

## LETTER

# DNN Aided Joint Source-Channel Decoding Scheme for Polar Codes\*

Qingping YU<sup>†a)</sup>, You ZHANG<sup>†b)</sup>, Zhiping SHI<sup>††c)</sup>, Xingwang LI<sup>†††d)</sup>, *Nonmembers*, Longye WANG<sup>†e)</sup>, *Member*, and Ming ZENG<sup>††††f)</sup>, *Nonmember*

**SUMMARY** In this letter, a deep neural network (DNN) aided joint source-channel (JSCC) decoding scheme is proposed for polar codes. In the proposed scheme, an integrated factor graph with an unfolded structure is first designed. Then a DNN aided flooding belief propagation decoding (FBP) algorithm is proposed based on the integrated factor, in which both source and channel scaling parameters in the BP decoding are optimized for better performance. Experimental results show that, with the proposed DNN aided FBP decoder, the polar coded JSCC scheme can have about 2–2.5 dB gain over different source statistics  $p$  with source message length  $N_{SC} = 128$  and 0.2–1 dB gain over different source statistics  $p$  with source message length  $N_{SC} = 512$  over the polar coded JSCC system with existing BP decoder.

**key words:** belief propagation (BP) decoding, deep neural network (DNN), polar codes

## 1. Introduction

The source-channel separation theorem indicates that source coding and channel coding can be optimized individually without the performance loss of the overall system. However, the performance of the separated source-channel coding degrades under limited delay and complexity in practical systems. To solve this, the joint source-channel coding (JSCC) is proposed in [1], which considers source decoding and

channel decoding integrally. It turns out that JSCC has better performance than separation-based coding schemes with the presence of delay and complexity limits.

Polar codes, proposed in [2] by Arikan, are the first coding scheme that can achieve the Shannon capacity with linear complexity. Polar codes have been widely studied in channel coding and it turns out that polar codes can have outperforming error-correcting performance than low-density parity-check (LDPC) codes at short code-lengths [3], [4]. Belief propagation (BP) decoding is a basic algorithm for polar codes with advantages of low latency and high throughput due to its inherent high parallelism. The min-sum approximation [5] method and an early stopping scheme [6] are proposed for BP decoding to get efficient hardware design and lower complexity. Besides, a bit mapping aided BP decoder is researched for concatenated polar codes in [7] and a deep neural network decoder with optimized BP decoder is presented in [8] to obtain faster convergence. Deep learning methods are also used in concatenated polar codes with BP decoding for better performance [9] and ResNet-like BP architecture is proposed for polar codes in [10] to improve standard BP decoding.

Apart from channel coding, polar codes also have been researched in JSCC field. In [11], a new language-based list decoder is proposed to exploit the language source redundancy for improved error-correcting performance. In [12], systematic polar codes are used for JSCC with correlated sources and a novel distributed joint source-channel list decoding is proposed which exploits sources correlation to get better performance. Then turbo-like BP (TL-BP) and joint successive cancellation list (J-SCL) decoders are proposed for the double-polar JSCC scheme in [13] and [14], respectively. TL-BP decoder is more favorable for low-latency applications because of the parallel decoding process. However, the performance of the TL-BP decoding is not optimal because of the fixed scaling parameter. Besides, the source and channel factor graphs in [13] are separately represented, it's not beneficial to use deep learning network [15] to optimize the integrate system model.

In this letter, we design the integrated factor graph and combined it with DNN structure to optimize scaling parameters for a better performance. We first design an integrated factor for the polar coded JSCC system. Based on this, a DNN aided flooding belief propagation (DNN-FBP) decoding algorithm is proposed, in which scaling parameters of both channel and source are optimized to improve perfor-

Manuscript received July 25, 2023.

Manuscript publicized August 23, 2023.

<sup>†</sup>The authors are with School of Electrical Engineer and Information, Southwest Petroleum University, Chengdu, 610500, China.

<sup>††</sup>The author is with the National Key Lab of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China.

<sup>†††</sup>The author is with School of Physics and Electronic Information Engineering, Henan Polytechnic University, Jiaozuo, 454150, China.

<sup>††††</sup>The author is with the Department of Electrical and Computer Engineering, Laval University, Quebec, QC G1V 0A6, Canada.

\*This work was supported by National Nature Science Foundation of China under Grant 62371101 and 62161047, National Key Research and Development Program under Grant 2022YFC3005702, the Natural Science Foundation of Sichuan Province under Grant 2022NSFSC0488, the scientific research starting project of SWPU (No.2021QHZ037) and Henan Province Key R&D and Promotion Special Project under Grant 212102210166 and 232102211073.

a) E-mail: yuqingping@swpu.edu.cn (Corresponding author)

b) E-mail: zhangyou\_swpu@126.com

c) E-mail: szp@uestc.edu.cn

d) E-mail: lixingwangbupt@gmail.com

e) E-mail: utibetwly@qq.com

f) E-mail: ming.zeng@gel.ulaval.ca

DOI: 10.1587/transfun.2023EAL2068

mance. Experimental results exhibit that the proposed DNN-FBP decoder achieves 2–2.5 dB gain over different source statistics  $p$  with source message length  $N_{SC} = 128$  and 0.2–1 dB gain over different source statistics  $p$  with source message length  $N_{SC} = 512$  and  $p=0.02$  of BER  $10^{-4}$  over that of the TL-BP decoder.

## 2. Preliminaries

### 2.1 Polar Codes

Based on the channel polarization theory, the encoding process for a polar code length of  $N$  can be viewed as transmitting source message over  $N$  polarized sub-channels. For a  $(N, K)$  polar code, the bit indexes of source message  $\mathbf{u}_1^N = (u_1, u_2, \dots, u_N)$  can be divided into two subsets: set  $A$  carrying  $K$  information bits over  $K$  reliable sub-channels and set  $A^c$  carrying  $N - K$  fixed bits with known values over remaining  $N - K$  sub-channels. The generation process of polar codewords  $\mathbf{x}_1^N$  can be denoted in form of matrix multiplication:

$$\mathbf{x}_1^N = (x_1, x_2, \dots, x_N) = \mathbf{u}_1^N \mathbf{G}, \quad (1)$$

where the  $N \times N$  generator matrix  $\mathbf{G}$  is obtained by  $\mathbf{G} = \mathbf{F}^{\otimes n}$ ,  $\mathbf{F}^{\otimes n}$  means the  $n$ -th Kronecker power of  $\mathbf{F}$ , with  $\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$  and  $n = \log_2 N$ .

Besides, systematic polar codes have lower bit error rate (BER) than non-systematic polar codes [16]. In the encoding of systematic polar codes,  $K$  information bits are assigned to a subset of codeword as the data carrier  $\mathbf{x}_B = (x_i | i \in B)$ , where  $B$  equals to  $A$ . Therefore, the completed codeword  $\mathbf{x}_1^N$  can be denoted as  $(\mathbf{x}_B, \mathbf{x}_{B^c})$ , where  $\mathbf{x}_{B^c} = (x_i | i \in B^c)$  can be calculated by:

$$\mathbf{x}_{B^c} = \mathbf{u}_A \mathbf{G}_{AB^c} + \mathbf{u}_{A^c} \mathbf{G}_{A^c B^c}, \quad (2)$$

where  $\mathbf{u}_A = (\mathbf{x}_B - \mathbf{u}_{A^c} \mathbf{G}_{A^c B}) (\mathbf{G}_{AB})^{-1}$ ,  $\mathbf{G}_{AB}$  denotes the submatrix of  $\mathbf{G}$  consisting of the array of elements  $(\mathbf{G}_{i,j})$ , with  $i \in A$  and  $j \in B$ . For details on the systematic polar code, please refer to [16].

### 2.2 Belief Propagation (BP) Decoding

BP is an important parallel decoding algorithm for polar codes, in which soft messages are transmitted iteratively based on the factor graph. For polar codes of rate  $R_{PC} = \frac{K}{N}$ , its factor graph consists of  $n$  stages and  $N * (n + 1)$  nodes, in which each node involves a right-to-left message  $L$  and a left-to-right message  $R$ . Specifically,  $L$  and  $R$  are defined as  $L_{i,j}^{(t)}$  and  $R_{i,j}^{(t)}$ , where  $i, j$  and  $t$  denote the stage index, the node index and the  $t$ -th iteration, respectively. During the BP decoding,  $L$  and  $R$  messages are propagated iteratively between adjacent nodes by

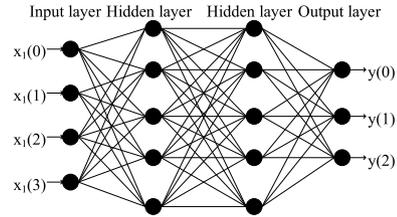


Fig. 1 A 4-layer deep neural network structure.

$$\begin{cases} R_{i+1,2j-1}^{(t)} = g(R_{i,j}^{(t)}, L_{i+1,2j}^{(t-1)} + R_{i,j+N/2}^{(t)}), \\ R_{i+1,2j}^{(t)} = g(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t-1)} + R_{i,j+N/2}^{(t)}), \\ L_{i,j}^{(t)} = g(L_{i+1,2j-1}^{(t)}, L_{i+1,2j}^{(t)} + R_{i,j+N/2}^{(t)}), \\ L_{i,j+N/2}^{(t)} = g(L_{i+1,2j-1}^{(t)}, R_{i,j}^{(t)}) + L_{i+1,2j}^{(t)}, \end{cases} \quad (3)$$

where  $g(x, y) = \ln \frac{(1+x)y}{(x+y)}$ . For lower computational complexity, the equation about  $g(x, y)$  can be approximated as  $g(x, y) \approx \alpha \cdot \text{sign}(x)\text{sign}(y)\min(|x|, |y|)$ . After the iteration number of BP decoding achieves the preset maximum number  $T$ , the estimation  $\hat{\mathbf{u}}_1^N$  is then judged by

$$\hat{u}_j = \begin{cases} 0, & \text{if } L_{1,j}^T + R_{1,j}^T \geq 0, \\ 1, & \text{if } L_{1,j}^T + R_{1,j}^T < 0, \end{cases} \quad (4)$$

where  $1 \leq j \leq N$ .

### 2.3 Deep Neural Network (DNN)

Figure 1 shows a 4-layer DNN, in which the feed-forward network structure can be described as a function that maps the input  $\mathbf{x}_0 \in \mathbb{R}^4$  to the output  $\mathbf{y} \in \mathbb{R}^3$ . In the  $l$ -th layer, the output  $\mathbf{x}_l$  can be obtained by the  $l$ -th layer mapping function with the last layer output  $\mathbf{x}_{l-1}$ . In all, the above mapping functions are described as

$$\begin{cases} \mathbf{y} = f(\mathbf{x}_0; \theta), \\ \mathbf{x}_l = f^{(l)}(\mathbf{x}_{l-1}; \theta_l), \quad l = 1, 2, \dots, L, \end{cases} \quad (5)$$

where  $\theta$  refers to the parameters of the system and  $\theta_l$  refers to the parameters of the  $l$ th layer. These parameters can be trained to approximate the mapping from input  $\mathbf{x}_0$  to output  $\mathbf{y}$ . To describe the non-linear relations between input and output, two common activation functions, rectified linear unit (ReLU) function and sigmoid function, are given as

$$\begin{cases} g_{\text{ReLU}}(s) = \max(0, s), \\ g_{\text{sigmoid}}(s) = \frac{1}{1 + e^{-s}}. \end{cases} \quad (6)$$

After constructing the DNN model, a loss function should be defined to evaluate the performance of the system. Mean square error (MSE) and cross entropy (CE) are defined as two common loss functions, as

$$\begin{cases} L_{\text{MSE}}(\mathbf{w}, \mathbf{o}) = \frac{1}{N} \sum_{i=1}^N (w_i - o_i)^2, \\ L_{\text{CE}}(\mathbf{w}, \mathbf{o}) = -\frac{1}{N} \sum_{i=1}^N w_i \log(o_i) + (1 - w_i) \log(1 - o_i), \end{cases} \quad (7)$$

where  $\mathbf{w}$  and  $\mathbf{o}$  are the label vector and prediction vector with length  $N$ . Optimizers in deep learning libraries can train the parameters efficiently by minimizing the loss function in the procedure of back-propagation algorithm. In order to train the DNN, we need to collect sufficient known input-output mappings for the training set. Fortunately, it is convenient to generate sufficient input-output mappings by simulation.

### 3. Proposed DNN-BP Decoder

#### 3.1 System Model

Figure 2 shows the system model of this polar coded JSCC system, in which a non-systematic polar encoder and a systematic polar encoder are involved for source and channel, respectively. At the transmitter, a binary independent identically distributed Bernoulli source with probability of success  $p < 0.5$  is considered. The source message  $\mathbf{u} = (u_1, u_2, \dots, u_{N_{SC}})$  is mapped to the source compressed vector  $\mathbf{v}_H = (v_1, v_2, \dots, v_{|H|})$  by the source encoder, where  $N_{SC} = 2^{n_{SC}}$  denotes the length of source message,  $H$  denotes indexes of high-entropy bits in the source codeword  $\mathbf{v}_1^{N_{SC}}$ , with  $|H| \leq N_{SC}$ . Then,  $\mathbf{v}_H$  is encoded by the systematic polar encoder and outputs codewords  $\mathbf{x} = (x_1, x_2, \dots, x_{N_{CC}})$ . So the rates of source code and channel code are  $R_{SC} = \frac{|H|}{N_{SC}}$  and  $R_{CC} = \frac{|H|}{N_{CC}}$ , respectively. Finally,  $\mathbf{x}$  is modulated and transmitted over the channel. At the receiver, a BP decoding is iteratively carried out for channel decoder and source decoder over the factor graph based on the received sequence  $\mathbf{y}$ .

#### 3.2 Designed Integrated Factor Graph for JSCC

Firstly, we design an integrated factor graph for the polar

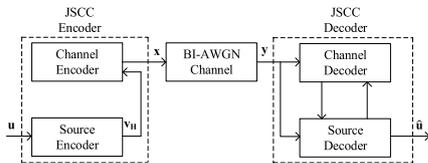


Fig. 2 The structure of JSCC system.

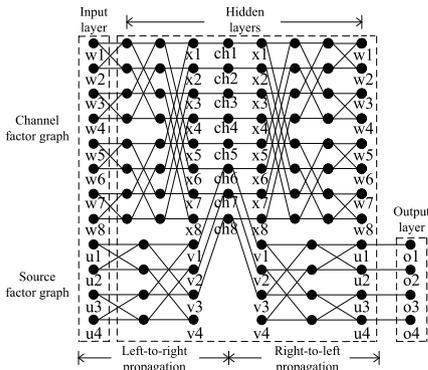


Fig. 3 The unfolded structure of the integrated factor graph.

coded JSCC system through connecting the equivalent variable nodes of channel factor graph and source factor graph. Figure 3 shows an example of the unfolded structure of integrated factor graph with  $N_{CC} = 8, N_{SC} = 4$  and  $|H| = 3$ . Besides,  $B = \{6, 7, 8\}$  and  $H = \{1, 2, 3\}$  for the JSCC system in Fig. 3. Note that,  $ch_1 \sim ch_8$  denote the nodes of the layer of received information and  $L_{ch}$  are the log-likelihood ratio (LLR) messages over these nodes. Based on the integrated factor graph, the flooding BP decoding is carried out iteratively as presented in Eq. (8) and Algorithm 1:

#### Algorithm 1 Proposed DNN-FBP Decoding.

- 1: **Input** : The received vector  $y_1^{N_{CC}}$ , the well-trained scaling parameters  $\alpha$  and  $\beta$ . The variance of i.i.d Gaussian noise in BI-AWGN channel  $\sigma^2$ . A preset maximum iteration number  $T$ .
- 2: **Output** : The decoded source sequence  $\hat{u}_1^{N_{SC}}$
- 3: //Initialization
- 4:  $R_{1,j}^{(0)} \leftarrow 0$  for  $j \in A$ ;  $R_{1,j}^{(0)} \leftarrow +\infty$  for  $j \in A^c$
- 5:  $r_{1,j}^{(0)} \leftarrow \ln \frac{1-p}{p}$ ;  $L_{ch(j)} \leftarrow \frac{-2y_j}{\sigma^2}$
- 6: **for**  $t=1:T$
- 7: Update  $R_{i,j}^{(t)}$  and  $r_{i,j}^{(t)}$  based on Eq. (8).
- 8:  $L_{m+1,j}^{(t)} \leftarrow L_{ch(j)} + r_{n+1,k}^{(t)}$  for  $j \in B, k \in H$
- 9:  $L_{m+1,j}^{(t)} \leftarrow L_{ch(j)}$  for  $j \in B^c$
- 10:  $l_{m+1,j}^{(t)} \leftarrow L_{ch(j)} + R_{m+1,k}^{(t)}$  for  $j \in H, k \in B$
- 11:  $l_{n+1,j}^{(t)} \leftarrow 0$  for  $j \in H^c$
- 12: Update  $L_{i,j}^{(t)}$  and  $l_{i,j}^{(t)}$  based on Eq. (8).
- 13: Make hard decision on  $\hat{\mathbf{u}}$  and  $\hat{\mathbf{v}}$ . If early stopping criterion  $\hat{\mathbf{u}}\mathbf{F}^{\otimes n} = \hat{\mathbf{v}}$  holds, the loop structure breaks.
- 14: **end for**
- 15: **return**  $\hat{\mathbf{u}}$

$$\left\{ \begin{array}{l} R_{i+1,2j-1}^{(t)} = \alpha \cdot g'(R_{i,j}^{(t)}, L_{i+1,2j}^{(t-1)} + R_{i,j+N_{CC}/2}^{(t)}), \\ R_{i+1,2j}^{(t)} = \alpha \cdot g'(R_{i,j}^{(t)}, L_{i+1,2j-1}^{(t-1)} + R_{i,j+N_{CC}/2}^{(t)}), \\ r_{i+1,2j-1}^{(t)} = \beta \cdot g'(r_{i,j}^{(t)}, l_{i+1,2j}^{(t-1)} + r_{i,j+N_{SC}/2}^{(t)}), \\ r_{i+1,2j}^{(t)} = \beta \cdot g'(r_{i,j}^{(t)}, l_{i+1,2j-1}^{(t-1)} + r_{i,j+N_{SC}/2}^{(t)}), \\ L_{i,j}^{(t)} = \alpha \cdot g'(L_{i+1,2j-1}^{(t)}, L_{i+1,2j}^{(t)} + R_{i,j+N_{CC}/2}^{(t)}), \\ L_{i,j+N_{CC}/2}^{(t)} = \alpha \cdot g'(L_{i+1,2j-1}^{(t)}, R_{i,j}^{(t)} + L_{i+1,2j}^{(t)}), \\ l_{i,j}^{(t)} = \beta \cdot g'(l_{i+1,2j-1}^{(t)}, l_{i+1,2j}^{(t)} + r_{i,j+N_{SC}/2}^{(t)}), \\ l_{i,j+N_{SC}/2}^{(t)} = \beta \cdot g'(l_{i+1,2j-1}^{(t)}, r_{i,j}^{(t)} + l_{i+1,2j}^{(t)}), \end{array} \right. \quad (8)$$

Note that, the scaling parameters  $\alpha$  and  $\beta$  in decoding algorithm have been well trained through deep learning training procession based on activation function Eq. (9) and loss function Eq. (10). In Eq. (8),  $g'(x, y) = \text{sign}(x)\text{sign}(y)\min(|x|, |y|)$ , the uppercase symbols  $R$  and  $L$  represent the LLR messages of channel factor graph, the lowercase symbols  $r$  and  $l$  represent the LLR messages of source factor graph,  $i$  denotes the stage index,  $j$  denotes the node index, and  $t$  denotes the  $t$ -th iteration.

The decoder needs to be initiated before decoding and the initialization rules are presented in lines 4-5 of Algorithm 1. Since the integrated factor graph contains both the

channel factor graph and source factor graph, the LLR messages in two directions of channel and source factor graphs need to be both updated based on Eq. (8) (line 7 and line 12, Algorithm 1). For the correlation between the source and channel factor graphs,  $\mathbf{L}_{ch}$  and two subsets of left-to-right messages from channel and source factor graphs are required to update the rightmost right-to-left messages of the source and channel factor graphs in the middle of each iteration (lines 8-11, Algorithm 1). Finally, the early stopping criterion  $\hat{\mathbf{u}}\mathbf{F}^{\otimes n} == \hat{\mathbf{v}}$  is applied. When the early stopping criterion is satisfied or  $t$  reaches the maximum iteration number  $T$ , the decoding ends and the estimation  $\hat{\mathbf{u}}$  is outputted as ultimate decoding result.

### 3.3 Proposed DNN-FBP Decoder for JSCC with Polar Codes

The unfolded structure of integrated factor graph can be well matched to a DNN because it has a similar structure with DNN, which contains an input layer, an output layer and several hidden layers, with many nodes in each hidden layer. Based on the similar structure, activation functions Eq. (6) and loss functions Eq. (7) in DNN can be used to optimize the scaling parameters ( $\alpha$  and  $\beta$  in Eq. (8)) for better performance. Specifically, the sigmoid function is selected as the activation function to rescale the negative of soft-decision of  $\mathbf{u}$  at the last iteration  $\mathbf{s}$  into the range  $[0, 1]$  as output  $\mathbf{o}$  by

$$\mathbf{o} = \frac{1}{1 + e^{-\mathbf{s}}}. \quad (9)$$

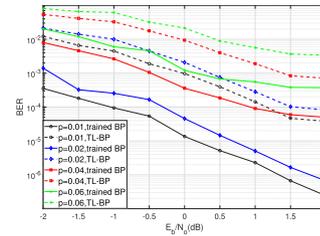
The cross entropy function is defined as the loss function to evaluate the decoding performance in which the source message  $\mathbf{u}$  is the label vector, as

$$L(\mathbf{u}, \mathbf{o}) = -\frac{1}{N_{SC}} \sum_{i=1}^{N_{SC}} u_i \log(o_i) + (1 - u_i) \log(1 - o_i). \quad (10)$$

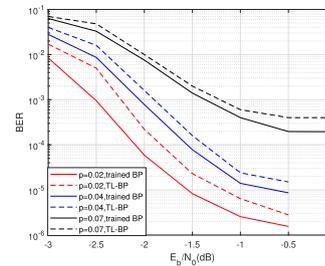
By minimizing the loss function, the trainable parameters  $\alpha$  and  $\beta$  are optimized in the back-propagation algorithm. We adopt the mini-batch stochastic gradient descent (SGD) method with  $M$  codewords in each batch to boost the training. The learning rate is  $\eta$ , which determines the step size at each iteration to minimize the loss function. Meanwhile, the adaptive moment estimation (Adam) method is applied to tune the step size during training. Finally, flooding BP decoding is carried out by supposing that the scaling parameters  $\alpha$  and  $\beta$  have been well trained, as described above. The proposed DNN-FBP decoding is presented in Algorithm 1.

## 4. Simulation Results

Next, we use the deep learning framework Tensorflow to train the model of the proposed decoder. To train the scaling parameters, we generate 9990 frames training data for each



**Fig. 4** BER of trained BP decoder and TL-BP decoder [13] with  $p = 0.01$ ,  $N_{SC} = 128$ ,  $R_{SC} = 3/5$  and  $R_{CC} = 3/10$ .



**Fig. 5** BER of trained BP decoder and TL-BP decoder [13] with  $p = 0.02$ ,  $N_{SC} = 512$ ,  $R_{SC} = 3/5$  and  $R_{CC} = 3/10$ .

**Table 1** The well-trained alphas and betas for different  $E_b/N_0$  situations with  $p=0.02$ ,  $N_{SC} = 128$ ,  $R_{SC} = 3/5$  and  $R_{CC} = 3/10$ .

$E_b/N_0$ (dB)	-2	-1.5	-1	-0.5	0	0.5	1	1.5	2
$\alpha$	0.9026	0.8822	0.8829	0.8421	0.8974	0.9065	0.9416	0.9806	0.9477
$\beta$	0.9481	0.9493	0.9120	0.9685	0.9374	0.9206	0.9267	0.8949	0.8774

SNR over an additive white Gaussian noise (AWGN) channel and the mini-batch size is 333 with  $M = 30$  codewords in each batch. It should be mentioned that the trainable parameters  $\alpha$  and  $\beta$  are both initialized to 1 and the learning rate  $\eta$  is 0.02.

We compare the BER performance of the trained BP decoder and TL-BP decoder [13] over different source statistics  $p$  with  $N_{SC} = 128$ ,  $R_{SC}=3/5$  and  $R_{CC}=3/10$ . The maximum iteration number of trained BP decoder equals 12,  $T=4$  and  $T_{SC} = T_{CC} = 3$  for TL-BP decoder in Fig. 4. As shown in Fig. 4, the trained BP decoder exhibits a better performance than the TL-BP decoder and the performance gain seems to increase as  $p$  decreases (note that a smaller  $p$  means that the source statistics offer more reliable information). In particular, the proposed trained BP decoder with  $p = 0.01$  can get about 2.15 dB performance gain at the BER of  $10^{-4}$ . Table 1 presents the well-trained alphas and betas for different  $E_b/N_0$  situations with  $p=0.02$ ,  $N_{SC} = 128$ ,  $R_{SC} = 3/5$  and  $R_{CC} = 3/10$ . Due to the randomness of deep neural network training, the well-trained alphas and betas are different for different  $E_b/N_0$  situations.

Moreover, the BER performance of the trained BP decoder and TL-BP decoder [13] over different source statistics  $p$  with  $N_{SC} = 512$ ,  $R_{SC} = 3/5$  and  $R_{CC} = 3/10$  are compared in Fig. 5. The maximum iteration number of trained BP decoder equals 1000,  $T=25$  and  $T_{SC} = T_{CC} = 40$ . It is shown that the trained BP decoder can outperform the TL-BP

decoder about 1 dB, 0.2 dB and 0.25 dB with  $p=0.07$ ,  $p=0.04$  and  $p=0.02$  at the BER of  $10^{-4}$ . Besides, it's observed from Fig. 4 and Fig. 5 that the scheme of source message length  $N_{SC} = 128$  has a better performance gain. We analyze one possible reason is that, polar codes of shorter code length suffer from a severer error propagation issue and it is more effective to alleviate error propagation by optimizing scaling parameters. In all, the proposed trained BP decoder can obtain a significant performance gain because the DNN can specially optimize the scaling parameters for different SNR situations. Thus, we can conclude that the proposed trained BP decoder strikes a better performance than the traditional polar coded JSCC system.

## 5. Conclusion

In this paper, we proposed the DNN-FBP decoder for polar coded JSCC system. In the DNN-FBP decoder, deep learning was used to optimize the scaling parameters of BP decoding for better performance. Simulation results showed that, with the proposed DNN-FBP decoder, the polar coded JSCC scheme could outperform double-polar coded JSCC with the existing BP decoder.

## References

- [1] J.L. Massey, "Joint source and channel coding," *Communications Systems and Random Process Theory*, vol.11, pp.279–293, 1978.
- [2] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol.55, no.7, pp.3051–3073, July 2009.
- [3] J. Piao, K. Niu, J. Dai, and C. Dong, "Approaching the normal approximation of the finite blocklength capacity within 0.025 dB by short polar codes," *IEEE Wireless Commun. Lett.*, vol.9, no.7, pp.1089–1092, July 2020.
- [4] Y. Shen, A. Balatsoukas-Stimming, X. You, C. Zhang, and A.P. Burg, "Dynamic SCL decoder with path-flipping for 5G polar codes," *IEEE Wireless Commun. Lett.*, vol.11, no.2, pp.391–395, Nov. 2022.
- [5] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," *Proc. 8th International Symposium on Wireless Communication Systems*, pp.437–441, Nov. 2011.
- [6] B. Yuan and K.K. Parhi, "Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders," *IEEE Trans. Signal Process.*, vol.62, no.24, pp.6496–6506, May 2014.
- [7] Q.P. Yu, Z.P. Shi, L. Deng, and X. Li, "An improved belief propagation decoding of concatenated polar codes with bit mapping," *IEEE Commun. Lett.*, vol.22, no.6, pp.1160–1163, June 2018.
- [8] W. Xu, Z. Wu, Y.L. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," *Proc. IEEE International Workshop on Signal Processing Systems (SiPS)*, pp.1–6, Nov. 2017.
- [9] W. Xu, X. Tan, Y. Be'ery, Y.-L. Ueng, Y. Huang, X. You, and C. Zhang, "Deep learning-aided belief propagation decoder for polar codes," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol.10, no.2, pp.189–203, June 2020.
- [10] J. Gao, D. Zhang, J. Dai, K. Niu, and C. Dong, "Resnet-like belief-propagation decoding for polar codes," *IEEE Wireless Commun. Lett.*, vol.10, no.5, pp.934–937, May 2021.
- [11] Y. Wang, M. Qin, K.R. Narayanan, A. Jiang, and Z. Bandic, "Joint source-channel decoding of polar codes for language-based sources," *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp.1–6, Dec. 2016.
- [12] L. Jin, P. Yang, and H. Yang, "Distributed joint source-channel decoding using systematic polar codes," *IEEE Commun. Lett.*, vol.22, no.1, pp.49–52, Jan. 2018.
- [13] Y. Dong, K. Niu, J. Dai, S. Wang, and Y. Yuan, "Joint source and channel coding using double polar codes," *IEEE Commun. Lett.*, vol.25, no.9, pp.2810–2814, Sept. 2021.
- [14] Y. Dong, K. Niu, J. Dai, S. Wang, and Y. Yuan, "Joint successive cancellation list decoding for the double polar codes," *IEEE Commun. Lett.*, vol.26, no.8, pp.1715–1719, Aug. 2022.
- [15] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G.V. Drissi- sche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of GO with deep neural networks and tree search," *Nature*, vol.529, pp.484–489, 2016.
- [16] E. Arıkan, "Systematic polar coding," *IEEE Commun. Lett.*, vol.15, no.8, pp.860–862, Aug. 2011.