Optimum Decoding of Modified Polar Codes: From Inactivation Decoding to Tree-Search

Mustafa Cemil Coșkun
{mustafa.coskun}@tum.de

Institute for Communications Engineering (LNT)
Technical University of Munich (TUM)

Ferienakademie 2021

Polar Codes

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 55, NO. 7, JULY 2009 3051

Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels

Erdal Arıkan, *Senior Member, IEEE*

***Abstract*—A method is proposed, called channel polarization, to construct code sequences that achieve the symmetric capacity $I(W)$ of any given binary-input discrete memoryless channel (B-DMC) W. The symmetric capacity is the highest rate achievable subject to using the input letters of the channel with equal probability. Channel polarization refers to the fact that it is pos-**

A. Preliminaries

We write $W : \mathcal{X} \rightarrow \mathcal{Y}$ to denote a generic B-DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition probabilities $W(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The input alphabet $\mathcal{X}$ will always be $\{0, 1\}$, the output alphabet and the transition probabilities may

- They are capacity-achieving on binary memoryless symmetric (BMS) channels with low encoding/decoding complexity [Arı09].

Polar Codes

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 55, NO. 7, JULY 2009 3051

Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels

Erdal Arıkan, *Senior Member, IEEE*

***Abstract*—A method is proposed, called channel polarization, to construct code sequences that achieve the symmetric capacity $I(W)$ of any given binary-input discrete memoryless channel (B-DMC) W. The symmetric capacity is the highest rate achievable subject to using the input letters of the channel with equal probability. Channel polarization refers to the fact that it is pos-**

A. Preliminaries

We write $W : \mathcal{X} \to \mathcal{Y}$ to denote a generic B-DMC with input alphabet $\mathcal{X}$, output alphabet $\mathcal{Y}$, and transition probabilities $W(y|x)$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$. The input alphabet $\mathcal{X}$ will always be $\{0, 1\}$, the output alphabet and the transition probabilities may

- They are capacity-achieving on binary memoryless symmetric (BMS) channels with low encoding/decoding complexity [Arı09].
- But successive cancellation (SC) decoding performs poorly for small blocks.

Successive List Cancellation Decoding

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 61, NO. 5, MAY 2015 2213

List Decoding of Polar Codes

Ido Tal, *Member, IEEE* and Alexander Vardy, *Fellow, IEEE*

***Abstract*— We describe a successive-cancellation list decoder for polar codes, which is a generalization of the classic successive-cancellation decoder of Arıkan. In the proposed list decoder, *L* decoding paths are considered concurrently at each decoding stage, where *L* is an integer parameter. At the end of the decoding process, the most likely among the *L* paths is selected as the single codeword at the decoder output. Simulations show that the resulting performance is very close to that of maximum-likelihood decoding, even for moderate values of *L*. Alternatively, if a genie is allowed to pick the transmitted codeword from the list, the results are comparable with the performance of current state-of-the-art LDPC codes. We show that such a genie can be easily implemented using simple CRC precoding. The specific list-decoding algorithm that achieves this performance doubles the number of decoding paths for each information bit, and then uses a pruning procedure to discard all but the *L* most likely paths. However, straightforward implementation of this**

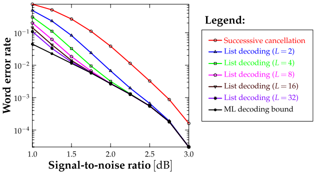


Fig. 1. List-decoding performance for a polar code of length $n = 2048$ and rate $R = 0.5$ on the BPSK-modulated Gaussian channel. The code was constructed using the methods of [15], with optimization for $E_b/N_0 = 2$ dB.

- SC list (SCL) decoding with CRC and large list-size performs very well and approaches maximum-likelihood (ML) decoding performance [TV15].

Successive List Cancellation Decoding

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 61, NO. 5, MAY 2015 2213

List Decoding of Polar Codes

Ido Tal, *Member, IEEE* and Alexander Vardy, *Fellow, IEEE*

***Abstract*—We describe a successive-cancellation list decoder for polar codes, which is a generalization of the classic successive-cancellation decoder of Arıkan. In the proposed list decoder, L decoding paths are considered concurrently at each decoding stage, where L is an integer parameter. At the end of the decoding process, the most likely among the L paths is selected as the single codeword at the decoder output. Simulations show that the resulting performance is very close to that of maximum-likelihood decoding, even for moderate values of L. Alternatively, if a genie is allowed to pick the transmitted codeword from the list, the results are comparable with the performance of current state-of-the-art LDPC codes. We show that such a genie can be easily implemented using simple CRC precoding. The specific list-decoding algorithm that achieves this performance doubles the number of decoding paths for each information bit, and then uses a pruning procedure to discard all but the L most likely paths. However, straightforward implementation of this**

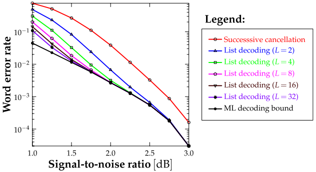


Fig. 1. List-decoding performance for a polar code of length $n = 2048$ and rate $R = 0.5$ on the BPSK-modulated Gaussian channel. The code was constructed using the methods of [15], with optimization for $E_b/N_0 = 2\,\text{dB}$.

- SC list (SCL) decoding with CRC and large list-size performs very well and approaches maximum-likelihood (ML) decoding performance [TV15].
- It can also be used to decode other codes (e.g., Reed–Muller codes).

Polar Codes with Dynamic Frozen Bits

254 IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 34, NO. 2, FEBRUARY 2016

Polar Subcodes

Peter Trifonov, *Member, IEEE*, and Vera Miloslavskaya, *Member, IEEE*

***Abstract*—An extension of polar codes is proposed, which allows some of the frozen symbols, called dynamic frozen symbols, to be data-dependent. A construction of polar codes with dynamic frozen symbols, being subcodes of extended BCH codes, is proposed. The proposed codes have higher minimum distance than classical polar codes, but still can be efficiently decoded using the successive cancellation algorithm and its extensions. The codes with Arikan, extended BCH and Reed-Solomon kernel are considered. The proposed codes are shown to outperform LDPC and turbo codes, as well as polar codes with CRC.**

RM codes, and are therefore likely to provide better finite length performance. However, there are still no efficient MAP decoding algorithms for these codes.

It was suggested in [17] to construct subcodes of RM codes, which can be efficiently decoded by a recursive list decoding algorithm. In this paper we generalize this approach, and propose a code construction "in between" polar codes and EBCH codes. The proposed codes can be efficiently decoded using the techniques developed in the area of polar coding, but provide

- Later, polar codes were extended with the concept of dynamic frozen bits, which enabled state-of-art designs.

Polar Codes with Dynamic Frozen Bits

254 IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 34, NO. 2, FEBRUARY 2016

Polar Subcodes

Peter Trifonov, *Member, IEEE*, and Vera Miloslavskaya, *Member, IEEE*

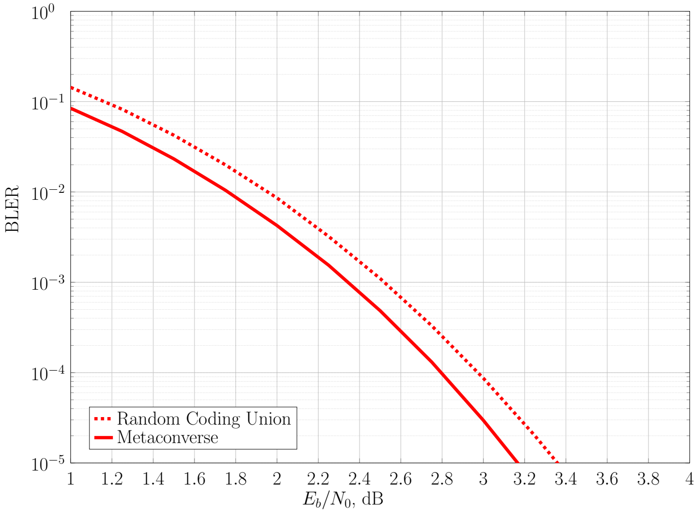
***Abstract*—An extension of polar codes is proposed, which allows some of the frozen symbols, called dynamic frozen symbols, to be data-dependent. A construction of polar codes with dynamic frozen symbols, being subcodes of extended BCH codes, is proposed. The proposed codes have higher minimum distance than classical polar codes, but still can be efficiently decoded using the successive cancellation algorithm and its extensions. The codes with Arikan, extended BCH and Reed-Solomon kernel are considered. The proposed codes are shown to outperform LDPC and turbo codes, as well as polar codes with CRC.**

RM codes, and are therefore likely to provide better finite length performance. However, there are still no efficient MAP decoding algorithms for these codes.

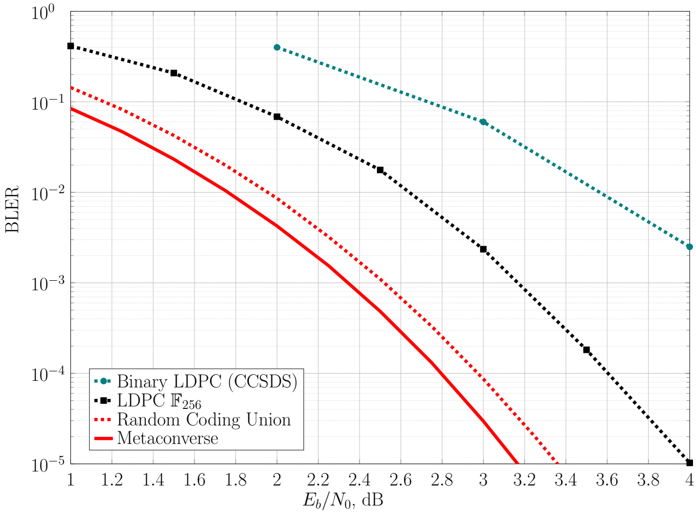
It was suggested in [17] to construct subcodes of RM codes, which can be efficiently decoded by a recursive list decoding algorithm. In this paper we generalize this approach, and propose a code construction "in between" polar codes and EBCH codes. The proposed codes can be efficiently decoded using the techniques developed in the area of polar coding, but provide

- Later, polar codes were extended with the concept of dynamic frozen bits, which enabled state-of-art designs.
- It is also shown that any code can be decoded using SCL decoding, but some require very large complexity for a good performance.

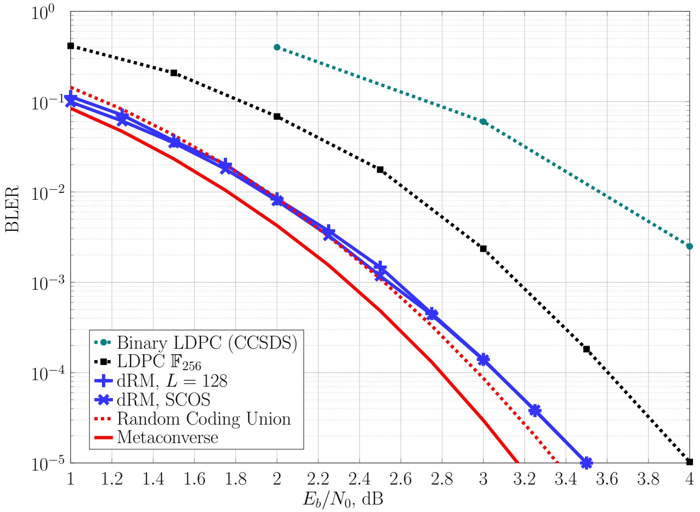
$N = 128, k = 64$


BLER
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 3.2 3.4 3.6 3.8 4
E_b/N_0, dB
Random Coding Union
Metaconverse

$N = 128, k = 64$


BLER
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 3.2 3.4 3.6 3.8 4
E_b/N_0, dB
Binary LDPC (CCSDS)
LDPC $\mathbb{F}_{256}$
Random Coding Union
Metaconverse

$N = 128, k = 64$


BLER
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
E_b/N_0, dB
Binary LDPC (CCSDS)
LDPC $\mathbb{F}_{256}$
dRM, $L = 128$
dRM, SCOS
Random Coding Union
Metaconverse

Outline

1. Overview of Polar Codes
2. Successive Cancellation Inactivation Decoding
3. Successive Cancellation Ordered Search Decoding
4. Conclusions

Outline

1 Overview of Polar Codes

2 Successive Cancellation Inactivation Decoding

3 Successive Cancellation Ordered Search Decoding

4 Conclusions

Easy Channels

Among all, there are channels for which it is easy to communicate optimally:

Easy Channels

Among all, there are channels for which it is easy to communicate optimally:

- Noiseless channels: The output Y determines the input X (i.e., $H(X|Y) \approx 0$).
- Useless channels: The output Y is independent from the input X (i.e., $H(X|Y) \approx 1$).

Easy Channels

Among all, there are channels for which it is easy to communicate optimally:

- Noiseless channels: The output Y determines the input X (i.e., $H(X|Y) \approx 0$).
- Useless channels: The output Y is independent from the input X (i.e., $H(X|Y) \approx 1$).

Channel polarization is a technique to convert any BMS channel to a mixture of easy channels, asymptotically in the block length.

Easy Channels

Among all, there are channels for which it is easy to communicate optimally:

- Noiseless channels: The output Y determines the input X (i.e., $H(X|Y) \approx 0$).
- Useless channels: The output Y is independent from the input X (i.e., $H(X|Y) \approx 1$).

Channel polarization is a technique to convert any BMS channel to a mixture of easy channels, asymptotically in the block length.

- The technique is lossless in terms of mutual information (required to achieve the capacity).

Easy Channels

Among all, there are channels for which it is easy to communicate optimally:

- Noiseless channels: The output Y determines the input X (i.e., $H(X|Y) \approx 0$).
- Useless channels: The output Y is independent from the input X (i.e., $H(X|Y) \approx 1$).

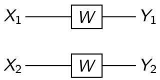
Channel polarization is a technique to convert any BMS channel to a mixture of easy channels, asymptotically in the block length.

- The technique is lossless in terms of mutual information (required to achieve the capacity).
- The technique is of low complexity (there exists an encoder-decoder pair, realizing the technique with $\mathcal{O}(N \log N)$ complexity, where N is the block length).

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$



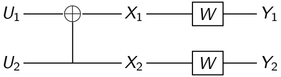
Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$



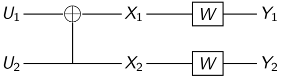
$$X_1^2 = U_1^2 \mathrm{G}_2$$

$$\mathrm{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$



we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 G_2$$

$$G_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \Bigg\{$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathrm{G}_2$$

$$\mathrm{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 \\ \\ \\ \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 \quad \checkmark \\ & \\ & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 \quad \checkmark \\ (?, U_2) & \text{w.p. } \epsilon(1-\epsilon) \\ & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2) & \text{w.p. } \epsilon(1-\epsilon) & \times \\ & & \\ & & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2) & \text{w.p. } \epsilon(1-\epsilon) & \times \\ (U_1 \oplus U_2, ?) & \text{w.p. } (1-\epsilon)\epsilon & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2) & \text{w.p. } \epsilon(1-\epsilon) & \times \\ (U_1 \oplus U_2, ?) & \text{w.p. } (1-\epsilon)\epsilon & \times \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0, 1\} \to \{0, 1, ?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2) & \text{w.p. } \epsilon(1-\epsilon) & \times \\ (U_1 \oplus U_2, ?) & \text{w.p. } (1-\epsilon)\epsilon & \times \\ (?, ?) & \text{w.p. } \epsilon^2 & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- Estimate U_1 by observing the output (Y_1, Y_2):

$$(Y_1, Y_2) = \begin{cases} (U_1 \oplus U_2, U_2) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2) & \text{w.p. } \epsilon(1-\epsilon) & \times \\ (U_1 \oplus U_2, ?) & \text{w.p. } (1-\epsilon)\epsilon & \times \\ (?, ?) & \text{w.p. } \epsilon^2 & \times \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.

$$X_1^2 = U_1^2 \mathrm{G}_2$$

$$\mathrm{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \Bigg\{$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 \\ & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 \quad \checkmark \\ & \\ & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 \quad \checkmark \\ (?, U_2, U_1) & \text{w.p. } \epsilon(1-\epsilon) \\ & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2, U_1) & \text{w.p. } \epsilon(1-\epsilon) & \checkmark \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 \quad \checkmark \\ (?, U_2, U_1) & \text{w.p. } \epsilon(1-\epsilon) \quad \checkmark \\ (U_1 \oplus U_2, ?, U_1) & \text{w.p. } (1-\epsilon)\epsilon \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2, U_1) & \text{w.p. } \epsilon(1-\epsilon) & \checkmark \\ (U_1 \oplus U_2, ?, U_1) & \text{w.p. } (1-\epsilon)\epsilon & \checkmark \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2, U_1) & \text{w.p. } \epsilon(1-\epsilon) & \checkmark \\ (U_1 \oplus U_2, ?, U_1) & \text{w.p. } (1-\epsilon)\epsilon & \checkmark \\ (?, ?, U_1) & \text{w.p. } \epsilon^2 & \end{cases}$$

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

- U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Assume now that U_1 is given. Estimate U_2 by observing (Y_1, Y_2, U_1):

$$(Y_1, Y_2, U_1) = \begin{cases} (U_1 \oplus U_2, U_2, U_1) & \text{w.p. } (1-\epsilon)^2 & \checkmark \\ (?, U_2, U_1) & \text{w.p. } \epsilon(1-\epsilon) & \checkmark \\ (U_1 \oplus U_2, ?, U_1) & \text{w.p. } (1-\epsilon)\epsilon & \checkmark \\ (?, ?, U_1) & \text{w.p. } \epsilon^2 & \times \end{cases}$$

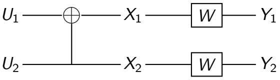
Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$



$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- The input U_1 is erased w.p. $(1-(1-\epsilon)^2)$.
- Given U_1, the input U_2 is erased w.p. $\epsilon)^2$.

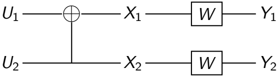
Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$



$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- The input U_1 is erased w.p. $(2\epsilon - \epsilon^2)$.
- Given U_1, the input U_2 is erased w.p. ϵ^2.

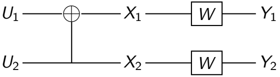
Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$



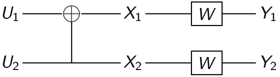
$$X_1^2 = U_1^2 \mathsf{G}_2$$
$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

- The input U_1 is erased w.p. $(2\epsilon - \epsilon^2)$, i.e., $H(U_1|Y_1^2) = 2\epsilon - \epsilon^2$.
- Given U_1, the input U_2 is erased w.p. ϵ^2, i.e., $H(U_2|Y_1^2 U_1) = \epsilon^2$.

Example: Binary Erasure Channel

Given two independent copies of a BEC(ϵ) $W : \{0,1\} \to \{0,1,?\}$, i.e.,

$$Y = \begin{cases} X & \text{w.p. } 1-\epsilon \\ ? & \text{w.p. } \epsilon \end{cases}$$



$$X_1^2 = U_1^2 \mathsf{G}_2$$

$$\mathsf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

we set

$$X_1 = U_1 \oplus U_2$$
$$X_2 = U_2$$

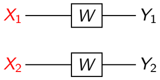
- The input U_1 is erased w.p. $(2\epsilon - \epsilon^2)$, i.e., $H(U_1|Y_1^2) = 2\epsilon - \epsilon^2$.
- Given U_1, the input U_2 is erased w.p. ϵ^2, i.e., $H(U_2|Y_1^2 U_1) = \epsilon^2$.

Hence, we have

$$2\epsilon - \epsilon^2 \geq H(X_1|Y_1) = \epsilon \geq \epsilon^2 \quad \text{with equality if and only if } \epsilon \in \{0,1\}$$

Polarized Synthetic Channels: General BMSCs

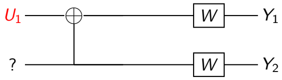
Given two independent copies of $W : \{0,1\} \to \mathcal{Y}$ with a capacity of $C(W)$, we obtain two synthetic channels:



Polarized Synthetic Channels: General BMSCs

Given two independent copies of $W : \{0,1\} \to \mathcal{Y}$ with a capacity of $C(W)$, we obtain two synthetic channels:

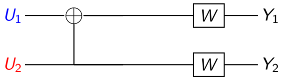
- A downgraded channel $W_2^{(1)} : \{0,1\} \to \mathcal{Y}^2$ having input U_1 and output Y_1^2 with $C(W_2^{(1)}) < C(W)$



Polarized Synthetic Channels: General BMSCs

Given two independent copies of $W : \{0,1\} \to \mathcal{Y}$ with a capacity of $C(W)$, we obtain two synthetic channels:

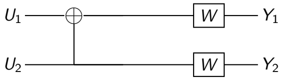
- A downgraded channel $W_2^{(1)} : \{0,1\} \to \mathcal{Y}^2$ having input U_1 and output Y_1^2 with $C(W_2^{(1)}) < C(W)$
- An upgraded channel $W_2^{(2)} : \{0,1\} \to \mathcal{Y}^2 \times \{0,1\}$ having input U_2 and output (Y_1^2, U_1) with $C(W_2^{(2)}) > C(W)$



Polarized Synthetic Channels: General BMSCs

Given two independent copies of $W : \{0,1\} \to \mathcal{Y}$ with a capacity of $C(W)$, we obtain two synthetic channels:

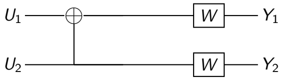
- A downgraded channel $W_2^{(1)} : \{0,1\} \to \mathcal{Y}^2$ having input U_1 and output Y_1^2 with $C(W_2^{(1)}) < C(W)$
- An upgraded channel $W_2^{(2)} : \{0,1\} \to \mathcal{Y}^2 \times \{0,1\}$ having input U_2 and output (Y_1^2, U_1) with $C(W_2^{(2)}) > C(W)$



This suggests that a successive decoding can be employed [Sto02] to achieve $C(W)$ [Arı09]:

Polarized Synthetic Channels: General BMSCs

Given two independent copies of $W : \{0,1\} \to \mathcal{Y}$ with a capacity of $C(W)$, we obtain two synthetic channels:



- A downgraded channel $W_2^{(1)} : \{0,1\} \to \mathcal{Y}^2$ having input U_1 and output Y_1^2 with $C(W_2^{(1)}) < C(W)$
- An upgraded channel $W_2^{(2)} : \{0,1\} \to \mathcal{Y}^2 \times \{0,1\}$ having input U_2 and output (Y_1^2, U_1) with $C(W_2^{(2)}) > C(W)$

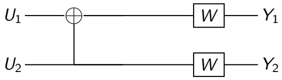
This suggests that a successive decoding can be employed [Sto02] to achieve $C(W)$ [Arı09]:

- Transmit at a rate $C(W_2^{(1)})$, where the decoder takes Y_1^2 as input and outputs $\hat{U}_1$.

Polarized Synthetic Channels: General BMSCs

Given two independent copies of $W : \{0, 1\} \to \mathcal{Y}$ with a capacity of $C(W)$, we obtain two synthetic channels:

- A downgraded channel $W_2^{(1)} : \{0, 1\} \to \mathcal{Y}^2$ having input U_1 and output Y_1^2 with $C(W_2^{(1)}) < C(W)$
- An upgraded channel $W_2^{(2)} : \{0, 1\} \to \mathcal{Y}^2 \times \{0, 1\}$ having input U_2 and output (Y_1^2, U_1) with $C(W_2^{(2)}) > C(W)$

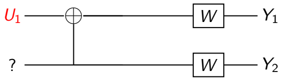


This suggests that a successive decoding can be employed [Sto02] to achieve $C(W)$ [Arı09]:

- Transmit at a rate $C(W_2^{(1)})$, where the decoder takes Y_1^2 as input and outputs $\hat{U}_1$.
- Then, transmit at a rate $C(W_2^{(2)})$, where the decoder uses $(Y_1^2, \hat{U}_1)$ to output $\hat{U}_2$.

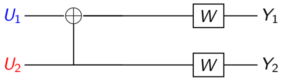
Genie-Aided vs. Real Successive Decoder

- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓

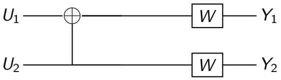


Genie-Aided vs. Real Successive Decoder

- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓
- The channel $W_2^{(2)}$ has the input U_2 and output (Y_1^2, U_1)!



Genie-Aided vs. Real Successive Decoder

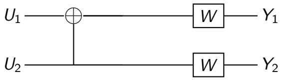


- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓
- The channel $W_2^{(2)}$ has the input U_2 and output (Y_1^2, U_1)!

It is possible to obtain $\hat{U}_1$ by first decoding $W_2^{(1)}$. What is the effect of using $\hat{U}_1$ instead of U_1 on the block error events?

Genie-Aided vs. Real Successive Decoder

- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓
- The channel $W_2^{(2)}$ has the input U_2 and output (Y_1^2, U_1)!



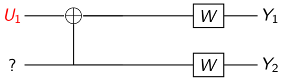
It is possible to obtain $\hat{U}_1$ by first decoding $W_2^{(1)}$. What is the effect of using $\hat{U}_1$ instead of U_1 on the block error events?

Genie-aided successive decoding:

Real successive decoding:

Genie-Aided vs. Real Successive Decoder

- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓
- The channel $W_2^{(2)}$ has the input U_2 and output (Y_1^2, U_1)!



It is possible to obtain $\hat{U}_1$ by first decoding $W_2^{(1)}$. What is the effect of using $\hat{U}_1$ instead of U_1 on the block error events?

Genie-aided successive decoding:

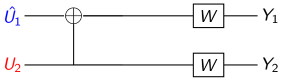
$$\tilde{U}_1 = f_1(Y_1^2)$$

Real successive decoding:

$$\hat{U}_1 = f_1(Y_1^2)$$

Genie-Aided vs. Real Successive Decoder

- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓
- The channel $W_2^{(2)}$ has the input U_2 and output (Y_1^2, U_1)!



It is possible to obtain $\hat{U}_1$ by first decoding $W_2^{(1)}$. What is the effect of using $\hat{U}_1$ instead of U_1 on the block error events?

Genie-aided successive decoding:

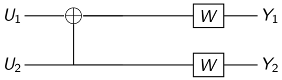
$$\tilde{U}_1 = f_1(Y_1^2)$$
$$\tilde{U}_2 = f_2(Y_1^2 U_1)$$

Real successive decoding:

$$\hat{U}_1 = f_1(Y_1^2)$$
$$\hat{U}_2 = f_2(Y_1^2 \hat{U}_1)$$

Genie-Aided vs. Real Successive Decoder

- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓
- The channel $W_2^{(2)}$ has the input U_2 and output (Y_1^2, U_1)!



$$\{\hat{U}_1^2 \neq U_1^2\} = \{\tilde{U}_1^2 \neq U_1^2\}$$

It is possible to obtain $\hat{U}_1$ by first decoding $W_2^{(1)}$. What is the effect of using $\hat{U}_1$ instead of U_1 on the block error events?

Genie-aided successive decoding:

$$\tilde{U}_1 = f_1(Y_1^2)$$
$$\tilde{U}_2 = f_2(Y_1^2 U_1)$$

Real successive decoding:

$$\hat{U}_1 = f_1(Y_1^2)$$
$$\hat{U}_2 = f_2(Y_1^2 \hat{U}_1)$$

The real decoder makes an error IF AND ONLY IF the genie-aided decoder makes an error!

Polar Transform - Recursive Application of the Basic Transform

Definition

The Kronecker product of two matrices X and Y is

$$\mathsf{X} \otimes \mathsf{Y} \triangleq \begin{bmatrix} x_{1,1}\mathsf{Y} & x_{1,2}\mathsf{Y} & \cdots \\ x_{2,1}\mathsf{Y} & x_{2,2}\mathsf{Y} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Then, a Kronecker power of a matrix is written as $\mathsf{X}^{\otimes n} = \mathsf{X}^{\otimes(n-1)} \otimes \mathsf{X}$, $\mathsf{X}^{\otimes 0} \triangleq 1$.

Polar Transform - Recursive Application of the Basic Transform

Definition

The Kronecker product of two matrices X and Y is

$$\mathsf{X} \otimes \mathsf{Y} \triangleq \begin{bmatrix} x_{1,1}\mathsf{Y} & x_{1,2}\mathsf{Y} & \dots \\ x_{2,1}\mathsf{Y} & x_{2,2}\mathsf{Y} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Then, a Kronecker power of a matrix is written as $\mathsf{X}^{\otimes n} = \mathsf{X}^{\otimes(n-1)} \otimes \mathsf{X}$, $\mathsf{X}^{\otimes 0} \triangleq 1$.

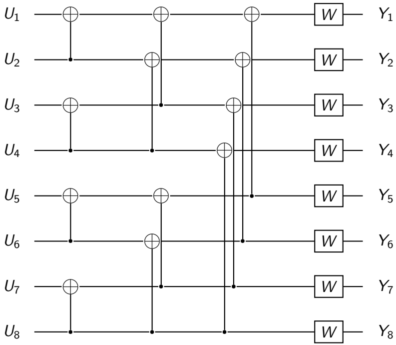
Example

Recall the matrix representing the basic transform $\mathsf{G}_2 \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Then, we write

$$\mathsf{G}_2^{\otimes 2} = \mathsf{G}_2 \otimes \mathsf{G}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

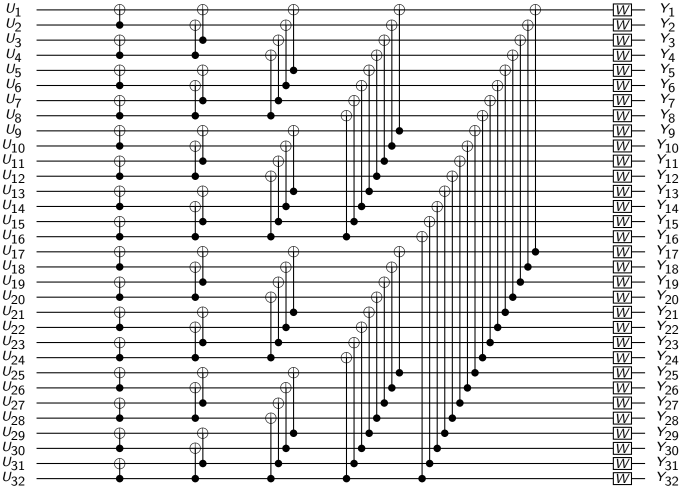
Polar Transform (N=8)

$$U_1^8 \mathsf{G}_2^{\otimes \log_2 8} = X_1^8$$



Polar Transform (N=32)

$$U_1^{32} \mathrm{G}_2^{\otimes \log_2 32} = X_1^{32}$$



Channel Polarization

For any fixed $\delta > 0$, the fraction of the mediocre channels vanishes as $N \to \infty$, i.e., we have

$$\lim_{N\to\infty} \frac{1}{N} \left| \left\{ i \in \{1, \ldots, N\} : \delta < H(W_N^{(i)}) < 1 - \delta \right\} \right| = 0.$$

Channel Polarization

For any fixed $\delta > 0$, the fraction of the mediocre channels vanishes as $N \to \infty$, i.e., we have

$$\lim_{N\to\infty} \frac{1}{N} \left| \left\{ i \in \{1, \ldots, N\} : \delta < H(W_N^{(i)}) < 1 - \delta \right\} \right| = 0.$$

Since the transform is information-lossless, we can write

$$\lim_{N\to\infty} \frac{1}{N} \left| \left\{ i \in \{1, \ldots, N\} : H(W_N^{(i)}) \le \delta \right\} \right| = C(W)$$

$$\lim_{N\to\infty} \frac{1}{N} \left| \left\{ i \in \{1, \ldots, N\} : H(W_N^{(i)}) \ge 1 - \delta \right\} \right| = 1 - C(W)$$

A Capacity-Achieving Scheme: Polar Codes

- Transmit uniformly distributed information bits over the good synthesized channels ($k \to N \cdot C(W)$).

A Capacity-Achieving Scheme: Polar Codes

- Transmit uniformly distributed information bits over the good synthesized channels ($k \to N \cdot C(W)$).
- Set the inputs of the bad synthesized channels to the constant values known to the decoder.

A Capacity-Achieving Scheme: Polar Codes

- Transmit uniformly distributed information bits over the good synthesized channels ($k \to N \cdot C(W)$).
- Set the inputs of the bad synthesized channels to the constant values known to the decoder.
- Decode the bits from U_1 to U_N successively.

A Capacity-Achieving Scheme: Polar Codes

- Transmit uniformly distributed information bits over the good synthesized channels ($k \to N \cdot C(W)$).
- Set the inputs of the bad synthesized channels to the constant values known to the decoder.
- Decode the bits from U_1 to U_N successively.
- $P_B \leq \sum_{i \in A} \delta$

A Capacity-Achieving Scheme: Polar Codes

- Transmit uniformly distributed information bits over the good synthesized channels ($k \rightarrow N \cdot C(W)$).
- Set the inputs of the bad synthesized channels to the constant values known to the decoder.
- Decode the bits from U_1 to U_N successively.
- $P_B \leq \sum_{i \in A} \delta$
- Indeed, polarization holds for $\delta = \mathcal{O}(2^{-\sqrt{N}})$ [AT09] (i.e., faster than $1/N$).

A Capacity-Achieving Scheme: Polar Codes

- Transmit uniformly distributed information bits over the good synthesized channels ($k \to N \cdot C(W)$).
- Set the inputs of the bad synthesized channels to the constant values known to the decoder.
- Decode the bits from U_1 to U_N successively.
- $P_B \leq \sum_{i \in A} \delta = N \cdot C(W) \cdot \delta$
- Indeed, polarization holds for $\delta = \mathcal{O}(2^{-\sqrt{N}})$ [AT09] (i.e., faster than $1/N$).

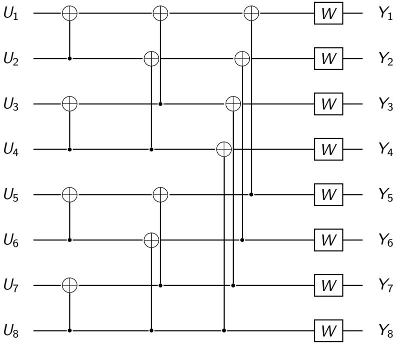
A Capacity-Achieving Scheme: Polar Codes

- Transmit uniformly distributed information bits over the good synthesized channels ($k \to N \cdot C(W)$).
- Set the inputs of the bad synthesized channels to the constant values known to the decoder.
- Decode the bits from U_1 to U_N successively.
- $P_B \leq \sum_{i \in A} \delta = N \cdot C(W) \cdot \delta \leq N \cdot C(W) \cdot 2^{-\sqrt{N}}$, resulting in $P_B \to 0$.
- Indeed, polarization holds for $\delta = \mathcal{O}(2^{-\sqrt{N}})$ [AT09] (i.e., faster than $1/N$).

Code Design

We want to design an (N, k) code, where $N = 2^n$ with $n \geq 1$.

.

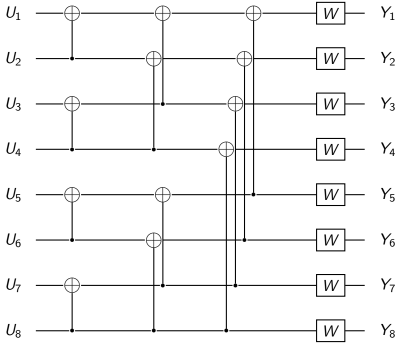


Code Design

We want to design an (N, k) code, where $N = 2^n$ with $n \geq 1$.
Equivalently, find a set $\mathcal{A} \in [N]$ of size k (information set).

1. Polar rule: For a target channel parameter, find the most reliable k positions for SC decoding.

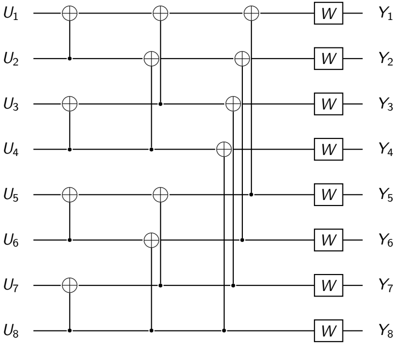
.



Code Design

We want to design an (N, k) code, where $N = 2^n$ with $n \geq 1$.
Equivalently, find a set $\mathcal{A} \in [N]$ of size k (information set).

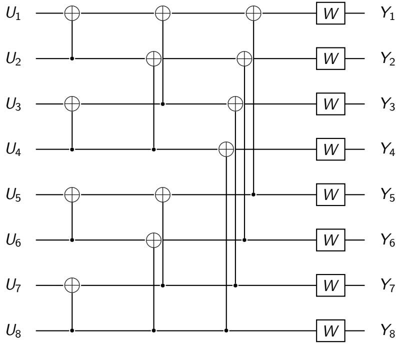
1. Polar rule: For a target channel parameter, find the most reliable k positions for SC decoding.
2. Reed-Muller (RM) rule: Find the indices of the k positions with the largest Hamming weight in $G_2^{\otimes n}$.



Code Design

We want to design an (N, k) code, where $N = 2^n$ with $n \geq 1$.
Equivalently, find a set $\mathcal{A} \in [N]$ of size k (information set).

1. Polar rule: For a target channel parameter, find the most reliable k positions for SC decoding.
2. Reed-Muller (RM) rule: Find the indices of the k positions with the largest Hamming weight in $\mathrm{G}_2^{\otimes n}$. Note that there is not an RM code for every k.

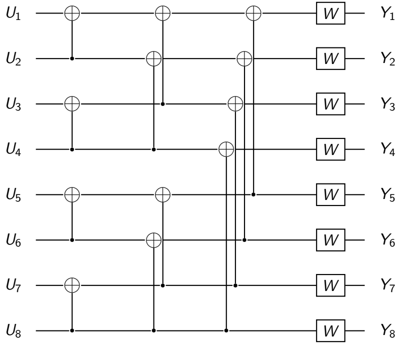


Code Design

We want to design an (N, k) code, where $N = 2^n$ with $n \geq 1$. Equivalently, find a set $\mathcal{A} \in [N]$ of size k (information set).

1. Polar rule: For a target channel parameter, find the most reliable k positions for SC decoding.
2. Reed-Muller (RM) rule: Find the indices of the k positions with the largest Hamming weight in $\mathrm{G}_2^{\otimes n}$. Note that there is not an RM code for every k.

The polar rule minimizes a tight upper bound on the error probability under SC decoding while the RM rule maximizes the minimum Hamming distance.



A Historical Remark

Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des Grades
Doktor-Ingenieur genehmigte

Dissertation

von

Dipl.-Ing. Norbert Stolte
Groß-Umstadt

- Observation: Reed-Muller (RM) codes perform poorly under low-complexity SC decoding.

A Historical Remark

Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des Grades
Doktor-Ingenieur genehmigte

Dissertation

von

Dipl.-Ing. Norbert Stolte
Groß-Umstadt

- Observation: Reed-Muller (RM) codes perform poorly under low-complexity SC decoding.
- Codes having Plotkin structure were optimized for SC decoding [Sto02].

A Historical Remark

Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des Grades
Doktor-Ingenieur genehmigte

Dissertation

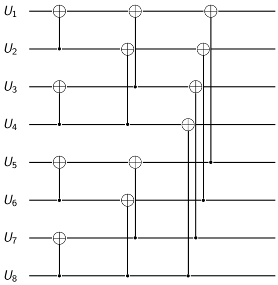
von

Dipl.-Ing. Norbert Stolte
Groß-Umstadt

- Observation: Reed-Muller (RM) codes perform poorly under low-complexity SC decoding.
- Codes having Plotkin structure were optimized for SC decoding [Sto02].
- They were shown to outperform RM codes under SC decoding.

Encoding

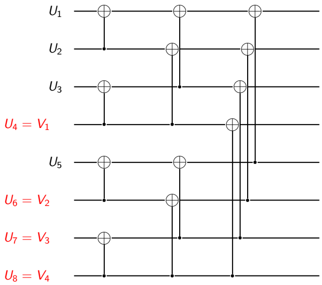
Let V_1^k denote the random information bits to be encoded:



Encoding

Let V_1^k denote the random information bits to be encoded:

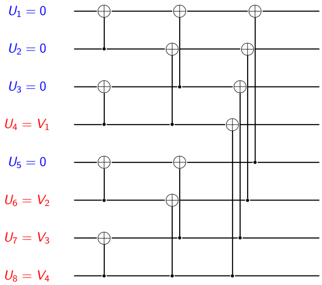
1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.



Encoding

Let V_1^k denote the random information bits to be encoded:

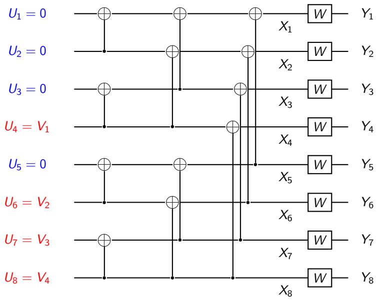
1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.
2. Set the remaining elements to 0, i.e., $U_{\mathcal{F}} = 0$ (frozen bits).



Encoding

Let V_1^k denote the random information bits to be encoded:

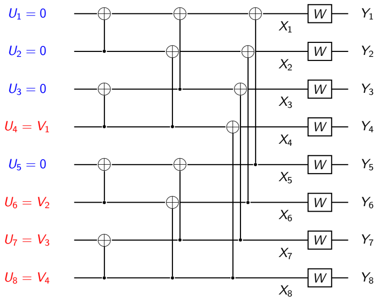
1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.
2. Set the remaining elements to 0, i.e., $U_{\mathcal{F}} = 0$ (frozen bits).
3. Apply polar transform of length$-N$, i.e., $X_1^N = U_1^N \mathrm{G}_2^{\otimes n}$.



Encoding

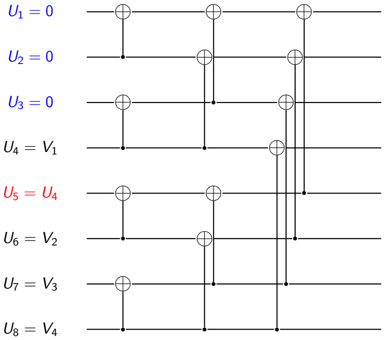
Let V_1^k denote the random information bits to be encoded:

1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.
2. Set the remaining elements to 0, i.e., $U_{\mathcal{F}} = 0$ (frozen bits).
3. Apply polar transform of length$-N$, i.e., $X_1^N = U_1^N \mathrm{G}_2^{\otimes n}$.
4. This can be done with a complexity of $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N^2)$.



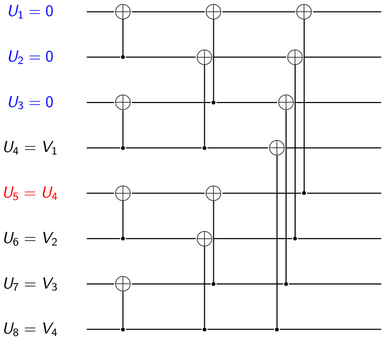
Dynamic Frozen Bits

- The value of a frozen bit can also be set to a linear combination of previous information bits (rather than a fixed 0 or 1 value) [TM16]



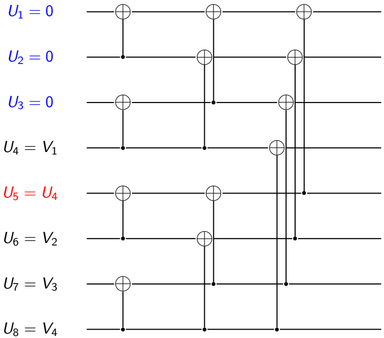
Dynamic Frozen Bits

- The value of a frozen bit can also be set to a linear combination of previous information bits (rather than a fixed 0 or 1 value) [TM16]
- A frozen bit whose value depends on past inputs is called dynamic.



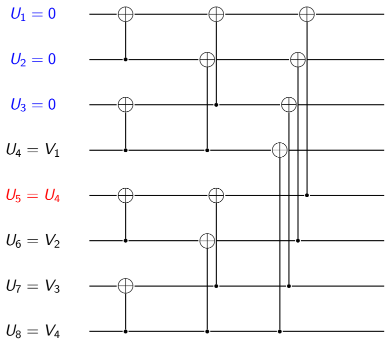
Dynamic Frozen Bits

- The value of a frozen bit can also be set to a linear combination of previous information bits (rather than a fixed 0 or 1 value) [TM16]
- A frozen bit whose value depends on past inputs is called dynamic.
- SC/SCL decoding easily modified for polar codes with dynamic frozen bits.

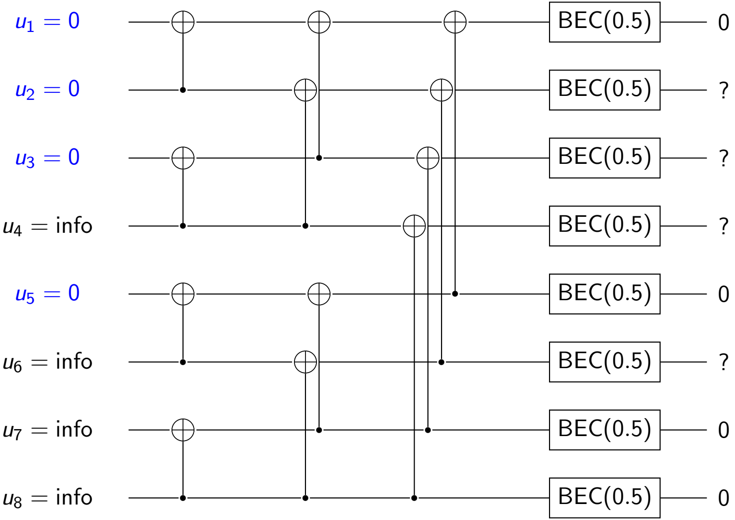


Dynamic Frozen Bits

- The value of a frozen bit can also be set to a linear combination of previous information bits (rather than a fixed 0 or 1 value) [TM16]
- A frozen bit whose value depends on past inputs is called dynamic.
- SC/SCL decoding easily modified for polar codes with dynamic frozen bits.
- Any binary linear block code can be represented as a polar code with dynamic frozen bits!

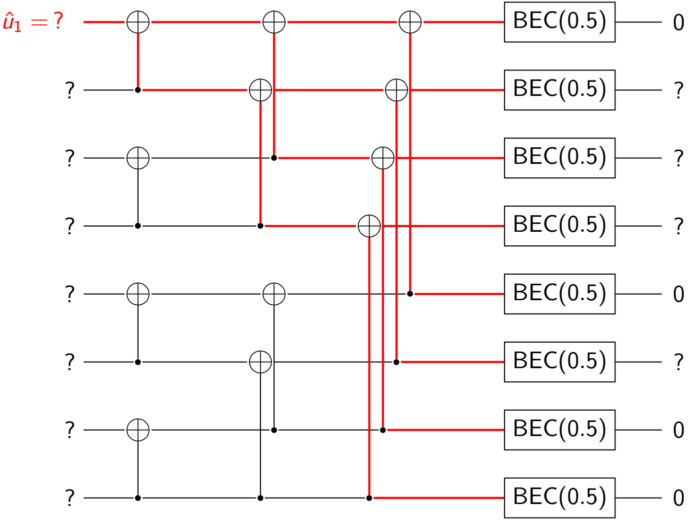


Successive Cancellation Decoding: BEC Example



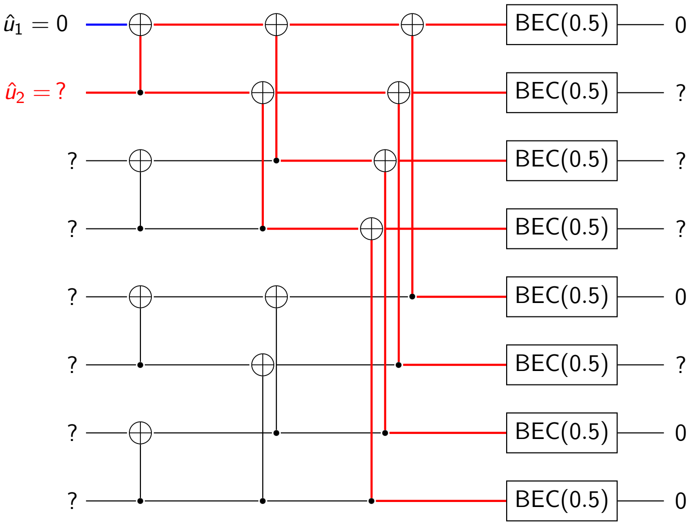
Successive Cancellation Decoding: BEC Example

$\hat{u}_1 = 0$ (frozen)



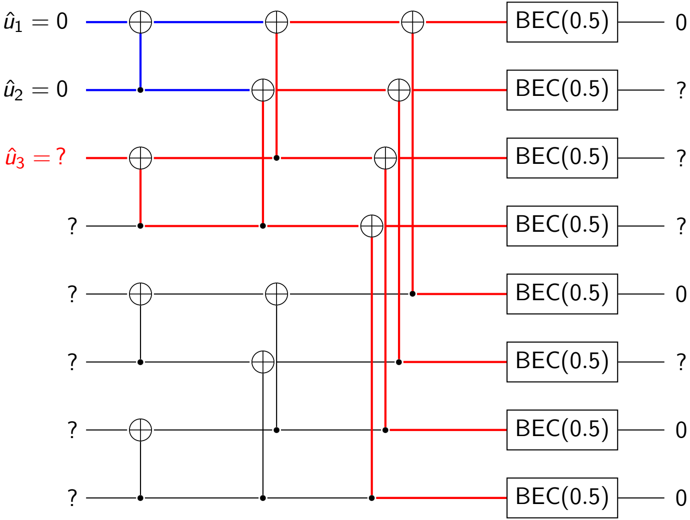
Successive Cancellation Decoding: BEC Example

$\hat{u}_2 = 0$ (frozen)

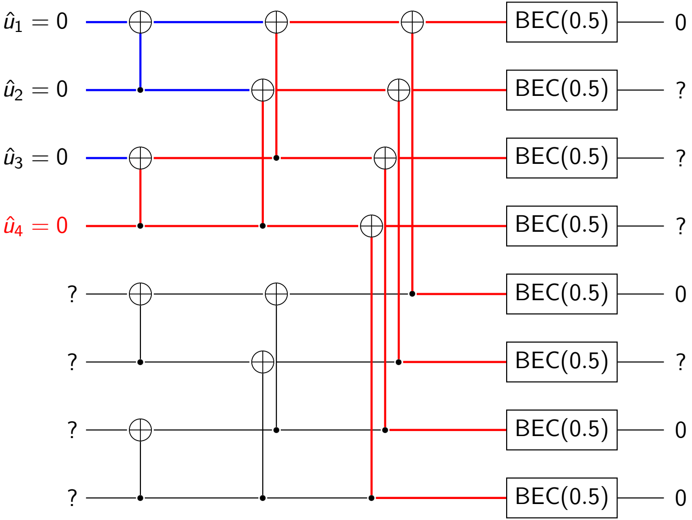


Successive Cancellation Decoding: BEC Example

$\hat{u}_3 = 0$ (frozen)

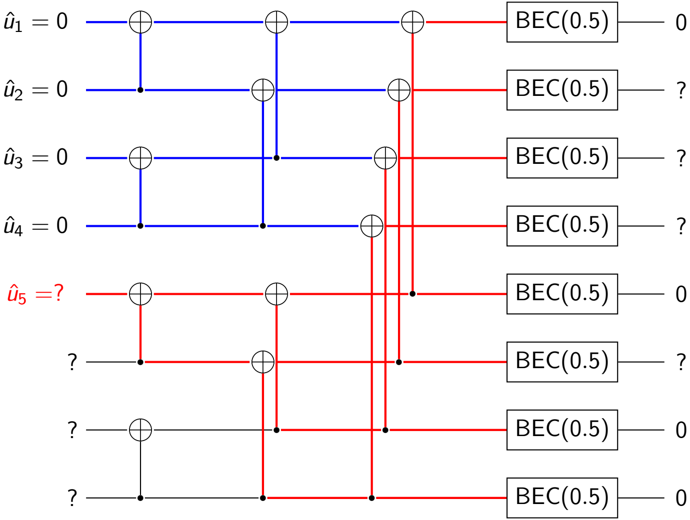


Successive Cancellation Decoding: BEC Example

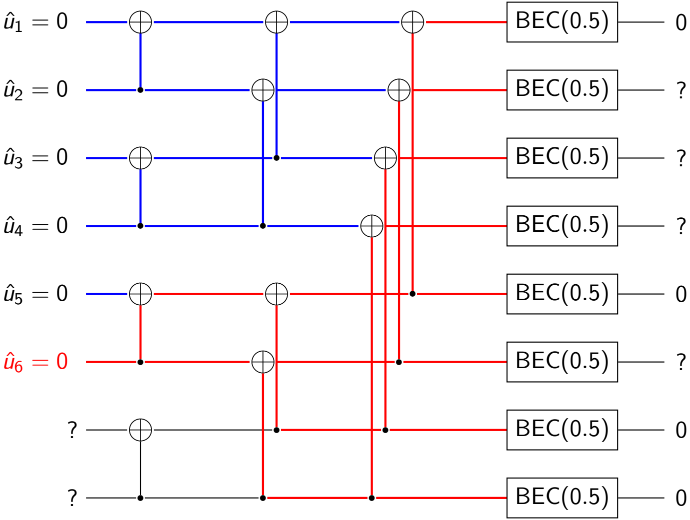


Successive Cancellation Decoding: BEC Example

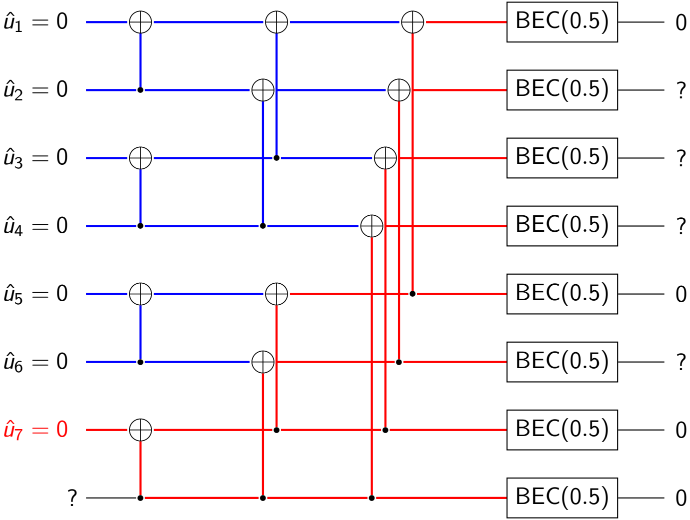
$\hat{u}_5 = 0$ (frozen)



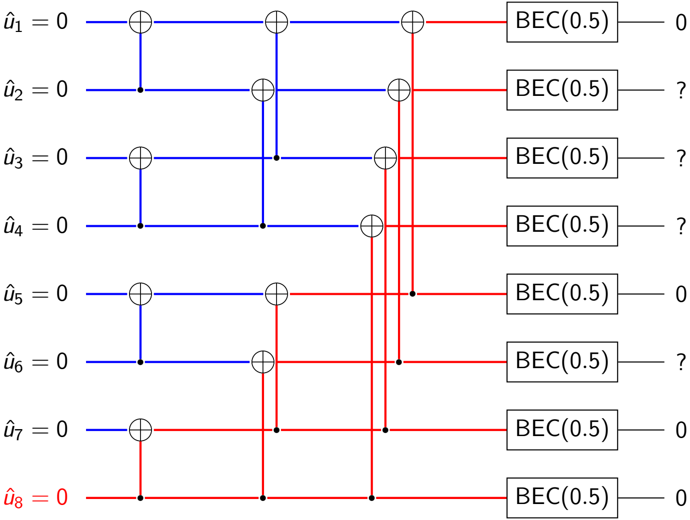
Successive Cancellation Decoding: BEC Example



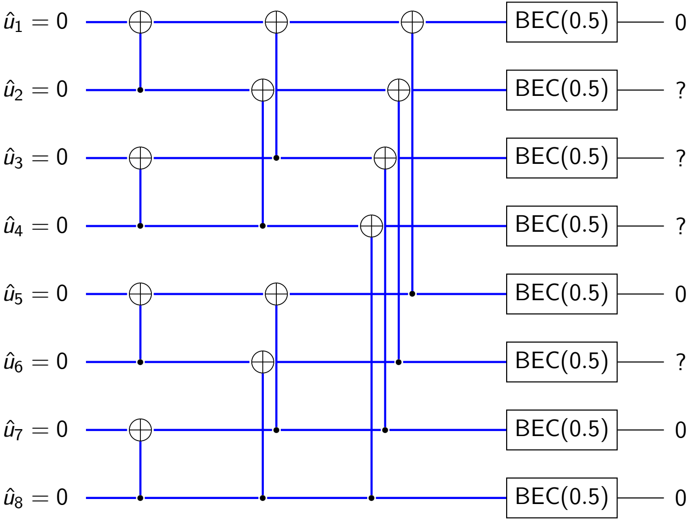
Successive Cancellation Decoding: BEC Example



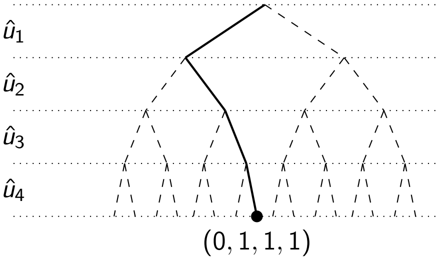
Successive Cancellation Decoding: BEC Example



Successive Cancellation Decoding: BEC Example



Successive Cancellation Decoding

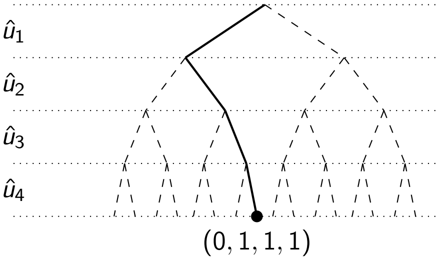


$$\hat{u}_i = \begin{cases} u_i & \text{if } i \in \mathcal{F} \\ f_i\left(y_1^N, \hat{u}_1^{i-1}\right) & \text{if } i \in \mathcal{A}. \end{cases} \quad (1)$$

$$f_i\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \begin{cases} 0 & \text{if } P_{U_i|Y^N,U^{i-1}}(0|y_1^N, \hat{u}_1^{i-1}) = 1 \\ ? & \text{if } P_{U_i|Y^N,U^{i-1}}(0|y_1^N, \hat{u}_1^{i-1}) = \frac{1}{2} \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

where a frame error occurs if $\hat{u}_i = ?$ for any $i \in \mathcal{A}$!

Successive Cancellation Decoding



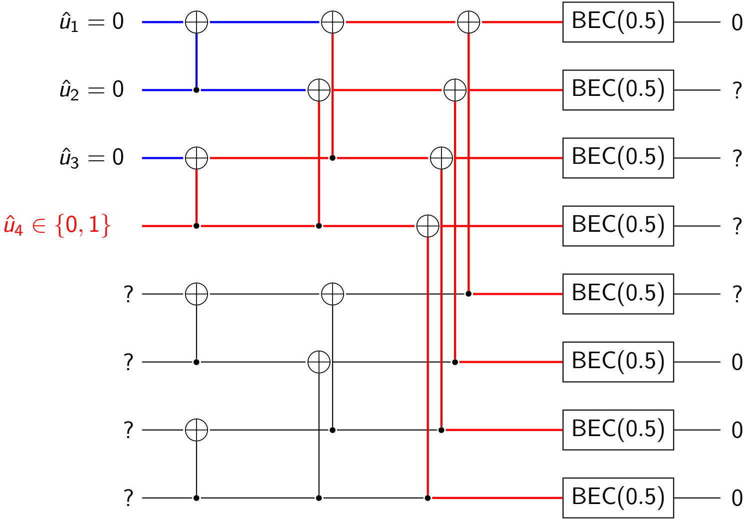
$$\hat{u}_i = \begin{cases} u_i & \text{if } i \in \mathcal{F} \\ f_i\left(y_1^N, \hat{u}_1^{i-1}\right) & \text{if } i \in \mathcal{A}. \end{cases} \tag{1}$$

$$f_i\left(y_1^N, \hat{u}_1^{i-1}\right) \triangleq \begin{cases} 0 & \text{if } P_{U_i|Y^N,U^{i-1}}(0|y_1^N, \hat{u}_1^{i-1}) = 1 \\ ? & \text{if } P_{U_i|Y^N,U^{i-1}}(0|y_1^N, \hat{u}_1^{i-1}) = \frac{1}{2} \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where a frame error occurs if $\hat{u}_i = ?$ for any $i \in \mathcal{A} \rightarrow$ successive cancellation list (SCL) decoding!

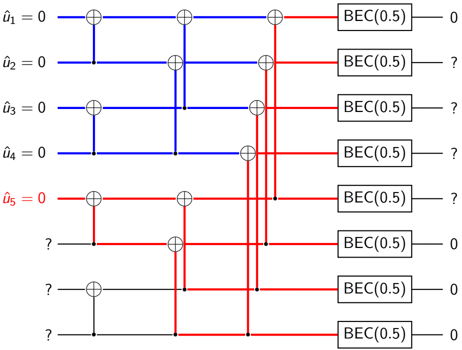
Successive Cancellation List Decoding: BEC Example

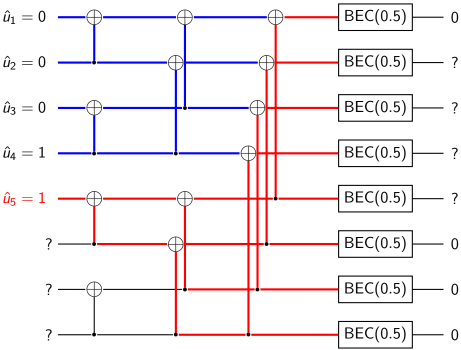
$|\mathcal{L}_3| = 1$



Successive Cancellation List Decoding: BEC Example

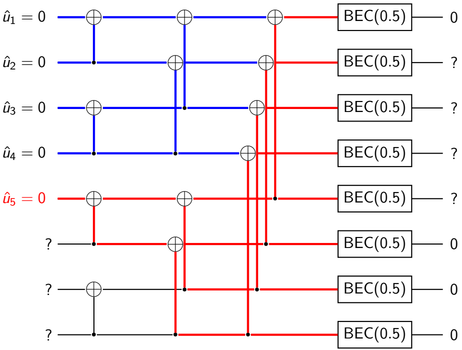
$$\mathcal{L}_4 = \{0000, 0001\}$$

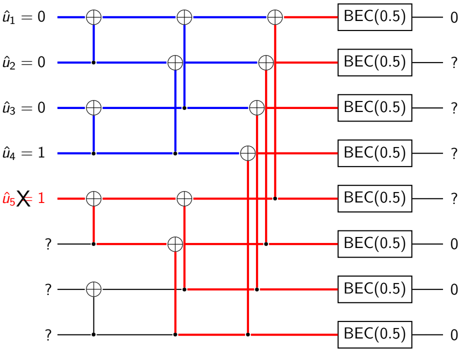




Successive Cancellation List Decoding: BEC Example

$$\mathcal{L}_4 = \{0000, 0001\}$$

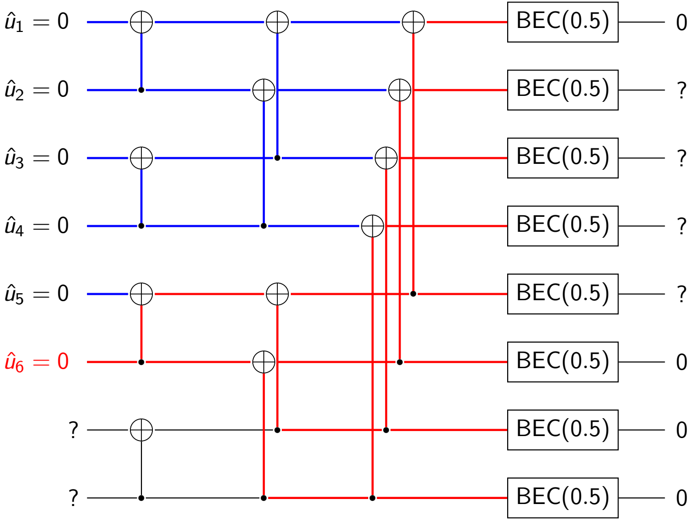




$\hat{u}_5 = 0$ (frozen) $\rightarrow P_{U_5|Y_1^8,U_1^4}\left(0|y_1^8,(0,0,0,1)\right) = 0$ tree pruning

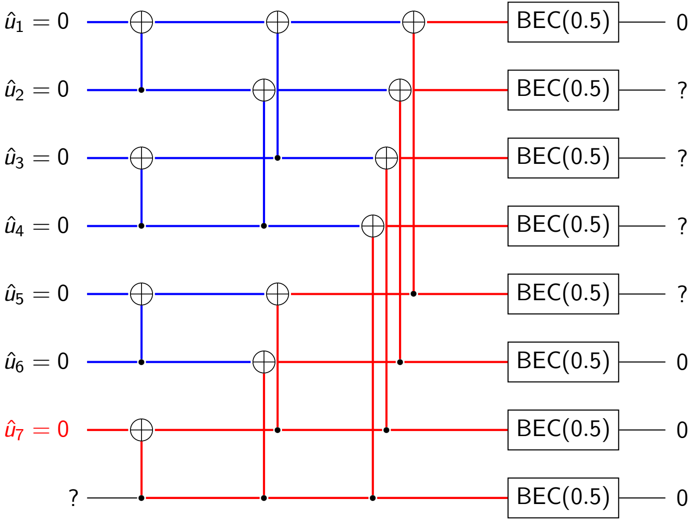
Successive Cancellation List Decoding: BEC Example

$|\mathcal{L}_5| = 1$



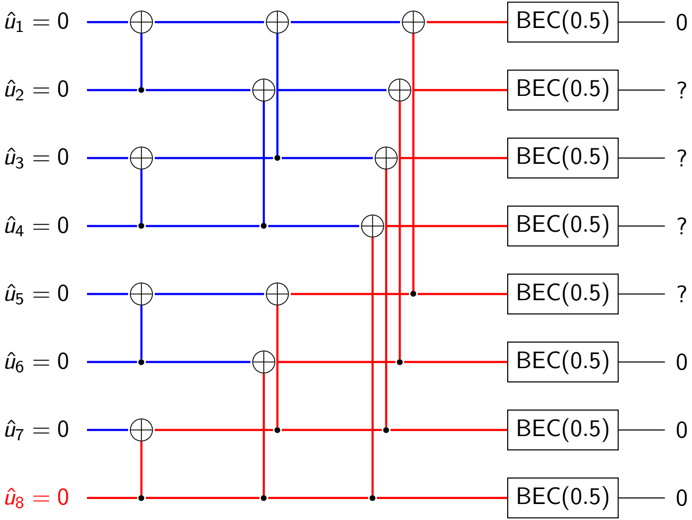
Successive Cancellation List Decoding: BEC Example

$|\mathcal{L}_6| = 1$

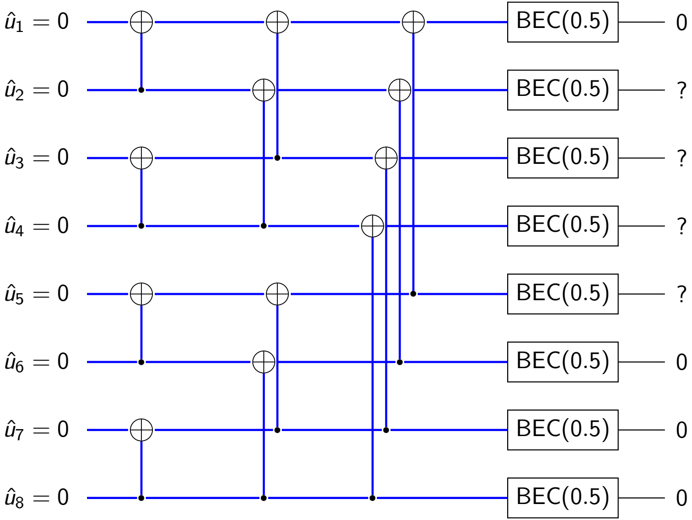


Successive Cancellation List Decoding: BEC Example

$|\mathcal{L}_7| = 1$



Successive Cancellation List Decoding: BEC Example



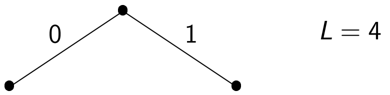
$|\mathcal{L}_8| = 1$

Successive Cancellation List Decoding: General BMSCs

Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.

Successive Cancellation List Decoding: General BMSCs

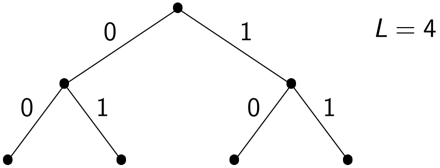
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.

Successive Cancellation List Decoding: General BMSCs

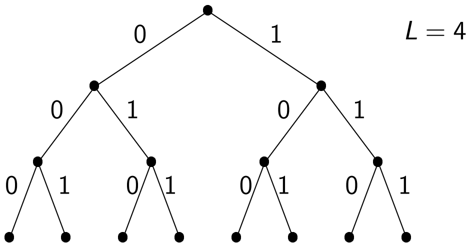
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.

Successive Cancellation List Decoding: General BMSCs

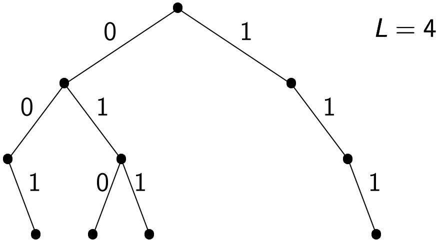
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.

Successive Cancellation List Decoding: General BMSCs

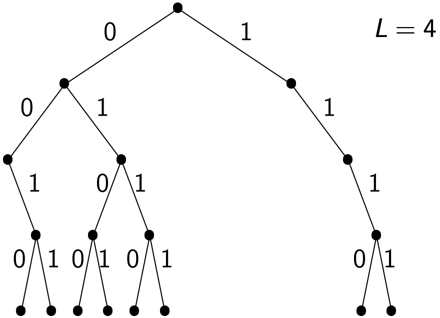
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.

Successive Cancellation List Decoding: General BMSCs

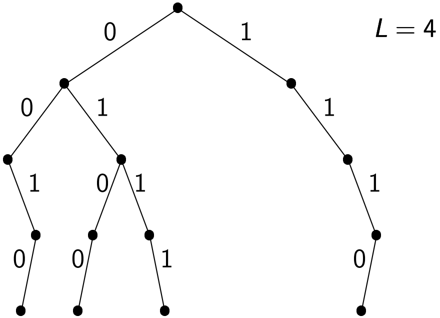
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.

Successive Cancellation List Decoding: General BMSCs

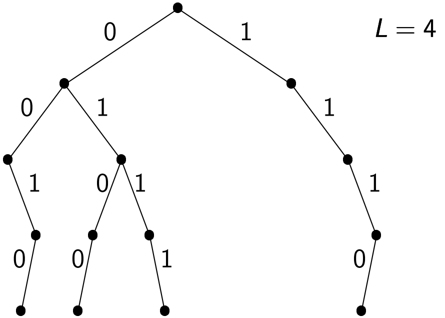
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.

Successive Cancellation List Decoding: General BMSCs

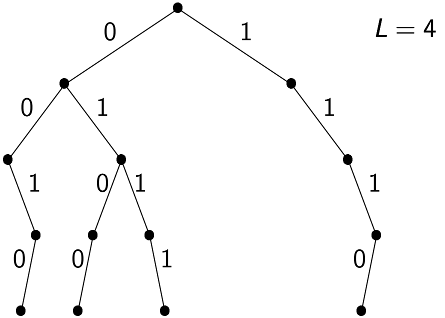
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.
- After N-th stage,
  $\hat{u}_1^N = \arg\max_{u_1^N \in \mathcal{L}_N} P_{U_N|Y_1^N, U_1^{N-1}}(u_N | y_1^N, u_1^{N-1}) = \arg\max_{u_1^N \in \mathcal{L}_N} \Pr(u_1^N | y_1^N)$.

Successive Cancellation List Decoding: General BMSCs

Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.
- After N-th stage,
  $\hat{u}_1^N = \arg\max_{u_1^N \in \mathcal{L}_N} P_{U_N|Y_1^N, U_1^{N-1}}(u_N|y_1^N, u_1^{N-1}) = \arg\max_{u_1^N \in \mathcal{L}_N} \Pr(u_1^N|y_1^N)$.
- Very similar ideas were applied to RM codes (see, e.g., [Sto02, DS06]).

Outline

1 Overview of Polar Codes

2 Successive Cancellation Inactivation Decoding

3 Successive Cancellation Ordered Search Decoding

4 Conclusions

Linear Codes over Erasure Channels

- For linear codes on the BEC, any uncertainty in the information bits always takes the form of an affine subspace.

Linear Codes over Erasure Channels

- For linear codes on the BEC, any uncertainty in the information bits always takes the form of an affine subspace.
- The SCL decoder on the BEC lists all valid paths in this subspace.
- Successive cancellation inactivation (SCI) decoding stores a basis instead.

Linear Codes over Erasure Channels

- For linear codes on the BEC, any uncertainty in the information bits always takes the form of an affine subspace.
- The SCL decoder on the BEC lists all valid paths in this subspace.
- Successive cancellation inactivation (SCI) decoding stores a basis instead.

Based on joint a work with Joachim Neu (Stanford) and Henry D. Pfister (Duke) [CNP20]

Related Works

- Inactivation-based decoders are efficient versions of Gaussian elimination.
- In the context of coding theory, an instance was proposed by E. Berlekamp in 1968 [Ber15].

Related Works

- Inactivation-based decoders are efficient versions of Gaussian elimination.
- In the context of coding theory, an instance was proposed by E. Berlekamp in 1968 [Ber15].
- Similar methods have been proposed for low-density parity-check (LDPC) [RU01, PF04, MMU08, PLMC12] and raptor [Sho06, LLB17] codes.

Related Works

- Inactivation-based decoders are efficient versions of Gaussian elimination.
- In the context of coding theory, an instance was proposed by E. Berlekamp in 1968 [Ber15].
- Similar methods have been proposed for low-density parity-check (LDPC) [RU01, PF04, MMU08, PLMC12] and raptor [Sho06, LLB17] codes.
- For polar codes, a BP decoder with inactivations was proposed [EP10], but it does not use SC decoding schedule. More activity this year [UB21, UMB21].

The Algorithm

- The SCI decoder has the same message passing schedule as the SC decoder.
- Whenever an information bit is decoded as erased, it is replaced by a dummy variable (i.e., inactivated).

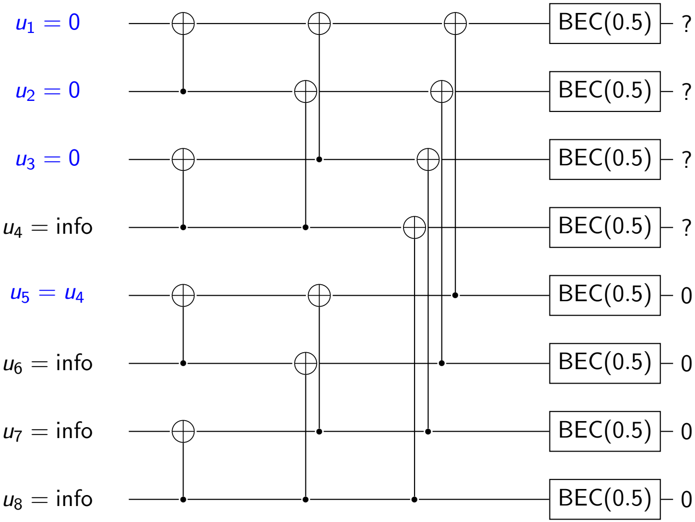
The Algorithm

- The SCI decoder has the same message passing schedule as the SC decoder.
- Whenever an information bit is decoded as erased, it is replaced by a dummy variable (i.e., inactivated).
- It continues decoding using SC decoding for the BEC, where the message values are allowed to be functions of all inactivated variables.
- The inactivated bits are resolved, later, using linear equations derived from decoding frozen bits.

Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

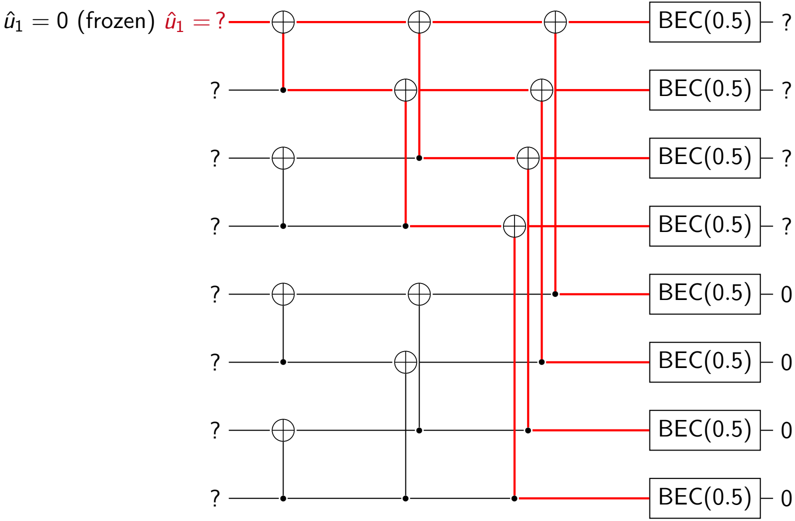
$g \triangleq$ total number of inactivations during a decoding attempt



Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

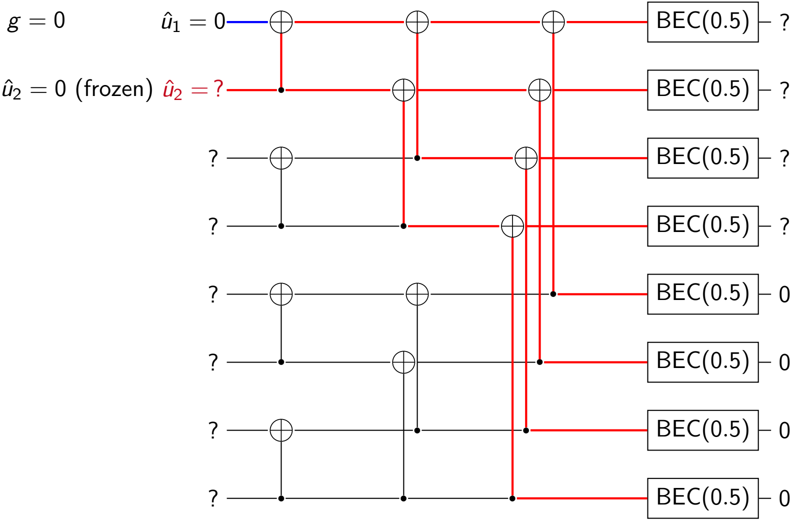
$g \triangleq$ total number of inactivations during a decoding attempt



Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$g \triangleq$ total number of inactivations during a decoding attempt

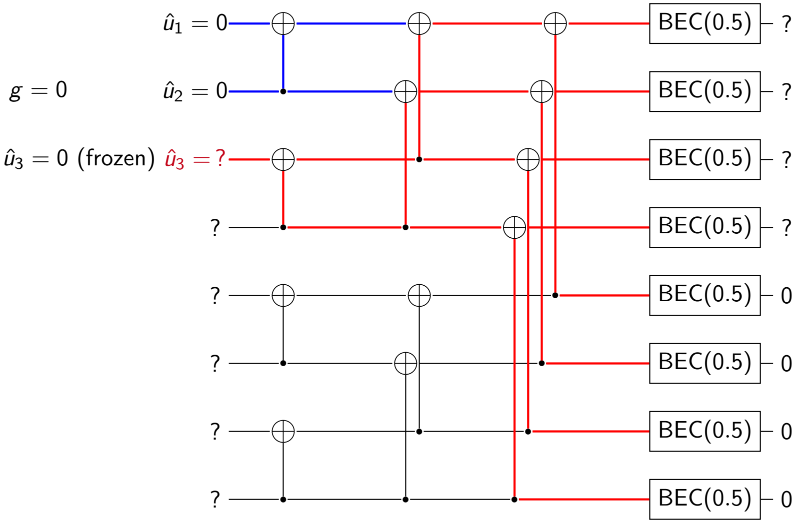


Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$g \triangleq$ total number of inactivations during a decoding attempt

$g = 0$

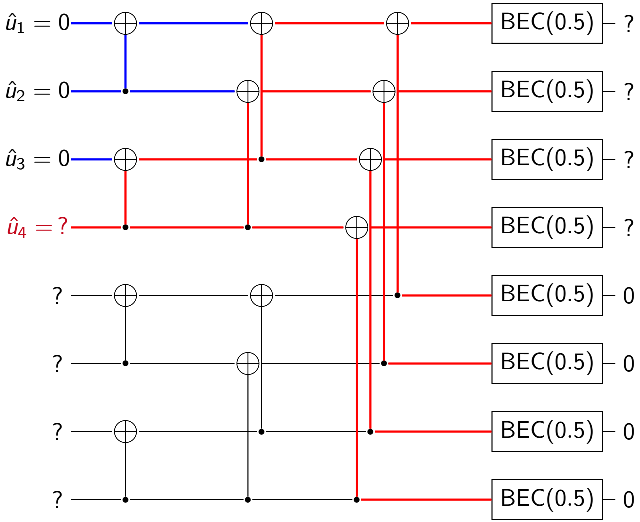


Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$g \triangleq$ total number of inactivations during a decoding attempt

$g = 0$

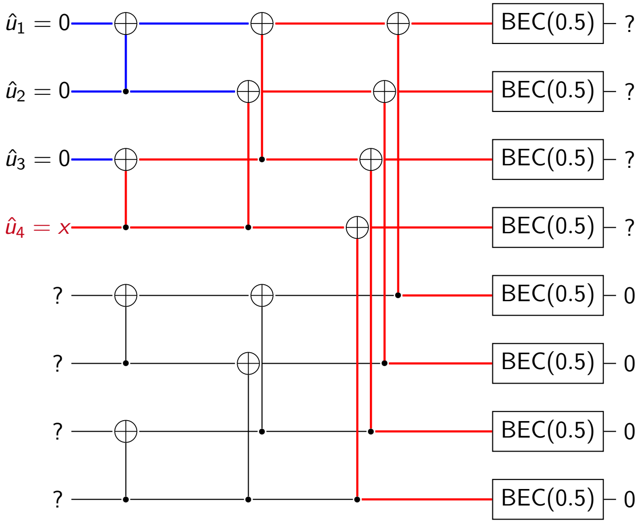


Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$g \triangleq$ total number of inactivations during a decoding attempt

$g = 0$



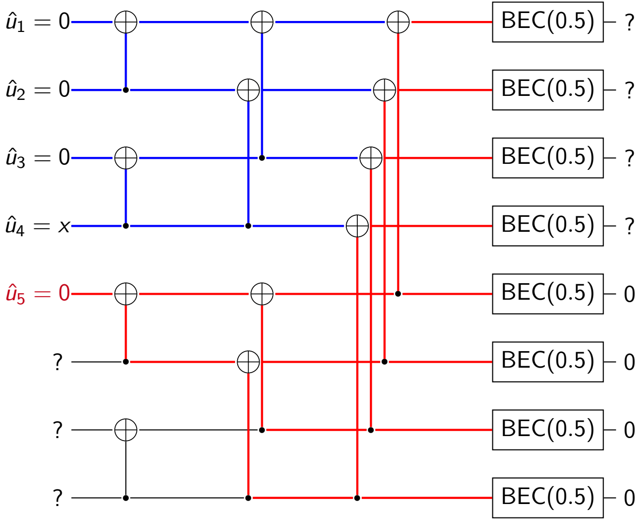
Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$g \triangleq$ total number of inactivations during a decoding attempt

$g = 1$

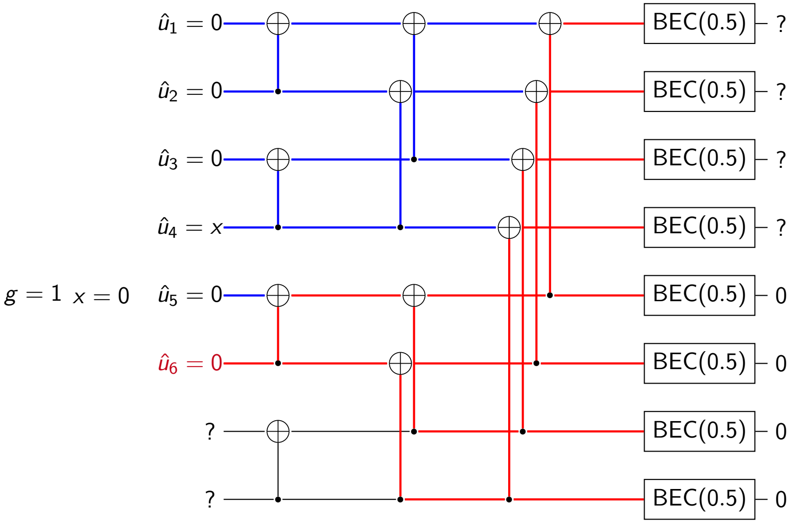
$\hat{u}_5 = \hat{u}_4$ (frozen)



Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

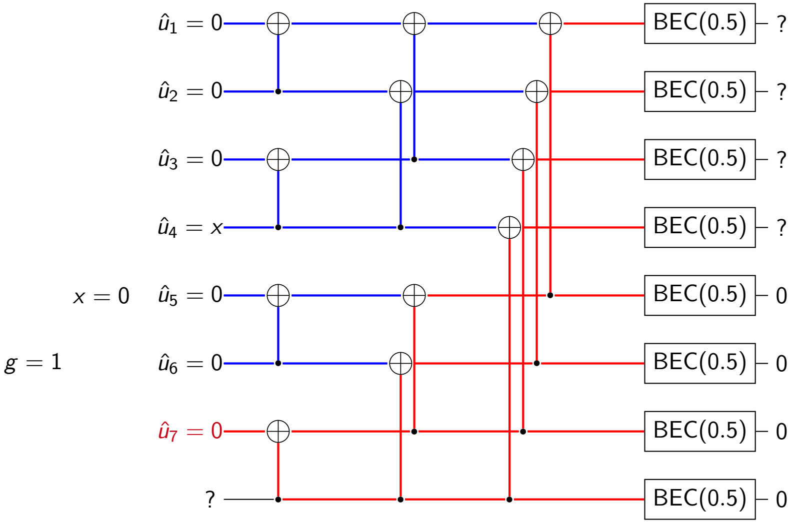
$g \triangleq$ total number of inactivations during a decoding attempt



Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

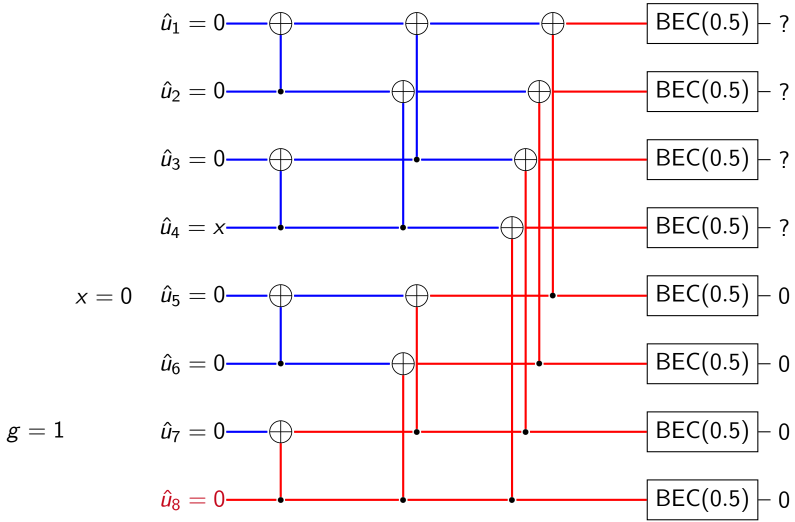
$g \triangleq$ total number of inactivations during a decoding attempt



Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

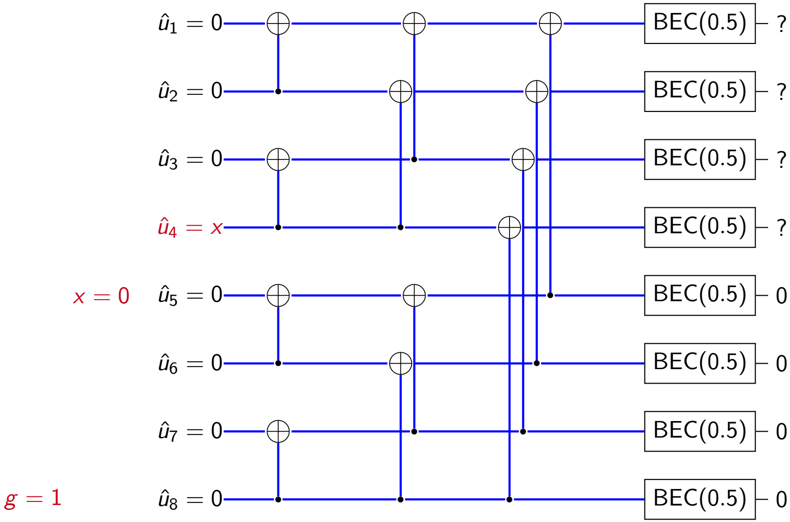
$g \triangleq$ total number of inactivations during a decoding attempt



Successive Cancellation Inactivation Decoding

Example: $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$g \triangleq$ total number of inactivations during a decoding attempt



Successive Cancellation Inactivation Decoding

- Assume that the SCI decoder inactivates g bits in total during a decoding attempt.
- The final step of SCI decoding is to solve a system of linear equations in g unknowns.
- This has a unique solution only if the equations obtained from frozen bits have rank g.

[1] Indeed, it delivers MAP decoding even when the input bits are not uniform by choosing the candidate maximizing the a-priori probability in the final list of candidates (if no unique solution).

Successive Cancellation Inactivation Decoding

- Assume that the SCI decoder inactivates g bits in total during a decoding attempt.
- The final step of SCI decoding is to solve a system of linear equations in g unknowns.
- This has a unique solution only if the equations obtained from frozen bits have rank g.
- It is a MAP decoder if there is no constraint on the number of inactivations.[1]

[1] Indeed, it delivers MAP decoding even when the input bits are not uniform by choosing the candidate maximizing the a-priori probability in the final list of candidates (if no unique solution).

SCI Decoding with Consolidations

- An inactivated bit may be resolved right after decoding a frozen bit whenever it provides an informative equation.
  - This event is referred to as consolidation.

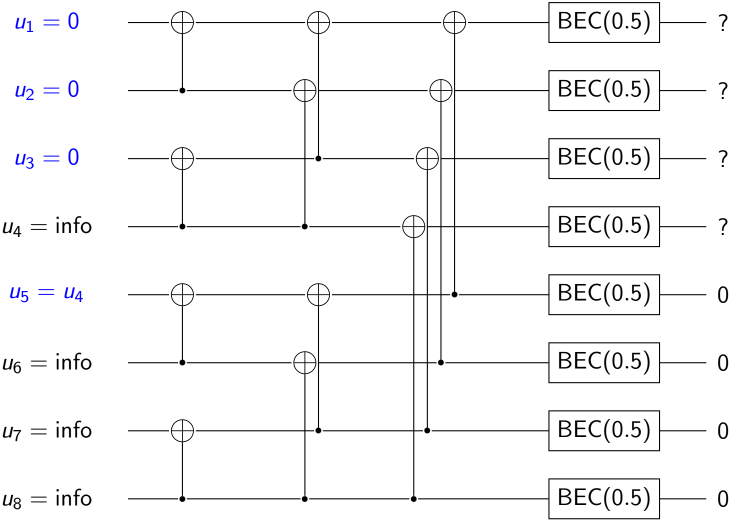
SCI Decoding with Consolidations

- An inactivated bit may be resolved right after decoding a frozen bit whenever it provides an informative equation.
  - This event is referred to as consolidation.
- The SCI decoder with consolidations mimics the path pruning stage of SCL decoding.

SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

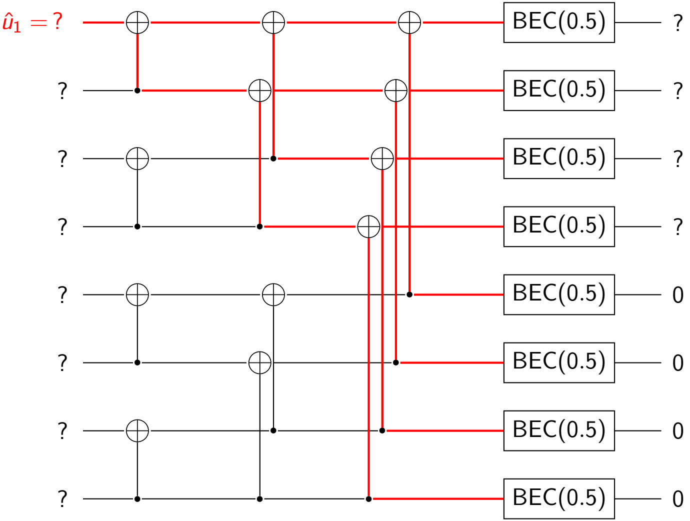


SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$\hat{u}_1 = 0$ (frozen)



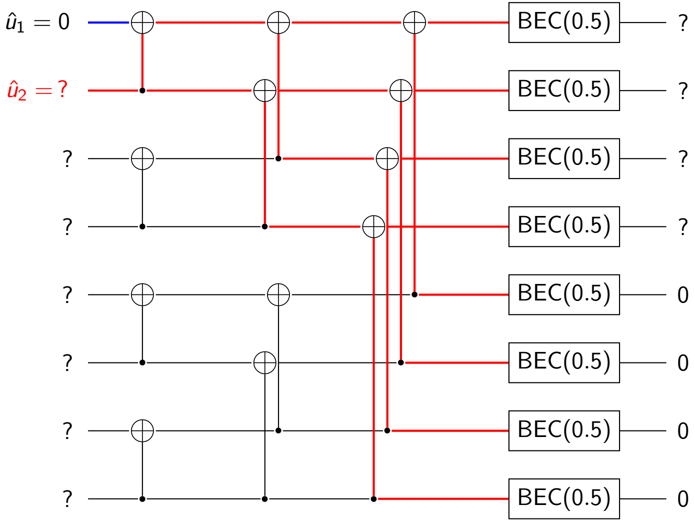
SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$d_1 = 0$

$\hat{u}_2 = 0$ (frozen)



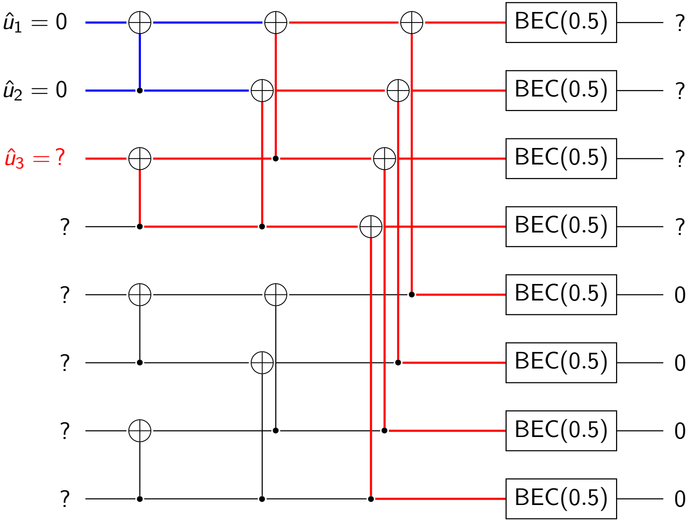
SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$d_2 = 0$

$\hat{u}_3 = 0$ (frozen)

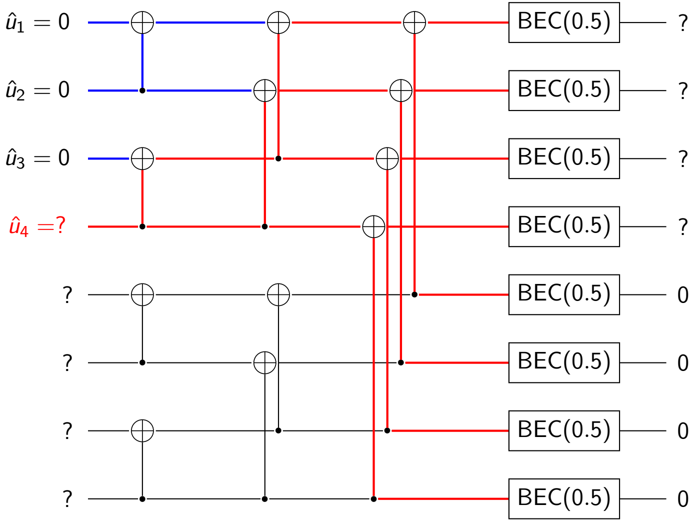


SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$d_3 = 0$

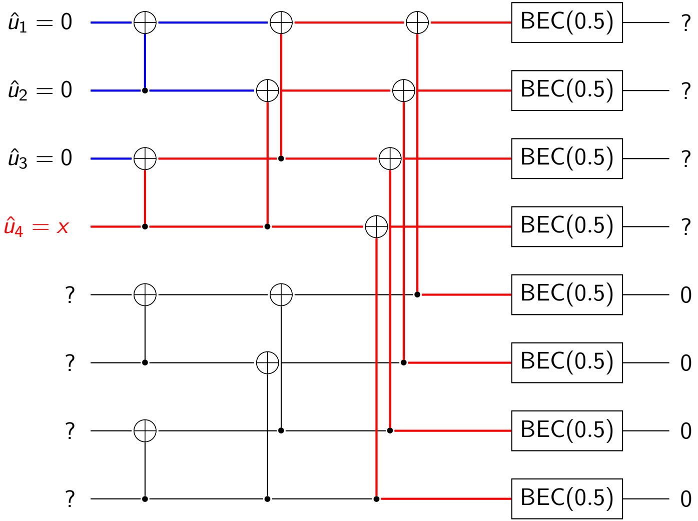


SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$d_3 = 0$



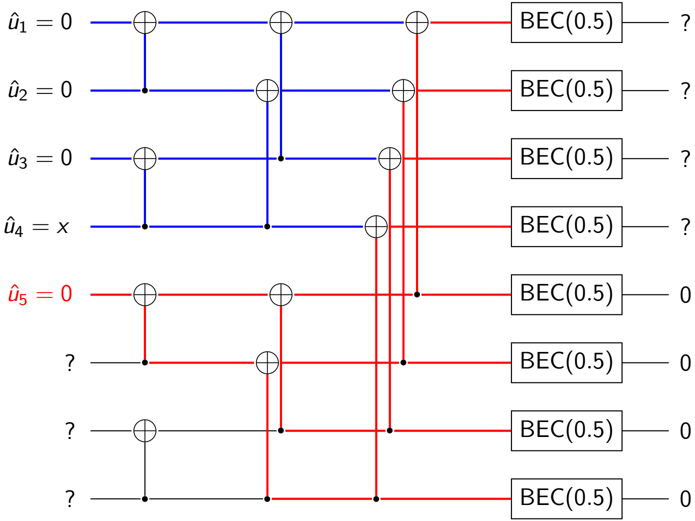
SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$d_4 = 1$

$\hat{u}_5 = \hat{u}_4$ (frozen)

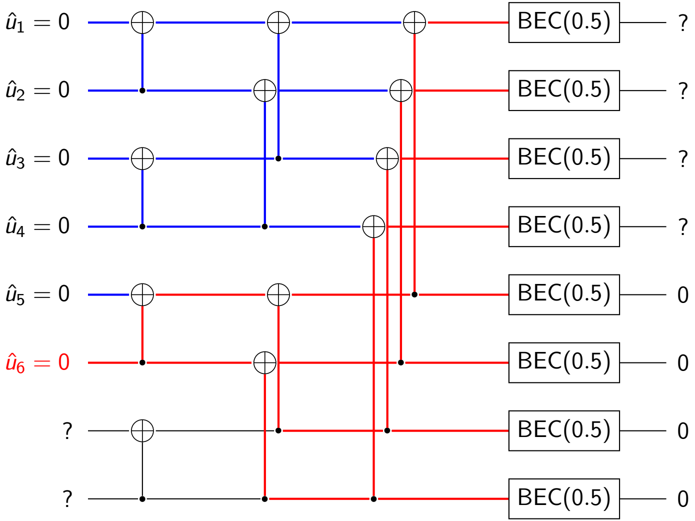


SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$d_5 = 0$

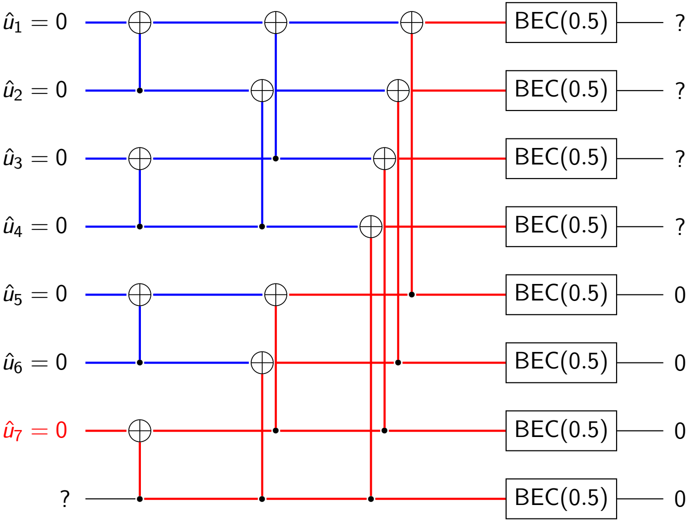


SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

$d_6 = 0$

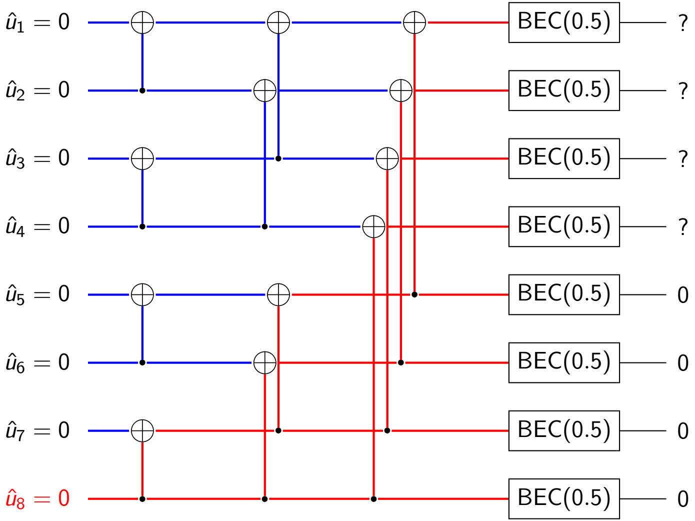


SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage

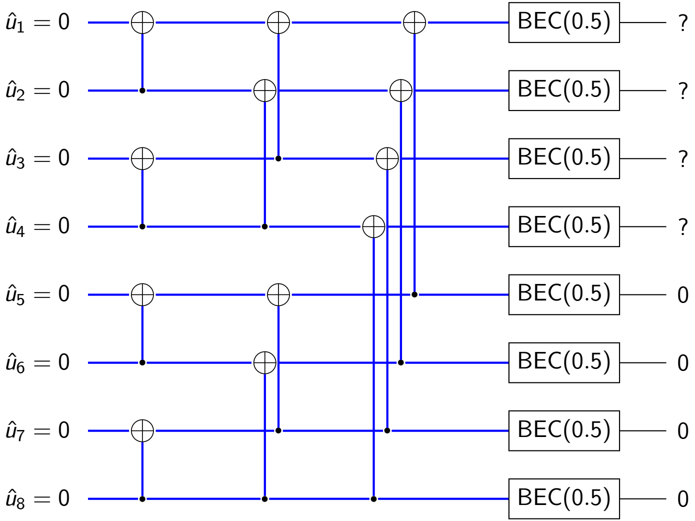
$d_7 = 0$



SCI Decoding with Consolidations

Example (Cont'd): $u_1 = u_2 = u_3 = 0$, $u_5 = u_4$ (frozen bits)

$d_i \triangleq$ number of unresolved inactivations (subspace dimension) at i-th decoding stage



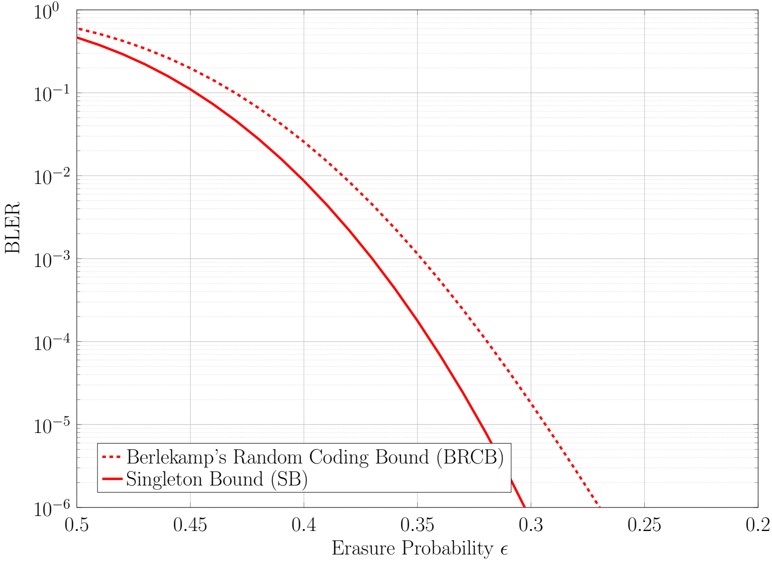
$d_8 = 0$

Analyzed Codes

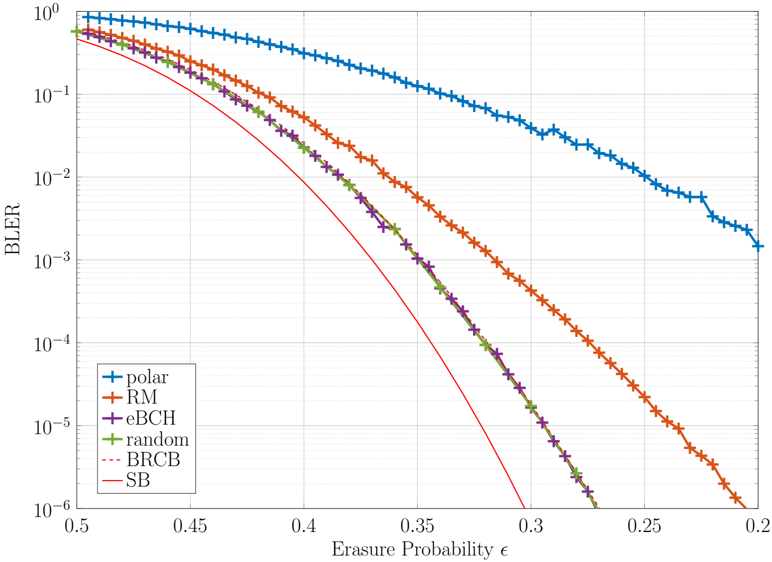
We first consider the following codes with parameters $(N = 128, k = 64)$:

1. A polar code designed for $\epsilon = 0.4$
2. The RM code
3. The eBCH code
4. A uniform random linear code: $\mathcal{A} = \{1, 2, \dots, 64\}$, where each frozen bit is a uniform random linear combination of bits u_i, $i \in \mathcal{A}$

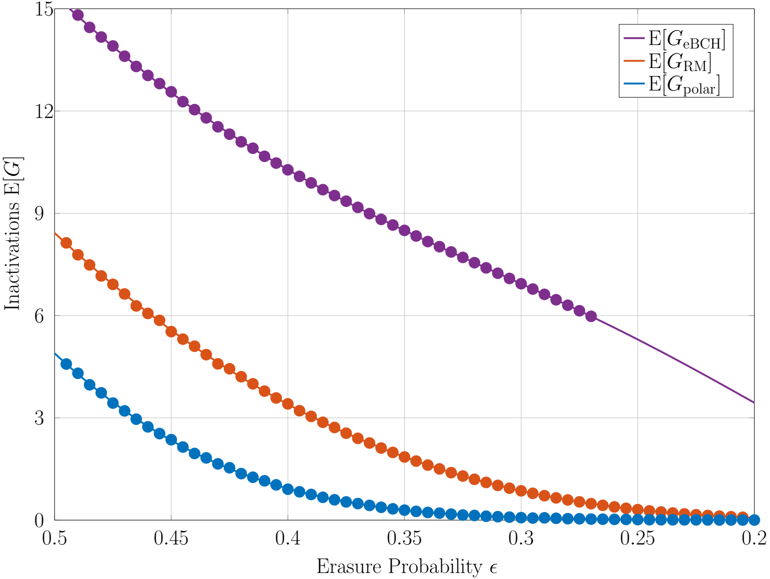
$(128, 64)$ Codes - MAP Performance



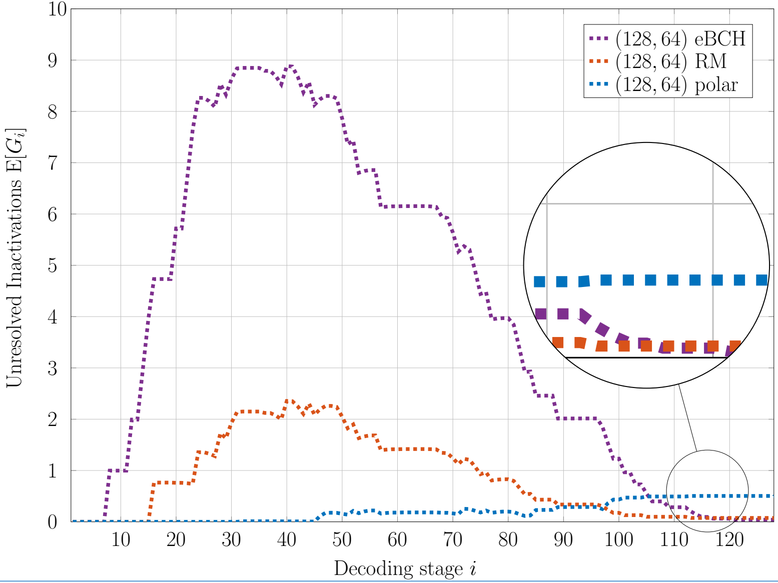
(128, 64) Codes - MAP Performance



$(128, 64)$ Codes - Expected Number of Inactivations $\mathrm{E}[G]$



(128, 64) Codes - Expected Subspace Dimension $\mathrm{E}[D_i]$ for $\epsilon = 0.4$



Dynamic Reed-Muller Code Ensemble

Since the $(N = 128, k = 64)$ RM code provides a good complexity vs. performance trade-off, we propose a modification to it:

- An instance from dynamic RM (dRM) code ensemble is obtained as follows:
  - Adopt the $\mathcal{A}$ of the RM code
  - Set each frozen bit to a random linear combination of all previous information bit(s)

Dynamic Reed-Muller Code Ensemble

Since the $(N = 128, k = 64)$ RM code provides a good complexity vs. performance trade-off, we propose a modification to it:

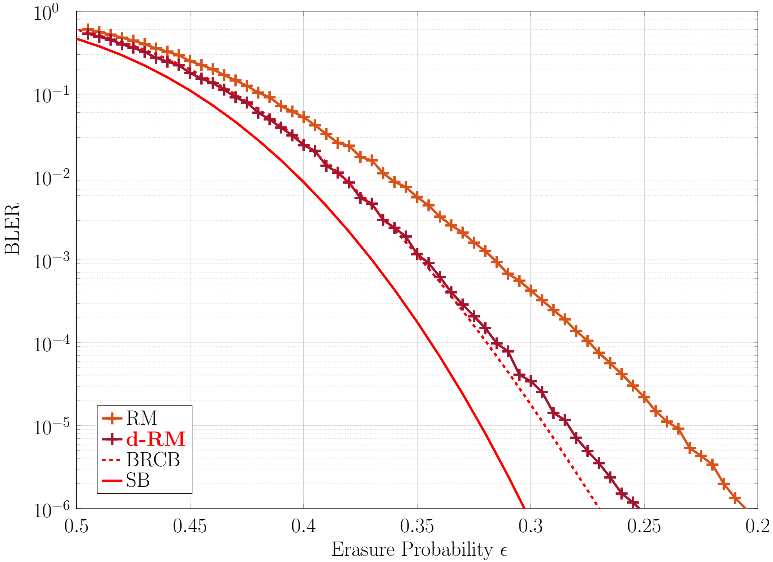
- An instance from dynamic RM (dRM) code ensemble is obtained as follows:
  - Adopt the $\mathcal{A}$ of the RM code
  - Set each frozen bit to a random linear combination of all previous information bit(s)
- Arıkan's PAC code [Arı19] is an instance form the ensemble:
  - PAC codes are defined by an information index set $\mathcal{A}$ and an upper-triangular Toeplitz matrix T

Dynamic Reed-Muller Code Ensemble

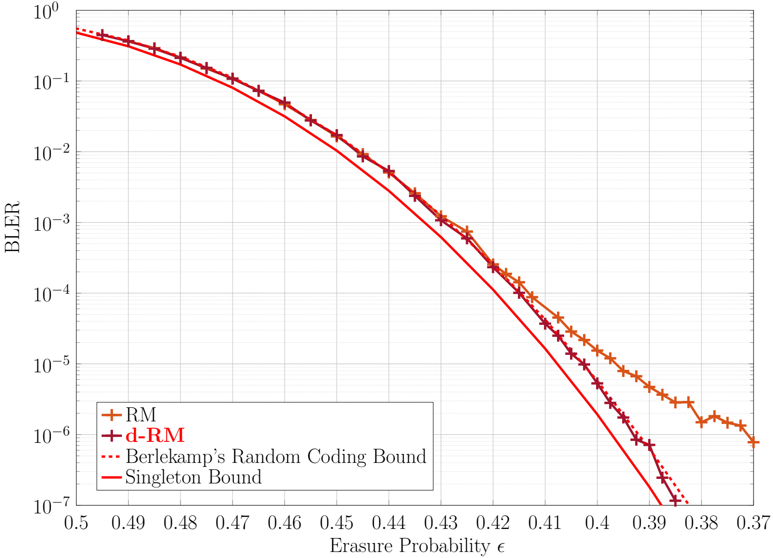
Since the $(N = 128, k = 64)$ RM code provides a good complexity vs. performance trade-off, we propose a modification to it:

- An instance from dynamic RM (dRM) code ensemble is obtained as follows:
  - Adopt the $\mathcal{A}$ of the RM code
  - Set each frozen bit to a random linear combination of all previous information bit(s)
- Arıkan's PAC code [Arı19] is an instance form the ensemble:
  - PAC codes are defined by an information index set $\mathcal{A}$ and an upper-triangular Toeplitz matrix T
  - PAC codes are polar codes with dynamic frozen bits, where information index set is $\mathcal{A}$ (Arıkan chooses $\mathcal{A}$ of the RM code) and dynamic frozen bits are specified by T [RBV20, YFV20]

$(128, 64)$ Codes - MAP Performance



$(512, 256)$ Codes - MAP Performance



Outline

1. Overview of Polar Codes
2. Successive Cancellation Inactivation Decoding
3. Successive Cancellation Ordered Search Decoding
4. Conclusions

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?
  - SCL decoding with unbounded list size has an exponential complexity :(

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?
  - SCL decoding with unbounded list size has an exponential complexity :(
  - Sequential decoders [Fan63, NC12, MT14, Tri18, JH19] require a parameter-optimization and do not guarantee ML decoding :(

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?
  - SCL decoding with unbounded list size has an exponential complexity :(
  - Sequential decoders [Fan63, NC12, MT14, Tri18, JH19] require a parameter-optimization and do not guarantee ML decoding :(
  - Flip decoders [ABSB14, CSD18], require an error-detection mechanism and do not guarantee ML decoding :(

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?
  - SCL decoding with unbounded list size has an exponential complexity :(
  - Sequential decoders [Fan63, NC12, MT14, Tri18, JH19] require a parameter-optimization and do not guarantee ML decoding :(
  - Flip decoders [ABSB14, CSD18], require an error-detection mechanism and do not guarantee ML decoding :(
- Successive cancellation ordered search (SCOS) decoding is an ML decoder, ...

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?
  - SCL decoding with unbounded list size has an exponential complexity :(
  - Sequential decoders [Fan63, NC12, MT14, Tri18, JH19] require a parameter-optimization and do not guarantee ML decoding :(
  - Flip decoders [ABSB14, CSD18], require an error-detection mechanism and do not guarantee ML decoding :(
- Successive cancellation ordered search (SCOS) decoding is an ML decoder, ...
  - ✓ ...which requires neither an error-detection mechanism nor a parameter optimization :)

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?
  - SCL decoding with unbounded list size has an exponential complexity :(
  - Sequential decoders [Fan63, NC12, MT14, Tri18, JH19] require a parameter-optimization and do not guarantee ML decoding :(
  - Flip decoders [ABSB14, CSD18], require an error-detection mechanism and do not guarantee ML decoding :(
- Successive cancellation ordered search (SCOS) decoding is an ML decoder, ...
  - ✓ ...which requires neither an error-detection mechanism nor a parameter optimization :)
  - ✓ ...whose complexity approaches (for a wide range of codes) to that of SC decoding for high-SNR regime :)

Motivating Question

- Is it possible to mimic the behaviour of inactivation decoding over more general BMSCs?
  - SCL decoding with unbounded list size has an exponential complexity :(
  - Sequential decoders [Fan63, NC12, MT14, Tri18, JH19] require a parameter-optimization and do not guarantee ML decoding :(
  - Flip decoders [ABSB14, CSD18], require an error-detection mechanism and do not guarantee ML decoding :(
- Successive cancellation ordered search (SCOS) decoding is an ML decoder, ...
  - ✓ ...which requires neither an error-detection mechanism nor a parameter optimization :)
  - ✓ ...whose complexity approaches (for a wide range of codes) to that of SC decoding for high-SNR regime :)

Based on a joint work with Peihong Yuan (TUM) [YC21]

A Preview of the Algorithm

A tree search algorithm, which...

- ... flips the bits of valid paths to find a leaf with higher likelihood than other leaves, if such a leaf exists.

A Preview of the Algorithm

A tree search algorithm, which...

- ... flips the bits of valid paths to find a leaf with higher likelihood than other leaves, if such a leaf exists.
- ... repeats until the ML decision is found.

A Preview of the Algorithm

A tree search algorithm, which...

- ... flips the bits of valid paths to find a leaf with higher likelihood than other leaves, if such a leaf exists.
- ... repeats until the ML decision is found.
- ... stores a list of branches that is updated progressively while running partial successive cancellation decoding by flipping the bits of the most likely leaf at each iteration.

A Preview of the Algorithm

A tree search algorithm, which...

- ... flips the bits of valid paths to find a leaf with higher likelihood than other leaves, if such a leaf exists.
- ... repeats until the ML decision is found.
- ... stores a list of branches that is updated progressively while running partial successive cancellation decoding by flipping the bits of the most likely leaf at each iteration.
- The search is ordered according to the probability that they provide the ML decision.

A Preview of the Algorithm

A tree search algorithm, which...

- ... flips the bits of valid paths to find a leaf with higher likelihood than other leaves, if such a leaf exists.
- ... repeats until the ML decision is found.
- ... stores a list of branches that is updated progressively while running partial successive cancellation decoding by flipping the bits of the most likely leaf at each iteration.
- The search is ordered according to the probability that they provide the ML decision.

Hence, successive cancellation ordered search decoding

Some Definitions

- The log-probability of a path $\tilde{u}_1^i$ via SC decoding [TV15]

$$M_i\left(\tilde{u}_1^i\right) \triangleq -\log P_{U_1^i|Y_1^N}\left(\tilde{u}_1^i|y_1^N\right). \tag{3}$$

Some Definitions

- The log-probability of a path $\tilde{u}_1^i$ via SC decoding [TV15]

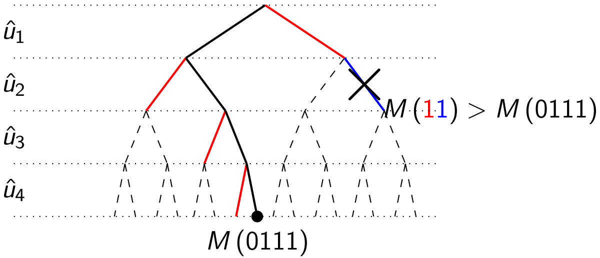
$$M_i\left(\tilde{u}_1^i\right) \triangleq -\log P_{U_1^i|Y_1^N}\left(\tilde{u}_1^i|y_1^N\right). \tag{3}$$

- A score function for a path $\tilde{u}_1^i$ [JH19]

$$S_i\left(\tilde{u}_1^i\right) \triangleq M_i\left(\tilde{u}_1^i\right) + \sum_{j=1}^{i} \log\left(1-p_j\right) \tag{4}$$

  where p_j is the probability of the event that the first bit error occurred for u_j in SC decoding and $S\left(\tilde{u}^0\right) \triangleq 0$.

Ordered Search Decoding: Example



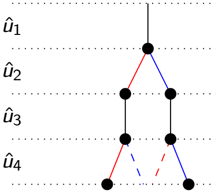
- Search priority: $S_i\left(\tilde{u}_1^i\right)$
- Decision & Pruning: $M_i\left(\tilde{u}_1^i\right)$

Ordered Search Decoding

1. SC path: 0111, $M(0111)$
   $M_1(1), M_2(00), M_3(010), M_4(0110)$
   $S(1), S(00), S(010), S(0110)$
2. Find the sub-path with lowest S.
3. Return to the Lowest Common Ancestor and re-start SC decoding.
4. $M\,(11) > M\,(0111)$
5. Repeat until it is impossible to find a more reliable path.

Decoding Complexity

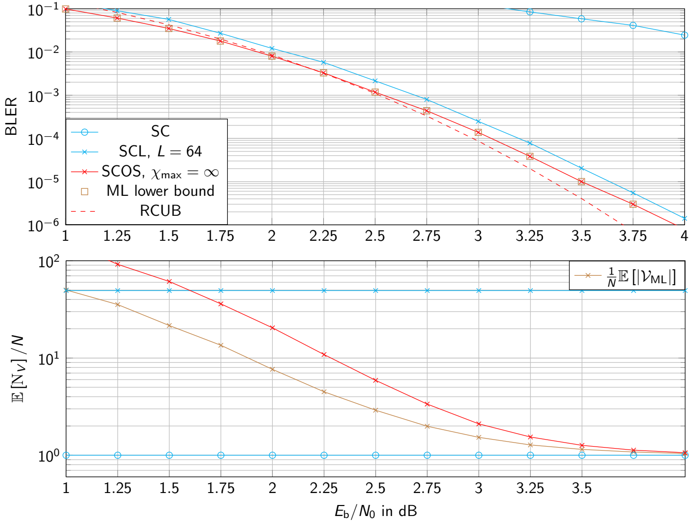
- Space complexity: $\mathcal{O}\left(N \log N\right)$
- N_V: Number of node-visits in the SC-decoding tree for SCOS decoding



$\mathrm{N}_V^{\mathrm{SCL}}\left(2, \{2,4\}\right) = 7$

- $\mathcal{V}_{\mathrm{ML}} \triangleq \bigcup_{i=1}^{N} \left\{u^i \in \{0,1\}^i : M\left(u^i\right) \leq M\left(\hat{\boldsymbol{u}}_{\mathrm{ML}}\right)\right\}$
- $\mathrm{N}_V \geq \left|\mathcal{V}_{\mathrm{ML}}\right|$ due to the redundant visits.

(128, 64) PAC code



Outline

1 Overview of Polar Codes

2 Successive Cancellation Inactivation Decoding

3 Successive Cancellation Ordered Search Decoding

4 Conclusions

Conclusions and Outlook

- Complexity-adaptive block-wise optimum decoding for BMSCs, where the complexity approaches very closely to that of SC decoding for wide range of codes (polar codes, short RM codes, their modifications, etc.) as the channel quality gets better.
- For much more, see [CNP20, CP21, YC21].
- A promising direction is to extend SCI decoding for codes over q-ary erasure channels.
- For SCOS decoding, optimizing the search schedule for the RM and/or PAC codes of larger blocklengths is promising. It has already been explored in [HDG+21] for RM codes of various lengths and rates.

References I

O. Afisiadis, A. Balatsoukas-Stimming, and A. Burg.
A low-complexity improved successive cancellation decoder for polar codes.
pages 2116–2120, 2014.

E. Arıkan.
Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels.
IEEE Trans. Inf. Theory, 55(7):3051–3073, July 2009.

E. Arıkan.
From sequential decoding to channel polarization and back again.
CoRR, abs/1908.09594, 2019.

E. Arıkan and E. Telatar.
On the rate of channel polarization.
In *IEEE Int. Symp. Inf. Theory*, pages 1493–1495, 2009.

E. R. Berlekamp.
Algebraic Coding Theory - Revised Edition.
World Scientific Publishing Co., Inc., USA, 2015.

References II

M. C. Coşkun, J. Neu, and H. D. Pfister.
Successive cancellation inactivation decoding for modified Reed-Muller and eBCH codes.
In *IEEE Int. Symp. Inf. Theory*, pages 437–442, 2020.

M. C. Coşkun and H. D. Pfister.
An information-theoretic perspective on successive cancellation list decoding: Analysis and code design (first revision).
2021.

L. Chandesris, V. Savin, and D. Declercq.
Dynamic-SCFlip decoding of polar codes.
66(6):2333–2345, 2018.

I. Dumer and K. Shabunov.
Soft-decision decoding of reed-muller codes: recursive lists.
IEEE Trans. Inf. Theory, 52(3):1260–1266, 2006.

A. Eslami and H. Pishro-Nik.
On bit error rate performance of polar codes in finite regime.
In *Proc. 48th Annu. Allerton Conf. on Commun., Control, and Comput.*, pages 188–194, September 2010.

References III

R. Fano.
A heuristic discussion of probabilistic decoding.
9(2):64–74, 1963.

S. A. Hashemi, N. Doan, W. J. Gross, J. Cioffi, and A. Goldsmith.
A tree search approach for maximum-likelihood decoding of reed-muller codes.
In *2021 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2021.

M. Jeong and S. Hong.
SC-Fano decoding of polar codes.
IEEE Access, 7:81682–81690, 2019.

F. Lázaro, G. Liva, and G. Bauch.
Inactivation decoding of LT and raptor codes: Analysis and code design.
65(10):4114–4127, October 2017.

C. Measson, A. Montanari, and R. Urbanke.
Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding.
54(12):5277–5307, December 2008.

References IV

V. Miloslavskaya and P. Trifonov.
Sequential decoding of polar codes.
18(7):1127–1130, 2014.

K. Niu and K. Chen.
CRC-aided decoding of polar codes.
16(10):1668–1671, 2012.

H. Pishro-Nik and F. Fekri.
On decoding of low-density parity-check codes over the binary erasure channel.
50(3):439–454, March 2004.

E. Paolini, G. Liva, B. Matuz, and M. Chiani.
Maximum likelihood erasure decoding of ldpc codes: Pivoting algorithms and code design.
IEEE Transactions on Communications, 60(11):3209–3220, 2012.

M. Rowshan, A. Burg, and E. Viterbo.
Polarization-adjusted convolutional (PAC) codes: Fano decoding vs list decoding.
CoRR, abs/2002.06805, 2020.

References V

T.J. Richardson and R.L. Urbanke.
Efficient encoding of low-density parity-check codes.
IEEE Transactions on Information Theory, 47(2):638–656, 2001.

A. Shokrollahi.
Raptor codes.
52(6):2551–2567, June 2006.

N. Stolte.
Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung.
PhD thesis, TU Darmstadt, 2002.

P. Trifonov and V. Miloslavskaya.
Polar subcodes.
IEEE J. Sel. Areas Commun., 34(2):254–266, Feb. 2016.

P. Trifonov.
A score function for sequential decoding of polar codes.
pages 1470–1474, 2018.

References VI

I. Tal and A. Vardy.
List decoding of polar codes.
IEEE Trans. Inf. Theory, 61(5):2213–2226, May 2015.

Y. Urman and D. Burshtein.
Efficient maximum likelihood decoding of polar codes over the binary erasure channel.
In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 1600–1605, 2021.

Y. Urman, G. Mogilevsky, and D. Burshtein.
Efficient belief propagation list ordered statistics decoding of polar codes.
CoRR, abs/2107.05965, 2021.

P. Yuan and M. C. Coşkun.
Complexity-adaptive maximum-likelihood decoding of modified $\boldsymbol{G}_n$-coset codes.
Inf. Theory Workshop (ITW), accepted, 2021.

H. Yao, A. Fazeli, and A. Vardy.
List decoding of arıkan's PAC codes.
CoRR, abs/2005.13711, 2020.