Deep-Learning-Aided Fast Successive Cancellation Decoding of Polar Codes

Haogang Feng, Haiyu Xiao, Shida Zhong, Zhuqing Gao, Tao Yuan, and Zhi Quan

***Abstract*—With the continuous evolution of 5G communication technology to B5G and the next generation of communication technology, Deep Learning technology will also lead the automation and intelligent transformation of communication systems. Existing research has shown that the combination of deep learning and communication technology is expected to break some performance bottlenecks of traditional communication algorithms and solutions. This paper explores the application of deep learning (DL) in polar decoding algorithms, proposing a DL-aided-FSC (DL-FSC) polar code decoder algorithm. For the DL-FSC decoding algorithm, the conventional successive cancellation (SC) decoder is partitioned into multiple sub-blocks, which are replaced by R0 nodes, R1 nodes and sub-DL decoder. The log-likelihood ratio (LLR) and frozen bit pattern are input to the sub-DL decoder to predict decode codewords under any decoding code rate. Through simulation verification, under the PBCH channel of 5G communication, the DL-FSC decoder achieves similar block error rate (BLER) performance to the SC decoder, even improving by about 1%. In order to verify the performance optimization effect of the proposed algorithm at the hardware level, the DL-FSC deocder circuit design was completed. Through FPGA synthesis, the proposed decoder achieves a throughput of about 4571 Mbps, which is 1.71× improvement in decoding throughput at the expense of increased logic resources.**

***Index Terms*—5G, deep learning, fast successive-cancellation decoding, list decoding, polar codes.**

I. Introduction

POLAR code, proposed by Erdal Arikan [1], is a channel coding algorithm that can be rigorously proven to reach channel capacity. In recent years, due to its deterministic construction method and being the only known channel coding method that can be strictly proven to reach channel capacity, it has received widespread attention. During the 3GPP RAN #87 meeting, polar codes were adopted for the control channel of the enhanced mobile broadband (eMBB) service category in 5th generation (5G) wireless communication systems [2]. The proposed polar coded NOMA (PC-NOMA) scheme can significantly improve the capacity of access users with low-complexity multi-user detection algorithms [3], and polar codes can also meet the large-capacity access requirements of 6G [4]. Successive cancellation (SC) and belief propagation (BP) are two traditional methods used in polar code decoding. When the code length in a binary memoryless channel is long enough, the Shannon capacity can be achieved using the SC decoding algorithm [5]. However, the serial nature of the SC decoding algorithm imposes data dependencies, resulting in a high decoding latency and low throughput [6]. Compared with their SC counterparts, polar BP decoders are more attractive for low-latency applications. However, due to their iterative nature, the required latency and energy dissipation of BP decoders increase linearly with the number of iterations [7].

In recent years, deep learning (DL) [8], also known as deep neural networks, has received widespread attention for its ability to solve complex tasks. Recently, researchers have attempted to apply deep learning techniques to channel coding problems [9]–[11]. This is because the deep learning network can complete any mapping from one vector space to another through learning, and it has the property of one-time decoding.

In polar codes with shorter code lengths, feed-forward neural networks are used for polar code decoding for the first time, where log-likelihood ratios (LLRs) serve as inputs and estimated positions serve as the outputs of the neural network [12]. Based on this, a joint learning system architecture consisting of a residual learning denoiser (RLD) and a neural network decoder (NND) is proposed, which uses the multi-task learning (MTL) strategy to jointly optimize the denoising loss function and decoding loss function of residual neural network decoder (RNND), resulting in better denoising and decoding performance [13].

However, neural network decoders for long polar codes encounter significant training challenges due to the high-dimensional space involved, with complexity exponentially increasing with the number of information bits. To address this issue, the integration of neural networks with traditional decoding algorithms, particularly through the substitution of certain decoding components, has been extensively explored. Within the conventional BP decoding framework, specific sub-blocks of the BP decoder have been replaced with BP neural network decoding (BP-NND) sub-blocks [14], thereby enhancing decoding performance. Similarly, a ResNet-like belief propagation structure has been employed to improve the effectiveness of traditional polar BP decoding. The proposed BP decoder with a ResNet-like architecture has similar block

Manuscript received July 3, 2024; revised September 19, 2024; approved for publication by Tarable, Alberto Division 1 Editor, November 15, 2024.

This work was supported in part by the National Key Research and Development Program, China, under Project No.2023YFB4403805 and No.2021YFB3302001, in part by the (Key Area) Project of Department of Education of Guangdong Province, China, under Project No.2021ZDZX4008.

H. Feng, H. Xiao (equal contribution), S. Zhong, T. Yuan, and Z. Quan are with the State Key Laboratory of Radio Ereguency Heterogeneous Integration, Shenzhen University, Shenzhen, 518060, Asia, China, email: xiaohaiyu2021@email.szu.edu.cn, fenghaogang@email.szu.edu.cn, shida.zhong@szu.edu.cn, yuantao@szu.edu.cn, zquan@szu.edu.cn.

Z. Gao is with the Beijing Xiaomi Mobile Sofeware Co., Ltd. Xiaomi Campus, No. 33 Xi erqi Middle Road, Haidian District, Beijing, 100085, China, email: gaozhuqing@xiaomi.com.

S. Zhong is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2024.000070

1229-2370/24/$10.00 © 2024 KICS

error rate (BLER) performance to the standard BP decoder, but with fewer iterations [15]. The neural successive cancellation (NSC) decoder is another solution that connects multiple neural network decoders through SC decoding [16]. The proposed NSC sub-block $N = 2$ SC decoding sub-block can effectively reduce the decoding time step, but the input-output data dimension is too small to limit the role of neural networks. A sub-NN decoder with tanh-based modified LLR is used to replace the $N = 4$ SC sub-block to reduce the decoding delay of polar codes on FSO turbulence channels [17]. However, these decoding algorithms that utilize DL primarily rely on polar codes with fixed code length and fixed code rate. By exploring different NNN recognition strategies, [18] introduces the last subcode NN-assisted decoding (LSNNAD) and the key-bit-based subcode NN-assisted decoding (KSNNAD) schemes, which can effectively handle Polar codes with long code lengths, although there is no simulation test under 5G channel in this work. Moreover, the BLER performance of the proposed algorithms has not been compared with that of the successive cancellation list (SCL) decoding algorithm.

In this paper, we propose a practical deep-learning-aided fast successive cancellation (DL-FSC) decoding algorithm. The DL-FSC decoding algorithm uses R0 nodes, R1 nodes and general $N = 8$ sub-DL decoders to replace the $N = 8$ sub-blocks in the traditional SC decoder. The sub-DL decoder can predict the probability of decoding codewords through a deep learning network. Among them, the calculation of R0 and R1 nodes relies on the traditional FSC decoding algorithm to achieve fast decoding. More specifically, the input to the sub-DL decoder consists of two-dimensional data, which includes 8 LLRs and the corresponding frozen bit pattern, allowing the decoding of sub-blocks with varied frozen bit information and code rates. Integrating deep learning techniques with traditional decoding methods not only enhances the performance of polar codes but also aligns with the evolving trend of incorporating deep learning into future communication systems.

Simulations under 5G channel conditions have demonstrated that the DL-FSC decoder achieves a BLER performance comparable to the traditional SC decoder. The results indicate that a well-trained sub-DL decoder enables the DL-FSC decoding algorithms to meet the performance standards of 5G. Additionally, the recursive nature of our scheme allows for the reuse of the DL-FSC across different parts of the decoding process. To assess the performance improvements at the hardware level, a hardware circuit design for the DL-FSC decoder was completed. Despite consuming more logical resources, a literature review reveals that the DL-FSC decoder's throughput has significantly increased. Moreover, the DL-FSC algorithms are adaptable to various channels within 5G and prospective 6G technologies, though the integration of deep learning introduces additional computational complexity.

The rest of this paper is organized as follows. Section II briefly introduces polar codes, SC, FSC decoding algorithms and deep learning decoding algorithms. The proposed DL-FSC decoder and hardware implementation will be described in detail in Section III. And the simulation process and results of our DL-FSC decoder in the 5G channel are presented in Section IV. Finally, Section V draws the main conclusions of this paper.

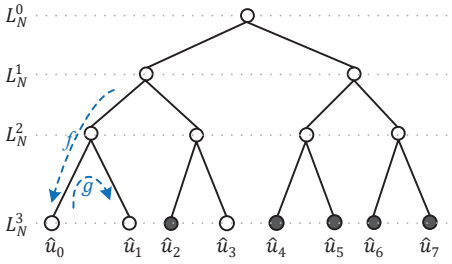


Fig. 1. SC decoding on a binary tree for P(8,5).

II. Preliminaries

A. Polar Codes

Polar code (N, K) of length $N = 2^n$ with K information bits is as $x = uF^{\otimes n}$, where $x = \{x_0, x_1, \cdots, x_{N-1}\}$ is codeword. After determining the code length N, the generator matrix of the polar code is uniquely determined and can be generated by the Arikan core matrix F [19]. $F^{\otimes n}$ denotes the nth Kronecker power of F, which can be recursively obtained from the Arikan core F [20].

Polar code is used as the channel coding scheme for the control channel in the 5G eMBB scenario. The coding schemes for the uplink and downlink control channels are different, and the specific coding scheme is determined according to the different information sequence lengths.

B. SC Decoding

SC decoding is one of the classic decoding algorithms for polar codes. The SC decoding algorithm uses LLR as the decision criterion, makes a hard decision for each bit, and decodes in the order of bit numbers from small to large. Fig. 1 shows a binary tree representation of a polar code P(8,5) and its corresponding SC decoding. For a node of length N, $L_i(0 \le i < N/2)$ represents the ith LLR value, and $B_N = \{b_0, \cdots, b_{N-1}\}$ represents frozen bit pattern.

The LLR L_{i+1} of left-child nodes can be computed as:

$$L_{i+1} = sign(L_i)sign(L_{i+N/2})min\{|L_i|, |L_{i+N/2}|\}. \quad (1)$$

The LLR L_{i+1} of right-child nodes can be computed as:

$$L_{i+1} = (1 - 2x_i)L_i + L_{i+N/2}, \quad (2)$$

whereas the estimated hard values x of the parent node are updated from those of the left and right-child nodes.

$$\hat{x}_i = \begin{cases} x_i, \ i < \frac{N}{2}, \\ x_i \oplus x_{i+N/2}, \ \text{otherwise}, \end{cases} \quad (3)$$

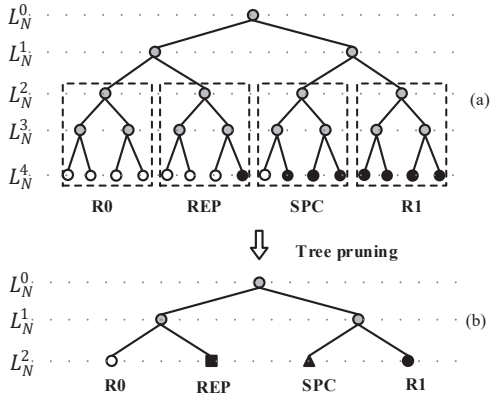


Fig. 2. (a) The full binary tree representation of P(16, 8), and (b) the pruned binary tree representation of the same polar code.

where $x_i = \hat{u}_i$, at the leaf node, $\hat{u}_i$ can be estimated as

$$\hat{u}_i = \begin{cases} 0, \ b_i == 0 \ or \ L_i \geq 0, \\ 1, \ L_i < 0. \end{cases} \tag{4}$$

The latency of SC decoding algorithm can be represented in terms of the number of time steps as

$$\mathcal{T}_{SC} = 2N - 2. \tag{5}$$

C. FSC Decoding

The FSCL algorithm in [21] provides efficient decoders for Rate-0, Rep, SPC, and Rate-1 nodes in SCL without traversing the decoding tree while guaranteeing the error-correction performance preservation. Fig. 2 shows the division of special nodes in P(16,8). The pruned decoding tree of the same polar code is shown in Fig. 2(b) which consists of R0 nodes, Rep nodes, SPC nodes, and Rate-1 nodes. The FSC algorithm in [21] provides efficient decoders in SC without traversing the decoding tree while guaranteeing the error-correction performance preservation. The definitions and decoding operations of each special node under FSCL decoding are given as follows

1) Rate 0: A polar code node of length N where all codewords $u_1, u_2, \cdots u_3$ are frozen bits, with no information bits, is referred to as an R0 Node.
2) Repetition: A polar code node of length N where only the u_N codeword is an information bit, and the rest $u_1, u_2, \cdots, u_{N-1}$ are frozen bits, is referred to as a Rep Node.
3) Single parity check: A polar code node of length N where only the u_1 codeword is a frozen bit, and the rest $u_2, u_3, \cdots, u_N$ are information bits, is referred to as a Rep Node.
4) Rate 1: A polar code node of length N where all codewords $u_1, u_2, \cdots, u_3$ are information bits, with no frozen bits, is referred to as an R1 Node.

D. Deep-learning Decoding

DL [22] is a new research direction in the field of ML. Generally speaking, by integrating more processing layers in a neural network, we are able to describe much more complicated algorithms with improved performance via deep learning. The fully connected neural network (FCNN) [23] is a deep neural network model based on multi-layer non-linear transformations. The input layer has N inputs and the output layer has K outputs. For each hidden layer i, n_i inputs and m_i outputs perform the mapper $f(i)$: $\mathbb{R}^{n_i} \rightarrow \mathbb{R}^{m_i}$, and it is composed of multiple neurons. In these neurons all of its weighted inputs are added up, a bias is optionally added, and the result is propagated through a nonlinear activation function. e.g. a sigmoid function or a rectified linear unit (ReLU), which are respectively defined as

$$sigmoid(z) = \frac{1}{1 + e^{-z}}, \quad relu(z) = max\{0, z\}. \tag{6}$$

Therefore, the input-output mapping of the whole DL decoder can be represented as a chain of functions, which is given by

$$w = f(v, \theta) = out(f^{(L-1)}(\cdots(f^{(0)}(v))), \tag{7}$$

where L gives the number of layers and is also called depth. It was shown in [23] that such a DL decoder and nonlinear activation functions can theoretically approximate any continuous function on a bounded region arbitrarily closely—if the number of neurons is large enough.

III. DL-FSC Decoding

A. DL-FSC Decoding Algorithm

In this paper, leveraging the design concepts of deep learning, we propose a deep learning-aided fast successive cancellation decoder. In the DL-FSC decoding scheme, the R0 node, the R1 node and the sub-DL decoder is used to replace the sub-block in the traditional SC decoding. The R0 node consists only of pure frozen bits, and decoding does not require any computational work, and a node length Nv vector of 0 is output as the decoded results. For Rate-1 node decoding, since there is no frozen bit, a hard decision (using (4)) can be made directly through the LLR of the top layer of the node to obtain X_{N_v}. Then multiply it by the corresponding polar transformation matrix $F^{\otimes s}$ to output the decoded data U_{N_v} of the corresponding node. The sub-DL decoder receives 8 internal LLRs and corresponding frozen bit pattern and predicts 8 output bits by DL network. The difference between DL-FSC decoder and SC decoding is that DL-FSC does not need to traverse all decoding trees and has a similar error correction performance.

The Fig. 4 shows a system overview of sub-DL decoder architecture. The LLRs and frozen bit pattern are input to the sub-DL decoder to predict decode codewords. Before inputting the LLR into the sub-DL decoder, a sigmoid-like function

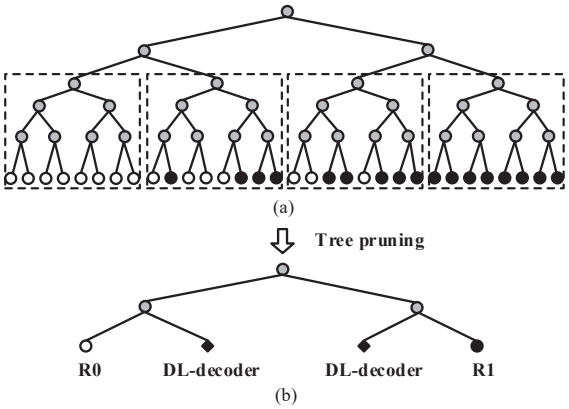


Fig. 3. (a) The division of sub-DL decoder in P(32,17), and (b) the pruned binary tree representation by the sub-DL decoder.

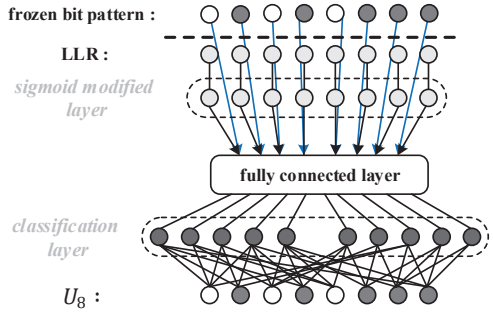


Fig. 4. A system overview of sub-DL decoder architecture.

Algorithm 1: DL-FSC decoding algorithm

Input: $L_N = \{l_0, \cdots, l_{N-1}\},\ B_N = \{b_0, \cdots, b_N\}$
Output: $\hat{U}_N = \{\hat{u}_0, \cdots, \hat{u}_N\}$
for $i = 0$ *to* $\frac{N}{8} - 1$ **do**
  / Calculate 8 LLRs using Eq1, Eq2 and Eq3 */*
  $L_8 \leftarrow SC(L_N, B_N, \hat{U}_N)$;
  / R0 node - decoding */*
  if $B_8 == 8\{0\}$ **then**
    $\hat{U}_8 = 8\{0\}$;
  / R1 node - decoding */*
  else if $B_8 == 8\{1\}$ **then**
    / The hard decision using Eq4 */*
    $\hat{X}_8 = hard\_decision\{L_8\}$;
    $\hat{U}_8 = \hat{X}_8 F^{\otimes log_2 8}$;
  / sub-DL decoding */*
  else
    / Normalized LLRs using Eq8, the scale parameters 's' have been trained */*
    V_8 = sigmoid(L_8, s);
    / Predict the probability of each codeword by DL */*
    $\hat{U}_8[256] \leftarrow \{V_8, B_8\}$;
    / Select the maximum probabiliy codeword */*
    $\hat{U}_8 = max(\hat{U}_8[256])$;
  end
  / Combine the decoding results of each sub-block */*
  $\hat{U}_N \leftarrow \hat{U}_8$;
end

is used in the sigmoid-modified layer to normalize LLR, as follows:

$$v = sigmoid(llr, s) = \frac{1}{1 + e^{-s*llr}}, \tag{8}$$

where s represents the scale parameter and v is the modified LLR, respectively. The range of v is limited to (0,1) by the sigmoid-modified layer. When LLR is close to zero, there is a greater unreliability for the channel transmission signal. Therefore, it is necessary to adjust the scale parameter s so that the sigmoid-like function has a higher resolution at the zero point, and the modified LLR v retains more information metrics.

The modified LLR is input into the neural network decoder. The fully connected neural network is adopted as the neural network decoder, which is composed of an input layer, a sigmoid-modified layer, a fully connected layer (weights: 256×16, bias: 256×1), and a classification layer. Therefore, the DL-FSC decoder can be seen as a mapper $f\{R(llr, frozen\ bit\ pattern) \rightarrow R(\hat{u})\}$. Algorithm 1 shows the decoding process of polar code using DL-FSC. The channel LLRs, the frozen bit pattern and the estimated codewords are denoted as $L_N = \{l_0, \cdots, l_{N-1}\},\ B_N = \{b_0, \cdots, b_{N-1}\}$ and $\hat{U}_N = \{\hat{u}_0, \cdots, \hat{u}_{N-1}\}$. Unlike other FSC decoders that need to be manually designed to decode special constituent codes [24], the DL-FSC decoder in this paper is trained to decode any node without considering any specific frozen bit pattern.

B. Training of the DL-FSC

As described above, the DL-FSC decoder for a certain channel polarization code is universal. Therefore, we only need to train one DL-FSC decoder. In this paper, we use the polarization code scheme of the PBCH channel under the 5G standard as shown to collect training data and verify the scheme. The sub-DL decoder is trained using gradient descent optimization method and backpropagation algorithm [25]. In order for the DL-FSC decoder to understand the LLR characteristics under the PBCH channel, the scale parameter s in the sigmoid-modified layer $\{v = sigmoid(llr, s)\}$ added by the DL-FSC will also participate in the training. The parameter s is typically set around 0.25, as determined through training.

The process of collecting training data for DL-FSC decoder is summarized in Algorithm 2. The training data of the DL-FSC decoder is collected by assuming that the SC decoder has perfect knowledge of the transmitted bits. Under the condition that all decodings are correct, compute the LLRs of

Algorithm 2: Collect training data for DL-FSC

Input: $L_N = \{l_0, \cdots, l_{N-1}\}$,
$B_N = \{b_0, \cdots, b_{N-1}\}$,
$U_N = \{u_0, \cdots, u_{N-1}\}$
Output: L_S, B_S, U_S

Initialization
for *Different SNR* **do**
  for $i = 0$ *to* $\frac{N}{8} - 1$ **do**
    for $j = 0$ *to* 8 **do**
      /* *Calculate LLRs using (1), (2), and (3)* */
      $L_{8i+j} \leftarrow SC(L_N, B_N, U_N)$;
      /* *Use the correct codewords for SC iteration* */
      $\hat{u}_{8i+j} = u_{8i+j}$;
    end
    if $\{b_{8i}, \cdots, b_{8i+7}\} \neq 8\{0\}$ **then**
      /* *Store the training data for DL-FSC* */
      Store $B_8 = \{b_{8i}, b_{8i+1}, \cdots, b_{8i+7}\}$;
      Store $U_8 = \{u_{8i}, u_{8i+1}, \cdots, u_{8i+7}\}$;
      /* *Store the n-3 stage LLRs* */
      Store $L_8 = \{l_{8i}, l_{8i+1}, \cdots, l_{8i+7}\}$;
    end
  end
end

the partitioned sub-blocks in the SC decoder. Finally, collect the LLR L_S, frozen bit pattern B_S, and correct codeword U_S for each sub-block with a non-zero code rate.

For instance, in the PBCH channel, the core block of the polar code is (512, 56), which can be decomposed into 64 polar code sub-blocks, among which only 16 sub-blocks have a non-zero code rate. The information bit numbers of these 16 sub-blocks are $\{0, 1, 3, 4, 6, 7, 8\}$ respectively. Sub-blocks without information bits (R0) do not need to be decoded, and other information sub-blocks can all serve as the training set for a single DL-FSC decoder. The *deepNetworkDesigner* in Matlab was used to help us quickly establish and train neural networks. *Bayesian optimization* was used for deep learning to find the optimal network hyperparameters and training options.

C. The DL-FSC Decoder Architecture

Although deep learning decoders theoretically exhibit one-shot decoding characteristics, the substantial matrix computations within deep learning networks necessitate a certain clock cycle for decoding predictions during hardware implementation. In this section, for the proposed DL-FSC decoding algorithm, we completed the hardware design of the polar code decoder with $N = 16$ to verify the performance of the proposed decoding algorithm in hardware.

The Fig. 5 shows the hardware structure of the DL-FSC decoder. The input of the decoder is mainly a 16-bit frozen bit pattern and the corresponding LLR value (using 8-bit quantization) to complete the decoding of the 16-bit codeword. The process of the decoder is mainly divided into two parts: the left node and the right node to complete the decoding. The decoding accelerator receives the LLR data and completes the 8 LLR_{left} calculations of the left node through the f operation of (1). Then the LLR_{left} is input into the R0 sub-block, the R1 sub-block and the sub-DL decoder for decoding, and

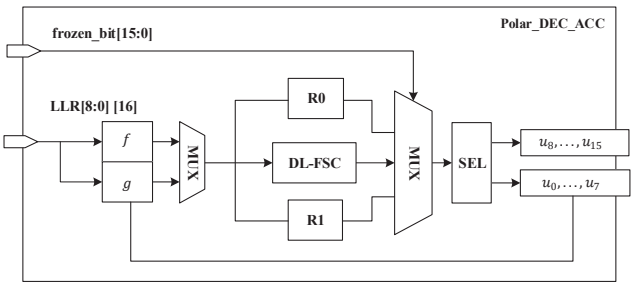


Fig. 5. A system overview of DL-FSC decoder hardware structure.

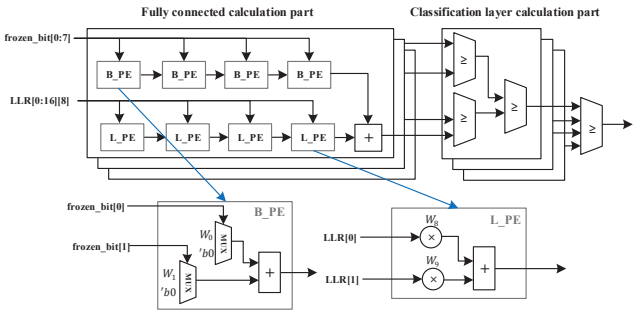


Fig. 6. The hardware structure of DL-FSC decoder core sub-block.

the corresponding result is selected as the decoding codeword U_{left} [7:0] according to the frozen bit pattern. The calculation of the right node is based on the g operation of (2), and the LLR_{right} calculation of the right node is completed, and then the decoding is completed through the sub-block.

In the DL-FSC decoder, the core part is the sub-DL decoder, which mainly completes polar code decoding based on the trained deep learning network. Through training, we determined that its core network is a fully connected network module, and int8 is used as data storage. Therefore, the module design of the sub-DL decoder is mainly divided into two parts:

1) Fully connected calculation part: Complete the calculation of each channel (corresponding to the codeword label) based on the weight and bias data obtained from training. The calculation formula for each channel Y_i is:

$$Y_i = \sum_{j=1}^{8} W_{i,j} \times B_j + \sum_{j=1}^{8} W_{i,8+j} \times L_j + bias_i, \qquad (9)$$

where B represents the forzen bit pattern and L are the LLRs, and $W_{i,j}$ $(i \in [0, 256])$ represents the weight of the ith row and jth column in the weight matrix.

2) Classification layer calculation part: Since sub-DL decoding only needs to output the label with the maximum prediction probability as the decoding codeword, the design in this part only needs to calculate the channel corresponding to the maximum value, and the traditional softmax layer is not required.

The Fig 6 shows the decoding circuit module of the DL-FSC core sub-block, which is mainly divided into the fully connected calculation part and the classification layer cal-

TABLE I
5G POLAR CODE PARAMETERS.

Parameters	**Uplink channel**		**Downlink cannel**	
Channel	PUCCH / PUSCH		PDCCH	PBCH
Data length (A)	$12 \leq A \leq 19$	$20 \leq A \leq 1706$	$12 \leq A \leq 140$	$A = 32$
Max rate Matching length (E)	8192	16384	8192	864
CRC length	6	11	24	
PC bits	3	N/A	N/A	
Encode schemes	PC-CA-Polar	CA-Polar	DCA-Polar	
Minimum code rate	$1/8$			

TABLE II
THE PARAMETERS OF THE POLAR CODE ENCODING SCHEME.

Channel	**Data length (A)**	**CRC length (C)**	**Message length (K)**	**Encoded block length (N)**	**Rate matching length (E)**	**Code block concatenation**
PBCH	32	24	56	512	864	Unsegmented
PUCCH	64	11	75	1024	1728	
	128		139	1024	3456	
	256		267	1024	3456	
	512		523	1024	864	
	1024		523	1024	6912	2-Segm

culation part. The channel accumulation module is mainly divided into two modules to complete the accumulation: In the B_{PE} module, since the frozen bit is a single-bit input, the multiplier can be optimized to a MUX selector to select the weight data or 0 to complete the accumulation. The L_{PE} module completes the accumulation of LLR and weight data through a constant multiplier. The input of the classification layer calculation part is the splicing signal of each channel value and the current channel sequence number, which can complete the size comparison and get the sequence number of the current channel. At the same time, in order to speed up the comparison speed, the three comparators used here are cascaded into a four-selection comparator to complete the comparison of multiple channels.

IV. SIMULATION RESULTS

A. Data Pre-Processing and Model Training

In order to obtain suitable training and validation datasets, we first generate 1 million random codewords and encode them using the corresponding polar code encoding scheme according to different 5G channels. Different noises are superimposed on the encoded data, and then the correctly decoded codewords are collected. The LLR values, frozen bit patterns, and correct codewords of non-R0 and R1 sub-blocks are collected in the SC decoder. These random codewords include sub-block decoding results under different frozen bit patterns and different E_b/N_0. Then 95% of the random codewords are used as training sets, and the remaining 5% are used as validation sets.

As mentioned in Section II-A, the specific coding scheme used in uplink or downlink channel depends on the information length. The specific parameters of the polar code coding schemes for the uplink and downlink channels are shown in Table I.

Due to the different message length (A) and rate matching length (E) in the 5G channel, the corresponding polar code encoding construction method is also different. Therefore, we selected some polarization code encoding schemes of 5G channels for simulation testing. Table II shows the detailed parameters of the polar code encoding scheme used in this paper.

After completing the collection of training dataset, we use the Bayesian optimization algorithm in MATLAB to select the type of neural network and related hyperparameters. Bayesian optimization is a hyperparameter search algorithm based on Bayesian theory. It can find the optimal hyperparameter combination by establishing a probability model representing the objective function. This experiment selects the following hyperparameters for optimization. The specific hyperparameters are shown in Table III. The squared error of the prediction dataset and validation dataset of the corresponding model used at the same time is used as the objective function of the Bayesian optimization algorithm.

TABLE III
THE HYPERPARAMETERS BY USING BAYESIAN OPTIMIZATION.

Hyperparameter	Range	Description
Mode	[1, 3]	This parameters defines three models: Fully connected network; LSTM; GRU.
Hidden_units	[1, 512]	The number of hidden units affects the model's expressive power.
Batch_size	[100, 1000]	Its size affects the degree of optimization and the speed of the model.
Initial_learn_rate	[0.01, 1]	It affects the convergence speed of training and the performance of the model.
L2Regularization	[1e-5, 1e-2]	It prevents model overfitting and improves generalization ability.
Scale_parameter s	[0.01, 1]	The self-determined parameter s in (7), determines the resolution of the LLR at zero.

TABLE IV
THE TRAINING RESULT OF DIFFERENT DEEP LEARNING NETWORKS.

Network	Fullyconnected	LSTM	GRU
Hidden_units	N/A	248	275
Convergence speed	2 epoch	16 epoch	21 epoch
Batch_size	508	785	698
Initial_learn_rate	0.12	0.16	0.31
L2Regularization	4.74e-5	2.95e-5	1.84e-5
Scale_parameter s	0.21	0.16	0.24
Validation probability	99.6%	98.7%	99.1%

TABLE V
THE PERFORMANCE COMPARISON OF DL-FSC DECODER AND SC DECODER UNDER 5G PBCH AND PUCCH CHANNELS.

Channel	Polar encode	Performance improvement (dB)
PBCH	$P(512, 56)$	0.046
PUCCH	$P(1024, 75)$	0.074
	$P(1024, 139)$	0.023
	$P(1024, 267)$	0.022
	$P(1024, 523)$	0.008
	$P(1024, 523)_{2-segm}$	0.011

TABLE IV shows the training result of different deep learning networks as the core network of the DL-FSC decoder. Although LSTM, GRU and CNN [18] can effectively solve problems such as long-term memory and gradients in back-propagation, fully connected networks have a huge advantage in terms of convergence speed, model size, and verification probability in this application scenario. Most importantly, the simpler network structure makes fully connected networks more hardware-friendly and easy to hardwareize. Therefore, a fully connected network is used as the core component of the deep learning-assisted SC decoder, which is conducive to achieving good decoding performance and faster response speed, even under limited computing resources. The specific network structure comprises an input layer, a fully connected network with weight dimensions of 256×16 and bias dimensions of 128×1, a softmax activation layer, and an output layer. Additionally, the data type for this network is INT8. This compact and efficient neural network architecture allows the deep learning-assisted SC decoder to balance decoding performance and computational complexity, making it suitable for practical implementation in resource-constrained 5G and beyond communication systems.

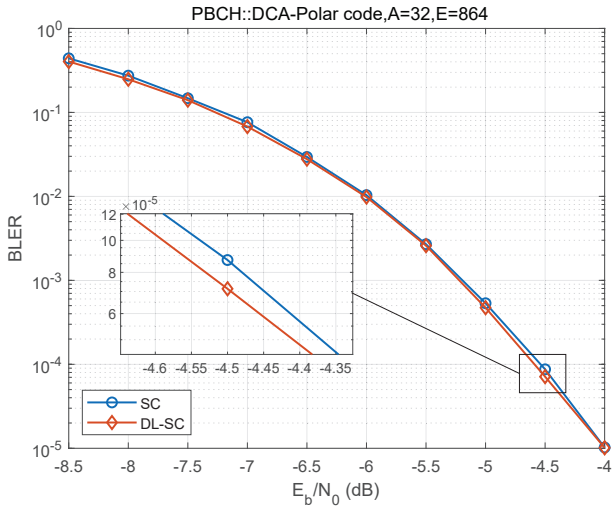
B. 5G Channel Simulation Results

To evaluate the error correction performance of the proposed DL-FSC decoder in a 5G communication context, we conduct simulations using BLER as the performance metric. The simulations are implemented in MATLAB, with the channel model set as AWGN and the modulation scheme as QPSK. To ensure a fair comparative analysis, the testing process for the proposed DL-FSC decoder is consistent with that of the other considered decoders, capturing a minimum of 50 error events. The comparison results of BLER versus E_b/N_0 performance are shown in Fig. 7, when using QPSK for communication over an AWGN channel.

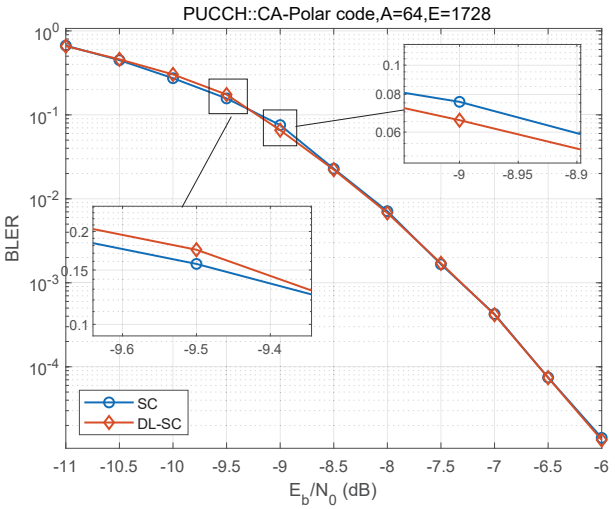
As shown in Table V, the experimental results under 5G communication channels show that compared with the traditional SC decoder, the decoding performance of DL-FSC decoder is improved by 1% (0.046 dB) on average in PBCH channel and 0.7% (0.028 dB) on average in PUCCH channel. In the context of polar code decoding, the DL-FSC decoder exhibits a higher degree of parallelism compared to the SC decoder. However, the incorporation of the DL component also introduces an associated increase in computational complexity.

C. Hardware Design and Comparison of DL-FSC Decoder

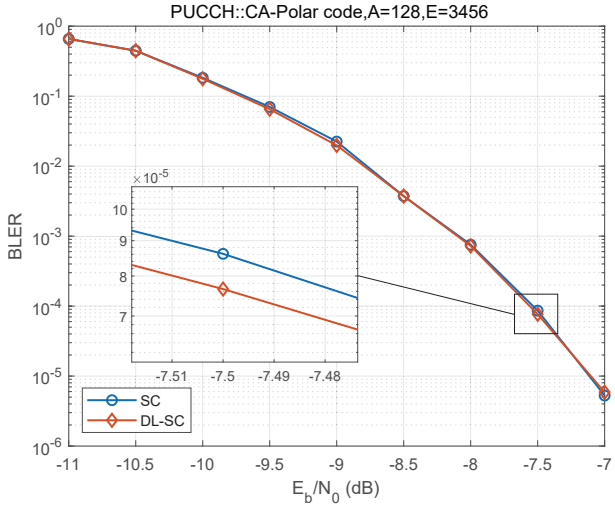
In terms of hardware, the proposed DL-FSC decoder achieves a decoding time of 14 clock cycles for $N = 16$ polar codes, without considering input data preparation time. Table VI presents the deep learning-based polar code decoder accelerator proposed in this work, compared to other polar code decoders. For ease of comparison, the results are based on FPGA synthesis. The deep learning-assisted FSC decoder in this study consumes more logic resources (look-up tables and flip-flops) on the FPGA compared to decoder [26] and decoder [27]. However, this design avoids using RAM and prioritizes more logic resources to achieve higher processing efficiency. Simultaneously, the proposed decoder significantly



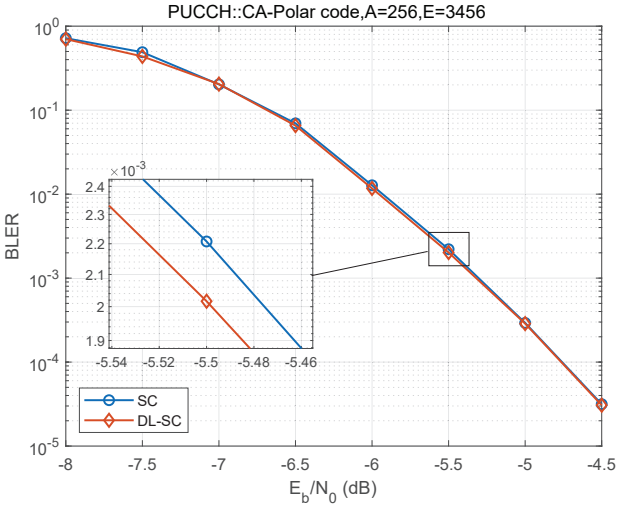
(a) $P(512,56)$



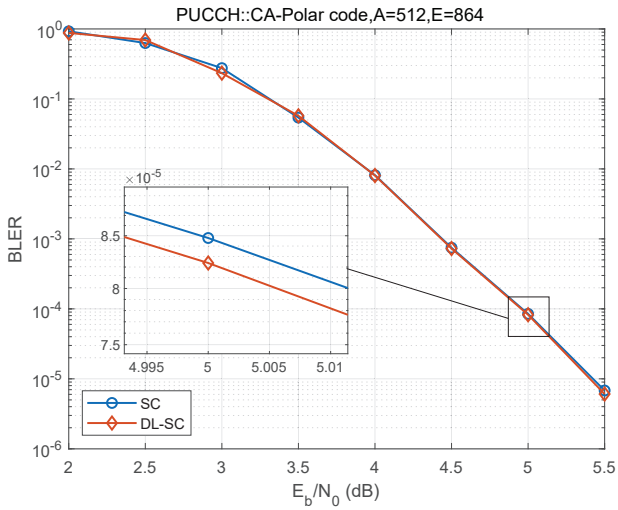
(b) $P(1024,75)$



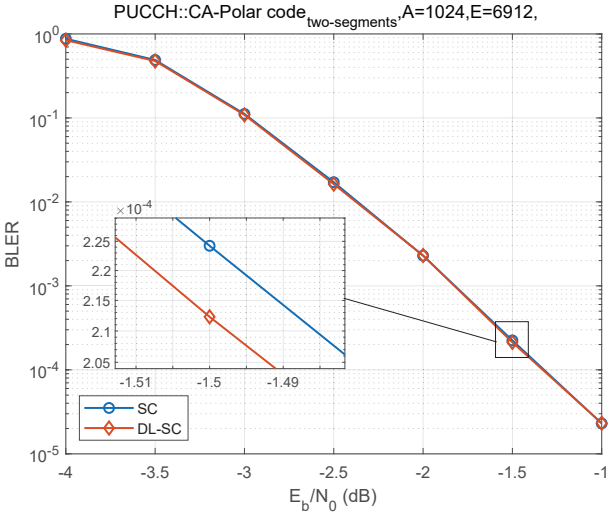
(c) $P(1024,139)$



(d) $P(1024,267)$



(e) $P(1024,523)$



(f) $P(1024,523)$

Fig. 7. The BLER performance comparison of the proposed DL-FSC decoder and SC decoder for the 5G polar code, when using QPSK for communication over an AWGN channel.

TABLE VI
COMPARING THE SYNTHESIS OUTCOMES FOR THE DECODER ON THE XILINX FPGA WITH CODE LENGTHS $N = 16$.

	Ours	**Y Ali [26]**	**SP Badar [27]**
LLR quantification	8 bit	5 bit	5 bit
LUTs	5798	1460	722
FFs	1723	261	N/A
RAM [bits]	N/A	114	127
Frequency [Mhz]	400	360	300
Throughput [Mbps]	4571	2672	1245

improves throughput by sacrificing some hardware resources, achieving a $1.71\times$ throughput improvement compared to the latest decoder [26]. In future communication systems and other high-speed data transmission scenarios with strict requirements for real-time performance and data processing speed, the proposed decoding accelerator can provide enhanced data processing capabilities.

V. CONCLUSION

This paper proposes a DL-FSC polar code decoder. The proposed DL-FSC decoder connects the sub-DL decoder through the SC decoder. The sub-DL decoder can be regarded as a mapper of the general decoder $f\{R(llr, frozen\ bit\ pattern) \rightarrow R(\hat{u})\}$, and the frozen bit pattern allows the DL-FSC decoder to support Polar codes with any code rate under a certain code length. We prove that the proposed DL-FSC decoder has slightly better BLER performance than the SC decoder. At the same time, the proposed deep learning accelerator improves the decoding rate from the hardware level. Compared with the latest literature, the proposed decoding accelerator improves the throughput by $1.71\times$. The proposed polar code encoding and decoding system based on deep learning is suitable for future intelligent communication systems. Combined with intelligent decoding based on deep learning, it can improve the reliability of communication links and data transmission efficiency. Our future work will design and implement the hardware architecture of the proposed DL-FSC decoder, and optimize the fully connected network of DL-FSC through quantization, pruning, and other algorithms to reduce decoding delays.

REFERENCES

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes," in *Proc. IEEE ISIT*, 2008.

[2] U. Equipment, "Technical specification group radio access network; NR; user equipment (UE) radio transmission and reception; part 1: Range 1 standalone (release 15)," *Part1: Range*, vol. 1.

[3] J. Dai, K. Niu, Z. Si, and J. Lin, "Polar coded non-orthogonal multiple access," in *Proc. IEEE ISIT*, 2016.

[4] A. K. Ahmed and H. S. Al-Raweshidy, "Performance evaluation of serial and parallel concatenated channel coding scheme with non-orthogonal multiple access for 6G networks," *IEEE Access*, vol. 10, pp. 39681–39690, 2022.

[5] E. Arikan, "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, 2009.

[6] L. Xiang, S. Zhong, R. G. Maunder, and L. Hanzo, "Reduced-complexity low-latency logarithmic successive cancellation stack polar decoding for 5G new radio and its software implementation," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12449–12458, 2020.

[7] S. M. Abbas, Y. Fan, J. Chen, and C.-Y. Tsui, "High-throughput and energy-efficient belief propagation polar code decoder," *IEEE IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 1098–1111, 2017.

[8] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[9] T. Wang *et al.*, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, 2017.

[10] H. Abdelbaki, E. Gelenbe, and S. El-Khamy, "Random neural network decoder for error correcting codes," in *Proc. IEEE IJCNN*, 1999.

[11] A. S. Hadi, "Linear block code decoder using neural network," in *Proc. IEEE IJCNN*, 2008.

[12] T. Gruber, S. Cammerer, J. Hoydis, and S. Ten Brink, "On deep learning-based channel decoding," in *Proc. IEEE CISS*, 2017.

[13] H. Zhu, Z. Cao, Y. Zhao, and D. Li, "Learning to denoise and decode: A novel residual neural network decoder for polar codes," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8725–8738, 2020.

[14] C. Wen, J. Xiong, L. Gui, and L. Zhang, "A BP-NN decoding algorithm for polar codes," in *Proc. WCSP*, 2019.

[15] 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, 2021.

[16] N. Doan, S. Ali Hashemi, and W. J. Gross, "Neural successive cancellation decoding of polar codes," in *Proc. IEEE SPAWC*, 2018.

[17] J. Fang *et al.*, "Neural successive cancellation polar decoder with Tanh-based modified LLR over FSO turbulence channel," *IEEE Photon. J.*, vol. 12, no. 6, pp. 1–10, 2020.

[18] H. Liu, L. Zhang, W. Yan, and Q. Ling, "Neural-network-assisted polar code decoding schemes," *Applied Sciences*, vol. 12, no. 24, p. 12700, 2022.

[19] E. Arikan, "Systematic polar coding," *IEEE Commun. Lett.*, vol. 15, no. 8, pp. 860–862, 2011.

[20] G. Sarkis *et al.*, "Flexible and low-complexity encoding and decoding of systematic polar codes," *IEEE Trans. Commun.*, vol. 64, no. 7, pp. 2732–2745, 2016.

[21] S. A. Hashemi, C. Condo, and W. J. Gross, "Fast and flexible successive-cancellation list decoders for polar codes," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5756–5769, 2017.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[23] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[24] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 5, pp. 946–957, 2014.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[26] Y. Ali *et al.*, "Efficient hardware realization of SC polar decoders using compound pipelined processing units and auxiliary registers," *IEEE Access*, vol. 12, pp. 23808–23826, 2024.

[27] S. P. Badar and K. Khanchandani, "Successive cancellation polar decoder implementation using processing elements," in *Proc. IEEE TENSYMP*, 2022.

Haogang Feng was born in Henan, China, 1997. He received the B.Sc. degree in Electronics Engineering from Shenzhen University, Shenzhen, China, in 2019. From 2019 on, he will pursue Ph.D. degree in Electronics and Electrical Engineering in Shenzhen University, Shenzhen, China. His current research interests include feld-programmable gate array (FPGA), joint algorithm and hardware codesign, and application-specified integrated circuit (ASIC) implementations for next-generation channel coding system.

Haiyu Xiao was born in Fujian, China in 1996. Obtained a bachelor's degree in Electronic Information Engineering from Chengdu University of Technology in 2019. Starting from 2021, he will study for a master's degree in Integrated Circuit Engineering from Shenzhen University. His current research interests include 5G polar code design, neural network accelerator design and application specific integrated circuit (ASIC) implementation of next-generation channel coding systems.

Shida Zhong received the B.Sc. degree in electronics engineering from Shenzhen University, Shenzhen, China, in 2008, and the M.Sc. and Ph.D. degrees in Electronics and Electrical Engineering from the University of Southampton, Southampton, U.K., in 2009 and 2013, respectively. He is currently an Assistant Professor with the College of Electronics and Information Engineering, Shenzhen University. His current research interests include low power IC design, FPGA and ASIC implementations for next-generation channel coding system.

Zhuqing Gao received the B.E. degree from Nanjing Institute of Technology in 2004 and joined Potevio in the same year for R&D of PCMs and optical terminals. Since 2005, he has been engaged in the development and design of handset products in Inventec, ZTE, and Xiaomi. He is dedicated to technology innovation and research, and has obtained more than 20 patents in semiconductor and product application.

Tao Yuan (M'19) received the B.E. degree in Electronic Engineering and the M.E. degree in Signal and Information Processing at Xidian University, Xi'an, Shaanxi, China, in 1999 and 2003, respectively, and the Ph.D. degree in Electrical and Computer Engineering at National University of Singapore, Singapore, in 2009. He is now a Distinguished Professor (2016–) with the College of Electronics and Information Engineering at Shenzhen University, Shenzhen, Guangdong, China. He is the Director of the Guangdong Provincial Mobile Terminal Microwave and Millimeter-Wave Antenna Engineering Research Center and the Deputy Director of the Guangdong-Hong Kong Joint Laboratory for Big Data Imaging and Communication. His current research interests include design and implementation of novel RF frontend chips/modules and integrated devices/antennas/circuits for 5G/6G applications.

Zhi Quan is a distinguished professor with the College of Electronic and Information Engineering, Shenzhen University, China. He received his Ph.D. in Electrical Engineering from University of California, Los Angeles (UCLA) with highest honors in 2009, and his B.E. in Communications Engineering from Beijing University of Posts and Telecommunications (BUPT), China in 1999. He worked as a Sr. System Engineer in Qualcomm Research Center (QRC) of Qualcomm Inc. (San Diego, CA) during 2008-2012, and as a RF System Architect with Apple Inc. (Cupertino, CA) during 2012-2015. Dr. Quan has been granted over 40 patents, and published over 70 papers in wireless communications and signal processing with more than 5000 citations from Google Scholar. Dr. Quan was the recipient of UCLA Outstanding Ph.D. Award in 2009, IEEE Signal Processing Society Best Paper Award in 2012, China National Excellent Young Scientist Foundation in 2016, and First Prize Technology Innovation Award by China Institute of Communications in 2020. His current research interests include wireless communication systems, RF system calibration and measurement, data-driven signal processing, and machine learning.