. [[CrossRef](#)]
- Chen, P.; Xie, Z.; Fang, Y.; Chen, Z.; Mumtaz, S.; Rodrigues, J.J. Physical-layer network coding: An efficient technique for wireless communications. *IEEE Netw.* **2020**, *34*, 270–276. [[CrossRef](#)]
- Dai, L.; Fang, Y.; Yang, Z.; Chen, P.; Li, Y. Protograph LDPC-coded BICM-ID with irregular CSK mapping in visible light communication systems. *IEEE Trans. Veh. Technol.* **2021**, *70*, 11033–11038. [[CrossRef](#)]
- Bioglio, V.; Condo, C.; Land, I. Design of polar codes in 5G new radio. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 29–40. [[CrossRef](#)]
- Zhang, H.; Li, R.; Wang, J.; Dai, S.; Zhang, G.; Chen, Y.; Luo, H.; Wang, J. Parity-check polar coding for 5G and beyond. In Proceedings of the IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–7.
- Tataria, H.; Shafi, M.; Molisch, A.F.; Dohler, M.; Sjöland, H.; Tufvesson, F. 6G wireless systems: Vision, requirements, challenges, insights, and opportunities. *Proc. IEEE* **2021**, *109*, 1166–1199. [[CrossRef](#)]
- 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](#)]
- Tal, I.; Vardy, A. How to construct polar codes. *IEEE Trans. Inf. Theory* **2012**, *59*, 6562–6582. [[CrossRef](#)]
- Trifonov, P. Efficient design and decoding of polar codes. *IEEE Trans. Commun.* **2012**, *60*, 3221–3227. [[CrossRef](#)]
- Alamdar-Yazdi, A.; Kschischang, F.R. A simplified successive-cancellation decoder for polar codes. *IEEE Commun. Lett.* **2011**, *15*, 1378–1380. [[CrossRef](#)]
- Tal, I.; Vardy, A. List decoding of polar codes. *IEEE Trans. Inf. Theory* **2015**, *61*, 2213–2226. [[CrossRef](#)]
- Niu, K.; Chen, K. CRC-aided decoding of polar codes. *IEEE Commun. Lett.* **2012**, *16*, 1668–1671. [[CrossRef](#)]
- Chen, K.; Niu, K.; Lin, J. Improved successive cancellation decoding of polar codes. *IEEE Trans. Commun.* **2013**, *61*, 3100–3107. [[CrossRef](#)]
- 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](#)]
- Leroux, C.; Raymond, A.J.; Sarkis, G.; Gross, W.J. A semi-parallel successive-cancellation decoder for polar codes. *IEEE Trans. Signal Process.* **2013**, *61*, 289–299. [[CrossRef](#)]
- Li, B.; Shen, H.; Tse, D. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check. *IEEE Commun. Lett.* **2012**, *16*, 2044–2047. [[CrossRef](#)]
- Balatsoukas-Stimming, A.; Parizi, M.B.; Burg, A. LLR-based successive cancellation list decoding of polar codes. *IEEE Trans. Signal Process.* **2015**, *63*, 5165–5179. [[CrossRef](#)]
- Niu, K.; Dai, J.; Tan, K.; Gao, J. Deep learning methods for channel decoding: A brief tutorial. In Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China, 28–30 July 2021; pp. 144–149.

23. Cammerer, S.; Gruber, T.; Hoydis, J.; Brink, S.T. Scaling deep learning-based decoding of polar codes via partitioning. In Proceedings of the IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
24. Doan, N.; Ali Hashemi, S.; Gross, W.J. Neural successive cancellation decoding of polar codes. In Proceedings of the IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), London, UK, 31 August–3 September 2018; pp. 1–6.
25. Wang, X.; Zhang, H.; Li, R.; Huang, L.; Dai, S.; Huangfu, Y.; Wang, J. Learning to flip successive cancellation decoding of polar codes with LSTM networks. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul, Turkey, 8–11 September 2019; pp. 1–5.
26. Xu, W.; Tan, X.; Be'ery, Y.; Ueng, Y.L.; Huang, Y.; You, X.; Zhang, C. Deep learning-aided belief propagation decoder for polar codes. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2020**, *10*, 189–203. [[CrossRef](#)]
27. Liu, X.; Wu, S.; Wang, Y.; Zhang, N.; Jiao, J.; Zhang, Q. Exploiting error-correction-CRC for polar SCL decoding: A deep learning-based approach. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 817–828. [[CrossRef](#)]
28. Kay, S. Can detectability be improved by adding noise? *IEEE Signal Process. Lett.* **2000**, *7*, 8–10. [[CrossRef](#)]
29. Sutura, A. Stochastic perturbation of a pure connective motion. *J. Atmos. Sci.* **2010**, *37*, 245–249. [[CrossRef](#)]
30. Arli, A.; Gazi, O. Noise-aided belief propagation list decoding of polar codes. *IEEE Commun. Lett.* **2019**, *23*, 1285–1288. [[CrossRef](#)]
31. Xiao, D.; Gu, Z. Dynamic perturbation decoding of polar-CRC cascaded code. In Proceedings of the IEEE International Wireless Communications, and Mobile Computing, Limassol, Cyprus, 5–19 June 2020; pp. 42–45.
32. Lee, H.; Kil, Y.S.; Jang, M.; Kim, S.H.; Park, O.S.; Park, G. Multi-round belief propagation decoding with impulsive perturbation for short LDPC codes. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1491–1494. [[CrossRef](#)]
33. Gerrar, N.K.; Zhao, S.; Kong, L. A CRC-aided perturbed decoding of polar codes. In Proceedings of the 14th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2018), Chongqing, China, 18–20 September 2018. [[CrossRef](#)]
34. Zhao, S.; Xu, P.; Zhang, N.; Kong, L. A decoding algorithm of polar codes based on perturbation with CNN. *J. Electron. Inf. Technol.* **2021**, *43*, 1900–1906.
35. Gerrar, N.K.; Zhao, S.; Kong, L. Error correction in data storage systems using polar codes. *IET Commun.* **2021**, *15*, 1859–1868. [[CrossRef](#)]
36. Kong, L.; Liu, H.; Hou, W.; Meng, C. A perturbed and directed neural-evolutionary decoding algorithm for polar codes. *Electron. Lett.* **2022**, Submitted.
37. Benzi, R.; Sutura, A.; Vulpiani, A. The mechanism of stochastic resonance. *J. Phys. A Math. Gen.* **1981**, *14*, L453–L457. [[CrossRef](#)]
38. McDonnell, M.D.; Abbott, D. What is stochastic resonance? definitions, misconceptions, debates, and its relevance to biology. *PLoS Comput. Biol.* **2009**, *5*, e1000348. [[CrossRef](#)] [[PubMed](#)]
39. Chen, H.; Varshney, P.K.; Michels, J.H.; Kay, S. Approaching near optimal detection performance via stochastic resonance. In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP), Toulouse, France, 14–19 May 2006; p. III. [[CrossRef](#)]
40. Shih, K.; Shiu, D. Perturbed decoding algorithm for concatenated error correcting and detecting codes system. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 11–14 September 2006; pp. 1–5.
41. Wang, W.; Shiu, D. The theory behind perturbed decoding algorithm. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Athens, Greece, 3–7 September 2007; pp. 1–5.
42. He, Z.; Li, X.; Wang, S.; Qi, Y. A noise-assisted polar code attempt decoding algorithm. In Proceedings of the IEEE International Conference on Computer Research and Development (ICCRD), Beijing, China, 5–7 January 2021; pp. 73–77.
43. Nishikawa, M.; Nakamura, Y.; Kanai, Y.; Osawa, H.; Okamoto, Y. A study on iterative decoding with LLR modulator by neural network using adjacent track information in SMR system. *IEEE Trans. Magn.* **2019**, *55*, 1–5. [[CrossRef](#)]
44. Petrović, V.L.; Marković, M.M.; El Mezeni, D.M.; Saranovac, L.V.; Radošević, A. Flexible high throughput QC-LDPC decoder with perfect pipeline conflicts resolution and efficient hardware utilization. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 5454–5467. [[CrossRef](#)]
45. Wang, X.; Ma, Q.; Li, J.; Zhang, H.; Xu, W. An improved SC flip decoding algorithm of polar codes based on genetic algorithm. *IEEE Access* **2020**, *8*, 222572–222583. [[CrossRef](#)]
46. Elkelesh, A.; Ebada, M.; Cammerer, S.; Brink, S.T. Decoder-tailored polar code design using the genetic algorithm. *IEEE Trans. Commun.* **2019**, *67*, 4521–4534. [[CrossRef](#)]
47. Zhou, H.; Gross, W.J.; Zhang, Z.; You, X.; Zhang, C. Low-complexity construction of polar codes based on genetic algorithm. *IEEE Commun. Lett.* **2021**, *10*, 3175–3179. [[CrossRef](#)]
48. Sun, C.; Cai, K.; Song, G.; Quek, T.Q.; Fei, Z. Belief propagation based joint detection and decoding for resistive random access memories. *IEEE Trans. Commun.* **2022**, *70*, 2227–2239. [[CrossRef](#)]
49. Huang, L.; Zhang, H.; Li, R.; Ge, Y.; Wang, J. AI coding: Learning to construct error correction codes. *IEEE Trans. Commun.* **2020**, *68*, 26–39. [[CrossRef](#)]
50. Moreloszaragoza, R.H. *The Art of Error Correcting Coding*, 2nd ed.; John Wiley & Sons: Chichester, UK, 2006; pp. 39–41.