Shortened PAC Codes and List Decoding

LIU Aolin[1,2], FENG Bowen[1], LIANG Chulong[3], XU Jin[3], ZHANG Qinyu[1,2]
(1. Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China;
2. Pengcheng Laboratory, Shenzhen 518055, China;
3. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202504010
https://kns.cnki.net/kcms/detail/34.1294.TN.20251205.1832.002.html, published online December 08, 2025
Manuscript received: 2024-08-14

Abstract: Shortening is a standard rate-matching method for polar codes in wireless communications. Since polarization-adjusted convolutional (PAC) codes also have a block length limited to the integer powers of two, they also require rate-matching. To this end, we first analyze the limitations of existing shortening patterns for PAC codes and explore their feasibility. Subsequently, we propose a novel shortening scheme for PAC codes based on list decoding, where the receiver is allowed to treat the values of the deleted bits as undetermined. This approach uses a specialized PAC codeword and activates multiple decoding paths during the initialization of list decoding, enabling it to achieve the desired reliability.

Keywords: polarization-adjusted convolutional codes; rate-matching; shortened codes; list decoding

Citation (Format 1): LIU A L, FENG B W, LIANG C L, et al. Shortened PAC codes and list decoding [J]. ZTE Communications, 2025, 23(4): 86–96. DOI: 10.12142/ZTECOM.202504010
Citation (Format 2): A. L. Liu, B. W. Feng, C. L. Liang, et al., "Shortened PAC codes and list decoding," *ZTE Communications*, vol. 23, no. 4, pp. 86–96, Dec. 2025. doi: 10.12142/ZTECOM.202504010.

1 Introduction

Polar coding is a high-reliability channel coding technique[1], which has been widely used for control information in 5G communication systems. In wireless communications, rate-matching is a key step for polar codes to adjust the block length and code rate to fit actual requirements, since the block length of polar codes follows $N = 2^n$. In 3GPP TS 38. 212[2], puncturing or shortening is applied to polar codes when the required block length E satisfies $N/2 < E < N$. The standard specifies the puncturing when the actual required code rate $K/E \leqslant 7/16$. In this case, $(N - E)$ bits are deleted from the mother codeword. These punctured bits are unknown to the receiver and can be considered as bits transmitted over zero-capacity channels. In contrast, the shortening of polar codes is applied when the actual required code rate $K/E > 7/16$. When constructing a shortened polar codeword, $(N - E)$ bits are deleted and their values are fixed by freezing the correlated source bits or by applying dynamic frozen bits (DFBs). Therefore, the corresponding $(N - E)$ sub-channels can be considered as one-capacity channels, and the shortening is also known as capacity-one puncturing, as described in Ref. [3].

Polarization-adjusted convolutional (PAC) coding is a representative pre-transformed polar coding scheme[4]. As shown in Ref. [5], with sequential or list decoding, PAC codes at short block lengths can achieve better block error rate (BLER) performance than their underlying polar codes. The construction of PAC codes involves rate-profiling and convolutional pre-transform prior to the code-rate-1 polar coding. PAC codes can be viewed as a concatenated coding scheme with polar coding as the inner code. Hence, the block length of PAC codewords is also $N = 2^n$. Consequently, rate-matching is also required for PAC codes. However, the pre-transform enables the information bits of PAC codes to utilize more sub-channels[6], thereby providing a concatenated gain. This makes it difficult to obtain capacity-one sub-channels for PAC codes via the conventional one-to-one bit freezing used in shortened polar codes. To implement shortened PAC codes without using DFBs, a sufficient number of one-capacity sub-channels must be reserved and cannot be allocated to information bits. This reservation reduces the channel utilization of the resulting PAC codes, thus diminishing the gain provided by the pre-transform and causing a performance loss.

In this paper, we focus on the construction of shortened

This work was supported in part by the National Natural Sciences Foundation of China (NSFC) under Grant Nos. 62301185 and 62027802, in part by ZTE Industry-University-Institute Cooperation Funds, and in part by The Major Key Project of PCL under Grant No. PCL2024A01.

PAC codes and the corresponding list decoding.

First, we discuss the limitations in constructing shortened PAC codes. We present three conventional approaches for constructing shortened PAC codes, by referencing the shortening schemes of polar codes. The first approach allocates more well-polarized locations to frozen bits, thereby obtaining enough one-capacity sub-channels. The second approach partially relaxes the pre-transform constraint to free sufficient one-capacity sub-channels. The third approach employs DFBs to ensure the deleted bits remain fixed.

Second, we propose a novel scheme for constructing shortened PAC codes by exploiting the multi-path nature of list decoding. Unlike conventional schemes where the deleted bits are restricted to fixed values, the values of the deleted bits are considered valid in at least one path during the initial stage of list decoding. In this case, the corresponding sub-channels can be regarded as conditional one-capacity channels. To aid decoding, several bits are frozen and used to carry pre-designed bits that depend on the values of the deleted bits.

Third, the corresponding list decoding is introduced, where multiple decoding paths are activated to accommodate the results derived from various possible combinations of values for the deleted bits. The corresponding values of the pre-designed bits are then calculated and assigned to each path.

With the slightly modified list decoding, the proposed shortened PAC codes achieve better BLER performance than comparable shortened or punctured polar codes.

2 Preliminaries

2.1 Rate-Matching of Polar Codes

The polar codeword with block-length $N = 2^n$ is generated by $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}_N$[1], where the generator matrix $\boldsymbol{G}_N = \boldsymbol{F}_2^{\otimes n}\boldsymbol{B}_N$, in which the involutory matrix $\boldsymbol{B}_N$ refers to bit-reversal operation and $\boldsymbol{F}_2^{\otimes n}$ refers to the n-order Kronecker product of the kernel matrix $\boldsymbol{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$.

The puncturing of polar codes has been extensively studied. Among the notable works, Ref. [7] introduces the quasi-uniform puncturing (QUP) for polar codes. The indices in the QUP pattern $\mathcal{P}_{\text{QUP}} = \beta_N(\mathcal{U}_{N-E})$ are located quasi-uniformly in the mother codeword. The reversal function is $\beta_{2^\eta}(\zeta) = \mathcal{U}_{2^\eta}\boldsymbol{B}_{2^\eta}(\mathcal{U}_{2^\eta}, \zeta)$. In Ref. [8], a worst quality puncturing (WQP) pattern is formed by the indices of the least reliable sub-channels. In contrast, Ref. [9] selects puncturing positions based on columns with the minimum column weight (MCW). A low-complexity method in Ref. [10] provides a bit-reversed QUP (BRP) pattern, given by $\mathcal{P}_{\text{BRP}} = \mathcal{U}_{N-E}$. Moreover, the changes in sub-channels' reliability caused by puncturing are discussed. A mandatory frozen subset $\beta_N(\mathcal{U}_{N-E}) \subseteq \mathcal{A}^c$ is provided to avoid transmitting information bits over zero-capacity sub-channels. Based on the changes in sub-channels' reliability, an information set approximation puncturing (ISAP) pattern is proposed for low-code-rate codewords[11], where certain guard bits are set as unpuncturable. Furthermore, Ref. [12] defines a minimum reliability score (MMRS) as the objective and devises a puncturing pattern to maximize this score. The MMRS pattern performs particularly well for high-code-rate codewords.

Research on shortening of polar codes has also yielded several important schemes. In Ref. [13], a criterion is established to obtain an $E \times E$ reduced-dimensional generator matrix by searching for columns with weight 1. The shortening pattern $\mathcal{S}$ consists of the indices of the deleted $(N - E)$ columns of the generator matrix. The index set of the deleted rows is called the mandatory frozen subset, denoted as $\mathcal{F}_{\text{polar}}$ with $\mathcal{F}_{\text{polar}} \subseteq \mathcal{A}^c$. This subset ensures the existence of $(N - E)$ one-capacity sub-channels. Since the unreversed generator matrix $\boldsymbol{F}_2^{\otimes n}$ is a lower-triangular matrix, Ref. [13] also provides a simple shortening pattern by setting the mandatory frozen subset as $\mathcal{F}_{\text{polar}} = \{E + 1, E + 2, \cdots, N\}$. Ref. [14] adapts the method in Ref. [13] and provides fundamental insights into the shortening of PAC systematic codes. In Ref. [15], the shortening pattern and the frozen set are jointly optimized to achieve lower bit-error probability, and the DFBs are applied to part of the bits of $u_{\mathcal{F}}$ to ensure that the values of the deleted bits are fixed. In Ref. [9], a low-complexity method is introduced to select both puncturing and shortening patterns, where a WANG-LIU bit-reversed shortening (BRS) pattern is used. The same pattern is also presented in Ref. [16]. In Refs. [17] and [18], a shortening pattern is introduced based on evaluating the reliability of sub-channels via Gaussian approximation (GA). Ref. [19] redesigns a subblock-wise interleaver to improve rate-matching performance, based on an analysis of the locations of zero-capacity and one-capacity sub-channels.

2.2 PAC Codes and List Decoding

In PAC codes, a convolutional transform is employed as a precoding stage before the polar transform, providing a significant concatenated gain[20–21]. The construction proceeds in two steps. First, the information bit vector $\boldsymbol{d}$ is mapped to a data carrier word $\boldsymbol{v} = \{\boldsymbol{v}_{\mathcal{A}}, \boldsymbol{v}_{\mathcal{A}^c}\}$ via rate-profiling, i.e., $\boldsymbol{v}_{\mathcal{A}} = \boldsymbol{d}$ and $\boldsymbol{v}_{\mathcal{A}^c} = \boldsymbol{0}$. Second, the interim codeword $\boldsymbol{u}$ is obtained by the convolutional pre-transform $\boldsymbol{u} = \boldsymbol{v}\boldsymbol{T}_N$. $\boldsymbol{T}_N = (t_{a,b})_{a,b \in \mathcal{U}_N}$ is the N-dimension generator matrix of convolutional codes, expanded from $\boldsymbol{c} = (c_1, c_2, \cdots, c_\varsigma)$, where

$$\begin{cases} t_{a,b} = c_{b-a+1}, & \text{if } a \leqslant b < a + \varsigma \\ t_{a,b} = 0, & \text{otherwise} \end{cases} \quad (1),$$

and where $\mathcal{U}_N$ refers to the integer sequence $\mathcal{U}_N = \{1, 2, \cdots, N\}$. Therefore, each interim bit can obtained by

$$u_i = \sum_{j=1}^{\varsigma} c_j v_{i-j+1} \quad (2).$$

Finally, the PAC codeword $\boldsymbol{x}$ is obtained by $\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}_N$[4].

In the list decoding of PAC codes, the Log-Likelihood Ratio (LLR) corresponding to bit u_i, $i \in \mathcal{U}_N$ on each decoding path is expressed as $L_i^n[\ell] = \ln \frac{\Pr\left(\boldsymbol{y}, \hat{\boldsymbol{u}}_{\mathcal{U}_{i-1}}[\ell] | \hat{u}_i[\ell] = 0\right)}{\Pr\left(\boldsymbol{y}, \hat{\boldsymbol{u}}_{\mathcal{U}_{i-1}}[\ell] | \hat{u}_i[\ell] = 1\right)}$, $\ell \in \mathcal{L}$, where $\boldsymbol{y}$ refers to the received symbols of the PAC codeword, and $\hat{u}_i[\ell]$ is obtained from Eq. (2). This LLR is derived according to the recursive formulas in Ref. [1], given by

$$\begin{cases} L^m_{2^{p+1}q+p}[\ell] = L^{m-1}_{2^{p+1}q+2p-1}[\ell] \boxplus L^{m-1}_{2^{p+1}q+2p}[\ell] \\ L^m_{2^p(2q+1)+p}[\ell] = \nabla[\ell] \cdot L^{m-1}_{2^{p+1}q+2p-1}[\ell] + L^{m-1}_{2^{p+1}q+2p}[\ell] \end{cases} \quad (3),$$

where the operation $a \boxplus b = 2\tanh^{-1}\left(\tanh(a/2)\tanh(b/2)\right)$, and the parameters are $m \in \mathcal{U}_n$, $p \in \mathcal{U}_{2^p}$, $q \in \mathcal{U}_{2^{m-1}} - 1$, $2^p = 2^{n-m}$ and $\nabla[\ell] = (-1)^{\hat{\boldsymbol{u}}_{q \cdot 2^{p+1} + \mathcal{U}_{2^p}}[\ell] \boldsymbol{G}_{2^p}(\mathcal{U}_{2^p}, p)}$. Path-metric (PM) is used to evaluate the reliability of each candidate path, which is expressed as:

$$PM_{\hat{v}_i[\ell]} = \sum_{j=1}^{i} BM_{\hat{v}_j[\ell]} = \sum_{j=1}^{i} \ln\left(1 + e^{-(1-2\hat{u}_j[\ell])L_j^n[\ell]}\right) \quad (4),$$

where $BM_{\hat{v}_i[\ell]}$ refers to the branch metric (BM) of the path ℓ at the i-th bit. Up to L paths with smaller PMs are retained.

3 Encoding and Decoding Schemes for Shortened PAC Codes

3.1 Limitations in Construction of Shortened PAC Codes

The unreversed generator matrix of PAC codes is $\boldsymbol{H}_N = \boldsymbol{T}_N \boldsymbol{F}_2^{\otimes n}$, which can be expressed as:

$$\boldsymbol{H}_N = \begin{bmatrix} \boldsymbol{A}_{(N-\varsigma+1)\times(\varsigma-1)} & \boldsymbol{C}_{(N-\varsigma+1)\times(N-\varsigma+1)} \\ \boldsymbol{D}_{(\varsigma-1)\times(\varsigma-1)} & \boldsymbol{E}_{(\varsigma-1)\times(N-\varsigma+1)} \end{bmatrix} \quad (5),$$

where the submatrix $\boldsymbol{C}$ is lower-triangular. The reversed generator matrix of PAC codes is $\boldsymbol{J}_N = \boldsymbol{H}_N \boldsymbol{B}_N$. The minimum column-weight of $\boldsymbol{J}_N$ is now $\sum_{i=1}^{\varsigma} c_i$, so there is no longer a column with weight 1. To make the existing shortening methods of polar codes applicable to PAC codes, we propose the following three straightforward modification schemes.

1) Expanding the mandatory frozen subset $\mathcal{F}$

According to Eq. (2), the value of u_i depends on the bits $\boldsymbol{v}_{i+1-\mathrm{supp}(\boldsymbol{c})}$, where supp(·) refers to the set-theoretic support function, i.e., the indices of the non-zero elements in the set. To obtain the same shortening pattern $\mathcal{S}$ as the shortened polar codes, the indices of all dependent bits of $\boldsymbol{u}_{\mathcal{F}_{\mathrm{polar}}}$ should be included in the mandatory frozen subset, i.e.,

$$\mathcal{F}_{\mathrm{PAC}} \supseteq \left\{f_i + 1 - \mathrm{supp}(\boldsymbol{c})\right\},\ \forall f_i \in \mathcal{F}_{\mathrm{polar}} \quad (6),$$

where $\mathcal{F}_{\mathrm{PAC}}$ refers to the mandatory frozen subset of PAC codes with $\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}} = \boldsymbol{0}$. Therefore, the information set $\mathcal{A}$ is narrowed down to $\left\{\mathcal{U}_N \backslash \mathcal{F}_{\mathrm{PAC}}\right\}$. The strengths and weaknesses of this scheme are as follows.

- Strengths: The decoding of the PAC code does not need to be modified.
- Weaknesses: Compared to shortened polar codes, more indices corresponding to unfavorably polarized sub-channels have to be selected to carry the information bits, which diminishes reliability. For the WANG-LIU pattern, the mandatory frozen subset is expanded to $\mathcal{F}_{\mathrm{PAC}} = \left\{E - \varsigma + 1, \cdots, E, E + 1, \cdots, N\right\}$. For the other shortening patterns, more elements are included in $\mathcal{F}_{\mathrm{PAC}}$, causing a significant performance loss. In particular, the BRS pattern is not directly applicable.

2) Reconstructing the pre-transform matrix $\boldsymbol{T}_N$

By reconstructing $\boldsymbol{T}_N$, we can avoid the expansion of the mandatory frozen subset caused by the pre-transform and make the original subset $\mathcal{F}_{\mathrm{polar}}$ valid, i.e. $\mathcal{F}_{\mathrm{PAC}} = \mathcal{F}_{\mathrm{polar}}$. The reconstructed pre-transform matrix $\tilde{\boldsymbol{T}}_N = \left(\tilde{t}_{a,b}\right)_{a,b \in \mathcal{U}_N}$ can be expressed as:

$$\begin{cases} \tilde{t}_{a,b} = c_{b-a+1}, & \text{if } a \leqslant b < a + \varsigma \,\&\, b \notin \mathcal{F}_{\mathrm{polar}} \\ \tilde{t}_{a,b} = 1, & \text{else if } a = b \,\&\, b \in \mathcal{F}_{\mathrm{polar}} \\ \tilde{t}_{a,b} = 0, & \text{otherwise} \end{cases} \quad (7).$$

In this case, the interim codeword generated from $\tilde{\boldsymbol{T}}_N$ satisfies $\boldsymbol{u}_{\mathcal{F}_{\mathrm{PAC}}} = \boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}}$, and the corresponding deleted bits $\boldsymbol{x}_{\mathcal{S}}$ are actually the polar code bits. The characteristics of this scheme are summarized below.

- Strengths: It retains an identical shortening pattern and mandatory frozen subset as conventional shortened polar codes, ensuring compatibility.
- Weaknesses: It requires a modified decoding procedure: standard polar code decoding rules are used for bits $\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}}$, while PAC code decoding rules for the remaining bits, thereby increasing receiver complexity.

3) Applying dynamic frozen bits to $\boldsymbol{v}_{\mathcal{F}}$

The expansion of the mandatory frozen subset can be prevented by applying DFBs, which constrain the deleted bits to zero, i.e., $\boldsymbol{x}_{\mathcal{S}} = 0$ and $\mathcal{F}_{\mathrm{PAC}} = \mathcal{F}_{\mathrm{polar}}$. The mandatory frozen bits $\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}}$ are designated as DFBs. Their values are derived from a linear combination of information bits. This combination can be expressed as:

$$\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}} = g(\boldsymbol{v}_{\mathcal{A}}) = \boldsymbol{v}_{\mathcal{A}} \boldsymbol{K}(:, \mathcal{A})^{\mathrm{T}} \quad (8),$$

where the matrix $\boldsymbol{K}$ is the reduced row echelon form obtained by applying Gaussian elimination to $\boldsymbol{J}_N(:, \mathcal{S})^{\mathrm{T}}$. The submatrix $\boldsymbol{K}(:, \mathcal{F}_{\mathrm{PAC}})$ should be an identity matrix, while $\boldsymbol{J}_N(:, \mathcal{S})$ refers to the submatrix consisting of the $\mathcal{S}$-th columns of $\boldsymbol{J}_N$.

For this scheme, the one-capacity sub-channels of shortening are conditionally present in the decoding graph. When the information bits $\boldsymbol{v}_{\mathcal{A}}$ are correctly decoded, the results of the DFBs are accurate, and the one-capacity sub-channels exist. Once the bits $\boldsymbol{v}_{\mathcal{A}}$ are decoded incorrectly, the DFB results become unreliable and the one-capacity sub-channels no longer exist.

The strengths and weaknesses of this scheme are discussed as follows.

• Strengths: The shortening pattern and the mandatory frozen subset are the same as those of shortened polar codes.

• Weaknesses: The decoding process needs to be modified. The linear function $g(\cdot)$ should be taken as the inputs for decoding, serving as an additional step to decode the DFBs. The function $g(\cdot)$ varies with different block-lengths and code-rates.

3.2 A Novel Shortening Scheme for PAC Codes

We provide a novel idea to allow $\boldsymbol{x}_{\mathcal{S}}$ to be undetermined at the receiver, aiming to reduce the size of $\mathcal{F}_{\mathrm{PAC}}$. A subset $\mathcal{I}$ is removed from $\mathcal{F}_{\mathrm{PAC}}$ and incorporated into the information set; thus, the mandatory frozen subset becomes $\{\mathcal{F}_{\mathrm{PAC}} \backslash \mathcal{I}\}$. All possible values of $\boldsymbol{x}_{\mathcal{S}}$ are assigned to the paths during the initialization of list decoding, ensuring that at least one path contains the true value.

Similar to the DFB scheme, the one-capacity sub-channels of shortening are conditionally present in the list decoding process. When the paths assigned with the true value of $\boldsymbol{x}_{\mathcal{S}}$ can be retained in the list during decoding, the one-capacity sub-channels exist in these paths and contribute to reliable decoding performance. Therefore, despite $\boldsymbol{x}_{\mathcal{S}}$ being undetermined at the receiver, this scheme is essentially a shortening scheme rather than a puncturing one.

The number of possible combinations of the value of $\boldsymbol{x}_{\mathcal{S}}$ is up to $2^{|\mathcal{I}|}$, denoted as $\hat{\boldsymbol{x}}_{\mathcal{S}}[\ell] = \hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]\boldsymbol{J}_N(\mathcal{I},\mathcal{S})$, $\ell \in \mathcal{U}_{2^{|\mathcal{I}|}}$, where $\boldsymbol{J}_N(\mathcal{I},\mathcal{S})$ refers to the submatrix consisting of the $\mathcal{I}$-th row and $\mathcal{S}$-th column of $\boldsymbol{J}_N$, and $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell] \in \{0,1\}^{|\mathcal{I}|}$. Therefore, the main task of this scheme is to eliminate the incorrectly assigned paths early in the decoding process while retaining the correct one.

1) Findings from list decoding

We can activate $2^{|\mathcal{I}|}$ paths in list decoding to contain the different results derived from various value-combinations of $\boldsymbol{x}_{\mathcal{S}}$, when $L \geqslant 2^{|\mathcal{I}|}$. The initial LLR corresponding to the ℓ-th path is expressed as:

$$L_i^0[\ell] = \begin{cases} (1-2\hat{x}_i[\ell])\cdot\infty, & \text{if } i \in \mathcal{S} \\ \ln\dfrac{\Pr(\boldsymbol{y}|\hat{x}_i[\ell]=0)}{\Pr(\boldsymbol{y}|\hat{x}_i[\ell]=1)}, & \text{otherwise} \end{cases} \quad (9).$$

Before starting path-killing, the PM derived from the correct value-combination $\hat{\boldsymbol{x}}_{\mathcal{S}}[\breve{\ell}]$ needs to be lower than the other PMs, to avoid killing the correct path $\breve{\ell}$. The expected values of the PMs of the correct path $\breve{\ell}$ and incorrect path $\widehat{\ell} \in \{\mathcal{U}_{2^{|\mathcal{I}|}} \backslash \breve{\ell}\}$ should have

$$\left\langle \Delta_{\widehat{\ell}} \right\rangle = \left\langle \mathrm{PM}_{\hat{v}_{\min(\mathcal{A})-1}[\widehat{\ell}]} - \mathrm{PM}_{\hat{v}_{\min(\mathcal{A})-1}[\breve{\ell}]} \right\rangle > 0 \quad (10),$$

where $\langle\cdot\rangle$ refers to the expected value function. To diminish the adverse impact of incorrect paths, the minimum value $\min\left(\left\langle \Delta_{\widehat{\ell}} \right\rangle\right)$ should be as large as possible.

According to the recursive formula (3), the LLR results $\boldsymbol{L}^n_{\mathcal{U}_{2^p}}[\ell]$ derived from the full codeword's initial LLRs $\boldsymbol{L}^0_{\mathcal{U}_N}[\ell]$ are equivalent to those derived from the sub-codeword's LLRs $\boldsymbol{L}^m_{\mathcal{U}_{2^p}}[\ell]$, where the sub-codeword is expressed as $\boldsymbol{x}^m_{\mathcal{U}_{2^p}}$ and the corresponding LLRs are

$$L_i^m[\ell] = \mathop{\boxplus}_{j=(i-1)M+1}^{iM} L_j^0[\ell],\ i \in \mathcal{U}_{2^p}, \ell \in \mathcal{U}_{2^{|\mathcal{I}|}} \quad (11).$$

Example 1: Consider an example in which the codeword is with the block length $N = 32$, and the initial LLRs are $\{L_1^0[\ell], L_2^0[\ell], \cdots, L_{32}^0[\ell]\}$. The LLRs of the sub-codeword $\boldsymbol{x}^2_{\mathcal{U}_8}$ can be obtained by $\{(L_1^2[\ell] = L_1^0[\ell] \boxplus L_2^0[\ell] \boxplus L_3^0[\ell] \boxplus L_4^0[\ell]), \cdots, (L_8^2[\ell] = L_{29}^0[\ell] \boxplus L_{30}^0[\ell] \boxplus L_{31}^0[\ell] \boxplus L_{32}^0[\ell])\}$.

For an integer Λ, where $2^{p-1} < \Lambda \leqslant 2^p$, we can obtain the reversal of $\mathcal{U}_\Lambda$, which is expressed as $\mathcal{R}_\Lambda = \beta_{2^p}(\mathcal{U}_\Lambda)$, where the reversal function is $\beta_{2^\eta}(\zeta) = \mathcal{U}_{2^\eta}\boldsymbol{B}_{2^\eta}(\mathcal{U}_{2^\eta}, \zeta)$. We define the coefficients of the sub-codeword bits $\boldsymbol{x}^m_{\mathcal{R}_\Lambda}$ in each path ℓ as:

$$h_i^m[\ell] \triangleq \hat{x}_i^m[\ell] \oplus \hat{x}_i^m[\breve{\ell}],\ i \in \mathcal{R}_\Lambda \quad (12).$$

The coefficients $h^m_{\mathcal{R}_\Lambda}[\ell]$ can be partitioned into two parts. One part, denoted as $\boldsymbol{h}^m_{\boldsymbol{r}}[\ell] = \{h^m_\delta[\ell]\}, \boldsymbol{r} \subseteq \mathcal{R}_\Lambda$, is influenced by the initial infinite LLRs $\boldsymbol{L}^0_{\mathcal{S}}[\ell]$. According to Eq. (11), $\boldsymbol{L}^0_{\mathcal{S}}[\ell]$ can be mapped to the sub-codeword's LLRs as:

$$L_\delta^m[\ell] = L_\delta^m[\breve{\ell}]\cdot\left(\prod_{s\in\mathcal{S}_\delta}\mathrm{sgn}\left(\mathrm{L}_s^0[\ell]\right)\mathrm{sgn}\left(L_s^0[\breve{\ell}]\right)\right) \quad (13),$$

where $\mathrm{sgn}(\cdot)$ denotes the sign function. The subset $\mathcal{S}_\delta \subseteq \mathcal{S}$ is expressed as:

$$\mathcal{S}_\delta = \left\{ s \in \mathcal{S}: \delta = \left\lceil \frac{s}{2^m} \right\rceil \right\} \quad (14),$$

where $\lceil\cdot\rceil$ denotes the ceiling function. These coefficients satisfy:

$$h_\delta^m[\ell] = \bigoplus_{s\in\mathcal{S}_\delta}\left(\hat{x}_s[\ell]\oplus\hat{x}_s[\breve{\ell}]\right),\ \forall\delta\in\boldsymbol{r} \tag{15}$$

The other part of the coefficients $\boldsymbol{h}^m_{\mathcal{R}_\Lambda\backslash\boldsymbol{r}}[\ell]$ are not affected by the initial infinite LLRs.

Example 2: Extend Example 1 where $\Lambda = 6$ and $E = 27$ by applying the WANG-LIU pattern $\mathcal{S} = \{32, 16, 24, 8, 28\}$. The coefficients $h^2_{\mathcal{R}_6}[\ell]$, $\mathcal{R}_6 = \{1, 5, 3, 7, 2, 6\}$ can be divided into two sets: set $\boldsymbol{h}^m_{\boldsymbol{r}}[\ell] = \left\{h^2_7[\ell], h^2_2[\ell], h^2_6[\ell]\right\}$, influenced by the infinite LLRs $\left\{L^0_{28}[\ell], L^0_8[\ell], L^0_{24}[\ell]\right\}$, and its complement, $\boldsymbol{h}^m_{\mathcal{R}_\Lambda\backslash\boldsymbol{r}}[\ell] = \left\{h^2_1[\ell], h^2_5[\ell], h^2_3[\ell]\right\}$, which is not affected by the infinite LLRs.

To better distinguish between correct and incorrect paths, we impose a structure on this part by freezing the bits $\boldsymbol{v}_{\mathcal{U}_\Lambda}$ to specific values.

The effect on this part is formulated as:

$$\boldsymbol{h}^m_{\mathcal{R}_\Lambda\backslash\boldsymbol{r}}[\ell] = \left(\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]\oplus\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\breve{\ell}]\right)\boldsymbol{J}_{2^\rho}\left(\mathcal{U}_\Lambda,\mathcal{R}_\Lambda\backslash\boldsymbol{r}\right) \tag{16}$$

In this case, we have the following lemma.

Lemma 1: A lower bound exists for the expected PM difference $\left\langle\Delta_{\widehat{\ell}}\right\rangle$ between the incorrect path $\widehat{\ell}$ and the correct path $\breve{\ell}$ before path-killing, which is expressed as:

$$\left\langle\Delta_{\widehat{\ell}}\right\rangle > \left|\left\langle L^n_\iota[\breve{\ell}]\right\rangle\right| \tag{17}$$

where $\iota = \min\left(\mathrm{supp}\left((\hat{\boldsymbol{x}}^m_{\mathcal{R}_\Lambda}[\widehat{\ell}]\oplus\hat{\boldsymbol{x}}^m_{\mathcal{R}_\Lambda}[\breve{\ell}])\boldsymbol{G}_{2^\rho}(\mathcal{R}_\Lambda,\mathcal{U}_\Lambda)\right)\right)$.

Proof: Since $\hat{u}_\iota[\widehat{\ell}] = 1 - \hat{u}_\iota[\breve{\ell}]$ denotes the first difference between the two result paths of the interim codeword, the corresponding resulting LLR satisfies $L^n_\iota[\breve{\ell}] = L^n_\iota[\widehat{\ell}]$. Therefore, we can obtain $\left\langle \mathrm{BM}_{\hat{v}_\iota[\widehat{\ell}]} - \mathrm{BM}_{\hat{v}_\iota[\breve{\ell}]}\right\rangle > \left|\left\langle \mathrm{L}^n_\iota[\breve{\ell}]\right\rangle\right|$ from Eq. (4). According to Lemma 1 in Ref. [22], a front erroneous decision reduces the magnitude of subsequent resulting LLRs, i.e., $\left|\left\langle L^n_\jmath[\breve{\ell}]\right\rangle\right| > \left|\left\langle L^n_\jmath[\widehat{\ell}]\right\rangle\right|$, where $\jmath\in\{\iota+1,\cdots,N\}$. Therefore, the corresponding BMs satisfy $\left\langle BM_{\hat{v}_\jmath[\widehat{\ell}]}\right\rangle > \left\langle BM_{\hat{v}_\jmath[\breve{\ell}]}\right\rangle$.

2) The selection of $\mathcal{I}$ and the value of $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$

The unfrozen set $\mathcal{I}$ can follow that $\mathcal{I}\subseteq\beta_N(\mathcal{S})$ to ensure the value-combinations $\hat{\boldsymbol{x}}_{\mathcal{S}}[\ell]$, $\ell\in\mathcal{U}_{2^{|\mathcal{I}|}}$ are independent. According to Lemma 1, we design the coefficients $\boldsymbol{h}^m_{\mathcal{R}_\Lambda}[\ell]$ to maximize the lower bound $\min\left(\left\langle\Delta_{\widehat{\ell}}\right\rangle\right)$. The sub-codeword is constructed as $\boldsymbol{x}^m_{\mathcal{R}_\Lambda}[\ell] = \boldsymbol{u}_{\mathcal{B}}[\ell]\boldsymbol{G}_{2^\rho}(\mathcal{B},\mathcal{R}_\Lambda)$, where $\boldsymbol{u}_{\mathcal{B}}[\ell] = \{0,1\}^{|\mathcal{I}|}$. GA is an ideal way to approximate the LLR magnitude $\left|\left\langle L^n_\iota[\breve{\ell}]\right\rangle\right|$ and thereby identify a small set $\mathcal{B}\subseteq\mathcal{U}_\Lambda$ to raise the lower bound. This set is composed of indices $b\in\beta_{2^\rho}(\boldsymbol{r})$ with the largest LLR magnitudes $\left|\left\langle L^n_b[\breve{\ell}]\right\rangle\right|$, which can be expressed as:

$$\mathcal{B} = \underset{\mathcal{B}'\subseteq\beta_{2^\rho}(\boldsymbol{r}),|\mathcal{B}'|=|\mathcal{I}|}{\arg\max}\sum_{b\in\mathcal{B}'}\left|\left\langle L^n_b[\breve{\ell}]\right\rangle\right| \tag{18}$$

Consequently, an upper bound of the block error probability, derived from Lemma 1, is given by

$$\Pr\left(\mathrm{PM}_{\hat{v}_N[\widehat{\ell}]} < \mathrm{PM}_{\hat{v}_N[\breve{\ell}]}\right) < \frac{1}{2}\mathrm{erfc}\left(\frac{1}{2}\sqrt{\min\left(\left|\left\langle L^n_{\mathcal{B}}[\breve{\ell}]\right\rangle\right|\right)}\right) \tag{19}$$

Example 3: Building on Example 2, where $\mathcal{B} = \{4, 6\}$ is always selected from $\beta_8(\boldsymbol{r}) = \{4, 5, 6\}$, the generator matrix of the sub-codewords becomes:

$$\boldsymbol{G}_8\left(\{4,6\},\{1,5,3,7,2,6\}\right) = \begin{bmatrix}1&1&1&1&0&0\\1&1&0&0&1&1\end{bmatrix} \tag{20}$$

The minimum distance of all the possible sub-codewords is 4.

An appropriate set Λ can be determined by lowering this upper bound while minimally impacting the selection of the information set $\mathcal{A}$. The unfrozen set $\mathcal{I}$ is then selected via a one-to-one mapping from $\mathcal{B}$.

Lemma 2: For each element $b\in\mathcal{B}$, there is always a corresponding element $\left(N + b - 2^\rho\right)\in\mathcal{F}_{\mathrm{PAC}}$.

Proof: Assume $\delta\in\beta_{2^\rho}(b)$. Then, the corresponding set $\mathcal{S}_\delta$ can be obtained from Eq. (14). From Lemma 1 in Ref. [23], the elements follows that $\boldsymbol{G}_N\left(N + b - 2^\rho,\mathcal{S}_\delta\right) = 1$. Therefore, u_{N+b-2^ρ} should be frozen bits or DFBs.

According to Lemma 2, we provide a simple way to obtain $\mathcal{I}$ from $\mathcal{B}$, which is expressed as:

$$\mathcal{I} = \mathcal{B} + N - 2^\rho \tag{21}$$

Once the unfrozen set $\mathcal{I}$ is determined, we need to examine whether all the coefficients in $\boldsymbol{h}^m_{\boldsymbol{r}}[\ell]$ are affected by the value of $\boldsymbol{x}_{\mathcal{S}}$.

Actually, due to the Kronecker-product construction of the generator matrix, there will be a part of elements $\boldsymbol{q} = \{\boldsymbol{r}\backslash\boldsymbol{r}_{\mathrm{sub}}\}$ whose corresponding coefficients $h^m_\varrho[\ell]$, $\varrho\in\boldsymbol{q}$ are fixed to 0 for each path $\ell\in\mathcal{U}_{2^{|\mathcal{I}|}}$, if $|\mathcal{S}_\varrho|$ is even. In other words, only the coefficients $\boldsymbol{h}^m_{\boldsymbol{r}_{\mathrm{sub}}}[\ell]$ are affected by the the initial infinite LLRs.

Proposition 1: For the WANG-LIU pattern, the indices of the actually affected part $\boldsymbol{h}^m_{\boldsymbol{r}_{\mathrm{sub}}}[\ell]$ can be obtained as follows:

$$\mathrm{supp}\left(\beta_{2^p}\left(\boldsymbol{r}_{\mathrm{sub}}\right)\right)=\begin{cases}\mathcal{U}_{E\%2^p}\cap\mathcal{U}_{\Lambda}, & \text{if}\left\lfloor E/2^p\right\rfloor\text{is even}\\ \mathcal{U}_{\Lambda}\backslash\mathcal{U}_{E\%2^p}, & \text{otherwise}\end{cases}\tag{22}$$

Proof: From the coefficient derivationin Eq. (15), we have $\bigoplus_{s\in\mathcal{S}_\delta}\hat{x}_s[\ell]=\bigoplus_{s\in\mathcal{S}_\delta}\hat{v}_{\mathcal{I}}[\ell]\boldsymbol{J}_N(\mathcal{I},s)$, where

$$\bigoplus_{s\in\mathcal{S}_\delta}\hat{v}_{\mathcal{I}}[\ell]\boldsymbol{J}_N(\mathcal{I},s)=\begin{cases}0, & \text{if}\left|\mathcal{S}_\delta\right|\text{is even}\\ \hat{v}_{\mathcal{I}}[\ell]\boldsymbol{J}_N(\mathcal{I},2^m\cdot\delta), & \text{otherwise}\end{cases}\tag{23}$$

Eq. (23) is obtained because $\hat{v}_{\mathcal{I}}[\ell]\boldsymbol{J}_N(\mathcal{I},\mathcal{S}_\delta)$ is an all-zero (or all-one) sequence when the WANG-LIU pattern is adopted. Therefore, we have $\boldsymbol{r}_{\mathrm{sub}}=\{\delta:\left|\mathcal{S}_\delta\right|\text{is odd}\}$. We assume that δ' is the bit-reversal result of δ, i.e., $\delta'=\beta_{2^p}(\delta)$. In this case, the event that $\left|\mathcal{S}_\delta\right|$ is odd is equivalent to the event that $\left|\mathcal{Q}_{\delta'}\right|$ is odd, where $\mathcal{Q}_{\delta'}=\left\{s\in\mathcal{F}:\delta'=s\%2^p\right\}$. Consequently, Eq. (22) is obtained.

Example 4: We examine a specific case of Example 2 where $E=20$. Under this condition, the actually affected coefficients are $\boldsymbol{h}^m_{\boldsymbol{r}_{\mathrm{sub}}}[\ell]=\left\{h^2_7[\ell],h^2_2[\ell],h^2_6[\ell]\right\}$. The indices of these coefficients are $\boldsymbol{r}_{\mathrm{sub}}=\{1,5,3,7\}$. Here, an additional coefficient $h^2_2[\ell]$ is affected by $\left\{L^0_6[\ell],L^0_8[\ell]\right\}$. Since $L^0_6[\ell]=L^0_8[\ell]$, the effects of $L^0_6[\ell]$ and $L^0_8[\ell]$ on $h^2_2[\ell]$ are offset.

We now introduce an effect on the unaffected part by assigning specific values to $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ for each path. Following Lemma 1, the corresponding sub-codeword bits of each path should be constructed to satisfy $\hat{\boldsymbol{x}}^m_{\mathcal{R}_\Lambda\backslash\boldsymbol{r}_{\mathrm{sub}}}[\ell]=\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]\boldsymbol{J}_N\left(\mathcal{I},2^m\left(\mathcal{R}_\Lambda\backslash\boldsymbol{r}_{\mathrm{sub}}\right)\right)$. According to Eq. (16), the specific bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ of each path $\ell\in\mathcal{U}_{2^{|\mathcal{I}|}}$ can be calculated as:

$$\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]=\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]\boldsymbol{J}_N\left(\mathcal{I},2^m\left(\mathcal{R}_\Lambda\backslash\boldsymbol{r}_{\mathrm{sub}}\right)\right)\boldsymbol{J}^{-1}_{2^p}\left(\mathcal{R}_\Lambda\backslash\boldsymbol{r}_{\mathrm{sub}},\mathcal{U}_\Lambda\right)\tag{24}$$

where $\boldsymbol{J}^{-1}_{2^p}=\boldsymbol{G}_{2^p}\boldsymbol{T}^{-1}_{2^p}$. The inverse matrix $\boldsymbol{T}^{-1}_{2^p}$ exists, since $\boldsymbol{T}_{2^p}$ is an upper triangular matrix.

Example 5: Revisit Example 2, where the sub-codewords should be constructed via Eq. (16). The part $\hat{\boldsymbol{x}}^m_{\boldsymbol{r}_{\mathrm{sub}}}[\ell]=\left\{\hat{x}^2_7[\ell],\hat{x}^2_2[\ell],\hat{x}^2_6[\ell]\right\}$ has been obtained by the effects from initial infinite LLRs, which can be expressed as:

$$\begin{aligned}&\left\{\hat{x}^2_7[\ell],\hat{x}^2_2[\ell],\hat{x}^2_6[\ell]\right\}=\left\{\frac{1}{2}\left(1-\mathrm{sgn}\left(L^0_{28}[\ell]\right)\right),\right.\\&\left.\frac{1}{2}\left(1-\mathrm{sgn}\left(L^0_8[\ell]\right)\right),\frac{1}{2}\left(1-\mathrm{sgn}\left(L^0_{24}[\ell]\right)\right)\right\}=\\&\left\{\hat{x}^0_{28}[\ell],\hat{x}^0_8[\ell],\hat{x}^0_{24}[\ell]\right\}\end{aligned}\tag{25}$$

Consequently, the bits $\hat{\boldsymbol{x}}^m_{\mathcal{R}_\Lambda\backslash\boldsymbol{r}_{\mathrm{sub}}}[\ell]=\left\{\hat{x}^2_1[\ell],\hat{x}^2_5[\ell],\hat{x}^2_3[\ell]\right\}$ should be constructed to satisfy:

$$\begin{aligned}&\left\{\hat{x}^2_1[\ell],\hat{x}^2_5[\ell],\hat{x}^2_3[\ell]\right\}=\\&\left\{\hat{v}_{28}[\ell],\hat{v}_{30}[\ell]\right\}\boldsymbol{J}_{32}\left(\{28,30\},\{4,20,12\}\right)\end{aligned}\tag{26}$$

Therefore, the specific bits $\hat{\boldsymbol{v}}_{\mathcal{U}_6}[\ell]$ must be set to $\hat{\boldsymbol{v}}_{\mathcal{U}_6}[\ell]=\left\{\hat{v}_{28}[\ell],\hat{v}_{30}[\ell]\right\}\boldsymbol{J}_{32}\left(\{28,30\},\{4,20,12\}\right)\boldsymbol{J}^{-1}_8\left(\{1,5,3\},\mathcal{U}_6\right)$ to ensure that Eq. (26) can be obtained.

In summary, when constructing the shortened PAC codes, the values of the frozen bits $\boldsymbol{v}_{\mathcal{U}_\Lambda}$ need to be calculated from the bits $\boldsymbol{v}_{\mathcal{I}}$ using Eq. (24). The proposed construction of shortened PAC codes is detailed in Algorithm 1.

Algorithm 1: Construction of shortened PAC codes

Input: Shortening pattern $\mathcal{S}$, information bits $\boldsymbol{d}$, $\boldsymbol{J}_N$ and Λ
Output: Shortened PAC codes $\boldsymbol{x}_{\mathcal{U}_N\backslash\mathcal{S}}$

Calculate the mandatory frozen subset $\mathcal{F}_{\mathrm{PAC}}$.
Select the small set $\mathcal{B}\subseteq\mathcal{U}_\Lambda$ based on Eq. (18).
Calculate $\mathcal{I}$ based on Eq. (21).
Select the information set $\mathcal{A}$ from $\left\{\left\{\mathcal{U}_N\backslash\mathcal{U}_\Lambda\right\}\backslash\left\{\mathcal{F}_{\mathrm{PAC}}\backslash\mathcal{I}\right\}\right\}$ via rate-profiling.
$\boldsymbol{v}_{\mathcal{A}}\leftarrow\boldsymbol{d}$.
Calculate $\boldsymbol{v}_{\mathcal{U}_\Lambda}$ from the bits $\boldsymbol{v}_{\mathcal{I}}$ by Eq. (24).
Construct the mother PAC codeword by $\boldsymbol{x}=\boldsymbol{v}\boldsymbol{J}_N$.

3.3 List Decoding for Proposed Shortened PAC Codes

Section 3.2 has provided a brief discussion on the list decoding of the proposed shortened PAC coding scheme. Here, we detail the list decoding of the proposed shortened PAC codes. The decoding is provided in Algorithm 2. In the initialization of list decoding, $2^{|\mathcal{I}|}$ paths are activated. For each activated path ℓ, a unique value combination is assigned to the bits $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]$, i.e., $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]=\left\{b_{|I|},b_{|I|-1},\cdots,b_1\right\}_j=\mathrm{d2b}(\ell-1)$, where $\mathrm{d2b}(\cdot)$ refers to a decimal-to-binary converter. Thereby, the initial LLRs $\boldsymbol{L}^0_{\mathcal{U}_N}[\ell]$ can be computed from $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]$. Simultaneously, the specific bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ are calculated using Eq. (24).

Algorithm 2: List decoding of shortened PAC codes

for $\ell\in\mathcal{U}_{2^{|\mathcal{I}|}}$ **do**
    Calculate the variable bits $\hat{\boldsymbol{x}}_{\mathcal{S}}[\ell]=\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]\boldsymbol{J}_N(\mathcal{I},\mathcal{S})$, where $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]=\left\{b_{|I|},b_{|I|-1},\cdots,b_1\right\}_j=\mathrm{d2b}(\ell-1)$.
    Calculate initial LLRs $\boldsymbol{L}^0_{\mathcal{U}_N}[\ell]$ according to Eq. (9).
    Calculate the specific bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ according to Eq. (24).
end
//Initialization differs from conventional one.
for $i=1$ **to** N **do**
    for $\ell\in\boldsymbol{p}$ **do**
        Derive the resulting LLR $L^n_i[\ell]$ according to Eq. (3).
        if $i\leqslant\Lambda$ **or** $i\in\mathcal{I}$ **then**
            $\hat{u}_i[\ell]\leftarrow\sum_{k=1}^{s}c_k\hat{v}_{i-k+1}[\ell]$.

//No need to make decisions.
else if $i \notin \mathcal{A}$ **then**
$\hat{v}_i[\ell] \leftarrow 0,\ \hat{u}_i[\ell] \leftarrow \sum_{k=1}^{s} c_k \hat{v}_{i-k+1}[\ell]$.
else
$\hat{v}_i[\ell] \leftarrow \{0,1\},\ \hat{u}_i[\ell] \leftarrow \sum_{k=1}^{s} c_k \hat{v}_{i-k+1}[\ell]$.
end
Calculate $\mathrm{PM}_{\hat{v}_i[\ell]}$ according to Eq. (4).
end
//PM-updating process is slightly changed.
if $i \in \mathcal{A}$ **then**
if the number of candidate paths exceeds L **then**
Retain L paths with smaller PMs $PM_{\hat{v}_i[\ell]}$.
else
Retain all candidate paths.
end
end
//Path-killing process is unchanged.
end
$\ell^\star \leftarrow \arg\min_{\ell \in \mathcal{L}} PM_{\hat{v}_N[\ell]}$.
return $\hat{\boldsymbol{v}}_{\mathcal{U}_N}[\ell^\star]$.

In the PM-updating process of the list decoding, decisions are unnecessary for bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ and $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]$, because they have been already assigned during initialization. For the remaining bits, PM-updating follows the conventional list decoding for PAC codes. The path-killing process is also unchanged.

According to Algorithm 2, the decoding complexity remains dominated by the derivation of resulting LLRs, although the computation of the bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ in the initialization introduces a small amount of addition computation. The complexity of the decoding can still be expressed as $O(LN\log_2 N)$.

3.4 Application of Dynamic Frozen Bits to Proposed Scheme

As described in Section 3.1, DFBs can be applied to enforce $\mathcal{F}_{\mathrm{PAC}} = \mathcal{F}_{\mathrm{polar}}$, which can also be applied in the proposed scheme. In this case, the bits $\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}}$ serve as DFBs. Since the subset $\mathcal{I} \subseteq \mathcal{F}_{\mathrm{PAC}}$ is also included in the information set $\mathcal{A}$, the actual information bits appearing in the data carrier word are $\boldsymbol{v}_{\mathcal{A}\backslash\mathcal{I}}$. The remaining $|\mathcal{I}|$ information bits are implied in the specific bits $\boldsymbol{v}_{\mathcal{U}_\Lambda}$.

The shortened codeword is constructed as follows. First, the data carrier word $\boldsymbol{v}$ is still mapped from $\boldsymbol{d}$, comprising the information bits $\boldsymbol{v}_{\mathcal{A}} = \boldsymbol{d}$ and the frozen bits $\boldsymbol{v}_{\mathcal{A}^c} = 0$. Assuming that $\mathcal{I}$ consists of the e-th index in $\mathcal{A}$ where $e \in \mathcal{E}$, i. e., $\mathcal{I} = \mathcal{A}_{\mathcal{E}}$, we have $\boldsymbol{v}_{\mathcal{I}} = \boldsymbol{d}_{\mathcal{E}}$. Second, the values of the special bits $\boldsymbol{v}_{\mathcal{U}_\Lambda}$ are calculated from $\boldsymbol{v}_{\mathcal{I}}$ via Eq. (24). Then, the values of DFBs $\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}}$ are updated as:

$$\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}} = \dot{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}} + \boldsymbol{v}_{\mathcal{U}_N\backslash\mathcal{F}_{\mathrm{PAC}}} \boldsymbol{K}\left(:, \mathcal{U}_N\backslash\mathcal{F}_{\mathrm{PAC}}\right)^{\mathrm{T}} \quad (27),$$

where the matrix $\boldsymbol{K}$ is identical to that in Eq. (8). The vector $\dot{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}}$ on the right-hand side of Eq. (27) is composed of three parts: $\left\{\dot{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}\cup\mathcal{U}_\Lambda} = \boldsymbol{d}_{\mathcal{E}}\boldsymbol{J}_N\left(\mathcal{I}, 2^m(\mathcal{R}_\Lambda\backslash\boldsymbol{r}_{\mathrm{sub}})\right)\boldsymbol{J}_{2^p}^{-1}\left(\mathcal{R}_\Lambda\backslash\boldsymbol{r}_{\mathrm{sub}}, \mathcal{F}_{\mathrm{PAC}}\cup\mathcal{U}_\Lambda\right), \dot{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}\backslash\{\mathcal{I}\cup\mathcal{U}_\Lambda\}} = \boldsymbol{0}, \dot{\boldsymbol{v}}_{\mathcal{I}} = \boldsymbol{d}_{\mathcal{E}}\right\}$.

Therefore, the values of the bits $\boldsymbol{v}_{\mathcal{I}}$ are updated, and $\boldsymbol{v}_{\mathcal{I}}$ do not appear as information bits. Finally, the PAC codeword is generated from $\boldsymbol{v}$ and then shortened using pattern $\mathcal{S}$. From Eq. (27), we obtain $\boldsymbol{v}\boldsymbol{K}^{\mathrm{T}} = \dot{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}}$. Therefore, the value of the deleted bits is given by:

$$\boldsymbol{x}_{\mathcal{S}} = \dot{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}}\boldsymbol{R}^{\mathrm{T}} \quad (28),$$

where $\boldsymbol{R}$ is an elementary matrix with $\boldsymbol{J}_N(:, \mathcal{S})^{\mathrm{T}} = \mathbf{RK}$. Unlike conventional DFB schemes, $\boldsymbol{x}_{\mathcal{S}}$ are not constrained to zeros in this design. Overall, the construction of the proposed shortened PAC codes with DFBs is summarized in Algorithm 3.

Algorithm 3: Construction of shortened PAC codes with DFBs

Input: Shortening pattern $\mathcal{S}$, information bits $\boldsymbol{d}$, $\boldsymbol{J}_N$ and Λ
Output: Shortened PAC codes $\boldsymbol{x}_{\mathcal{U}_N\backslash\mathcal{S}}$
Execute Lines 1 to 6 of Algorithm 1.
Update the DFBs $\boldsymbol{v}_{\mathcal{F}_{\mathrm{PAC}}}$ using Eq. (27).
Execute Line 7 of Algorithm 1.

Accordingly, the decoding is modified. During initialization, $2^{|\mathcal{I}|}$ paths are activated. Different from conventional decoding, the length of the data carrier word result is $N + |\mathcal{I}|$ for each path, i.e., $\hat{\boldsymbol{v}}_{\mathcal{U}_{N+|\mathcal{I}|}}[\ell]$. For each path, the specific bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ are calculated from $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]$ according to Eq. (24), where $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell] = \mathrm{d2b}(\ell - 1)$. Moreover, the values of $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]$ need to be copied into $\hat{\boldsymbol{v}}_{N+\mathcal{U}_{|\mathcal{I}|}}[\ell]$, i. e., $\hat{\boldsymbol{v}}_{N+\mathcal{U}_{|\mathcal{I}|}}[\ell] = \hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]$. The value combination of $\hat{\boldsymbol{x}}_{\mathcal{S}}[\ell]$ for each path is then determined, and the initial LLRs are calculated accordingly. According to Eq. (28), we have $\hat{\boldsymbol{x}}_{\mathcal{S}}[\ell] = \hat{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}}[\ell]\boldsymbol{R}^{\mathrm{T}}$ for each path $\ell \in \mathcal{U}_{2^{|\mathcal{I}|}}$.

In the PM-updating process, no decision is required when reaching the specific bits. For each DFB $\hat{v}_f[\ell], f \in \mathcal{F}_{\mathrm{PAC}}$, where f is the j-th element in $\mathcal{F}_{\mathrm{PAC}}$, its value is calculated as:

$$\hat{v}_f[\ell] = \hat{\boldsymbol{v}}_{\mathcal{U}_f}[\ell]\boldsymbol{K}\left(j, \mathcal{U}_f\right)^{\mathrm{T}} \quad (29).$$

The decision processes for the frozen bits $\hat{\boldsymbol{v}}_{\mathcal{A}^c\backslash\mathcal{F}_{\mathrm{PAC}}}[\ell]$ and information bits $\hat{\boldsymbol{v}}_{\mathcal{A}\backslash\mathcal{I}}[\ell]$ remain unchanged.

Upon completing the PM-updating and path-killing processes, the values $\hat{\boldsymbol{v}}_{N+\mathcal{U}_{|\mathcal{I}|}}[\ell]$ are copied back into $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell]$, because $\hat{\boldsymbol{v}}_{N+\mathcal{U}_{|\mathcal{I}|}}[\ell]$ match with the specific bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$. The decoding procedure is formalized in Algorithm 4.

Algorithm 4: List decoding of shortened PAC codes with DFBs

for $\ell \in \mathcal{U}_{2^{|\mathcal{I}|}}$ **do**

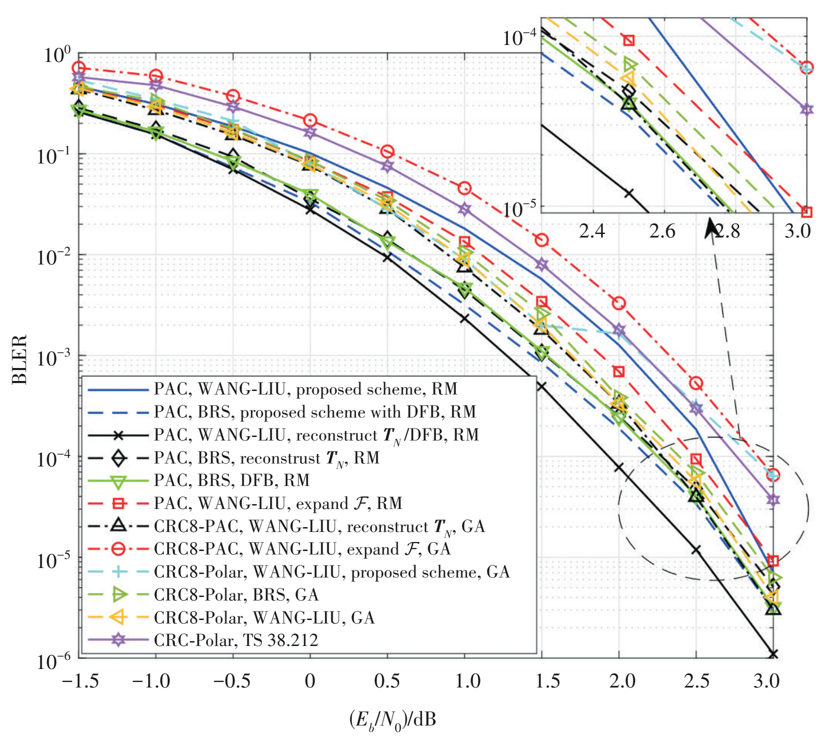
$\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell] = \hat{\boldsymbol{v}}_{N+\mathcal{U}_{\mathcal{I}}}[\ell] = \{b_{|\mathcal{I}|}, b_{|\mathcal{I}|-1}, \cdots, b_1\}_j = \mathrm{d2b}(\ell - 1)$.
    Calculate the specific bits $\hat{\boldsymbol{v}}_{\mathcal{U}_\Lambda}[\ell]$ according to Eq. (24).
    Calculate the variable bits by $\hat{\boldsymbol{x}}_{\mathcal{S}}[\ell] = \hat{\boldsymbol{v}}_{\mathcal{F}_{\mathrm{PAC}}}[\ell]\boldsymbol{R}^{\mathrm{T}}$.
    Calculate initial LLRs $\mathbf{L}^0_{\mathcal{U}_N}[\ell]$ according to Eq. (9).
end
$j = 1$.
for $i = 1$ **to** N **do**
    for $\ell \in \boldsymbol{p}$ **do**
        Derive the resulting LLR $L_i^n[\ell]$ according to Eq. (3).
        if $i \in \{\mathcal{U}_\Lambda \backslash \mathcal{F}_{\mathrm{PAC}}\}$ **or** $i \in \mathcal{I}$ **then**
            $\hat{u}_i[\ell] \leftarrow \sum_{k=1}^{s} c_k \hat{v}_{i-k+1}[\ell]$.
            //No need to make decisions.
        else if $i \in \mathcal{F}_{\mathrm{PAC}}$ **then**
            $\hat{v}_i[l] \leftarrow \hat{\boldsymbol{v}}_{\mathcal{U}_i}[l]\boldsymbol{K}(j, \mathcal{U}_i)^{\mathrm{T}}$,
            $j \leftarrow j + 1$,
            $\hat{u}_i[\ell] \leftarrow \sum_{k=1}^{s} c_k \hat{v}_{i-k+1}[\ell]$.
        else if $i \in \{\mathcal{A}^c \backslash \mathcal{F}_{\mathrm{PAC}}\}$ **then**
            $\hat{v}_i[\ell] \leftarrow 0$, $\hat{u}_i[\ell] \leftarrow \sum_{k=1}^{s} c_k \hat{v}_{i-k+1}[\ell]$.
        else
            $\hat{v}_i[\ell] \leftarrow \{0, 1\}$,
            $\hat{u}_i[\ell] \leftarrow \sum_{k=1}^{s} c_k \hat{v}_{i-k+1}[\ell]$.
        end
        Calculate $\mathrm{PM}_{\hat{v}_i[\ell]}$ according to Eq. (4).
    end
    if $i \in \mathcal{A}$ **then**
        if the number of candidate paths exceeds L **then**
            Retain L paths with smaller PMs $\mathrm{PM}_{\hat{v}_i[\ell]}$.
        else
            Retain all candidate paths.
        end
    end
end
for $\ell = 1$ **to** $2^{|\mathcal{I}|}$ **do**
    $\hat{\boldsymbol{v}}_{\mathcal{I}}[\ell] \leftarrow \hat{\boldsymbol{v}}_{N+\mathcal{U}_{\mathcal{I}}}[\ell]$.
end
$\ell^\star \leftarrow \arg\min_{\ell \in \mathcal{L}} \mathrm{PM}_{\hat{v}_N[\ell]}$.
return $\hat{\boldsymbol{v}}_{\mathcal{U}_N}[\ell^\star]$.

4 Simulation Results

4.1 Comparison of Shortened Schemes of PAC and Polar Codes

We simulate the performance of the proposed shortened PAC codes over a binary-input additive white Gaussian noise (BI-AWGN) channel. The set $\boldsymbol{c} = (1, 0, 1, 1, 0, 1, 1)$ is applied to construct $\boldsymbol{T}_N$. List decoding is employed with a list size $L = 8$. Parameters are set as $|\mathcal{I}| = 3$ and $\Lambda = 28$. Since the proposed scheme can be readily adapted for polar codes, its performance under this adaptation is also simulated for comparison. We evaluate the three schemes described in Section 3.1: expanding $\mathcal{F}$, reconstructing $\boldsymbol{T}_N$, and applying DFB. All simulations utilize the WANG-LIU shortening pattern[13]. For further comparison, we also include simulation results for the rate-matching of polar codes specified in the 3GPP TS 38.212 (Release 17) standard[2], employing both cyclic redundancy check (CRC)-8 and the standard CRC.

Fig. 1 illustrates the BLER performance under the conditions of $E = 104$ and $R = 1/3$. For PAC codes, under WANG-LIU pattern and RM rate-profiling, the reconstruct $\boldsymbol{T}_N$ and DFB schemes achieve optimal performance, yielding a gain exceeding 0.2 dB over other schemes. Under the BRS pattern and RM rate-profiling, the schemes utilizing DFB outperform those without DFB. Additionally, all proposed schemes for PAC codes surpass the CRC-Polar scheme from the TS 38.212 standard. Fig. 2 presents the BLER performance under the conditions of $E = 104$ and $R = 1/2$. For PAC codes, under the WANG-LIU pattern and RM rate-profiling, the pro-



BLER: Block Error Rate
BRS: Bit-Reversed WANG-LIU Shortening
CRC: Cyclic Redundancy Check
DFB: Dynamic Frozen Bit
GA: Gaussian Approximation
PAC: Polarization-Adjusted Convolutional
RM: Reed-Muller

Figure 1. BLER performance of shortened PAC and polar codes with $E = 104$ and $K = 35$

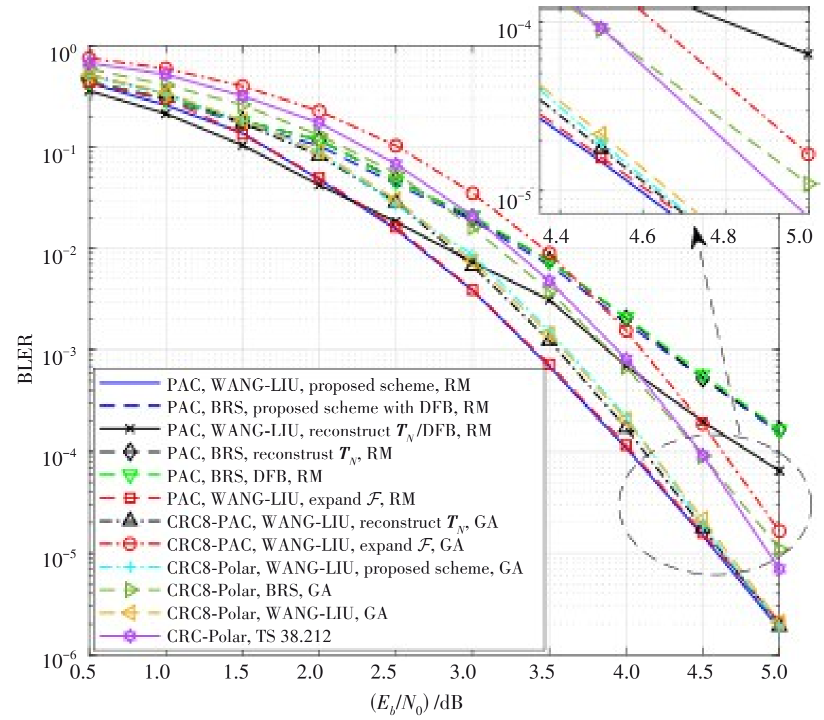
posed scheme and the reconstruct $\boldsymbol{T}_N$ deliver optimal performance, offering an approximate 0.3 dB gain over the CRC-Polar scheme from the TS 38.212 standard at BLER = 1×10^{-5}. Fig. 3 shows the BLER performance under the conditions of E = 104 and R = 3/5. For PAC codes, under the WANG-LIU pattern and RM rate-profiling, the proposed scheme and the expand $\mathcal{F}$ achieve optimal performance, providing about a 0.25 dB gain compared to the CRC-Polar scheme from the TS 38.212 standard at BLER = 1×10^{-5}.

Fig. 4 depicts the BLER performance under the conditions of $E \in \{84, 120, 180\}$ and R = 1/3. A total of six proposed shortening schemes are evaluated. Fig. 4a presents the schemes without DFB, while Fig. 4b shows those with DFB. As seen in Fig. 4a, the proposed scheme, reconstruct $\boldsymbol{T}_N$ and expand $\mathcal{F}$ have the same property that the performance improves gradually as the block length increases at a fixed code rate. However, Fig. 4b reveals an inconsistency in the performance trend of the DFB-based schemes as the block length grows.

Based on the above analysis, a promising hybrid approach is to adopt the proposed scheme with DFB for low code rates and the proposed scheme without DFB for medium to high code rates. This combination is expected to enhance the overall performance of shortened PAC codes.

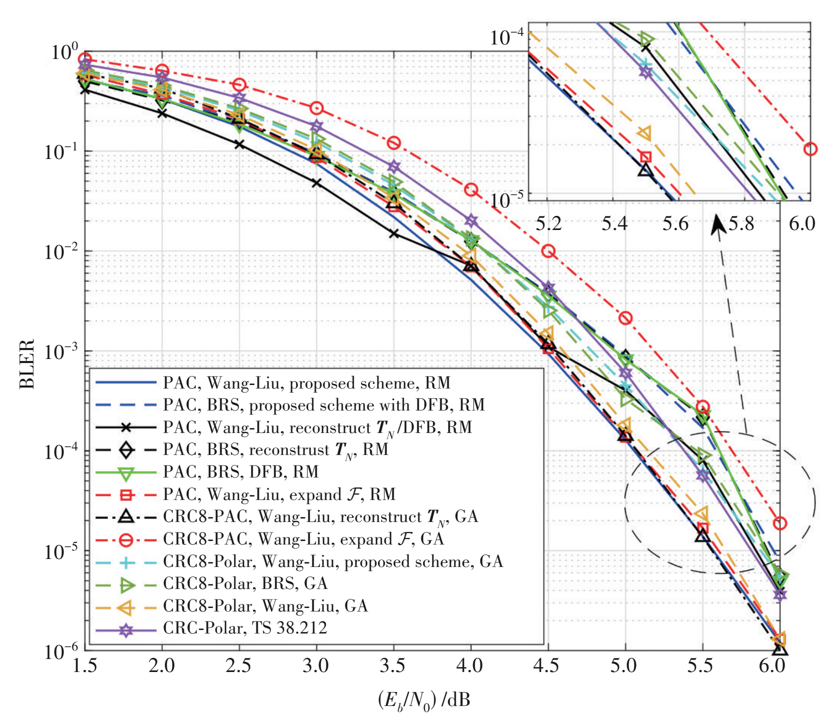
5 Conclusions

This paper explores the feasibility of applying existing shortening patterns of polar codes to PAC codes. A novel shortening scheme for PAC codes is proposed, which incorporates specially designed bits into the encoding process and introduces a minor modification to the list decoding algorithm. For short block lengths and over a portion of the low-to-medium rate range, the proposed shortened PAC coding scheme demonstrates superior BLER performance compared to existing shortened polar coding schemes, the standard rate-matching scheme, and conventional shortened PAC coding schemes.



BLER: Block Error Rate
BRS: Bit-Reversed WANG-LIU Shortening
CRC: Cyclic Redundancy Check
DFB: Dynamic Frozen Bit
GA: Gaussian Approximation
PAC: Polarization-Adjusted Convolutional
RM: Reed-Muller

Figure 2. BLER performance of shortened PAC and polar codes with E=104 and K=52



BLER: Block Error Rate
BRS: Bit-Reversed WANG-LIU Shortening
CRC: Cyclic Redundancy Check
DFB: Dynamic Frozen Bit
GA: Gaussian Approximation
PAC: Polarization-Adjusted Convolutional
RM: Reed-Muller

Figure 3. BLER performance of shortened PAC and polar codes with E=104 and K=62

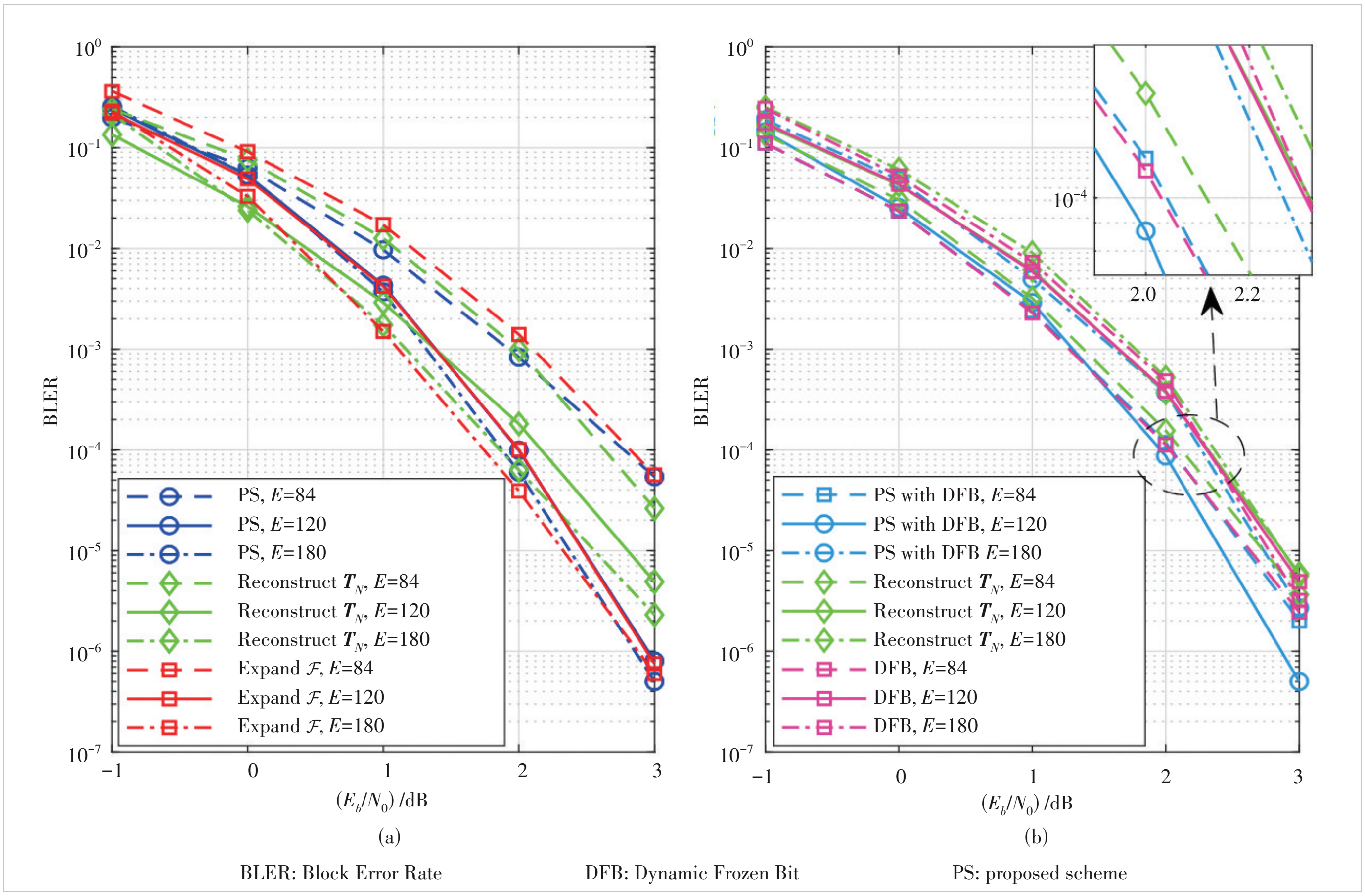


Figure 4. BLER performance of shortened PAC codes with R = 1/3

References

[1] ARIKAN E. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels [J]. IEEE transactions on information theory, 2009, 55(7): 3051 – 3073. DOI: 10.1109/TIT.2009.2021379

[2] 3GPP. 3rd generation partnership project; technical specification group radio access network; NR; multiplexing and channel coding (release 17): TS 38.212 [S]. 2022

[3] NIU K, DAI J, CHEN K, et al. Rate-compatible punctured polar codes: optimal construction based on polar spectra [EB/OL]. [2024-12-03]. https://arxiv.org/abs/1612.01352

[4] ARIKAN E. From sequential decoding to channel polarization and back again [EB/OL]. [2024-09-09]. https://arxiv.org/abs/1908.09594

[5] ROWSHAN M, BURG A, VITERBO E. Polarization-adjusted convolutional (PAC) codes: sequential decoding vs list decoding [J]. IEEE transactions on vehicular technology, 2021, 70(2): 1434 – 1447. DOI: 10.1109/TVT.2021.3052550

[6] LIU W X, CHEN L, LIU X C. A weighted sum based construction of PAC codes [J]. IEEE communications letters, 2023, 27(1): 28 – 31. DOI: 10.1109/LCOMM.2022.3209082

[7] NIU K, CHEN K, LIN J R. Beyond turbo codes: rate-compatible punctured polar codes [C]//Proc. IEEE International Conference on Communications (ICC). IEEE, 2013: 3423 – 3427. DOI: 10.1109/ICC.2013.6655078

[8] LI L P, SONG W, NIU K. Optimal puncturing of polar codes with a fixed information set [J]. IEEE access, 2019, 7: 65965 – 65972. DOI: 10.1109/ACCESS.2019.2918346

[9] SOLIMAN T, YANG F. Cooperative punctured polar coding (CPPC) scheme based on plotkin's construction [J]. Radioengineering, 2016, 25(3): 482 – 489. DOI: 10.13164/re.2016.0482

[10] BIOGLIO V, GABRY F, LAND I. Low-complexity puncturing and shortening of polar codes [C]//Proc. IEEE Wireless Communications and Networking Conference Workshops (WCNCW). IEEE, 2017: 1 – 6. DOI: 10.1109/WCNCW.2017.7919040

[11] ZHAO J H, ZHANG W, LIU Y Y. A novel puncturing scheme of low rate polar codes based on fixed information set [J]. IEEE communications letters, 2021, 25(7): 2104 – 2108. DOI: 10.1109/LCOMM.2021.3072050

[12] HAN S, KIM B, HA J. Rate-compatible punctured polar codes [J]. IEEE communications letters, 2022, 26(4): 753 – 757. DOI: 10.1109/LCOMM.2022.3144695

[13] WANG R X, LIU R K. A novel puncturing scheme for polar codes [J]. IEEE communications letters, 2014, 18(12): 2081 – 2084. DOI: 10.1109/LCOMM.2014.2364845

[14] ARIKAN E. Systematic encoding and shortening of PAC codes [J]. Entropy, 2020, 22(11): 1301. DOI: 10.3390/e22111301

[15] MILOSLAVSKAYA V. Shortened polar codes [J]. IEEE transactions on information theory, 2015, 61(9): 4852 – 4865. DOI: 10.1109/TIT.2015.2453312

[16] CHAKI P, KAMIYA N. On the properties of bit-reversal shortening in polar codes [C]//Proc. International Symposium on Information Theory and Its Applications (ISITA). IEEE, 2018: 437 – 441. DOI: 10.23919/ISITA.2018.8664298

[17] OLIVEIRA R M, DE LAMARE R C. Rate-compatible polar codes based on polarization-driven shortening [J]. IEEE communications letters, 2018, 22(10): 1984 – 1987. DOI: 10.1109/LCOMM.2018.2863375

[18] OLIVEIRA R M, DE LAMARE R C. Design of rate-compatible polar codes based on non-uniform channel polarization [J]. IEEE access, 2021, 9: 41902 – 41912. DOI: 10.1109/ACCESS.2021.3065816

[19] JANG M, AHN S K, JEONG H, et al. Rate matching for polar codes based on binary domination [J]. IEEE transactions on communications, 2019, 67(10): 6668–6681. DOI: 10.1109/TCOMM.2019.2930502
[20] ROWSHAN M, VITERBO E. On convolutional precoding in PAC codes [C]//Proceedings of IEEE Globecom Workshops (GC Wkshps). IEEE, 2021: 1–6. DOI: 10.1109/gcwkshps52748.2021.9681987
[21] SUN H, VITERBO E, LIU R K. Analysis of polarization-adjusted convolutional codes (PAC): a source-channel coding method [C]//Proc. IEEE Globecom Workshops (GC Wkshps). IEEE, 2021: 1–6. DOI: 10.1109/gcwkshps52748.2021.9682079
[22] FENG B W, YANG Y, JIAO J, et al. On tail-biting polarization-adjusted convolutional (TB-PAC) codes and small-sizes list decoding [J]. IEEE communications letters, 2023, 27(2): 433–437. DOI: 10.1109/LCOMM.2022.3225177
[23] ROWSHAN M, DAU S H, VITERBO E. On the formation of min-weight codewords of polar/PAC codes and its applications [J]. IEEE transactions on information theory, 2023, 69(12): 7627–7649. DOI: 10.1109/TIT.2023.3319015

Biographies

LIU Aolin received his BS degree in communication engineering from Harbin Institute of Technology (Shenzhen), China in 2023, where he is currently pursuing the PhD degree in information and communication engineering. He is also affiliated with Pengcheng Laboratory, China. His research interests include coding theory, error control codes, and satellite communications.

FENG Bowen (fengbowen@hit.edu.cn) received his BS, MS, and PhD degrees in information and communication engineering from the Harbin Institute of Technology, China in 2014, 2016, and 2021, respectively. From 2021 to 2024, he was a postdoctoral research fellow with Harbin Institute of Technology (Shenzhen) (HITSZ), China. He is currently an associate professor with the Guangdong Provincial Key Laboratory of Aerospace Communication and Networking Technology, HITSZ. His research interests include error control codes, satellite communications, and machine-to-machine communications.

LIANG Chulong received his BE degree in communication engineering and PhD degree in communication and information systems from Sun Yat-sen University, China in 2010 and 2015, respectively. He was a post-doctoral fellow with the Department of Electronic Engineering, City University of Hong Kong, China from July 2015 to June 2018, where he was a research fellow, from June 2018 to May 2019. He is currently a senior engineer with the Algorithm Department, ZTE Corporation. His current research interests include channel coding theory and its applications to communication systems.

XU Jin received his PhD degree from the School of Electronic Information, Wuhan University, China. He is currently a senior pre-research engineer with ZTE Corporation. His research interests include error correction codes, digital signal processing, semantic communication, and machine learning.

ZHANG Qinyu received his bachelor's degree in communication engineering from the Harbin Institute of Technology (HIT), China in 1994 and PhD degree in biomedical and electrical engineering from the University of Tokushima, Japan in 2003. From 1999 to 2003, he was an assistant professor with the University of Tokushima. He has been with Harbin Institute of Technology (Shenzhen) (HITSZ), China since 2003; he is currently a full professor and and serves as the Vice President of HITSZ. He has received the National Science Fund for Distinguished Young Scholars, and been recognized as a Young and Middle-Aged Leading Scientist of China and a Chinese New Century Excellent Talent in University, among other honors. He has also obtained three scientific and technological awards from government authorities. His research interests include aerospace communications and networks, and wireless communications and networks.