

## Research Article

# Channel Noise Optimization of Polar Codes Decoding Based on a Convolutional Neural Network

Ming Yan <sup>1,2</sup>, Xingrui Lou,<sup>2</sup> and Yan Wang <sup>1,3</sup>

<sup>1</sup>State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China

<sup>2</sup>School of Information and Communications Engineering, Communication University of China, Beijing 100024, China

<sup>3</sup>School of Data Science and Intelligent Media, Communication University of China, Beijing 100024, China

Correspondence should be addressed to Yan Wang; wy@cuc.edu.cn

Received 9 August 2021; Revised 2 October 2021; Accepted 26 October 2021; Published 9 November 2021

Academic Editor: Chunguo Li

Copyright © 2021 Ming Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Polar code has the characteristics of simple coding and high reliability, and it has been used as the control channel coding scheme of 5G wireless communication. However, its decoding algorithm always encounters problems of large decoding delay and high iteration complexity when dealing with channel noise. To address the above challenges, this paper proposes a channel noise optimized decoding scheme based on a convolutional neural network (CNN). Firstly, a CNN is adopted to extract and train the colored channel noise to get more accurate estimation noise, and then, the belief propagation (BP) decoding algorithm is used to decode the polar codes based on the output of the CNN. To analyze and verify the performance of the proposed channel noise optimized decoding scheme, we simulate the decoding of polar codes with different correlation coefficients, different loss function parameters, and different code lengths. The experimental results show that the CNN-BP concatenated decoding can better suppress the colored channel noise and significantly improve the decoding gain compared with the traditional BP decoding algorithm.

## 1. Introduction

As an excellent channel coding technology, the polar code can be theoretically proven to reach the channel capacity based on the phenomenon of channel polarization [1]. The polarization phenomenon is the result of channel merging and channel splitting. During the polarization process, multiple separate uses of the input channel are converted into the continuous use of a set of binary composite channels. After the channel is polarized, some of the channels are nearly perfect, and the channel capacity is close to 1, while the other part is close to 0. Although channel polarization is based on binary discrete memoryless channels (B-DMCs), the polarization phenomenon is widespread. Correlation noise is everywhere in the channel, and the signal is always affected by noise when it is transmitted through the channel.

At present, the two most common polar code decoding algorithms are the successive-cancellation (SC) algorithm [2–4] and belief propagation (BP) algorithm [5, 6]. The SC decoding algorithm has low computational complexity and

very good error correction performance. However, its serial arithmetic structure will cause a longer decoding delay [7]. Compared with SC, the BP algorithm has a parallel data processing structure, which can be repeatedly propagated in the factor graph and has a higher decoding throughput [8]. There are many operations in the improved BP algorithm mentioned in [9–12]; once a short loop appears, it will seriously affect the decoding performance. The research of polar codes is usually based on the ideal channel environment of Gaussian white noise. There are few references on using the knowledge of machine learning to decode polar codes under correlated noise. Correlated noise is the most common noise in actual communication systems, and it is also a major aspect that seriously affects channel coding performance [13–15]. The traditional method of dealing with this noise is to whiten it, that is, to convert the colored noise into white noise or to estimate the probability density distribution of the noise, and then use the estimated joint distribution to construct an optimal decoder [16]. However, the complexity of these two methods is relatively high, which

is not conducive to optimize the spectrum efficiency [17] and energy efficiency [18, 19] of the whole wireless communication network. In particular, the process of whitening involves matrix multiplication, so this is undoubtedly highly complicated for polarized codes with longer codewords.

With the application of deep learning algorithm in face recognition, visual localization, and image processing becoming more and more mature, researchers also began to try to use deep learning algorithm such as convolutional neural network (CNN) in the field of channel coding [20, 21]. The trained CNN is combined with the standard BP decoding algorithm to optimize the noise in the decoding process of low-density parity check (LDPC) code. This idea has achieved good decoding effect in suppressing channel noise [20]. The existing experimental results show that the proposed algorithm framework further reduces the bit error rate (BER) of iterative decoding. In addition, relevant studies show that the application of long- and short-term memory neural network decoder to polar codes in Markov Gaussian memory impulse noise channel can not only reduce the decoding complexity but also obtain more optimized BER performance [22, 23].

Based on the above algorithm ideas, this paper implements a polar code noise optimization algorithm based on a CNN for channel decoding under correlated noise. We use the powerful prediction model of CNN to extract and train the relevant characteristics of noise [24, 25], estimate and whiten channel noise more accurately, calculate a more reliable log-likelihood ratio (LLR), and reduce the impact of noise on decoding performance. Compared with the literature mentioned above [20], the main contribution of this paper is to apply the concatenated algorithm of CNN and BP to the decoding of polar codes. By offline training the CNN, linear simulation is carried out according to the generated values, and the specific decoding gain is analyzed, so as to optimize the channel noise. The experimental results show that the noise optimized concatenated structure based on CNN has obvious gain in BER performance.

The remainder of the paper is organized as follows. Section 2 briefly introduces the basic principles of the traditional BP decoding algorithm for polar codes. Section 3 proposes a CNN-based polar code noise optimization decoding structure. Section 4 shows the performance analysis of this algorithm and its comparison with other methods. Finally, Section 5 summarizes the paper and presents the research prospects.

## 2. Traditional BP Decoding Algorithm

As an iterative algorithm based on the idea of information transmission, BP decoding algorithm has corresponding theoretical research under different types of coding methods [26]. At present, BP decoding algorithm has been widely used in the decoding of LDPC codes, because it has the characteristic of keeping the confidence value of nodes convergent all the time. The process of BP decoding is bidirectional. In some specific cases, it can stop the iteration before reaching the artificially set maximum number of iterations.

In [27], Professor Arıkan proposed a factor graph to represent the polar code, and the BP algorithm was used to decode the polar code [28]. If a coset code is expressed as  $(N, K, A, u_{A^c})$ , where  $N$  is the code length and  $N = 2^n$ ,  $K$  is the dimension,  $A$  is the information set determined by the channel polarization, and  $u_{A^c}$  is the fixed bit [29]. The total number of nodes in the factor graph is  $N(n+1)$ , and each node is represented by an integer pair  $(i, j)$ . Figure 1 shows the factor diagram of the polar code when  $N = 8$ . There are three stages, and each stage includes four processing units [20, 30]. The information  $L_{n+1,j}$  (e.g.,  $L_{2,1}$  to  $L_{2,8}$ ) in the figure is allocated by the LLR of the channel output, and the information  $R_{1,j}$  in the first column on the left is the predecoded value of the LLR. The information transfer process of each basic processing unit is shown in Figure 2.

As shown in Figure 2, in each basic processing unit, the transfer and update of information are carried out between adjacent nodes [31]. The symbol  $L_{i,j}$  in the figure represents the update of the information node to the left;  $R_{i,j}$  indicates the update of the information node to the right. One iteration of the BP decoding algorithm is that the information is first transmitted from the rightmost node to the leftmost node and then from the leftmost node to the rightmost node [32, 33]. In this process, the transmission of information is expressed in the form of LLR. The traditional iterative formula is as follows:

$$\begin{cases} L_{i,j} = g(L_{i+1,j}, L_{i+1,j+2^{n-i}} + R_{i,j+2^{n-i}}), \\ L_{i,j+2^{n-i}} = g(R_{i,j}, L_{i+1,j}) + L_{i+1,j+2^{n-i}}, \\ R_{i+1,j} = g(R_{i,j}, L_{i+1,j+2^{n-i}} + R_{i,j+2^{n-i}}), \\ R_{i+1,j+2^{n-i}} = g(R_{i,j}, L_{i+1,j}) + R_{i+1,j+2^{n-i}}, \end{cases} \quad (1)$$

where  $1 \leq i \leq n$ ,  $1 \leq j \leq 2^{n-1}$ , and  $g(x, y) = 2 \tanh^{-1}(x/2) \cdot \tanh^{-1}(y/2)$ . The information from the channel  $L_{n+1,j}$  ( $1 \leq j \leq N$ ) is defined as follows:

$$L_{n+1,j} = \ln \frac{P(y_j | x_i = 0)}{P(y_j | x_i = 1)}. \quad (2)$$

The initialization information  $R_{i,j}$  ( $1 \leq j \leq N$ ) is defined as follows:

$$R_{i,j} = \begin{cases} 0, & j \in A, \\ \infty, & j \in A^c. \end{cases} \quad (3)$$

The steps of the BP decoding algorithm based on polar codes are as follows:

- (a) Calculate the value  $L_{n+1,j}$  in Equation (2) and use it as the initial channel information, where  $(1 \leq j \leq N)$
- (b) In each iteration, the node information of each processing unit is updated by the above formula

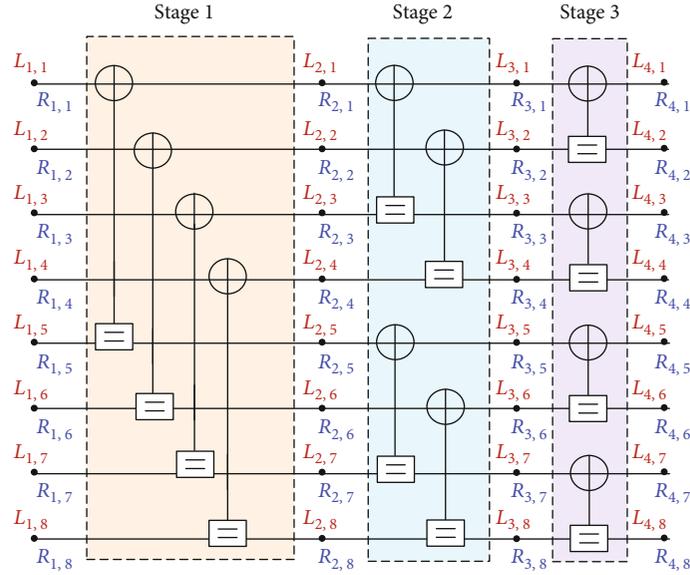
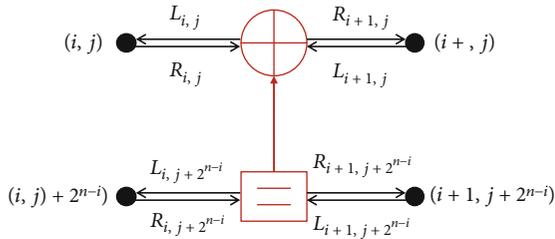

 FIGURE 1: Factor graph of  $N = 8$  polar codes.


FIGURE 2: Transfer process of the basic processing unit.

according to the iteration rules of the BP algorithm of the polar code [34]

- (c) When the number of iterations reaches the set value, a decision is made: if  $j \in A^c$  or  $j \in A$  and  $L_{i,j} > 0$ , the decision  $u_i$  is 0; otherwise, the decision is 1

Although the effect of BP decoding algorithm has been widely proved to be excellent, it is still very difficult to implement in specific hardware. Some scholars have made many improvements on the basis of traditional BP decoding algorithm, including early termination strategy based on mean and variance and approximate replacement with minimum sum algorithm [5–7]. Here, we mainly introduce the traditional BP decoding algorithm in detail.

### 3. Proposed CNN Noise Optimization Algorithm

In general, the research of polar codes is usually based on the ideal channel environment of additive white Gaussian noise (AWGN) channel. But in the actual communication system, due to the interference of atmospheric environment, the actual error of dynamic model does not belong to Gaussian white noise sequence, but belongs to the colored noise with certain temporal characteristics. The common solution to

this kind of colored noise is whitening process. Since the whitening process involves matrix multiplication, it has a high complexity for polar codes with long codewords.

Although the typical SC decoding algorithm has good bit error rate performance, but because of its bit by bit decoding characteristics, when the code length is large, the decoding delay will be a big problem. BP decoding algorithm does not have the above problems because of its parallel data processing structure, but its bit error rate performance under the influence of channel noise is not satisfactory. Therefore, the optimized channel noise CNN-BP concatenated system of polar codes is introduced in this chapter. Firstly, the framework of the whole concatenated decoding system is briefly introduced, and then, the Gaussian colored noise that may appear in the channel is modeled. In addition, the generation of noise sample data, the design of convolution neural network model, and the training of the network are also discussed.

**3.1. System Framework.** The flowchart of the optimized structure of channel noise based on a CNN is shown in Figure 3. It should be noted that this architecture mainly refers to the iterative BP-CNN architecture for channel decoding proposed earlier [20]. The information symbol  $u$  is encoded as  $x$  by the encoder. After binary phase-shift keying (BPSK) modulation and mapping, the symbol vector  $s$  is obtained. The modulated  $s$  is transmitted through the channel with additive noise.

In Figure 3,  $n$  is the autocorrelation matrix of Gaussian random vectors. At the decoding receiving end, the received signal  $y$  can be expressed as  $y = s + n$ , and the result  $s'$  is obtained through judgment.  $n'$  represents a preliminary estimate of noise, and the estimated value can be expressed as  $n' = y' - s'$ . Taking  $n'$  as the input of the CNN, we can obtain a more accurate noise estimation and make the residual noise follow a Gaussian distribution as much as possible.

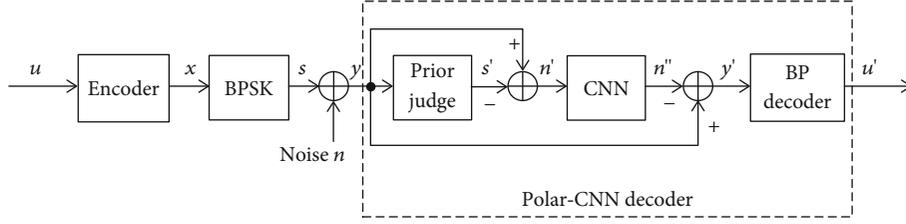


FIGURE 3: Flowchart based on the CNN noise optimization structure.

$n''$  is used to represent the output of the CNN, and the vector  $y'$  after noise suppression can be expressed as  $y' = y - n''$ .

Before decoding, we also corrected the LLR value according to the received value and residual noise distribution.

$$\text{LLR}_i = \log \frac{P_{0,i} + 10^{-7}}{P_{1,i} + 10^{-7}}, \quad (4)$$

where  $P_{0,i}$  and  $P_{1,i}$  represent the probability that the  $i$ th element of  $y'$  is 0 or 1. Finally, the more reliable LLR is input into the polar codes decoding to obtain the final information symbol estimate  $u'$ .

**3.2. Establishment of Channel Noise Model.** After the encoding operation, the code information of the transmitter is transmitted by BPSK modulation. In the process of transmission, the signal is destroyed because of the inevitability of noise, and then, the signal at the transmitter is interfered by noise to form the received signal.

In this paper, the colored noise in the channel is modeled by a first-order impulse response filter. Through an infinite impulse response (IIR) digital filter, AWGN be regarded as the colored noise commonly encountered in communication system, and its Z transform is as follows:

$$F_C(z) = \frac{a}{1 + bz^{-1}}, \quad (5)$$

where  $a$  represents a gain factor, which generally refers to the amplification factor, i.e., output divided by input.  $b$  represents a correlation coefficient,  $-1 < b < 1$ . When  $b = 0$ , the digital filter becomes an all-pass filter without any correlation, and the result of filtering is still white Gaussian noise. As can be seen from Equation (5), the IIR filter associates each white Gaussian noise sample with the preceding noise sample. Additive colored Gaussian noise (ACGN) can be obtained when the white Gaussian noise passes through an IIR filter. The ACGN of the  $j$ th sample can be expressed as follows:

$$n_C(j) = a \sum_{i=0}^j (-b)^i n_w(j-i). \quad (6)$$

It can be seen from Equation (6) that as the value of  $b$  gets larger, the interference between the noise samples before and after will be greater, so the filter coefficient  $b$  is generally called the correlation coefficient. If the power of the white noise before filtering needs to be the same as that of the col-

ored noise after filtering, then the square deviation of the two noise samples needs to be the same. Therefore, it is necessary to limit the value range of gain factor  $a$ , which is generally set to  $a = \sqrt{1 - b^2}$ . In this way, the power of noise can be expressed as follows:

$$\sigma_w^2 = \sigma_c^2 = \frac{1}{2} (10^{-\text{SNR}/10} \times E_b), \quad (7)$$

where SNR represents the signal-to-noise ratio (SNR) in dB,  $E$  represents the energy, and  $\sigma_w^2$  and  $\sigma_c^2$  represent the square deviation of AWGN and ACGN, respectively. Figure 4 shows the amplitude frequency response of  $F_C(z)$  for different values of  $b$ . It can be observed that when  $b > 0$ , with the increase of the value of  $b$ ,  $F_C(z)$  becomes higher pass, which is a good noise for high-frequency signal, and the energy of low-frequency signal is smaller while that of high-frequency signal is more concentrated. When  $b < 0$ ,  $F_C(z)$  becomes lower pass as the absolute value of  $b$  increases. At this time, the colored noise will become more and more close to the red noise, which will selectively absorb the signal with higher frequency. To avoid confusion, we only consider the system performance when  $b > 0$  in the following simulation experiments.

**3.3. CNN Structure.** In the system architecture proposed in this paper, the network structure adopted is shown in Figure 5. Assuming the CNN structure has  $l$  layers (the input layer is not counted), the number of convolution kernels in each layer is  $f_1, f_2, f_3, \dots, f_l$ , and the number of feature maps is  $k_1, k_2, k_3, \dots, k_l$ . The structure of the CNN is determined by the number of convolutional layers, the size of filters, and the number of feature maps in each layer. These parameters need to be determined before training the network. Our CNN structure contains a total of four convolutional layers. The first layer uses 64 filters, and the size of the convolution kernel is 9. The second layer uses 32 filters, and the size of the convolution kernel is 3. The third layer uses 16 filters, and the convolution kernel size is 3. The last layer uses 1 filter, and the convolution kernel size is 15. Unlike the network structure used for low-level tasks in image denoising and superresolution, our network input is a one-dimensional vector instead of a two-dimensional image.

The performance of neural networks largely depends on the choice of loss function. The loss function measures the difference between the actual output and the expected output of the CNN, which needs to be carefully defined according to the specific tasks of the network. In the framework of this paper, CNN is mainly used for channel noise estimation,

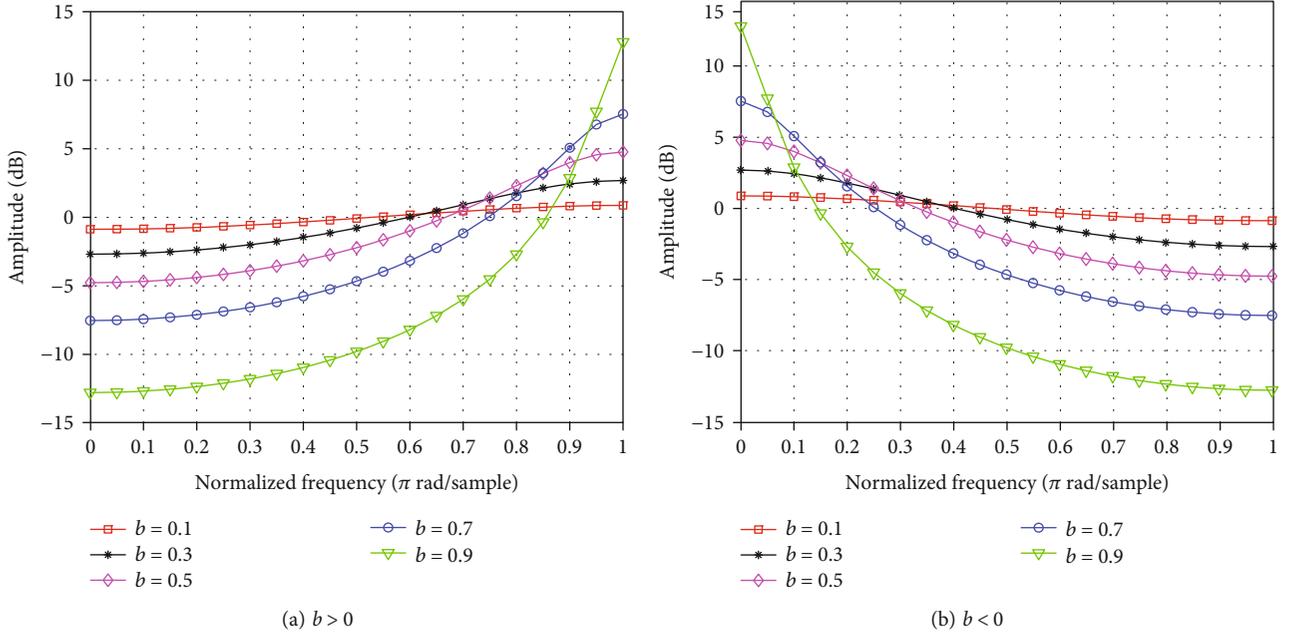


FIGURE 4: Amplitude frequency response under different correlation coefficients.

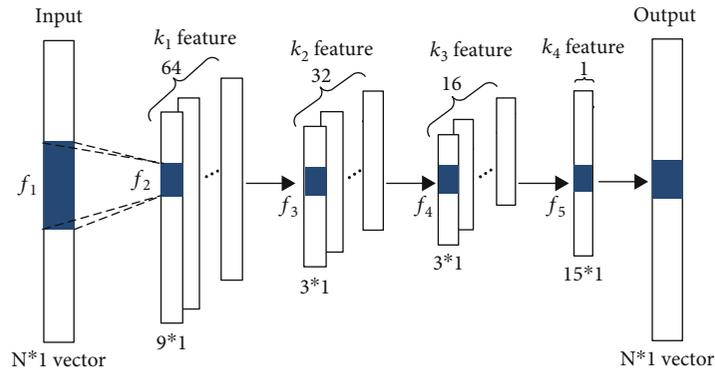


FIGURE 5: The CNN structure of the system model in this paper.

and its output will affect the performance of the next decoding. We introduced the loss function of the normal test [20, 30] so that the possibility of residual noise following the Gaussian distribution can be measured. The loss function is defined as follows:

$$\text{Loss} = \frac{\|n - n'\|^2}{N} + \lambda \left( S^2 + \frac{1}{4}(C - 3)^2 \right). \quad (8)$$

The first term on the right side of the equation is used to measure the power of residual noise, and the second term uses the Jarque-Bera test. In statistics,  $\lambda$  is a scale factor that balances the two terms. It should be noted that the choice of  $\lambda$  value is very important for the training model and the final decoding result. Therefore, in the experiment, we will use different  $\lambda$  values as much as possible to compare and analyze the decoding performance of polar codes. In addition,

$S$  and  $C$  are skewness and kurtosis, respectively.  $S$  and  $C$  are defined as follows:

$$\begin{cases} S = \frac{1/N \sum_{i=1}^N (r_i - \bar{n})^3}{\left(1/N \sum_{i=1}^N (r_i - \bar{n})^2\right)^{3/2}}, \\ C = \frac{1/N \sum_{i=1}^N (r_i - \bar{n})^4}{\left(1/N \sum_{i=1}^N (r_i - \bar{n})^2\right)^2}, \end{cases} \quad (9)$$

where  $r_i$  represents the  $i$ th element in the residual noise vector and  $\bar{n}$  represents the sample mean. Although the normality test method is not optimal, its function is derivable, and training is relatively simple.

The power of noise is reduced by subtracting the estimated value of the channel by the received value, which is equivalent to improving the signal-to-noise ratio (SNR).

The residual noise following the Gaussian distribution is equivalent to whitening the noise, which is more convenient and reliable for calculating LLR. Experiments also show that the loss function further reduces the BER of iterative decoding compared with the standard quadratic function.

#### 4. Performance Analysis and Comparison

In this section, we compare the performance of polar code decoding algorithms under different conditions. The simulation experiments in this paper are based on Python development environment, implemented in TensorFlow version 1.10.0 framework, and data simulation analysis is carried out on MATLAB. The CNN structure is {4; 9, 3, 3, 15; 64, 32, 16, 1}, and the random gradient descent method is used to train CNN until the value of loss function does not decrease. At the same time, Adam optimization method is used, and the training data is generated under different channel SNR values, such as {0, 0.5, 1, 1.5, 2, 2.5, 3} dB. Except for the special statement, the length of the polar codes in the simulation experiment in this paper is 256 and the code rate is 0.5. In order to understand the influence of different parameter settings on decoding performance, in our simulation experiment, multiple correlation coefficients  $b$  (0.5, 0.6, 0.7, 0.8, 0.9) and corresponding different scale factors  $\lambda$  were tested. For example, when  $b=0.6$ , a wide range of  $\lambda$  values are tested, ranging from 0.01 to 100.

*4.1. Decoding Performance with Different Correlation Coefficients and Scale Factors.* The effects of different correlation coefficient  $b$  and scale factor  $\lambda$  on the performance of CNN-BP concatenated decoding and traditional BP decoding are compared to select the optimal parameter values. At present, the selection of  $\lambda$  is still at the experimental level, and the most appropriate value has not been determined. A very small  $\lambda$  may not guarantee that the residual noise follows the Gaussian distribution, while a very large  $\lambda$  may not reduce the power of the residual noise. Therefore, the selection of scale factor  $\lambda$  will indirectly affect the decoding performance of polar codes.

By fixing the correlation coefficient  $b$ , CNN models are trained for different  $\lambda$  values to obtain different training models. Then, the decoding performance of different models is simulated to select the corresponding  $\lambda$  value with the lowest BER. Figure 6 shows the simulation results of CNN-BP concatenated decoding with different  $\lambda$  values when the correlation coefficient  $b$  is 0.6 and standard BP when the number of iterations is 12. It can be observed that the decoding performance of CNN-BP concatenated scheme is not optimal when the  $\lambda$  is large or small. For example, when  $\lambda = 10$  or  $\lambda = 100$ , the decoding performance of CNN-BP is poor, and its BER is higher than that of standard BP algorithm. When  $\lambda = 0.1$  or  $\lambda = 1$ , the performance of CNN-BP decoding is better than that of standard BP. In addition, when  $\lambda$  takes other values, such as 0.01, 0.03, or 0.07, the corresponding BER is higher than that when  $\lambda = 0.1$  or  $\lambda = 1$ . Therefore, selecting an appropriate  $\lambda$  value will help to improve the decoding performance of CNN-BP concatenated scheme.

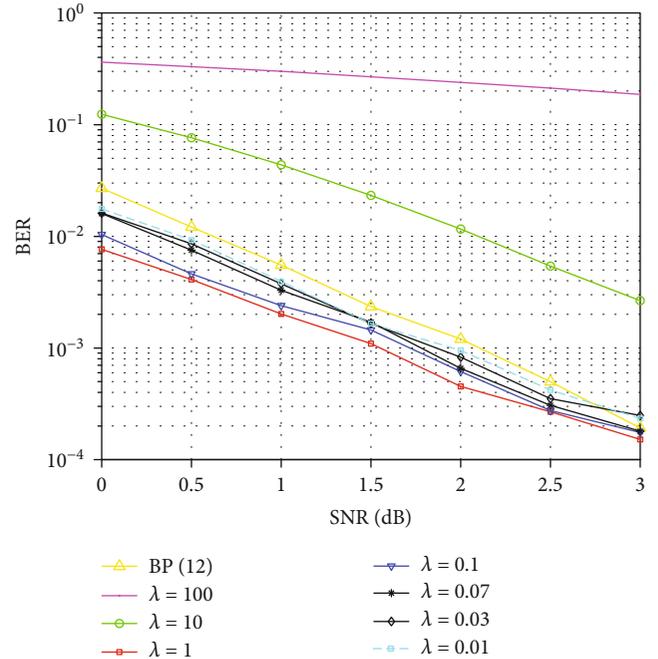


FIGURE 6: Simulation results of CNN-BP concatenated decoding with different  $\lambda$  and standard BP decoding.

According to the above conclusions, only the simulation results of CNN-BP when  $\lambda = 0.1$  and  $\lambda = 1$  are considered in the following experiments. Figure 7 shows the BER performance of CNN-BP concatenated decoding scheme with these two scale factors. The experimental data with correlation coefficients of 0.5, 0.6, 0.7, 0.8, and 0.9 are given here. It can be observed that the BER will decrease with the increase of correlation coefficient. In addition, when the correlation coefficient is small, the BER of CNN-BP concatenated decoding scheme with  $\lambda = 1$  is lower than that with  $\lambda = 0.1$ . However, when the correlation coefficient is large, the BER of CNN-BP concatenated decoding scheme with  $\lambda = 1$  is higher than that with  $\lambda = 0.1$ .

It can be noted that as the noise correlation intensity increases, it tends to choose smaller  $\lambda$ . This is due to that there is more information to reduce the input error of neural network when the channel noise has strong correlation. Therefore, choosing a smaller  $\lambda$  can make the network focus on reducing the power of residual noise. However, when the noise shows weak correlation, relatively little information can be used. In this case, the neural network needs to pay more attention to the distribution of residual noise, so it tends to choose a larger  $\lambda$ .

In addition, to reflect more differences, a separate simulation experiment is carried out on the BER performance of the standard BP decoding with different correlation coefficients. Figure 8 shows the performance of the standard BP decoding algorithm.

With the increase of channel noise correlation intensity, the standard BP decoding algorithm cannot resist the noise interference, resulting in the increase of BER of decoding. In the decoding algorithm using concatenated CNN, the influence of channel noise can be adjusted by the scale factor  $\lambda$ .

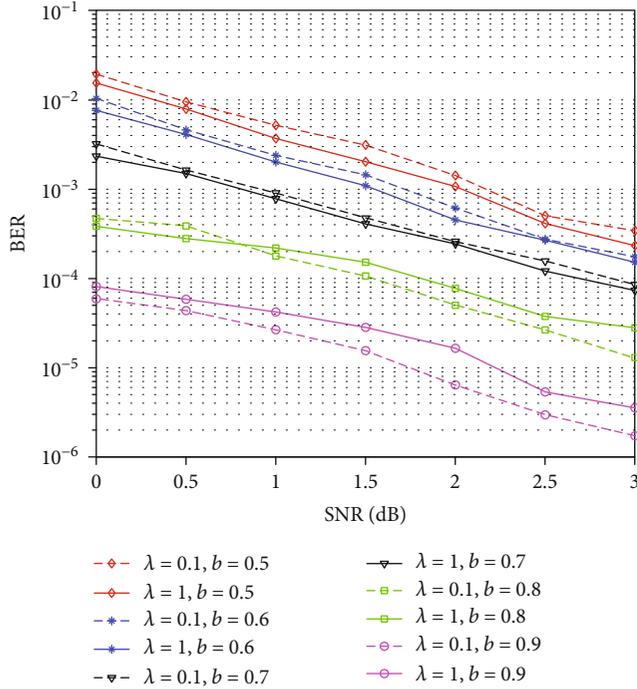


FIGURE 7: Performance comparison of CNN-BP concatenated decoding scheme with different parameters.

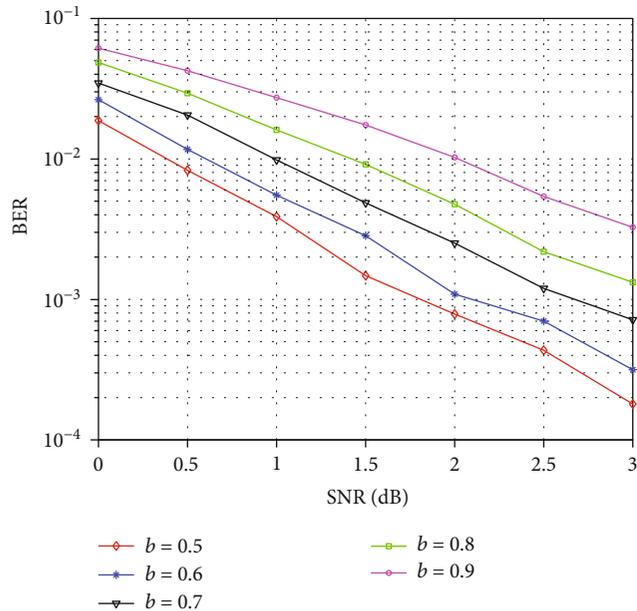


FIGURE 8: Performance of the standard BP decoding algorithm with different correlation coefficients.

When the channel noise has strong correlation, there can be more information to reduce the input error of neural network, so as to reduce the BER of decoding. In addition, comparing Figure 7 with Figure 8, it can be observed that when the correlation coefficient  $b \geq 0.6$ , the performance of CNN-BP concatenated decoding is significantly improved compared with the standard BP decoding. Table 1 shows the gain of

TABLE 1: BER gain of CNN-BP concatenated decoding.

$b$ value	$\lambda$ value	Gain (dB)
0.5	0.1	-0.6
	1	-0.3
0.6	0.1	0.5
	1	0.65
0.7	0.1	1.8
	1	2
0.8	0.1	4
	1	4
0.9	0.1	5
	1	5

CNN-BP concatenated decoding compared with standard BP decoding when the BER reaches  $10^{-3}$ . For example, when  $b = 0.6$  and  $\lambda = 0.1$ , CNN-BP concatenated decoding has a BER gain of 0.5 dB compared with the standard BP decoding.

It can be observed from Table 1 that when the correlation coefficient is relatively large, CNN-BP concatenated decoding has obvious gain compared with standard BP decoding. It should be noted that in the standard BP decoding, the gain cannot be compensated by more iterations. When the correlation coefficient is small, such as  $b = 0.5$ , the BER of CNN-BP concatenated decoding is higher than that of standard BP decoding. This is due to that when the correlation of colored noise is weak, only a few characteristics of the channel noise can be extracted, and the advantage of CNN is not obvious. When the correlation of colored noise is strong, CNN network can better extract the characteristics of noise, which makes CNN-BP concatenated decoding can significantly improve the decoding performance of polar codes.

*4.2. Decoding Performance with Different Code Lengths.* In this subsection, the simulation experiments of polar codes with different code lengths are carried out. In the experiment, the bit rate of polar codes is 0.5, the number of iterations is 12, and the correlation coefficient of channel noise is 0.8 (colored noise is strongly correlated). Considering the comparison and analysis with the standard BP decoding algorithm, two scale factor values  $\lambda = 0.1$  and  $\lambda = 1$  are still selected for simulation experiments. For further comparison, the decoding when  $\lambda = 0$  and  $b = -0.8$  are simulated.

Figure 9 shows the BER performance when the code length of polar codes is 64, 128, and 512, respectively. It can be observed that the decoding performance of CNN-BP concatenated decoding is about 2 orders of magnitude better than that of standard BP decoding when the code length is 64. With the increase of code length, the improvement of decoding performance is more obvious. In addition, with the increase of SNR, the decoding performance of CNN-BP concatenated decoding is obviously improved.

In addition, the decoding performance of the standard BP algorithm is almost the same when  $b = -0.8$  and  $b = 0.8$  with the same code length. However, the results of CNN-BP concatenated decoding are significantly different in these

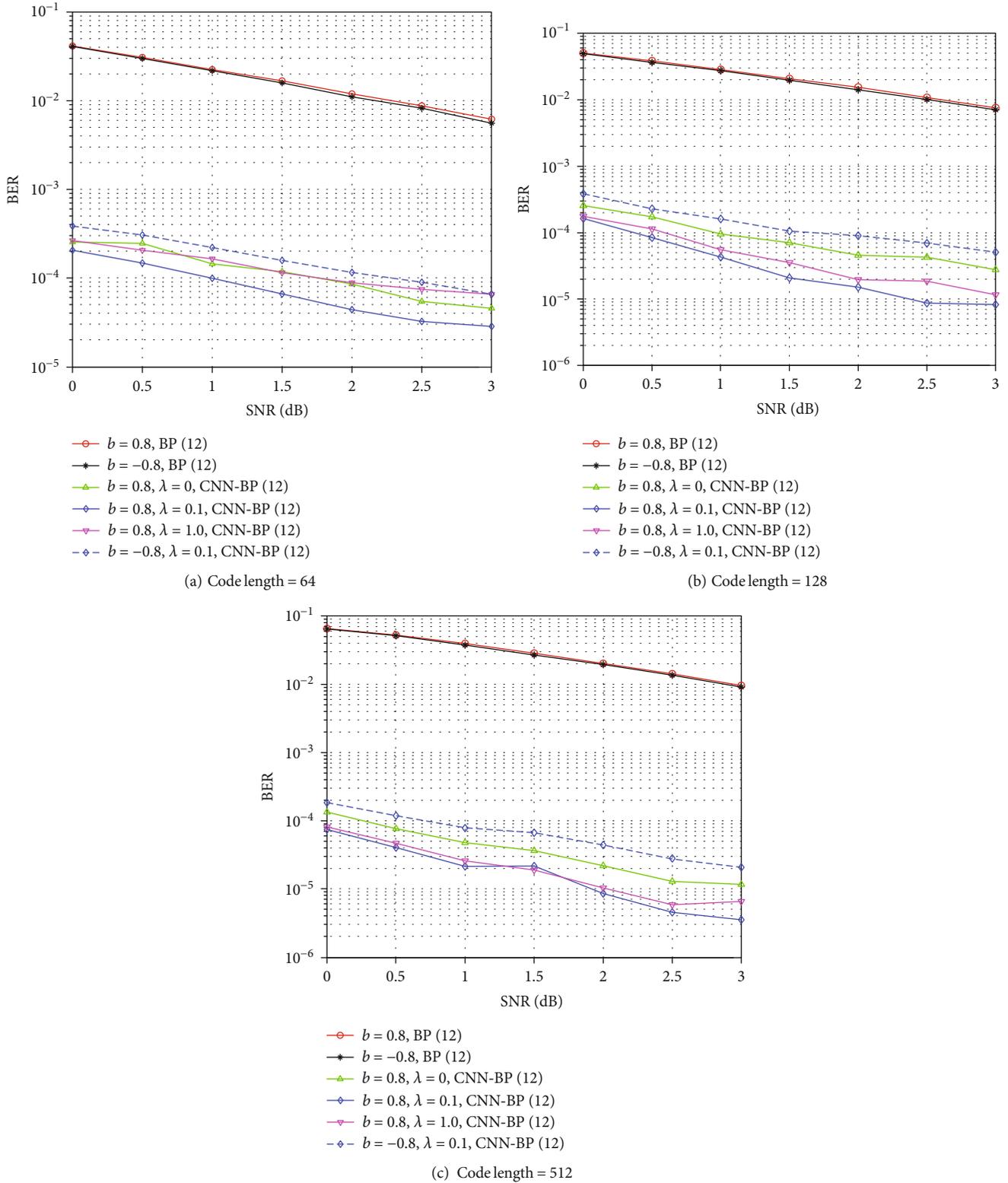


FIGURE 9: BER performance with different code lengths.

two cases. This shows that the standard BP algorithm cannot distinguish different noise scenes with the certain correlation intensity, and the latter can distinguish. In addition, this subsection also simulates the CNN-BP concatenated decod-

ing scheme when  $\lambda = 0$ .  $\lambda = 0$  means that the normal loss function has become a common loss function in neural network regression. It can be observed that no matter what kind of loss function is used to form CNN-BP concatenated

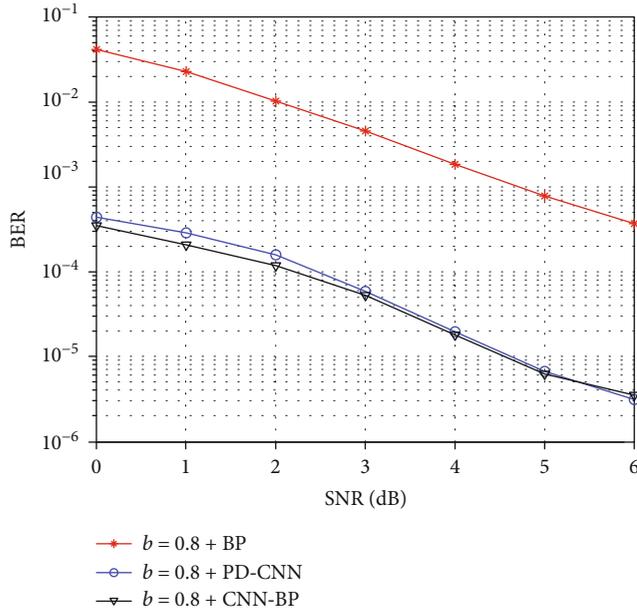


FIGURE 10: Performance comparison between different decoding algorithms.

decoding, the BER is lower than that of standard BP algorithm.

**4.3. Performance Comparison with Other Polar Codes Decoding Algorithms.** To show the difference between the proposed polar codes decoding algorithm and other related algorithms, the prior decision CNN (PD-CNN) decoding algorithm is selected to compare the performance. The PD-CNN decoding algorithm uses deep learning to decode the polar codes. It mainly makes a simple prediction of 0 or 1 for the information sequence at the receiving end and then uses CNN for decoding [35]. Figure 10 shows the performance comparison between CNN-BP concatenated scheme and other decoding algorithms when the correlation coefficient is 0.8 and the code rate is 0.9.

It can be observed that the BER of CNN-BP concatenated decoding scheme is lower when SNR is small. This is due to that the noise is relatively large when the SNR is low, and the prediction error rate of information sequence in PD-CNN decoding algorithm is relatively high. When SNR is more than 5 dB, the decoding performance of CNN-BP concatenated algorithm of polar codes is slightly lower than that of PD-CNN algorithm. The two CNN-based methods have better performance than the standard BP decoding algorithm, and in general, the proposed CNN-BP concatenated scheme has stronger SNR adaptability.

## 5. Conclusions

With the advent of 5G era, mobile communication technology is accelerating, and the decoding algorithm of polar codes has attracted more and more attention of researchers. At present, the processing method of colored noise in the channel is not perfect, and there are some problems, such

as high BER and high time delay, which are difficult to meet the actual needs in specific applications.

In this paper, we use the regression function of CNN to realize a polar code decoding structure that can more accurately estimate channel noise and more reliably calculate LLR. The comprehensive results show that with the increase in SNR, the BER curve of the decoding algorithm proposed in this paper has a good downward trend. CNNs can be trained offline, so this architecture has significant advantages in reducing complexity. Since the proposed architecture is a general solution based on CNN to optimize noise operation, it can also be applied to any other type of CNN-based polar code decoders.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

The authors would like to thank the authors of the paper entitled “An Iterative BP-CNN Architecture for Channel Decoding” which is available at <https://www.arxiv-vanity.com/papers/1707.05697/>. In this paper, the iterative BP-CNN architecture for channel decoding proposed by them is transplanted from LDPC decoding to polar codes decoding, and this transplantation obtains good decoding performance. This paper was supported in part by the Beijing Municipal Natural Science Foundation (Grant No. 9202018), the National Natural Science Foundation of China (Grant No. 61971382), and the Fundamental Research Funds for the Central Universities (Grant Nos. CUC21GZ005 and CUC210B022).

## References

- [1] E. Arıkan, “Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [2] K. Lay and C. Chang, “Enhanced successive cancellation for decoding of polar codes with very low latency through multiple parity checks,” in *IEEE International Conference on Intelligent Green Building and Smart Grid*, pp. 366–370, Hubei, China, 2019.
- [3] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, “LLR-based successive cancellation list decoding of polar codes,” *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5165–5179, 2015.
- [4] C. Gao, R. Liu, and X. Han, “Path splitting selecting strategy-aided successive cancellation list algorithm for polar codes,” *IEEE Communications Letters*, vol. 23, no. 3, pp. 422–425, 2019.

- [5] B. Feng and R. Liu, "Efficient-memory and low-latency BP decoding algorithm for polar codes," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1236–1239, 2020.
- [6] Q. Zhang, A. Liu, and X. Tong, "Early stopping criterion for belief propagation polar decoder based on frozen bits," *Electronics Letters*, vol. 53, no. 24, pp. 1576–1578, 2017.
- [7] Y. Chen, J. Chen, X. Yu, G. Xie, C. Zhang, and C. Zhang, "Belief propagation decoding of polar codes using intelligent post-processing," *Journal of Signal Processing Systems*, vol. 92, no. 5, pp. 487–497, 2020.
- [8] M. P. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Transactions on Communications*, vol. 47, no. 5, pp. 673–680, 1999.
- [9] J. Sha, X. Liu, Z. Wang, and X. Zeng, "A memory efficient belief propagation decoder for polar codes," *China Communications*, vol. 12, no. 5, pp. 34–41, 2015.
- [10] M. Xu, X. Liang, B. Yuan, Z. Zhang, X. You, and C. Zhang, "Stochastic belief propagation polar decoding with efficient re-randomization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6771–6776, 2020.
- [11] M. A. Hanif and S. Vafi, "A novel scheme of product polar codes with high efficiency and rate compatibility," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 7573147, 13 pages, 2020.
- [12] Q. Yu, Z. Shi, L. Deng, and X. Li, "An improved belief propagation decoding of concatenated polar codes with bit mapping," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1160–1163, 2018.
- [13] C. Li, S. Zhang, P. Liu, F. Sun, J. M. Cioffi, and L. Yang, "Overhearing protocol design exploiting intercell interference in cooperative green networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 441–446, 2016.
- [14] M. Yan, S. Li, C. A. Chan, Y. Shen, and Y. Yu, "Mobility prediction using a weighted Markov model based on mobile user classification," *Sensors*, vol. 21, no. 5, p. 1740, 2021.
- [15] X. Liu, X. Zhang, M. Jia, L. Fan, W. Lu, and X. Zhai, "5G-based green broadband communication system design with simultaneous wireless information and power transfer," *Physical Communication*, vol. 28, pp. 130–137, 2018.
- [16] C. Wen, J. Xiong, L. Gui, Z. Shi, and Y. Wang, "A novel decoding scheme for polar code using convolutional neural network," in *2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 1–5, Jeju, Korea (South), 2019.
- [17] C. Li, P. Liu, C. Zou, F. Sun, J. M. Cioffi, and L. Yang, "Spectral-efficient cellular communications with coexistent one- and two-hop transmissions," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6765–6772, 2016.
- [18] C. Li, Y. Li, K. Song, and L. Yang, "Energy efficient design for multiuser downlink energy and uplink information transfer in 5G," *Science China Information Sciences*, vol. 59, no. 2, pp. 1–8, 2016.
- [19] M. Yan, H. Yuan, Z. Li, Q. Lin, and J. Li, "Energy savings of wireless communication networks based on mobile user environmental prediction," *Journal of Environmental Protection and Ecology*, vol. 22, no. 1, pp. 206–217, 2021.
- [20] F. Liang, C. Shen, and F. Wu, "An iterative BP-CNN architecture for channel decoding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 144–159, 2018.
- [21] W. Xu, X. Tan, Y. Be'ery et al., "Deep learning-aided belief propagation decoder for polar codes," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 2, pp. 189–203, 2020.
- [22] S. M. Tseng, W. C. Hsu, and D. F. Tseng, "Deep learning based decoding for polar codes in Markov Gaussian memory impulse noise channels," *Wireless Personal Communications*, pp. 1–17, 2021.
- [23] J. Gao, D. Zhang, J. Dai, K. Niu, and C. Dong, "ResNet-like belief-propagation decoding for polar codes," *IEEE Wireless Communications Letters*, vol. 10, no. 5, pp. 934–937, 2021.
- [24] Q. Zhu, P. Zhang, and X. Ye, "A new loss function for CNN classifier based on predefined evenly-distributed class centroids," *IEEE Access*, vol. 8, pp. 10888–10895, 2020.
- [25] D. Xiong and B. Tian, "Deep learning method of polar codes under colored noise," in *2019 IEEE International Conference on Communications in China*, pp. 677–682, Changchun, China, 2019.
- [26] Y. Ren, W. Xu, and C. Zhang, "Efficient belief propagation list decoding of polar codes," in *2019 IEEE 13th International Conference on ASIC*, pp. 1–4, Chongqing, China, 2019.
- [27] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447–449, 2008.
- [28] G. D. Forney, "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, 2001.
- [29] Y. Meng, Y. Fang, C. Zhang, and L. Li, "LLR processing of polar codes in concatenation systems," *China Communications*, vol. 16, no. 9, pp. 201–208, 2019.
- [30] Q. Wang, P. Fu, and S. Zhang, "A comparison of concatenated polar codes with different interleaving and decoding schemes," in *2020 5th International Conference on Computer and Communication Systems*, pp. 570–574, Shanghai, China, 2020.
- [31] H. Saber, Y. Ge, R. Zhang, W. Shi, and W. Tong, "Convolutional polar codes: LLR-based successive cancellation decoder and list decoding performance," in *2018 IEEE International Symposium on Information Theory*, pp. 1480–1484, Vail, CO, USA, 2018.
- [32] A. Cao, L. Zhang, and Y. He, "An LLR-based segmented flipped SCL decoding algorithm for polar codes," in *2019 IEEE International Conference on Communications in China*, pp. 724–729, Changchun, China, 2019.
- [33] Y. El-Kurdi, D. Fernández, and W. J. Gross, "Acceleration of the finite-element Gaussian belief propagation solver using minimum residual techniques," *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 1–4, 2016.
- [34] Y. Shen, W. Song, and C. Zhang, "Enhanced belief propagation decoder for 5G polar codes with bit-flipping," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 5, pp. 901–905, 2020.
- [35] P. Xu, L. Kong, S. Zhao, and B. Zheng, "A decoding algorithm for polar codes based on PD-CNN," *Journal of Signal Processing*, vol. 35, no. 10, pp. 1652–1660, 2019.