Joint Successive Cancellation List Decoding for the Double Polar codes

Yanfei Dong, Kai Niu, Jincheng Dai, Sen Wang, and Yifei Yuan

Abstract—As a new joint source-channel coding scheme, the double polar (D-Polar) codes have been proposed recently. In this letter, a novel joint source-channel decoder, namely the joint successive cancellation list (J-SCL) decoder, is proposed to improve the decoding performance of the D-Polar codes. We merge the trellis of the source polar code and that of the channel polar code to construct a compound trellis. In this compound trellis, the joint source-channel nodes represent both of the information bits and the high-entropy bits. Based on the compound trellis, the J-SCL decoder is designed to recover the source messages by combining the source SCL decoding and channel SCL decoding. The J-SCL decoder doubles the number of the decoding paths at each decoding level and then reserves the list size of paths at each JSC node and low-entropy node. When the number of decoding paths exceeds the list size, a pruning procedure is used to select the candidate paths with the smallest joint path-metric (JPM) from the list. For the JSC node, the JPM is updated considering both the channel decision log-likelihood ratios (LLRs) and the source decision LLRs. Simulation results show that the J-SCL decoder outperforms the turbo-like BP (TL-BP) decoder with low complexity.

Index Terms—Double polar codes, joint source-channel decoder, joint successive cancellation list decoder.

I. INTRODUCTION

Polar codes [1], invented by Arikan, are the first theoretically provable capacity-achieving error correction codes for any binary-input discrete memoryless channels with low encoding and decoding complexity. The successive cancellation (SC) [1] and belief propagation (BP) decoders [2] are two commonly decoding methods of polar codes. However, the finite-length performance of polar codes is unsatisfying under the SC and BP decoders. The successive cancellation list (SCL) decoder [3] has been proposed to improve the performance of polar codes. Moreover, the source polarization has been introduced and investigated in [7] as the complement of channel polarization. Polar codes are proved to be optimal for lossy source coding [8], which can achieve the rate-distortion bound for a binary symmetric source. For lossless compression, a polar encoding algorithm achieving the optimal compression rate asymptotically is developed in [9].

Polar codes have been used in the field of joint source and channel coding (JSCC). By modeling the source redundancy as a sequence of t-erasure correcting block codes, polar codes have proven to be a good candidate to exploit the benefit of source redundancy in a JSCC scheme [10]. For correlated sources, a joint decoding scheme using systematic polar codes is proposed in [11], which exploits the correlation among the sources. In [12], a JSCC scheme using quasi-uniform systematic polar code is proposed by introducing additional bit-swap coding to modify original polar coding. Inspired by the double low-density parity-check (LDPC) codes [13]–[15], a JSCC scheme using double polar (D-Polar) codes is proposed in our previous work [16], in which the source is first compressed by a source polar code, and then the compressed bits are protected by a systematic polar code (SPC). And a turbo-like BP (TL-BP) decoder is also designed for the decoding of D-Polar codes, which suffers from severe error floor and high complexity.

In this letter, we propose a joint successive cancellation (J-SCL) decoding algorithm to improve the performance of the D-Polar codes. In the D-Polar codes, the source is first compressed by a polar code, and then another polar code is employed to protect the compressed source. We find that the information bit part of channel polar code is a copy of the high entropy bit part of source polar code. Therefore, a compound trellis can be constructed by using the joint source-channel (JSC) nodes to represent both of the information bits and the high-entropy bits, merging the trellis of the source polar code and that of the channel polar code. In this combined trellis, the variable nodes corresponding to frozen bits and low-entropy bits are referred to as frozen nodes and low-entropy nodes, respectively.

Based on the compound trellis, a JSCC scheme, namely the J-SCL decoder, is proposed, which doubles the number of paths at each JSC node and low-entropy node. When the number of decoding paths exceeds the list size, a pruning procedure is used to select the L candidate paths with the smallest joint path-metric (JPM) from the list. For the JSC node, the update of the JPM considers both the channel decision log-likelihood ratios (LLRs) and the source decision LLRs. For the low-entropy node, the update of the JPM considers only the source decision LLRs. Simulation results show that the J-SCL decoder gains 0.42 dB compared to the TL-BP decoder, while the error floor of bit error rate (BER) is lowered from $10^{-4}$ to $10^{-6}$.

II. JOINT SUCCESSIVE CANCELLATION LIST ALGORITHM

In this section, we first introduce the D-Polar JSCC system model. Secondly, we give the method to fuse the trellis of the source polar code and that of channel polar code into a compound trellis. Finally, we propose the J-SCL decoder for decoding the D-Polar codes.

A. Notational Conventions

In this letter, the calligraphic characters, such as $\mathcal{X}$, are used to denote sets. We write the notation $a_i^n$ to denote a row vector $(a_1, \ldots, a_n)$. Given such a vector $a_i^n$, we use $a_{i,j}^n$, $1 \leq i, j \leq N$, to denote the subvector $(a_i, \ldots, a_j)$, and $a_i$ denotes the $i$-th element in $a_i^n$. For positive integer $N$, $[N] \triangleq \{1, 2, \ldots, N\}$. Given $A \subseteq [N]$, we write $a_A$ to denote the subvector $(a_i : i \in A)$ of $a_i^n$. 

This work was supported by the National Key R&D Program of China (No. 2018YFE0205501), National Natural Science Foundation of China (No. 62071058 & No. 62001049), and Beijing University of Posts and Telecommunications - China Mobile Research Institute Joint Innovation Center. The authors are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (email: niukai@bupt.edu.cn).
B. System Model

Fig. 1 illustrates the framework of the D-Polar codes JSCC system. Let $S \sim \text{Ber}(p)$ denote a binary Bernoulli random variable and $P_S(1) = p$. The source polar code can be identified by a parameter vector $(N_s, K, \mathcal{H})$, where $N_s = 2^{n_s}$ is the length of the source vector, and $K$ specifies the size of the high-entropy set $\mathcal{H}$. The ratio $K/N_s$ is called the compression rate $R_s$. The sequence of $N_s$ i.i.d. source symbols $s_N^{N_s}$ generated from $S$ is first compressed into $c_H$ by using a source polar code. In the source encoding process, we compute $c_H^{N_s} = s_N^{N_s}G_{N_s}$ and output $c_H$ as the compressed word, where $G_{N_s} = F^{\otimes N_s}$ equals the $n_s$-th Kronecker power of the matrix $F^{[1\ 0]}$. The channel polar coding is executed by

$$x_{1} = u_{1}^{N_s}G_{N_s} = u_{A}G_{N_s}(A) \oplus u_{A^c}G_{N_s}(A^c)$$

where $G_{N_s} = F^{\otimes N_s}$ and $G_{N_s}(A)$ denotes the submatrix of $G_{N_s}$ formed by the rows with indices in $A$. The codeword $x_{1}$ is then transmitted through a channel. Having received $y_{1}^{N_s}$ and known the $P_S(1) = p$, the proposed J-SCL decoder will build an estimate $\hat{s}_{1}^{N_s}$ of $s_{1}^{N_s}$.

C. Source-Channel Trellis Merging

The trellis of the source polar code and that of the channel polar code can be merged into a compound trellis. From Fig. 1, we have known $u_{A} = c_H$, which indicates that the information bits are duplicates of the high-entropy bits. The merging of the source trellis and the channel trellis is based on the fusion of the high-entropy bits of the source polar code and the information bits of the channel polar code.

For the sake of understanding, we give an example of the compound trellis, as shown in Fig. 2. The source polar code is identified by $(N_s = 4, K = 2, \mathcal{H} = \{1, 3\})$ and the channel polar code is indicated by $(N_c = 4, K = 2, \mathcal{A} = \{2, 4\})$. In Fig. 2, the channel polar code is represented by a blue trellis, and the source polar code is described by a red trellis. In this compound trellis, the leftmost variable nodes can be divided into three types of nodes, i.e., frozen nodes, JSC nodes, and low-entropy nodes. The frozen node and the low-entropy node correspond to the frozen bit of the channel polar code and the low-entropy bit of the source polar code, respectively. And the JSC node corresponds to the information bit of the channel polar code as well as the high-entropy bit of the source polar code.

Algorithm 1: The construction of the set $J$ and $W$

Input: $N_s, N_c, K, \mathcal{H}, \mathcal{A}$

Output: $J$, $W$

1 for $i = 1, 2, \ldots, K$ do
2 $\epsilon = \sum_{k=1}^{i}(h_{k} - h_{k-1} - 1)$;
3 $j_i \leftarrow a_i + \epsilon$;
4 Set $\mathcal{H}^c \leftarrow [N_s] \setminus \mathcal{H}$;
5 for $i = 1, 2, \ldots, N_s - K$ do
6 $\tau = \sum_{k=1}^{i}(a_{k} - a_{k-1} - 1)$;
7 $w_i \leftarrow h_{i}^c + \tau$;
8 return $J$, $W$.

Given the source polar code and the channel polar code, Algorithm 1 shows the construction method of the set $J$ and $W$. Let $a_i$, $h_i$, and $j_i$ denote the $i$-th element of $\mathcal{A}$, $\mathcal{H}$, and $J$, respectively. For the leftmost variables of the channel trellis, the construction of the compound trellis is equivalent to inserting the low-entropy node into the channel trellis. The number of low-entropy bits before the high-entropy bit $c_h$ can be calculated by $\epsilon = \sum_{k=1}^{i}(h_{k} - h_{k-1} - 1)$, where $h_0$ is set to 0. As shown in Line 3 of Algorithm 1, we can obtain $j_i$ by adding $\epsilon$ to $a_i$.

Similarly, the construction of the compound trellis is equivalent to the insertion of the frozen node into the source trellis. The set $\mathcal{H}^c$ is the complement of the high-entropy set $\mathcal{H}$, which...
is called the low-entropy set. The \( i \)-th element of \( h^{c} \) is denoted by \( h^{c}_{i} \). The number of high-entropy bits before the low-entropy \( c_{h} \) can be computed by 
\[
\varepsilon = \sum_{k=1}^{i} (h^{c}_{k} - h^{c}_{k-1} - 1),
\]
where \( h^{c}_{0} \) is set to 0. In the D-Polar codes, the high-entropy \( c_{n} \) is carried by the information bit \( u_{a} \). The number of frozen bits between \( u_{a} \) and \( u_{d} \) can be obtained by \( a_{i} - a_{i-1} - 1 \), where \( a_{0} \) is set to 0. Then, we can calculate the number of frozen bits before the information bit \( u_{a} \) by \( \tau = \sum_{k=1}^{a_{k}} (a_{k} - a_{k-1} - 1) \). As shown in Line 7 of Algorithm 1, the likelihood ratios (LLRs) need to consider the channel decision log-likelihood ratios (LLRs) \( L_{n}^{(i)}[\varphi] \) and \( L_{n}^{(i)}[\varphi] \) together, where \( \varphi \in J \cup W \). The decision LLRs \( L_{n}^{(i)}[\varphi] \) and \( L_{n}^{(i)}[\varphi] \) can be computed by the recursions (1), (6).

\[
L_{n}^{(2i)}[\varphi] = f_{n-1}^{(2i-2)}[\varphi] L_{n-1}^{(2i-2)}[\varphi], \quad L_{n}^{(2i+1)}[\varphi] = g_{n-1}^{(2i+1)}[\varphi] U_{n-1}^{(2i+1)}[\varphi],
\]
where the operation \( f \) and \( g \) are defined as
\[
f(\alpha, \beta) \triangleq \ln \left( \frac{e^{\alpha + \beta} + 1}{e^{\alpha} + e^{\beta}} \right),
\]
\[
g(\alpha, \beta, u) \triangleq (-1)^{u} e^{\alpha + \beta},
\]
respectively. For the channel SC decoding, the recursions terminate at
\[
L_{0}^{(i)}[\varphi] \triangleq \ln \frac{W(y|0)}{W(y|1)}, \quad i \in [N_c],
\]

### Algorithm 2: J-SCL Decoder

**Input:** \( N, J, W, p, y^{N_c} \)

**Output:** \( s_{1}^{N_c} \)

1. Initialize \( \mathcal{L} \leftarrow \{0\}, \) \( J_{PM}^{(0)} \leftarrow 0, \) \( i_c \leftarrow 0, \) \( i_s \leftarrow 0; \)
2. for \( \varphi = 1, 2, \ldots, N \) do
   3. for \( \ell \in \mathcal{L} \) do
      4. if \( \varphi \in J \) then
         5. Compute \( L_{N}^{(i)}[\varphi] \) and \( L_{N}^{(i)}[\varphi] \);
         6. Copy the path \( \ell \) into a new path \( \ell' \neq \mathcal{L} : (v_{\ell'}[\varphi], J_{PM}^{(i)} \leftarrow (0, \phi(J_{PM}^{(i-1)}) L_{n}^{(i)}[\varphi], L_{n}^{(i)}[\varphi], 0));
         7. \( (v_{\ell'}[\varphi], J_{PM}^{(i)} \leftarrow (1, \phi(J_{PM}^{(i-1)}) L_{n}^{(i)}[\varphi], L_{n}^{(i)}[\varphi], 1));
         8. \( i_c \leftarrow i_c + 1, \) \( i_s \leftarrow i_s + 1; \)
      9. else if \( \varphi \in W \) then
         10. Compute \( L_{N}^{(i)}[\varphi]; \)
         11. Copy the path \( \ell \) into a new path \( \ell' \neq \mathcal{L} : (v_{\ell'}[\varphi], J_{PM}^{(i)} \leftarrow (0, \phi(J_{PM}^{(i-1)}) L_{n}^{(i)}[\varphi], 0));
         12. \( (v_{\ell'}[\varphi], J_{PM}^{(i)} \leftarrow (1, \phi(J_{PM}^{(i-1)}) L_{n}^{(i)}[\varphi], 1));
         13. \( i_c \leftarrow i_c + 1; \)
      14. else
         15. Compute \( L_{N}^{(i)}[\varphi] ;\)
         16. \( (v_{\ell'}[\varphi], J_{PM}^{(i)} \leftarrow (v_{\ell} \cdot \phi(J_{PM}^{(i-1)}) L_{n}^{(i)}[\varphi], 0));
         17. \( i_c \leftarrow i_c + 1; \)
      18. Discard all but \( L \) paths with the smallest \( J_{PM}^{(i)} \);
      19. \( \ell' \leftarrow \arg\min_{\ell} J_{PM}^{(i)} \);
      20. \( \hat{s}_{1}^{N_c} \leftarrow \hat{v}_{\ell'}[\ell'] G_{N_c} \); 
      21. return \( \hat{s}_{1}^{N_c} \).

where \( W(y|x) \) is the probability distribution function of the output letter \( y \) when \( x \) is transmitted. Correspondingly, for the source SC decoding, we have
\[
L_{0}^{(i)}[\varphi] \triangleq \ln \frac{1 - P_{S}(1)}{P_{S}(1)}, \quad i \in [N_c].
\]
The calculation of partial sums is written as
\[
\begin{align*}
    u^{(2i-2)}[\varphi] & \triangleq u^{(2i-2)}[\varphi] \triangleq u^{(2i+1)}[\varphi] \triangleq u^{(2i+1)}[\varphi] \triangleq u^{(2i+1)}[\varphi],
\end{align*}
\]
which is computed starting from \( u^{(1)}[\varphi] \). Since the JSC nodes connect both the trellis of channel polar code and that of source polar code, the update of the JPM needs to merge the channel decision LLRs \( L_{n}^{(i)}[\varphi] \) and the source decision LLRs \( L_{n}^{(i)}[\varphi] \). When the decision LLRs \( L_{n}^{(i)}[\varphi] \) and \( L_{n}^{(i)}[\varphi] \) are obtained, for the bits \( \{v_{\varphi} | \varphi \in J \} \), the update of JPM \( \phi \) is written as
\[
J_{PM}^{(i)}[\varphi] = \phi(J_{PM}^{(i-1)}), \quad L_{n}^{(i)}[\varphi], L_{n}^{(i)}[\varphi], \hat{v}_{\varphi}[\ell]),
\]
where the function $\tilde{\phi}$ is defined as

$$
\tilde{\phi}(\mu, \lambda, v) \triangleq \mu + \ln(1 + e^{(2v-1)\lambda}) + \ln(1 + e^{(2v-1)\lambda})
$$

(8)

2) Low-entropy node decoding: The decoding operation of the low-entropy node is shown in Lines 11-15 of Algorithm 2. Similar to the level $\varphi \in J$, we copy the path $\ell$ into a new path $\ell' \notin \mathcal{L}$ at the level $\varphi \notin \mathcal{V}$. The path $\ell$ and $\ell'$ have the same prefix, e.g., $\tilde{v}^{(\varphi)}_i[\ell]$. The last bit of the path $\ell$ is set to “0”, and the last bit of path $\ell'$ is set to “1”. Since $v_\varphi$ correspond to a low-entropy node, the update of $\text{JPM}_\ell^{(\varphi)}$ only need to consider the source decision LLR $L_n^{(\varphi)}(\ell)$, which can be written as

$$
\text{JPM}_\ell^{(\varphi)} = \phi(\text{JPM}_\ell^{(\varphi-1)}, L_n^{(\varphi)}(\ell), \tilde{v}_\varphi[\ell]).
$$

(9)

The function $\phi(\mu, \lambda, v)$ is defined as

$$
\phi(\mu, \lambda, v) \triangleq \mu + \ln(1 + e^{(2v-1)\lambda}).
$$

(10)

3) Frozen node decoding: The decoding operation of the frozen node is shown in Lines 17-19 of Algorithm 2. Let $\tilde{A} = J \cup W$. The subvector $v_{\tilde{A}^c}$ equals the frozen bits $u_{\tilde{A}^c}$, which is known both to the transmitter and the receiver. Therefore, the $\tilde{v}_\varphi[\ell]$ can be directly set to $v_\varphi$ at each level $\varphi \notin (J \cup W)$. The update of $\text{JPM}_\ell^{(\varphi)}$ only need to consider the channel decision LLR $L_n^{(\varphi)}(\ell)$, which can be written as

$$
\text{JPM}_\ell^{(\varphi)} = \phi(\text{JPM}_\ell^{(\varphi-1)}, L_n^{(\varphi)}(\ell), v_\varphi).
$$

(11)

When the decoding level $\varphi$ reaches $N$, the path $\ell$ with the smallest $\text{JPM}_\ell^{(N)}$ is selected as the final decoding path $\ell^*$. The estimate $\tilde{s}^N_1$ of $s^N_1$ equals $\tilde{v}_{\tilde{A}}[\ell^*]$, and the source decision $s^N_1$ can be calculated by

$$
\tilde{s}^N_1 = \tilde{v}_{\tilde{A}}[\ell^*]G_{N_s}
$$

(12)

which is shown in Line 21 Algorithm 2.

III. PERFORMANCE EVALUATION

In this section, we provide simulation results and analysis of the proposed J-SCL decoder over the binary phase shift keying (BPSK) modulated additive white Gaussian noise (AWGN) channel. The polar code used in simulations is constructed via Gaussian approximation [17]. The length of source vector is $N_s = 512$, and the overall rate $R$ is set to 1/2. The maximum of simulated frame number is $10^8$, and the simulation terminates when the error frame number reaches $10^2$ at each $E_b/N_0$. Note that $E_b$ refers to energy per source bit.

A. Simulation Results

In Fig. 3 we compare the BER performance of various decoding schemes for D-Polar codes with $p = 0.07$ and $R_s = 0.6$. The TL-BP decoder can be referred to [16]. And the Separate SCL (Sep. SCL) decoder refers to a channel SCL decoder followed by an independent source SCL decoder. From this figure, we can see that the performance gain of J-SCL ($L = 4$) decoding versus Sep. SCL ($L = 32$) decoding can reach 0.96 dB at $BER = 10^{-4}$. When the list size $L$ increases to 8 and 32, the gain of J-SCL decoding compared to the Sep. SCL ($L = 32$) decoding increases to 1.22 dB and 1.54 dB at $BER = 10^{-4}$, respectively. The gain of J-SCL decoding with respect to Sep. SCL decoding is due to the residual redundancy of the source code considered in the channel decoding. The residual redundancy that is not removed by the source polar encoder gives additional information to the channel SCL decoding. For the Sep. SCL decoder, the independent channel SCL decoder cannot benefit from the residual redundancy. Compared with existing JSCD scheme of D-Polar codes, i.e., TL-BP decoder [16], the J-SCL ($L = 32$) can obtain 0.36 dB gain at $BER = 10^{-3}$. One of the weaknesses of the TL-BP decoder is the high error floor, which is due to the residual decoding errors introduced by the limited length of source code. In contrast, a significant advantage of the J-SCL ($L = 32$) is the reduction of the error floor from $10^{-4}$ to $10^{-6}$. The list size $L$ affects the error floor of the J-SCL decoding. When the list size is reduced to 8 and 4, the error floor will be higher than $10^{-5}$. The lower error floor of

![Fig. 3. BER performance of TL-BP decoder, Separate SCL decoder and J-SCL decoder.](image-url)

![Fig. 4. BER performance of J-SCL ($L = 32$) decoding and TL-BP decoding with various $P_0(1) = p$ and the compression rate $R_e$.](image-url)
the J-SCL decoder is due to the better performance of source decoding using SCL decoder than BP decoder. Moreover, this figure gives the impact on the performance of the TL-BP decoder caused by replacing the SPC in D-Polar codes with the non-systematic polar code (NSPC). The TL-BP decoder for D-Polar codes using NSPC has 0.42 performance loss compared with the TL-BP decoder for original D-Polar codes \[16\] at BER = 10\(^{-3}\), while the error floor is close. The performance gain of J-SCL (\(L = 32\)) decoding versus TL-BP (NSPC) decoding can reach 0.78 dB at BER = 10\(^{-3}\).

In Fig. 4, we compare the BER performance of J-SCL decoding and TL-BP decoding at different \(p\) and \(R_s\). In this simulation, the list size \(L\) is set to 32. The J-SCL decoder and TL-BP decoder use the same transmitter. From this figure, we can see that the J-SCL outperforms the TL-BP decoder with various \(p\) and \(R_s\). The performance gap between the J-SCL decoding and TL-BP decoding widens as the \(E_b/N_0\) increases. For the J-SCL decoder with \(p = 0.07\), the performance gain of \(R_s = 0.5\) versus \(R_s = 0.6\) reaches 0.84 dB. Given a constant overall rate \(R\), this additional gain is due to the lower channel coding rate \(R_c\). However, the compression rate of \(R_s = 0.5\) causes the error floor to rise to 10\(^{-3}\). The compression rate \(R_s\) trades off the coding gain and the error floor. Moreover, this figure shows that a lower entropy will result in a better BER performance. For the J-SCL decoder with \(R_s = 0.5\), the performance gain of \(p = 0.04\) versus \(p = 0.07\) reaches 0.32 dB, and the error floor falls from 10\(^{-3}\) to 10\(^{-5}\). Moreover, Fig. 5 provides a comparison of the J-SCL (\(L=32\)) decoder and the TL-BP decoder with the overall rate \(R = 1\). The length of source vector is increased to 1024, and \(R_s (R_c)\) is set to 1/2. For \(p = 0.07\), the J-SCL decoder achieves 1.47 dB gain compared with the TL-BP decoder at BER = 10\(^{-3}\). For \(p = 0.04\), the J-SC decoder outperforms the TL-BP decoder by about 0.64 dB at BER = 10\(^{-4}\). Compared with the error floor of the TL-BP decoder at 10\(^{-5}\), the BER curve of the J-SCL decoder remains without an error floor at 10\(^{-7}\).

**B. Complexity Analysis**

Recall Section II, we can see that the calculation of \(I_{\text{NSP}} (c)\) involves a channel SC decoding, and the calculation of \(I_{\text{NSP}} (c)\) involves a source SC decoding. Therefore, the complexity of J-SCL decoding is \(O(LN_c \log N_e + LN_s \log N_s)\). As shown in \[16\], the TL-BP decoder contains a channel BP decoder and a source BP decoder. Let \(f\) denote the maximum number of iterations of TL-BP. Then, the complexity of the TL-BP decoder can be represented by \(O(IN_c \log N_e + IN_s \log N_s)\). Although the TL-BP decoder can use the early stopping criterion to obtain a very low average number of iterations in the high \(E_b/N_0\) region, the iteration number is still \(f\) in the worst case. The J-SCL decoder outperforms in terms of compressibility as long as \(I\) is greater than \(L\).

**IV. Conclusion**

In this letter, we propose a J-SCL decoder for the D-Polar codes JSCC scheme. To facilitate the design of the J-SCL decoder, we replaced the SPC in the D-Polar codes with an NSPC. Then, we give a method to combine the trellis of the channel polar code and that of the source polar code into a compound trellis, where a JSC node represents both the information bit and the high-entropy bit. Based on the compound trellis, the J-SCL decoder is proposed by performing path splitting on the JSC nodes and the low-entropy nodes. Simulation results show that the proposed JSCD outperforms the TL-BP decoder.

**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, July 2009.

[2] E. Arıkan, “A performance comparison of polar codes and reed-muller codes,” IEEE Comm. Lett., vol. 12, no. 6, pp. 447–449, jun 2008.

[3] K. Chen, K. Niu, and J. Lin, “List successive cancellation decoding of polar codes,” Electron. Lett., vol. 48, no. 9, p. 500, 2012.

[4] K. Niu and K. Chen, “CRC-Aided decoding of polar codes,” IEEE Commun. Lett., vol. 16, no. 10, pp. 1668–1671, 2012.

[5] I. Tal and A. Vardy, “List decoding of polar codes,” IEEE Trans. Inf. Theory, vol. 61, no. 5, pp. 2213–2226, 2015.

[6] 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, 2015.

[7] E. Arıkan, “Source polarization,” in Proc. IEEE Int. Symp. Inf. Theory. IEEE, June 2010.

[8] S. B. Korada and R. L. Urbanke, “Polar codes are optimal for lossy source coding,” IEEE Trans. Inf. Theory, vol. 56, no. 4, pp. 1751–1768, April 2010.

[9] H. S. Cronie and S. B. Korada, “Lossless source coding with polar codes,” in Proc. IEEE Int. Symp. Inf. Theory, 2010, pp. 904–908.

[10] Y. Wang, K. R. Narayanan, and A. A. Jiang, “Exploiting source redundancy to improve the rate of polar codes,” in 2017 IEEE Int. Symp. Inf. Theory (ISIT), 2017, pp. 864–868.

[11] L. Jin, P. Yang, and H. Yang, “Distributed joint source-channel decoding using systematic polar codes,” IEEE Commun. Lett., vol. 22, no. 1, pp. 49–52, 2018.

[12] L. Jin and H. Yang, “Joint source-channel polarization with side information,” IEEE Access, vol. 6, pp. 7340–7349, 2018.

[13] M. Fresia, F. Perez-Cruz, H. V. Poor, and S. Verdú, “Joint source and channel coding,” IEEE Signal Process. Magazine, vol. 27, no. 6, pp. 104–113, 2010.

[14] Jiguang He, Lin Wang, and Pingping Chen, “A joint source and channel coding scheme based on simple protograph structured codes,” in Proc. Int. Symp. Commun. Inf. Technol. (ISCIT), 2012, pp. 65–69.

Fig. 5. BER performance of J-SCL (\(L = 32\)) decoding and TL-BP decoding with the overall rate \(R = 1\).
[15] S. Liu, L. Wang, J. Chen, and S. Hong, “Joint component design for
the JSCC system based on DP-LDPC codes,” IEEE Trans. Commun.,
vol. 68, no. 9, pp. 5808–5818, 2020.

[16] Y. Dong, K. Niu, J. Dai, S. Wang, and Y. Yuan, “Joint source and channel
coding using double polar codes,” IEEE Commun. Lett., vol. 25, no. 9,
pp. 2810–2814, sep 2021.

[17] P. Trifonov, “Efficient design and decoding of polar codes,” IEEE Trans.
Commun., vol. 60, no. 11, pp. 3221–3227, 2012.