IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received October 9, 2019, accepted October 28, 2019, date of publication October 31, 2019, date of current version November 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950766

A Low-Complexity Belief Propagation Based Decoding Scheme for Polar Codes - Decodability Detection and Early Stopping Prediction

YAOHAN WANG[1,2,3], **SHUNQING ZHANG**[1,2,3], **(Senior Member, IEEE), CHUAN ZHANG**[4], **(Member, IEEE), XIAOJING CHEN**[1,2,3], **AND SHUGONG XU**[1,2,3], **(Fellow, IEEE)**

[1]Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
[2]Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai University, Shanghai 200444, China
[3]Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, Shanghai 200444, China
[4]National Mobile Communications Research Laboratory, Southeast University, Nanjing 211189, China

Corresponding author: Shunqing Zhang (shunqing@shu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61701293 and Grant 61871262, in part by the National Science and Technology Major Project under Grant 2018ZX03001009, in part by the National Key Research and Development Program of China under Grant 2017YFE0121400, in part by the Huawei Innovation Research Program (HIRP), and in part by the research funds from Shanghai Institute for Advanced Communication and Data Science (SICS).

ABSTRACT In the 5G communication systems, polar code has been adapted as the control channel coding solution in the enhanced mobile broadband (eMBB) scenario. Although different decoding schemes, including belief propagation (BP) and successive cancellation (SC) based algorithms, have been proposed, the decoding complexity as well as the latency are still significant. To address this critical issue, several low-complexity schemes, e.g., the use of simplified decoding operation and stop the decoding operation in earlier stage, have been proposed recently. However, conventional early stopping strategies have to check a pre-defined metric in each iteration, and the associated decoding delay is significant. In this paper, to address this challenge, we proposed a low-complexity BP based decoding scheme, which contains the decodability detection stage and the early stopping prediction stage. The decodability detection stage can identify the codewords in the deep channel fading environment and eliminate the unnecessary decoding operations to reduce the decoding complexity, while the early stopping prediction stage can directly predict the required number of iterations rather than checking the metric in each iteration to avoid the associated decoding delay. Through the above two approaches, our proposed scheme is shown to achieve 71% decoding delay reduction and maintain the same decoding performance as traditional BP, *G-matrix*, *MinLLR* schemes.

INDEX TERMS Polar codes, deep learning, BP decoding, decodability detection, early stop prediction.

I. INTRODUCTION

Polar code, invented by E. Arıkan in [1], has been regarded as the first capacity achieving code for the binary discrete memoryless channels. Due to the capacity achieving property, polar code has been adopted as the channel coding scheme for transmitting control signals in the fifth generation (5G) wireless communication systems [2]. Although the encoding delay and complexity for polar code is similar with traditional turbo code [3] and low density parity check (LDPC) code, the decoding delay and complexity for polar code are in general much more significant, which triggers a great amount of research efforts recently [4]–[9].

Based on the existing literature, the decoding schemes for polar code can be mainly divided into two categories, i.e., *successive cancellation* (SC) based [1] and *belief propagation* (BP) based decoding algorithms [10]. In the SC based scheme, since the cancellation relies on the previous decoded information, the optimal detection strategy needs to iterate bits by bits, which incurs significant decoding delay. For example, successive cancellation list (SCL) and cyclic redundancy check (CRC) aided SCL (CA-SCL) as

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

proposed in [4], [11] require the decoding delay increases exponentially with code length. To reduce the decoding delay, successive cancellation stack (SCS) has been proposed in [5] to reduce the searching depth of the potential trellis paths. However, due to the serial decoding nature of all the above SC based schemes, the decoding delay is still significant and the practical hardware throughput is usually limited as reported in [6], [7].

To address this issue, the BP based scheme utilizes a parallel structure to achieve high throughput and low latency as mentioned in [8]. Using the BP based decoding strategy, log-likelihood ratios (LLRs) can be propagated and updated stage by stage in the corresponding factor graph of polar code, which can be easily implemented via a pipelined structure in very large scale integration (VLSI) design. However, the propagation and updating process in the BP based decoding usually involves highly nonlinear operations as shown later, and the decoding complexity is still significant. To reduce the associated decoding complexity, min-sum (MS) [8] and normalized min-sum (NMS) [9] decoding schemes have been proposed to approximate the original nonlinear operations using some basic operations. Another possible solution is to stop the BP iterations at the earlier stage instead of running the whole iteration processes [12]–[14]. For example, a matrix multiplication based method named "*G-matrix*" has been proposed in [12], which checks the amplitude of LLR values after each iteration. In [13], [14], the early stopping strategy has been improved to use the signs and magnitudes of LLRs, and the corresponding checking complexity can be reduced. To further reduce the decoding complexity, the following two questions will be critical based on the current investigation.

- **Is it necessary to perform all the decoding process at any time?** If the transmitted polar codeword suffers from a deep channel fading, the received side can not recover the original information bits no matter how many iterations are executed in the BP decoding process. Therefore, a more reasonable policy is to eliminate the whole decoding process and immediately ask the transmitter to re-transmit again. With this in mind, if the decodability of received codewords can be detected in advance, we can minimize the associated complexity as well as power consumption in the BP decoding iterations and quickly generate a feedback signal for re-transmission to minimize the potential delay. As far as we are aware, the above questions for polar code are still open.
- **Can we directly predict the number of iterations required for early stopping?** Another interesting problem is whether it is possible to predict the number of iterations in advance. Conventional early stopping strategies as mentioned before, rely on checking the corresponding stopping criterion during each iteration.[1] As the criterion checking can be regarded as an interruption to the continuous BP iterations, the pipelined decoding flow in the hardware implementation has to be paused during each iteration, which causes significant delay overhead when the number of iterations is not small.

Since the exact decoding function between the received symbols and the decoded information bits is difficult to characterize in general, it can be even more challenging to describe the above problems using standard mathematical frameworks. To solve this issue, a model free based decoding technology has been proposed recently by applying the deep learning based approaches. For example, a standard three-layer perceptron has been proposed in [15], [16] to model the decoding process for polar codes. In [17], [18], the similar idea has been proposed to improve the performance of traditional BP or MS algorithms by adjusting the weights in the factor graph using stochastic gradient descent (SGD). A unified decoding framework for polar and LDPC codes has been proposed in [19], which utilizes the generalization capability of neural networks.

Motivated by the aforementioned examples, we try to provide some preliminary answers to the above critical issues in this paper. To be more specific, we propose a low-complexity BP based decoding framework for polar codes by applying the deep learning technology, which consists of the decodability detection and the early stopping prediction procedures. Nevertheless, as we will show later, to design and implement them without domain knowledge in wireless communications is never straight-forward, and a careful labelling mechanism to reflect the wireless channel conditions is necessary. With sufficient training data and carefully designed schemes, the decodability detection stage is able to extract the intrinsic features of LLRs and successfully predict the decodability of received symbols.[2] Meanwhile, the early stopping precidtion stage can estimate the required number of iterations by learning the average/total improvment of LLRs during BP iterations. Combine the above two stages, we can avoid unnecessary decoding processes for polar codes in the low signal-to-noise ratio (SNR) regime,such as deep fading channel.[3] Based on the numerical experiments, we show that the proposed low-complexity BP based decoding scheme can achieve 10 times complexity reduction compared with BP [9] in code length $N = 1024$ and per bit SNR $E_b/N_0 = 5$ dB while maintaining the same BLER performance.

The rest part of this paper is organized as follows. Section II introduces some basic knowledge related to the BP based low-complexity polar decoding scheme and the deep learning technology. In Section III, a low-complexity BP based

[1] In this paper, our main target is to figure out a low complexity decoding scheme on top of the conventional BP decoding algorithms. Since the convergence property of BP algorithms has already been proven in [10].

[2] As for the decodability detection, for those codewords in the deep fading environments (e.g. the absolute values of LLRs are close to zero), the codeword are likely to be undecodable, no matter which kind of code is applied.

[3] As for the early stopping prediction, since we only perform the numerical simulations for polar codes, we cannot provide a concrete conclusion for other "structured" codes and the extension to other coding schemes are still unknown. However, we believe the proposed scheme can be extended to linear block codes as the LLRs have strong correlations among neighboring symbols/LLRs in general.

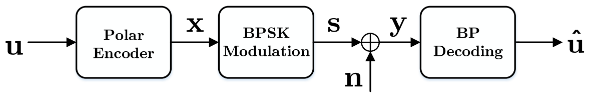


FIGURE 1. An overview of polar encoding/decoding system.

decoding framework with decodability detection and early stopping prediction is proposed and the corresponding complexity analysis is provided in Section IV. The experimental configurations and results are illustrated in Section V, respectively. Finally, we draw the conclusion in VI.

II. BACKGROUND

In this section, we briefly describe the background information for polar code and the deep learning technology.

A. POLAR CODE AND BP DECODING

Polar code is typically regarded as one type of linear block codes, which is constructed on the basis of channel polarization. Consider an (N, K) polar code with rate $R = K/N$ as shown in FIGURE 1, where N bits sequence $\mathbf{u} = [u_1, u_2, \ldots, u_N]$ consisting of K information bits and $N-K$ frozen bits are jointly encoded into $\mathbf{x} = [x_1, x_2, \ldots, x_N]$. Denote $\mathbf{G}_p$ to be the corresponding generation matrix and the N-bit coded information $\mathbf{x}$ is generated at the transmitter side through $\mathbf{x} = \mathbf{u} \cdot \mathbf{G}_p$, where $\mathbf{G}_p$ can be obtained via a continuous Kronecker product of $\mathbf{F}$, e.g.,

$$\mathbf{G}_p = \mathbf{F}^{\otimes \log_2 N}, \text{ and } \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{1}$$

In the above equation, $\mathbf{F}^{\otimes n}$ represents the n-th Kronecker power of $\mathbf{F}$. Without loss of generality, we assume binary phase shift keying (BPSK) and the transmitted symbols $\mathbf{s} = [s_1, s_2, \ldots, s_N]$ can be generated through,

$$s_j = \begin{cases} 1, & x_j = 1 \\ -1, & x_j = 0. \end{cases} \tag{2}$$

At the receiver side, the observed symbols, $\mathbf{y}$, are corrupted by the additive white Gaussian noise (AWGN) denoted by $\mathbf{n} = [n_1, n_2, \ldots, n_N]$, and the equivalent mathematical expression is given by $\mathbf{y} = \mathbf{s} + \mathbf{n}$. To eliminate the effect of different modulation schemes, per-bit LLRs, e.g, $\hat{\mathbf{b}} = [\hat{b}_1, \ldots, \hat{b}_N]$ are often calculated before the decoding process, which are defined as, $\hat{b}_j = \ln \frac{P(x_j=0|\mathbf{y})}{P(x_j=1|\mathbf{y})}$, for all $1 \leq j \leq N$. In the conventional BP decoding process, two types of LLRs, i.e., left-to-right messages, $\mathbf{r}_i^{(t)} = [r_{i,1}^{(t)}, \ldots, r_{i,N}^{(t)}]$, and right-to-left messages, $\mathbf{l}_i^{(t)} = [l_{i,1}^{(t)}, \ldots, l_{i,N}^{(t)}]$, are utilized, where i and t denote the stage and iteration indices, respectively. As shown in FIGURE 2, the BP decoding procedures rely on the bi-directional update of the left-to-right and right-to-left messages according to the following recursive relations,

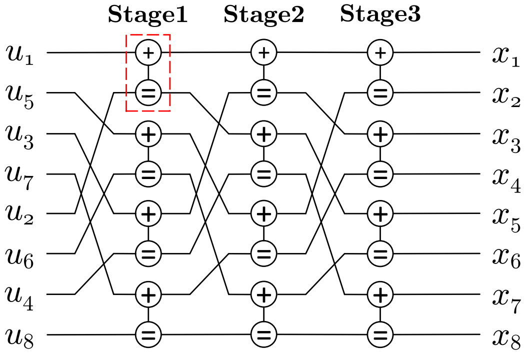


FIGURE 2. A factor graph of polar code with the block length $N = 8$. The dashed block represents a basic processing element (PE).

$$\begin{cases} l_{i,j}^{(t)} = g\left(l_{i+1,2j-1}^{(t)}, l_{i+1,2j}^{(t)} + r_{i,j+N/2}^{(t-1)}\right) \\ l_{i,j+N/2}^{(t)} = g\left(r_{i,j}^{(t-1)}, l_{i+1,2j-1}^{(t)}\right) + l_{i+1,2j}^{(t)} \\ r_{i+1,2j-1}^{(t)} = g\left(r_{i,j}^{(t)}, l_{i+1,2j}^{(t)} + r_{i,j+N/2}^{(t)}\right) \\ r_{i+1,2j}^{(t)} = g\left(r_{i,j}^{(t)}, l_{i+1,2j-1}^{(t)}\right) + r_{i,j+N/2}^{(t)}, \end{cases} \tag{3}$$

where the function, $g(a, b) = \ln \frac{1+e^{a+b}}{e^a+e^b}$, is usually approximated as $g(a, b) \approx \alpha \cdot \mathrm{sgn}(a)\mathrm{sgn}(b)\min(|a|, |b|)$ according to [9]. Where $\mathrm{sgn}(\cdot)$ is sign function. We use this NMS method as the BP decoding scheme in the rest of the paper. Denote $N_s = \log_2 N$ to be the total number of stages and the initial condition for the above recursive update is given by,

$$\begin{cases} \mathbf{l}_{N_s+1}^{(0)} = \hat{\mathbf{b}} \\ \mathbf{r}_1^{(0)} = [r_{1,1}^{(0)}, \ldots, r_{1,j}^{(0)}, \ldots, r_{1,N}^{(0)}], \end{cases} \tag{4}$$

where $r_{1,j}^{(0)}$ equals to 0, if the j^{th} bit is the information bit, and equals to a large value otherwise. After $N_{\max}$ (the preset maximum number of iterations) rounds of BP iterations, the decoded bits $\hat{\mathbf{u}} = [\hat{u}_1, \ldots, \hat{u}_N]$ are obtained via,

$$\hat{u}_j = \begin{cases} 0, & \text{if } l_{1,j}^{N_{\max}} \geq 0 \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

B. EARLY STOPPING SCHEMES

In practical applications, the BP decoding scheme may not need $N_{\max}$ rounds of iterations before it can converge. Therefore, a reasonable low-complexity decoding scheme is to stop the iteration processes when the decoder can recover the original transmitted sequence $\mathbf{u}$. As proposed in [12], two types of early stopping criteria, called *G-matrix* and *MinLLR*, have been widely used, where both of them can reduce the number of iterations during the BP decoding. For the *G-matrix* early stopping scheme, the decoder estimates the information sequence $\hat{\mathbf{u}}^{(t)}$ and the encoded bits $\hat{\mathbf{x}}^{(t)}$ by manipulating the sum value of $\mathbf{l}_1^{(t)}$ and $\mathbf{r}_1^{(t)}$ as well as $\mathbf{l}_{N_s+1}^{(t)}$ and $\mathbf{r}_{N_s+1}^{(t)}$, respectively, and checks the equality $\hat{\mathbf{u}}^{(t)} \cdot \mathbf{G}_p = \hat{\mathbf{x}}^{(t)}$ during each iteration. If the above equation

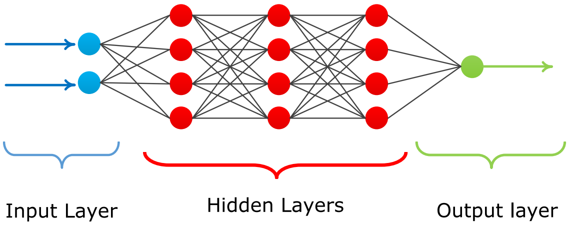


FIGURE 3. An example of three-layer DNN used in [15] and [16]. This type of neural network architectures generally includes the input layer, three hidden layers, and the output layer.

holds, the decoding process is assumed to be successful, and otherwise, the iteration process continues. For the *MinLLR* scheme, most of the above procedures are identical except for the early stopping criterion. During each iteration, *MinLLR* computes the minimum absolute value of the output vector $\{|l_{1,j}^{(t)} + r_{1,j}^{(t)}|\}$ and compares them with a predefined threshold.

Both *G-matrix* and *MinLLR* can reduce the number of iterations in the BP decoding processes. However, the decoding delay may still be quite significant when the required number of iterations is large. This is because computing the early stopping criterion during each iteration usually require complicated matrix multiplication and comparison, and hence, a novel approach to estimate the required number of iterations is more desirable.

C. DEEP LEARNING FOR CHANNEL CODING

Deep learning [20] has been considered as a new approach to describe nonlinear input and output relations and has been widely applied to describe some challenging relations in wireless communications. In the area of channel coding, since there are many nonlinear relations among different parameters, the deep learning techniques begin to play an important role recently. For example, it has been applied in [17] to optimize the scaling parameters in the BP decoding processes, and in [21], a recurrent neural network (RNN) based polar decoder has been invented to further quantize those parameters for efficient hardware processing.

Instead of optimizing the decoding parameters, another type of deep learning based scheme focuses on modeling the entire nonlinear decoding functions, where [15], [16] adopts three-layer DNN to achieve similar BER performance with BP or SCL decoding as shown in FIGURE 3. In [19], a unified polar-LDPC decoder using the deep learning based structure has been proposed, which achieve similar decoding performance for both polar and LDPC decoders simultaneously.

Although the above schemes provide several forward-looking deep learning based approaches for efficient channel decoding, the associated decoding complexities, especially with the additional deep learning functions, have not been carefully studied according to the existing literature.

III. LOW-COMPLEXITY BP BASED DECODING

In this section, a low-complexity BP based decoding framework for polar code is presented, which includes the decodability detection and the early stopping prediction.

A. OVERVIEW

To reduce the decoding complexity, a straightforward idea is to minimize the decoding efforts when the coded packets are suffering the deep channel fading. If we use the subscript k to indicate the index of coded blocks and denote $T_k^{\min}$ to be the required minimum number of iterations for decoding the k^{th} block, a total number of N_{BL} coded blocks can be classified as follows,

$$\begin{aligned} N_{BL} &= \sum_{k=1}^{N_{BL}} \left(\mathcal{I}\left(\mathbf{u}_k = \hat{\mathbf{u}}_k^{(T_k^{\min})}\right) + \mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(T_k^{\min})}\right) \right) \\ &= \sum_{k=1}^{N_{BL}} \Bigg(\underbrace{\mathcal{I}\left(\mathbf{u}_k = \hat{\mathbf{u}}_k^{(T_k^{\min})}\right)}_{\text{Correct Blocks}} \\ &\quad + \underbrace{\mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(T_k^{\min})} | T_k^{\min} = N_{\max}\right)}_{\text{Type I block errors}} \\ &\quad + \underbrace{\mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(T_k^{\min})} | T_k^{\min} < N_{\max}\right)}_{\text{Type II block errors}} \Bigg) \end{aligned} \tag{6}$$

where $\mathcal{I}(\cdot)$ denotes the indicator function, which equals to one when the inner condition holds and zero otherwise. The corresponding BLER, is defined as $\sum_{k=1}^{N_{BL}} \mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(T_k^{\min})}\right) / N_{BL}$.

As shown in Eq. (6), we can categorize the block decoding errors into two types, where "Type I block errors" represent the decoding errors happened after $N_{\max}$ rounds of iterations, and "Type II block errors" occur when the early stopping criterion is triggered. If the "Type I block errors" can be identified before the entire decoding process, we can directly declare the block error and save the associated decoding power without loss of the BLER performance. Meanwhile, if the required number of BP iterations, $T_k^{\min}$, can be estimated in advance, we can minimize the associated complexity for computing the early stopping criterion as well.

Motivated by the above observations, we propose a low-complexity BP based decoding framework as shown in FIGURE 4, which includes the decodability detection and the early stopping prediction blocks. The decodabilty detection block is mainly targeting for identifying "Type I block errors" and directly declaring transmission errors through a negative acknowledgement (NACK). With this mechanism, the transmitters can perform re-transmission immediately if needed. The early stopping prediction block is cascaded afterwards, which estimates the number of iterations required for BP iterations. As the early stopping prediction is not perfect in general, we also propose a compensation part after the predicted rounds of BP iterations to avoid unnecessary decoding errors, which triggers the compensation process when the CRC check is not satisfied. The detailed descriptions are provided in what follows.

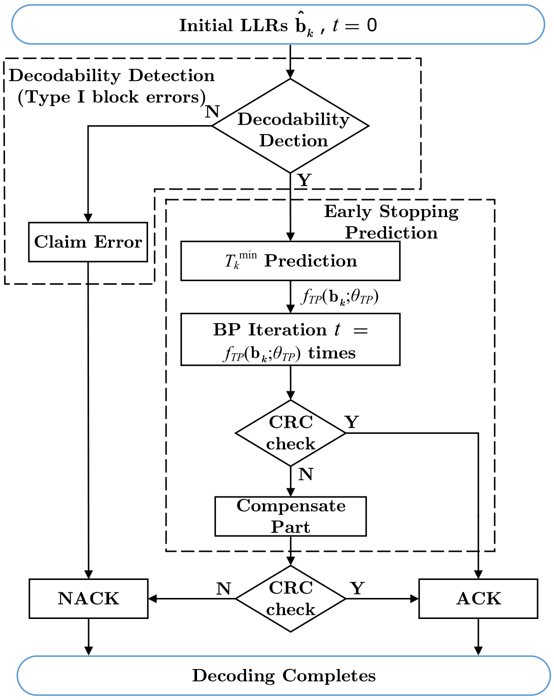

Initial LLRs $\hat{\mathbf{b}}_k$, $t = 0$
Decodability Detection
(Type I block errors)
N
Decodability Dection
Y
Claim Error
Early Stopping Prediction
$T_k^{\min}$ Prediction
$f_{TP}(\mathbf{b}_k;\theta_{TP})$
BP Iteration t = $f_{TP}(\mathbf{b}_k;\theta_{TP})$ times
CRC check
Y
N
Compensate Part
NACK
N
CRC check
Y
ACK
Decoding Completes

FIGURE 4. An overview of the proposed low-complexity BP based decoding framework, which contains the decodability detection stage and the early stopping prediction stage.

B. DECODABILITY DETECTION

In order to identify the "Type I block errors" based on the observed symbols, $\mathbf{y}_k$, or the equivalent LLRs, $\hat{\mathbf{b}}_k$, we need to find a function $f_{DD}(\cdot)$, which is able to model the behaviors of decodability detection. Mathematically, it can be expressed as the following non-convex optimization problem.

*Problem 1 (*BLE based Formulation):

$$\underset{\theta_{DD}}{\text{minimize}} \sum_{k=1}^{N_{BL}} \left| f_{DD}(\hat{\mathbf{b}}_k;\theta_{DD}) - \mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(N_{\max})}\right)\right|,$$
$$\text{subject to } \hat{\mathbf{u}}_k^{(N_{\max})} = f_{BP}(\hat{\mathbf{b}}_k; N_{\max}), \quad |\hat{\mathbf{b}}_k| \leq \epsilon_{\mathbf{b}}, \quad (7)$$

where $f_{BP}(\cdot)$ denotes the BP decoding process, and $\epsilon_{\mathbf{b}}$ reflects the maximum LLR value supported by the practical systems.

By numerically generating different LLRs, $\hat{\mathbf{b}}_k$, and the corresponding "label", $\mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(N_{\max})}\right)$, we can apply the conventional machine learning approach in [19] to minimize the absolute value between $f_{DD}(\hat{\mathbf{b}}_k; \theta_{DD})$ and the event $\mathcal{I}(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(N_{\max})})$ to predict the "Type I block errors". However, the above approach to solve Problem 1 may not be a practical solution due to the following two reasons. First, the indicator function is non-differentiable and non-continuous, and a smooth sigmoid based activation function [22], [23] at the output layer to model $\mathcal{I}(\cdot)$ may cause the inaccurate prediction. As we directly claim decoding errors in the proposed scheme, the inaccurate prediction can result in BLER performance degradation as shown in FIGURE 5. Second, since the block error event, e.g., $\mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(N_{\max})}\right)$ may not be sufficient to describe different wireless fading environments, a more reliable scheme to understand different block error events is desired.

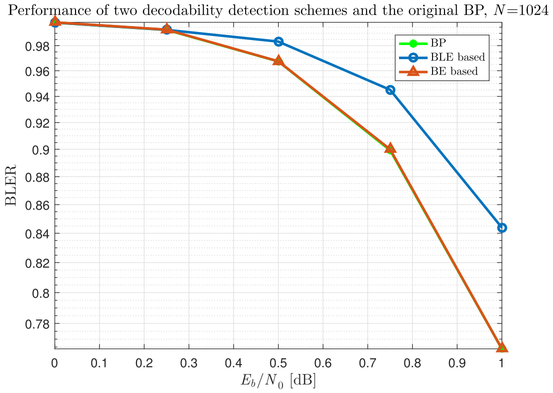


FIGURE 5. BLER versus per-bit SNR E_b/N_0 performance for different decoding schemes under the polar coding length $N = 1024$. As shown in this figure, the BE based formulation achieve the same BLER performance with the original BP scheme; while the BLE based formulation has the performance degradation due to the erroneous claim of "Type I block errors".

In order to control the BLER performance degradation under decodability detection schemes, a more reasonable approach is to introduce an auxiliary variable $\gamma_{\mathbf{u}}$ to describe the difference between the transmitted information bits $\mathbf{u}_k$ and the decoded bits $\hat{\mathbf{u}}_k^{(N_{\max})}$. Mathematically, we define $\gamma_{\mathbf{u}} = \mathbf{1}_N^T \cdot \text{abs}\left(\mathbf{u}_k - \hat{\mathbf{u}}_k^{(N_{\max})}\right)$ with abs$(\cdot)$ denoting the element-wise absolute value of the inner vector, and by tuning the value of $\gamma_{\mathbf{u}}$, we can control the prediction accuracy as summarized in the following lemma.

Lemma 1: The prediction accuracy,[4] defined as the correctly predicted error blocks over the total number of predicted error blocks, has the monotonically non-decreasing relation with respect to $\gamma_{\mathbf{u}}$.

Proof: Please refer to Appendix A for the proof. ■

Based on the above lemma, we propose the bit error (BE) based scheme by jointly considering the block error events and the number of error bits in each block together, where the mathematical formulation is given as follows.

*Problem 2 (*BE based Formulation):

$$\underset{\theta_{DD}}{\text{minimize}} \sum_{k=1}^{N_{BL}} \Bigg| f_{DD}(\hat{\mathbf{b}}_k;\theta_{DD}) - \mathcal{I}\left(\mathbf{1}_N^T \cdot \text{abs}\left(\mathbf{u}_k - \hat{\mathbf{u}}_k^{(N_{\max})}\right) \geq \gamma_{\mathbf{u}}\right)\Bigg|,$$
$$\text{subject to } \hat{\mathbf{u}}_k^{(N_{\max})} = f_{BP}(\hat{\mathbf{b}}_k; N_{\max}), \quad |\hat{\mathbf{b}}_k| \leq \epsilon_{\mathbf{b}}, \quad (8)$$

Based on the above BE based formulation, we can obtain a more reliable decodability detection of the "Type I block errors" by tuning the threshold $\gamma_{\mathbf{u}}$. By increasing the value of $\gamma_{\mathbf{u}}$, the neural networks are able to learn the intrinsic features of block error events with more error bits, and the decodability detection will be more reliable in

[4] The prediction accuracy in this paper is equivalent to the *precision* concept in the machine learning area.

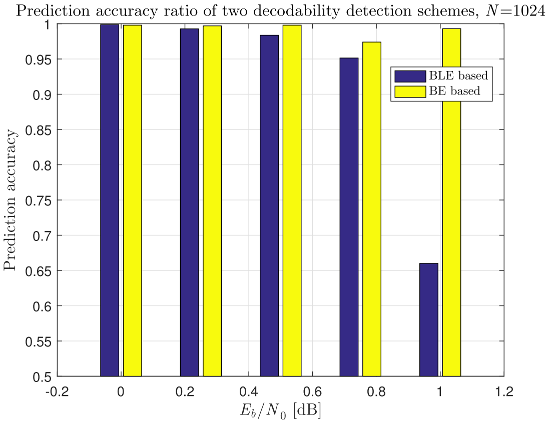


FIGURE 6. Prediction accuracy versus per-bit SNR E_b/N_0 relation for different decodability detection schemes under the code length $N = 1024$ and the pre-defined threshold $\gamma_{\mathbf{u}} = N/2$.

terms of the prediction accuracy. As shown in FIGURE 5, the BE based approach can achieve the same performance as compared with traditional BP detection, and provides 0.1 dB performance gain at BLER $=$ 0.9 as compared with the BLE based scheme. This is because the prediction accuracy has been improved by increasing the value of $\gamma_{\mathbf{u}}$ and the erroneous claim of "Type I block errors" diminishes accordingly.[5]

In FIGURE 6, we compare the detection accuracy under different per-bit SNR values to obtain a better understanding of the proposed schemes. As shown in this figure, if the value of $\gamma_{\mathbf{u}}$ exceeds half of the block length, e.g., $N/2$, the BE based formulation can provide a more reliable prediction accuracy than the BLE based scheme, which is thus adopted in the following evaluations.

C. EARLY STOPPING PREDICTION

After the decodability detection process, most of the remaining decoding blocks require less than $N_{\max}$ BP iterations, and therefore, another possible method to reduce the decoding complexity on top of the conventional early stopping schemes is to eliminate the possible computation of early stopping criteria, such as *G-Matrix*. Motivated by this goal, we formulate the non-convex optimization problem to predict the minimum required number of BP iterations, $f_{TP}(\cdot)$, as well as $f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP})$, as follows.

Problem 3 ($T_k^{\min}$ Prediction):

$$\begin{aligned}\underset{\theta_{TP}}{\text{minimize}}\ & \sum_{k=1}^{N_{BL}} \left| f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP}) - T_k^{\min}\right|^p, \\ \text{subject to } & T_k^{\min} = \left\{ \underset{t\in[1,N_{\max}]}{\arg\min} f_{BP}(\hat{\mathbf{b}}_k;t) = \mathbf{u}_k \right\}, \\ & |\hat{\mathbf{b}}_k| \le \epsilon_{\mathbf{b}}. \end{aligned} \tag{9}$$

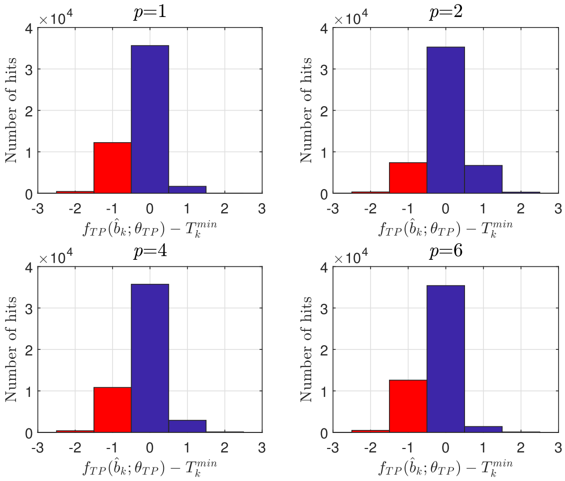


FIGURE 7. Histogram of $(f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP}) - T_k^{\min})$ for the coding length $N = 1024$ and SNR = 5 dB under different values of p. As shown in this figure, $p = 2$ provides the best prediction accuracy.

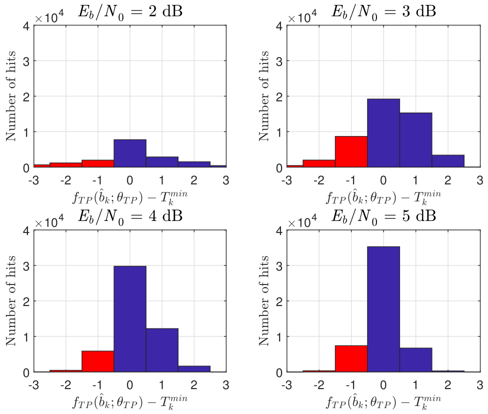


FIGURE 8. Histogram of $(f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP}) - T_k^{\min})$ for the coding length $N = 1024$ and $p = 2$ under different per-bit SNR values. As shown in this figure, insufficient BP iterations due to inaccurate predictions (red bar) may still happen under different SNR cases.

In the above formulation, we introduce an auxiliary variable p instead of choosing $p = 1$ in the decodability detection stage. This is because the decodability detection is a typical binary classification problem where the potential loss due to error prediction is equal to 1, while the early stopping prediction belongs to a multi-classification task. In this case, we need to use the auxiliary variable p to control the inter-class losses as illustrated in the following parts.

FIGURE 7 and FIGURE 8 illustrate the prediction accuracy under different values of p and per-bit SNR E_b/N_0 by accumulating the histogram of $(f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP}) - T_k^{\min})$. Once the predicted results, $f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP})$, are greater than or equal to the minimum required number of iterations, $T_k^{\min}$, we can still decode the block without errors. However, if $(f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP}) - T_k^{\min})$ is less than zero, we will suffer from the block decoding errors due to insufficient BP iterations. As shown in FIGURE 7 and FIGURE 8, $p = 2$ provides the best prediction accuracy when the per-bit SNR E_b/N_0 equals to 5 dB, and the proposed $T_k^{\min}$ estimation strategy still suffer from inaccurate prediction in different per-bit SNR cases.

[5] By increasing the value of $\gamma_{\mathbf{u}}$, the probability of miss detection will be increased, since the condition, $\mathbf{1}_N^T \cdot \text{abs}(\mathbf{u}_k - \hat{\mathbf{u}}_k^{(N_{\max})}) \ge \gamma_{\mathbf{u}}$, is more difficult to be satisfied. However, those miss detected blocks will go through the conventional BP decoding process, which only increases the decoding complexity and does not affect the BLER performance.

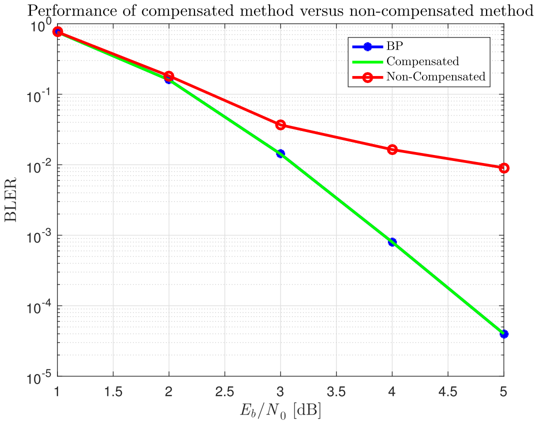


FIGURE 9. BLER versus per-bit SNR E_b/N_0 performance for different early stopping prediction schemes under the polar coding length $N = 1024$. As shown in this figure, the early stopping prediction with a BP based compensate part can achieve the same BLER performance with the original BP scheme, while the non-compensated scheme results in a BLER error flow.

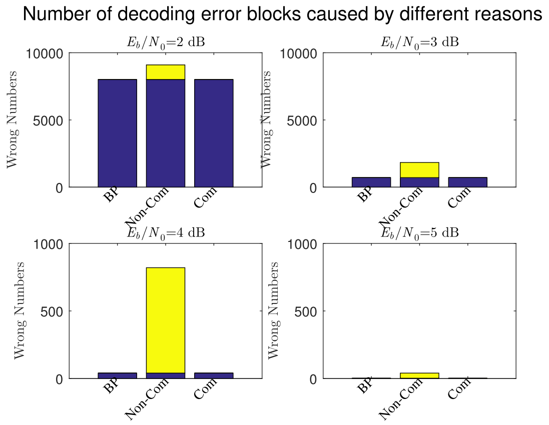


FIGURE 10. Number of block decoding errors for the coding length $N = 1024$ under different per-bit SNR values. As shown in this figure, insufficient BP iterations due to inaccurate $T_k^{\min}$ prediction lead to significant BLER performance degradation as shown in FIGURE 9.

In order to have an end-to-end performance point of view, we plot the BLER versus per-bit SNR curves for the traditional BP scheme with only $T_k^{\min}$ prediction (denoted as "non-compensated") and with compensated method as shown in FIGURE 9. As we can conclude from this figure, the major drawback of this scheme is the significant BLER performance degradation when E_b/N_0 exceeds 2 dB. In FIGURE 10, we provide an in-depth analysis on the block decoding errors. As we have explained before, the block decoding errors can come from conventional BP decoding errors (blue bar), directly claimed errors due to incorrect decodability detection (green bar),[6] and insufficient BP iterations due to inaccurate $T_k^{\min}$ prediction (yellow bar), and according to FIGURE 10, the last one becomes the major issue.

To deal with this issue, we propose some compensations as shown in FIGURE 11. First, by setting the label of the neural network as $T_k^{\min}$, we can obtain a trained DNN and then using the network to predict an iteration number $f_{TP}(\hat{\mathbf{b}}_k; \theta_{TP})$. The detailed parameters of the neural network is shown in Appendix B. As can be seen in FIGURE 7, many '-1' exist which does not meet the minimum required iteration number. Thus, the predicted iteration number can be added by one to ensure the decoding performance without significant overhead. Specifically, after $f_{TP}(\hat{\mathbf{b}}_k; \theta_{TP})$ rounds of BP iterations, CRC is then performed to understand the decoding result. If it fails to pass the CRC check, a compensation process is initiated, which contains $(N_{\max} - f_{TP}(\hat{\mathbf{b}}_k; \theta_{TP}))$ rounds of BP iterations. Through this compensation process, an entire $N_{\max}$ rounds of BP iterations are offered to decode this block, and a final round of CRC check is then used to complete the decoding process. We summarize the proposed low-complexity BP based decoding algorithm in Algorithm 1, and as shown in FIGURE 9, it eventually provides limited performance loss as compared with the traditional BP decoding performance with $N_{\max}$ rounds of iterations.

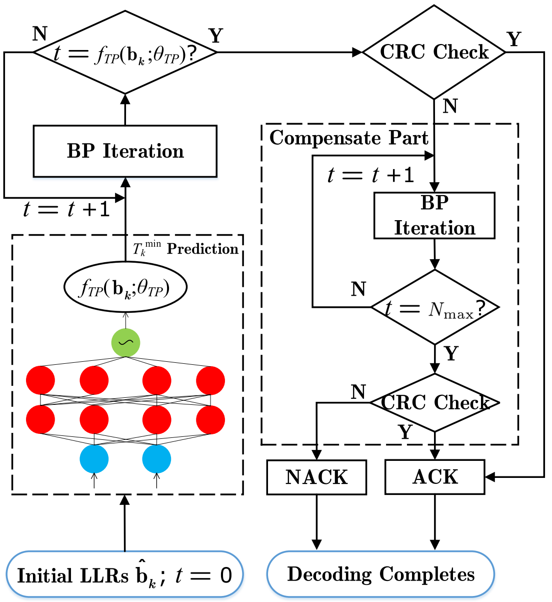


FIGURE 11. An overview of the entire early stopping prediction procedures with a BP based compensate part.

IV. COMPLEXITY ANALYSIS

In this section, we compare the decoding complexity of the proposed low-complexity BP based decoding scheme with three reference systems, including the traditional BP decoding [9] with $N_{\max}$ iterations, the *G-matrix* based scheme as well as the *MinLLR* based scheme [12]. We denote N_G, N_M, and N_p as the actual numbers of BP iterations required for the *G-matrix* based, the *MinLLR* based, and the low-complexity BP based decoding schemes, respectively. Also, as for the networks for the decodability detection and $T_k^{\min}$ prediction, we denote N_H as the number of nodes in the first hidden layer, whose number satisfies the empirical formula as shown in [24]. Although obtaining a closed-form expression for

[6] There are almost no directly claimed errors at these SNRs.

TABLE 1. Complexity comparison for different decoding schemes.

	Additions	Multiplications	Comparisons	Total
BP [9]	$2N_{\max}N(\log_2 N-1)$	$6N_{\max}N(\log_2 N-1)$	$2N_{\max}N(\log_2 N-1)$	$\mathcal{O}(N_{\max}N\log_2 N)$
G-matrix [12]	$2N_GN(\log_2 N-1)+2N_GN$	$6N_GN(\log_2 N-1)+N_GN^2$	$2N_GN(\log_2 N-1)+3N_GN$	$\mathcal{O}(N_GN^2)$
MinLLR [12]	$2N_MN(\log_2 N-1)+N_MN$	$6N_MN(\log_2 N-1)$	$2N_MN(\log_2 N-1)+2N_MN$	$\mathcal{O}(N_MN\log_2 N)$
Proposed	$7/4N_H+1+(2N_pN(\log_2 N-1)+7/4N_H+N_{\max})(1-\psi)$	$5/8N_H^2+NN_H+(6N_pN(\log_2 N-1)+5/8N_H^2+NN_H)(1-\psi)$	$7/4N_H+(2N_pN(\log_2 N-1)+7/4N_H)(1-\psi)$	$\mathcal{O}(N_pN\log_2 N)$

Algorithm 1 Overall Procedures for the Proposed Scheme

```
Input:
    Initial LLRs $\hat{\mathbf{b}}$;
    Predefined maximum iteration number $N_{\max}$.
Output:
    Decoded bits $\hat{\mathbf{u}}$;
Use $\hat{\mathbf{b}}$ to do the decodability detection $f_{DD}(\cdot)$ to obtain the decodability detection result.
Use the decodability detection result to do the $T^{\min}$ prediction $f_{TP}(\cdot)$ to obtain the predefined iteration number: $f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP})$;
for t = $1, 2, \ldots, f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP})$ do
    Update $\mathbf{l}_i^{(t)}$ and $\mathbf{r}_i^{(t)}$ based on Eq. (3);
end for
Use CRC criterion to check whether the decoding is successful;
if CRC check is met then
    Decoding is assumed to be successful and output $\hat{\mathbf{u}}$;
    Send ACK to the transmitter.
else
    for t = $f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP})+1, f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP})+2, \ldots, N_{\max}$ do
        Update $\mathbf{l}_i^{(t)}$ and $\mathbf{r}_i^{(t)}$ based on Eq. (3);
    end for
    Output $\hat{\mathbf{u}}$ based on $\mathbf{l}_1^{(t)}$;
    Do another CRC check and determine send ACK or NACK to the transmitter;
end if
```

them is mathematically intractable,[7] we provide the overall complexity in terms of addition, comparison and multiplication for the proposed scheme in the following lemma and compare with other decoding schemes in TABLE 1.[8]

Lemma 2: The overall complexity[9] of the proposed low-complexity BP based decoding scheme is given by,

$$\begin{aligned}&((7/2+N)N_H+5/8N_H^2)(2-\psi)\\&\quad+(10N_pN(\log_2 N-1)+1+N_{\max})(1-\psi)\end{aligned}\tag{10}$$

where ψ is the percentage of undecodable blocks predicted after decodability detection and $(1-\psi)$ is the percentage that actually needs to be decoded after decodability detection. Note that $f_{TP}(\hat{\mathbf{b}}_k;\theta_{TP}) \leq N_p \leq N_{\max}$ and the complexity scales with the order of $\mathcal{O}(N_pN\log_2 N)$.

Proof: Please refer to Appendix B for the proof. ■

From Lemma 2, we can observe that the complexity reduction of the proposed scheme mainly comes from the following two parts. The first part consits of directly claiming decoding errors without performing $N_{\max}$ rounds of BP iterations, and the second part is by eliminating the early stopping criteria computation for each round of BP iteration. It is worth mentioning that the size of neural networks in terms of the number of neurons for both decodability detection and $T_k^{\min}$ prediction does not scale with the block length N as well as the required number of BP iterations N_p. As summarized in TABLE 1, the early stopping criteria computations actually scale with the order of $\mathcal{O}(N_GN^2)$ and $\mathcal{O}(N_MN\log_2 N)$ for *G-matrix* and *MinLLR* [12] schemes, respectively, which grows significantly with the increase of the block length N and the required numbers of BP iterations, i.e, N_G and N_M. For our proposed low-complexity scheme, since the addition, multiplication, and comparison operations scale with the same order of $\mathcal{O}(N_pN\log_2 N)$, we conclude that the overall computational complexity scales with $\mathcal{O}(N_pN\log_2 N)$ as well.

[7] It has been reported in [12] that even for given observed LLR $\hat{\mathbf{b}}_k$, the actual number of BP iterations for the *G-matrix* and *MinLLR* based schemes are difficult to obtain.

[8] In TABLE 1, the total computational complexity scales with the highest order of addition, multiplication, and comparison operations.

[9] In the complexity derivation, we treat the addition, multiplication, and comparison operations to be the same for simplicity, since the incurred delays of the above three operations in our evaluation environment (e.g., Matlab2016a) are nearly the same, and we believe the extension to the scenario when three operations incur different computational complexities is straight forward.

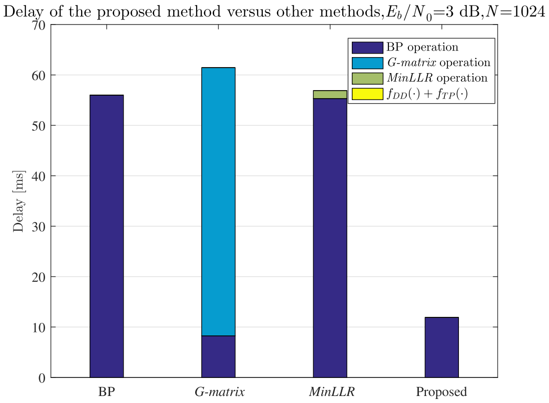


FIGURE 12. The decoding delay comparison for different low-complexity decoding scheme. As shown in this figure, the proposed scheme achieves more than 70% delay reduction if compared with other baseline schemes.

Based on the above analysis, we now compare the decoding delays[10] for different schemes in FIGURE 12, where $N = 1024$ and $E_b/N_0 = 3$ dB. The average decoding delay can be classified as follows, e.g., the normalized BP operation, *G-matrix*, and *MinLLR* operations, as well as the operations for the decodability detection and the early stopping prediction. As shown in FIGURE 12, in terms of BP operations, the delay of the original BP algorithm contains the whole BP operation and leads to a high decoding delay. The *G-matrix* method requires relative fewer BP iterations but it takes more time to calculate the matrix multiplication and also results in a high decoding delay. The *MinLLR* needs more BP iterations and leads to a high decoding delay. However, our proposed low-complexity method only needs the operations for the decodability detection and the $T_k^{\min}$ prediction and N_p times of BP iterations, which shows a lower decoding delay compared with other methods. Note that the DNN has a highly parallelized structure, the delay for $f_{DD}(\cdot)$ and $f_{TP}(\cdot)$ is very low. More details on decoding delay will be presented in Section V.

V. EXPERIMENTAL RESULTS

In this section, we provide several numerical examples to demonstrate the effectiveness of the proposed low-complexity BP based decoding scheme. TABLE 2 lists the detailed experiment setup.[11] In the following evaluation, we introduce the empirical results from three different aspects, including the effect of decodability detection, the effect of early stopping prediction, and the overall decoding delay compared with traditional methods.

A. EFFECT OF DECODABILITY DETECTION

The prediction accuracy ϕ of the decodability detection by our proposed scheme is plotted in FIGURE 13 under different code lengths and SNRs.

TABLE 2. Detailed experiment setups.

Training Platform	Keras
Code Rate R	0.5
Modulation	BPSK
Channel Model	AWGN
Training Set	2×10^6 samples
Testing Set N_{BL}	5×10^4 samples
Hidden Layers	$(128, 64, 32)$
Loss function	MSE
Learning Rate	10^{-3}
Epoches	10
Optimizer	Adam
Dropout	None

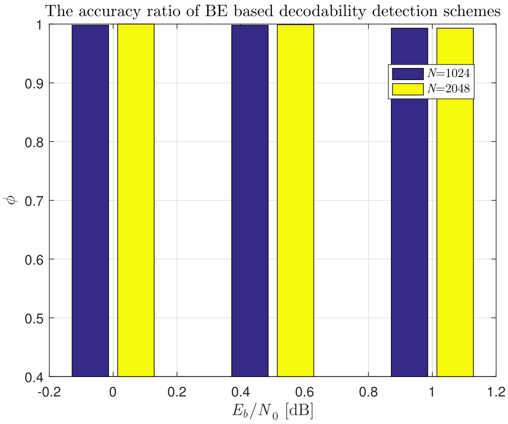


FIGURE 13. Prediction accuracy versus per-bit SNR E_b/N_0 performance for different coding length $N = 1024$ and 2048. The pre-defined threshold γ_u is chosen to be $N/2$ and the BE based formulation is adapted.

We use the BE based method mentioned in section III-B to understand the performance as well as the suitable code length of the decodability detection and the results are shown in FIGURE 13. It is shown that with the relative large code length, the accuracy ϕ of decodability detection can be very high. The reason is that the number of $\hat{\mathbf{b}}$ is relative sufficient so that enough information can be well learned to get the mapping relationship between $\hat{\mathbf{b}}$ and $\mathcal{I}\left(\mathbf{u}_k \neq \hat{\mathbf{u}}_k^{(N_{\max})}\right)$. On the contrary, we find that when the code length is relatively small (≤ 1024), the accuracy ϕ of the detection is relatively low due to the insufficient information bits for the neural network to learn.

In our experiment, we also find that not all situations are suitable for decodability detection. For example, when the SNR increases, the average BER in each block decreases so that there is not many labeled undecodable blocks when using BE based decodability detection method. So, the relative low SNR (lower than 1dB in this paper) is suitable for the

[10] In the following evaluation, we directly use the decoding delay as the complexity measure since the processing delays for additions, multiplications, and comparisons in our evaluation environment are nearly the same with less than 10% variations.

[11] We use default settings in Keras platform and did not apply other early stopping techniques in the training stage.

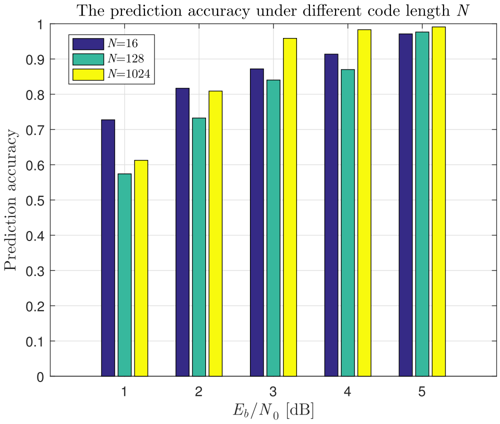


FIGURE 14. Prediction accuracy versus per-bit SNR E_b/N_0 performance under different coding length $N = 16$, 128 and 1024. As shown in this figure, the prediction accuracy improves as the per-bit SNR increases.

decodability detection. And when the SNR goes higher we don't need this step at all.

B. EFFECT OF EARLY STOPPING PREDICTION

After the decodability detection process, we use the early stopping prediction part to predict the iteration number of the remaining blocks and give a compensate method if necessary. The accuracy and BLER performance of the early stopping prediction are given as follows.

1) ACCURACY

FIGURE 14 shows the accuracy of the $T_k^{\min}$ prediction, which is defined as the ratio of the number of blocks correctly judged by the CRC check to the total number of simulated blocks N_{BL} by using the $T_k^{\min}$ prediction.

From FIGURE 14, we can observe that the accuracy increases as SNR increases given the code length N. This is because with a small SNR, the number required for early stopping $f_{TP}(\hat{\mathbf{b}}_k; \theta_{TP})$ increases and the fluctuation of $T_k^{\min}$ increases. On the other hand, at low SNR region, the codeword may not be decoded no matter how many iterations the BP operates. With a large SNR, the DNN can predict the number of correct iterations $T_k^{\min}$ with a large probability as $T_k^{\min}$ is small and uniform in this case.

2) BLER

BLER is an important indicator to show system reliability. FIGURE 15 compares BLER of different methods under different SNRs. It can be seen from FIGURE 15 that our proposed scheme can achieve a similar BLER performance compared with traditional BP [9], *G-matrix*, and *MinLLR* [12] when the code length is 16, 128, and 1024. Specifically, in the case of $N = 1024$ and SNR $= 1$ dB, the decodability detection can correctly identify "Type I block errors" with no performance degradation if compared with the conventional BP decoding method.

The reason of this phenomenon is as follows. First, in the decodability detection stage, we apply the BE based formulation to achieve a highly reliable decodability prediction result with marginal performance degradation as explained in Appendix A. Second, in the early stopping prediction stage, we propose to use the compensation mechanism as explained in Section section III-C to deal with the incorrect prediction cases. Therefore, combining the above two effects, the proposed scheme can achieve a similar BLER performance if compared with traditional BP decoding algorithms.

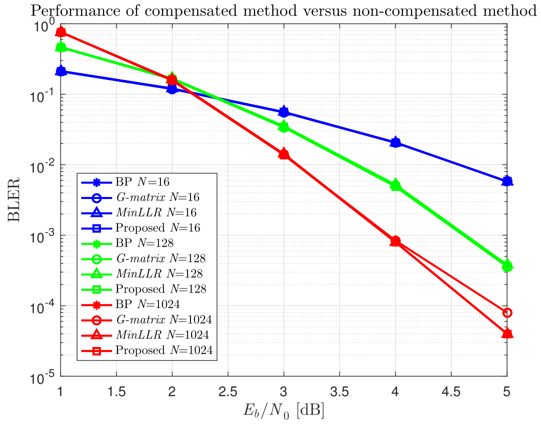


FIGURE 15. BLER versus per-bit SNR E_b/N_0 performance for different low-complexity decoding schemes under different coding lengths $N = 16$, 128 and 1024. As shown in this figure, the proposed scheme achieves the similar BLER performance as the conventional BP scheme [9] and the other low-complexity decoding schemes, including *G-matrix*, and *MinLLR* [12].

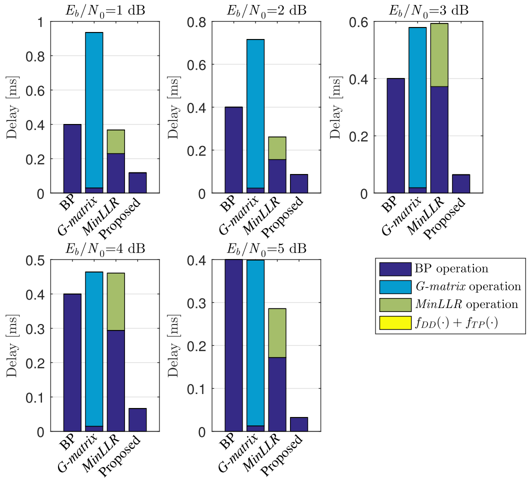


FIGURE 16. Decoding delay about our scheme and conventional BP [9], *G-matrix*, and *MinLLR* [12] decoders, $N = 16$.

C. OVERALL DECODING DELAY

FIGURE 16, FIGURE 17, and FIGURE 18 depict the decoding delay of different methods.

We can see that our method can realize a lower decoding delay compared with other three methods. Note that when $N = 1024$ and $E_b/N_0 = 1$ dB, the ψ in TABLE 1 created by the decodability detection is about 0.01 and can save around 1% of decoding delay. The time consumed by $f_{DD}(\hat{\mathbf{b}}_k;\theta_{DD})$ and $f_{TP}(\hat{\mathbf{b}}_k; \theta_{TP})$ is ignorable since they are executed by the highly parallelized neural network.

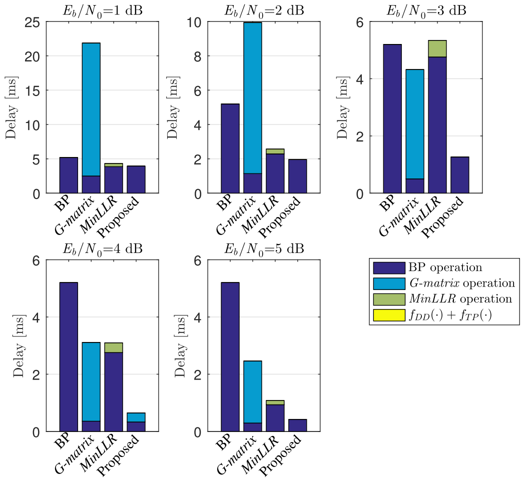


FIGURE 17. Decoding delay about our scheme and conventional BP [9], *G-matrix*, and *MinLLR* [12] decoders, $N = 128$.

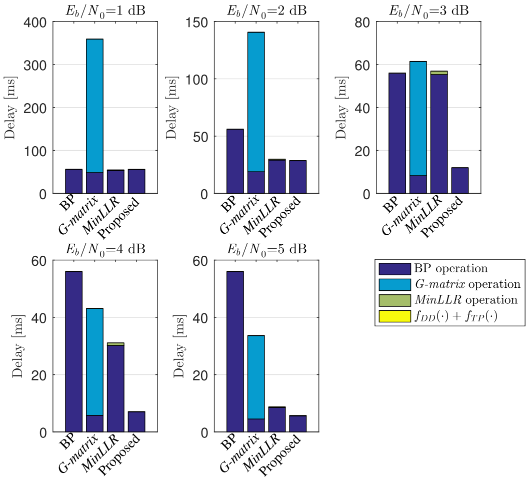


FIGURE 18. Decoding delay about our scheme and conventional BP [9], *G-matrix*, and *MinLLR* [12] decoders, $N = 1024$.

It can be concluded that the proposed method can not only guarantee the performance without deterioration but also achieve similar or lower time consumption compared with existing decoding methods.

VI. CONCLUSION

In summary, we proposed a low-complexity belief propagation based decoding scheme for polar codes. By detecting the decodable codewords and predicting the iteration number from the received signal, we can claim error directly and decode using the predicted number directly without any judgment, and the performance loss is compensated by BP properly. Results show that the decodability detection can detect some error blocks accurately and the BLER performance of our method is the same compared with traditional methods. Meanwhile, the overall decoding delay is lower than that of traditional methods.

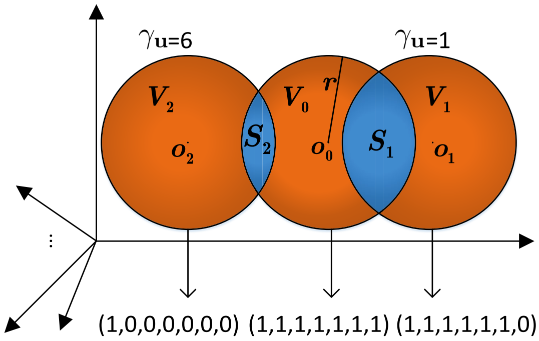


FIGURE 19. An example ($K = 7$) of different hyperspaces with the intersecting space represents their same LLR distribution, where $\gamma_u = 6$ for V_2 and $\gamma_u = 1$ for V_1.

APPENDIX A
PROOF OF LEMMA 1

To prove Lemma 1, we use K-dimensional hyperspaces to represent the LLR distribution in the training stage. As illustrated in FIGURE 19, we denote $\mathbf{V}_n$ as a set of K-dimensional hyperspaces, as given by

$$\{\mathbf{V}_n \in \mathbf{R}^K | r_{V_k,o_k} < r_{\max}\}(0 < n < 2^K) \tag{11}$$

Here, where $\mathbf{V}_n := \{V_0, V_1, \ldots, V_{2^K}\}, K = 1, 2, \cdots,$ denotes the set of all possible LLR hyperspaces with information bits of K. o_k is the center of V_k, which represents the possible codeword in the noise-free state. Due to the influence of noise, the hyperspace V_k expands from the center o_k and reforms to a sphere with radius r_{V_k,o_k}, which cannot exceed a maximum value $r_{\max}$. Note that $\mathbf{V}_n$ is the space that the neural networks can recognize.

Suppose that the center $o_0(1, 1, 1, 1, 1, 1, 1)$ of V_0 is the correct codeword to be decoded, see FIGURE 19. The centers of hyperspaces V_1 and V_2 beside V_0 are $o_1(1, 1, 1, 1, 1, 1, 0)$ and $o_2(1, 0, 0, 0, 0, 0, 0)$, respectively. We denote the intersecting spaces of V_0 and the other two hyperspaces as

$$\begin{cases} \mathbf{S}_1 = \mathbf{V}_0 \cap \mathbf{V}_1 \\ \mathbf{S}_2 = \mathbf{V}_0 \cap \mathbf{V}_2. \end{cases} \tag{12}$$

S_1 and S_2 represent the similar LLR distribution between the correct codeword o_0 and wrong codeword o_1 or o_2. We can conclude that the more space V_0 intersects with V_1 or V_2, the lower prediction accuracy of the neural networks is. Note that for V_1, the number of different bits of the codeword from V_0, denoted as γ_u, is 1. It means the centers of the two hyperspace is very close and the probability of wrong prediction is $\frac{\upsilon(S_1)}{\upsilon(V_0)}$ (denote $\upsilon(\cdot)$ as the volume function). For V_2, as γ_u increases to 6, the distance of the centers of V_0 and V_2 increases. In turn the probability of wrong prediction decreases to $\frac{\upsilon(S_2)}{\upsilon(V_0)}$. In other words, the prediction accuracy of the neural network shows a monotonically non-decreasing relation with respect to γ_u.

TABLE 3. DNN for both strategies.

Layers	Input	Hidden1	Hidden2	Hidden3	Output
Size (Decodability detection)	N	N_H	$N_H/2$	$N_H/4$	1
Size ($T_k^{\min}$ prediction)	N	N_H	$N_H/2$	$N_H/4$	$N_{\max}$
Activation Function (Decodability detection)	ReLU	ReLU	ReLU	ReLU	Sigmoid
Activation Function ($T_k^{\min}$ prediction)	ReLU	ReLU	ReLU	ReLU	Softmax

APPENDIX B
PROOF OF LEMMA 2

In the training part, both decodability detection and $T_k^{\min}$ prediction use a three-hidden layer DNN to represent the non-linear relationship of decodability detection and $T_k^{\min}$ prediction. We illustrate the two networks parameters in TABLE 3, where N_H is set to 128.

In the fully-connected networks, since the output of one neuron can be formulated as $\sigma(\sum w_i d_i + w_b)$, where σ, w_i, d_i, w_b denote the activation function, network weight, input and bias, respectively. This function needs several add, multiplications and comparison operations. In our experiment, we use three hidden layers $(N_H, N_H/2, N_H/4)$ to map the two tasks $f_{DD}(\hat{\mathbf{b}}_k; \theta_{DD})$ and $f_{TP}(\hat{\mathbf{b}}_k; \theta_{TP})$.

For the add and multiplication operations, as is shown in the output of neuron, the multiplication will be executed several times and the bias will be added after every neuron. Thus, the total number of the add operation is $7/4N_H + 1$ for the $f_{DD}(\hat{\mathbf{b}}_k; \theta_{DD})$ and $7/4N_H + N_{\max}$ for $f_{TP}(\hat{\mathbf{b}}_k; \theta_{TP})$. For the multiplication, both the two methods need $5/8N_H^2 + NN_H$ times.

For the comparison operation, since the activation function of the hidden layers of the two networks is Rectified Linear Unit (ReLU) whose output can be formulated as $\max\{0, x\}$ and this function can be treated as a comparison. Thus, the total number of comparison operation is $7/4N_H$ for both two methods.

Through the above calculation, the total operation numbers of the proposed method can be written as Eq. (10) by adding the corresponding BP operations.

REFERENCES

[1] 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, Jul. 2009.

[2] N. Reno, *Final Report of 3GPP TSG RAN1 #86bis v1.0.0*, document R1-1611081, 3GPP, Nov. 2016.

[3] B. Sklar, "A primer on turbo code concepts," *IEEE Commun. Mag.*, vol. 35, no. 12, pp. 94–102, Dec. 1997.

[4] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul./Aug. 2011, pp. 1–5.

[5] K. Niu and K. Chen, "Stack decoding of polar codes," *Electron. Lett.*, vol. 48, no. 12, pp. 695–697, Jun. 2012.

[6] C. Zhang and K. K. Parhi, "Low-latency sequential and overlapped architectures for successive cancellation polar decoder," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2429–2441, May 2013.

[7] C. Zhang, B. Yuan, and K. K. Parhi, "Reduced-latency SC polar decoder architectures," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 3471–3475.

[8] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," in *Proc. 8th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Nov. 2011, pp. 437–441.

[9] B. Yuan and K. K. Parhi, "Architecture optimizations for BP polar decoders," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 2654–2658.

[10] E. Arıkan, "A performance comparison of polar codes and Reed–Müller codes," *IEEE Commun. Lett.*, vol. 12, no. 6, pp. 447–449, Jun. 2008.

[11] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5165–5179, Oct. 2015.

[12] 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, Dec. 2015.

[13] C. Simsek and K. Turk, "Simplified early stopping criterion for belief-propagation polar code decoders," *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1515–1518, Aug. 2016.

[14] Y. Ren, C. Zhang, X. Liu, and X. You, "Efficient early termination schemes for belief-propagation decoding of polar codes," in *Proc. IEEE 11th Int. Conf. ASIC (ASICON)*, Nov. 2015, pp. 1–4.

[15] T. Gruber, S. Cammerer, J. Hoydis, and S. T. Brink, "On deep learning-based channel decoding," in *Proc. IEEE Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.

[16] S. Cammerer, T. Gruber, J. Hoydis, and S. ten Brink, "Scaling deep learning-based decoding of polar codes via partitioning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[17] W. Xu, Z. Wu, Y.-L. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2017, pp. 1–6.

[18] B. Dai, R. Liu, and Z. Yan, "New min-sum decoders based on deep learning for polar codes," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2018, pp. 252–257.

[19] Y. Wang, Z. Zhang, S. Zhang, C. Shan, and S. Xu, "A unified deep learning based polar-LDPC decoder for 5G communication systems," in *Proc. IEEE 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Hangzhou, China, Oct. 2018, pp. 1–6.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

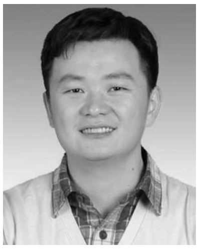
[21] C.-F. Teng, C.-H. D. Wu, A. K.-S. Ho, and A.-Y. A. Wu, "Low-complexity recurrent neural network-based polar decoder with weight quantization mechanism," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1413–1417.

[22] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, and S. Venkatraman, "Robust intelligent malware detection using deep learning," *IEEE Access*, vol. 7, pp. 46717–46738, 2019.

[23] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[24] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 274–281, Mar. 2003.

YAOHAN WANG received the B.E. degree from Hebei University, Baoding, China, in 2017. He is currently pursuing the master's degree in information and communication engineering with Shanghai University, China. His research fields include polar codes, machine learning, deep learning, and rate matching technique in the PHY layer.



SHUNQING ZHANG (S'05–M'09–SM'14) received the B.S. degree from the Department of Microelectronics, Fudan University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, in 2009.

He was with the Communication Technologies Laboratory, Huawei Technologies, as a Research Engineer and then a Senior Research Engineer, from 2009 to 2014, and a Senior Research Scientist of Intel Collaborative Research Institute on Mobile Networking and Computing, Intel Labs, from 2015 to 2017. Since 2017, he has been with the School of Communication and Information Engineering, Shanghai University, Shanghai, China, as a Full Professor. He has published over 60 peer-reviewed journals and conference papers, and over 50 granted patents. His current research interests include energy efficient 5G/5G of communication networks, hybrid computing platform, and joint radio frequency and baseband design. He has received the National Young 1000-Talents Program and received the paper award for Advances in Communications from IEEE Communications Society, in 2017.

CHUAN ZHANG (S'07–M'13) received the B.E. degree *(summa cum laude)* in microelectronics and the M.E. degree (Hons.) in verylarge scale integration (VLSI) design from Nanjing University, Nanjing, China, in 2006 and 2009, respectively, and the M.S.E.E. and Ph.D. degrees from the Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities (UMN), USA, in 2012. He is currently the Excellence Professor and the Purple Mountain Professor with Southeast University. He is also with the LEDAS, National Mobile Communications Research Laboratory, Quantum Information Center of Southeast University, and the Purple Mountain Laboratories, Nanjing, China. His current research interests include low-power high-speed VLSI design for digital signal processing and digital communication, bio-chemical computation and neuromorphic engineering, and quantum communication. Dr. Zhang is also a member of the Seasonal School of Signal Processing and Design and Implementation of Signal Processing Systems TC of the IEEE Signal Processing Society, and Circuits and Systems for Communications TC, VLSI Systems and Applications TC, and Digital Signal Processing TC of the IEEE Circuits and Systems Society. He received two Best (Student) Paper Awards of the IEEE International Conference on ASIC, in 2015 and 2017, respectively, the Best Paper Award Nomination of the IEEE Workshop on Signal Processing Systems, in 2015, the Best Paper Award, in 2016, the Best (Student) Paper Award of the IEEE International Conference on DSP in 2016, the three Excellent Paper Awards and two Excellent Poster Presentation Awards of the International Collaboration Symposium on Information Production and Systems, from 2016 to 2018, the Outstanding Achievement Award of the Intel Collaborative Research Institute, in 2018, the Best Contribution Award of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), in 2018, and the Merit (Student) Paper Award of the IEEE APCCAS, in 2008. He also received the Three-Year University-Wide Graduate School Fellowship of UMN and the Doctoral Dissertation Fellowship of UMN. He serves as an Associate Editor for the IEEE Transactions on Signal Processing. He is also the Secretary-Elect of the Circuits and Systems for Communications TC of the IEEE Circuits and Systems Society.

XIAOJING CHEN received the B.E. degree in communication science and engineering and the Ph.D. degree in electromagnetic field and microwave technology from Fudan University, China, in 2013 and 2018, respectively, and the Ph.D. degree in engineering from Macquarie University, Australia, in 2019. She is currently a Lecturer with Shanghai University, China. Her research interests include wireless communications, energy-efficient communications, stochastic network optimization, and network functions virtualization.

SHUGONG XU (M'98–SM'06–F'16) was graduated from Wuhan University, China, in 1990. He received the master's degree in pattern recognition and intelligent control and the Ph.D. degree in EE from the Huazhong University of Science and Technology (HUST), China, in 1993 and 1996, respectively. Prior to joining Huawei in 2008, he was with Sharp Laboratories of America as a Senior Research Scientist. Before joining Intel in September 2013, he was a Research Director and a Principal Scientist at the Communication Technologies Laboratory, Huawei Technologies. Among his responsibilities at Huawei, he founded and directed Huawei's green radio research program, Green Radio Excellence in Architecture and Technologies (GREAT). He was the center Director and a Intel Principal Investigator of the Intel Collaborative Research Institute for Mobile Networking and Computing (ICRI-MNC), prior to December 2016 when he joined Shanghai University. He is currently a Professor with Shanghai University, the Head of the Shanghai Institute for Advanced Communication and Data Science (SICS). Before that, he conducted research as a Research Fellow in City College of New York, Michigan State University, and Tsinghua University. He published over 100 peer-reviewed research articles in top international conferences and journals. One of his most referenced articles has over 1400 Google Scholar citations, in which the findings were among the major triggers for the research and standardization of the IEEE 802.11S. He has over 20 U.S. patents granted. Some of these technologies have been adopted in international standards, including the IEEE 802.11, 3GPP LTE, and DLNA. His current research interests include wireless communication systems and machine learning. Dr. Xu was awarded 'National Innovation Leadership Talent' by China Government in 2013, was elevated to IEEE Fellow in 2015 for contributions to the improvement of wireless networks efficiency. He is also the winner of the 2017 Award for Advances in Communication from IEEE Communications Society. He was also the Chief Scientist and PI for the China National 863 project on End-to-End Energy Efficient Networks. He was one of the co-founders of the Green Touch consortium together with Bell Labs etc., and he served as the Co-Chair for the Technical Committee for three terms in this international consortium.