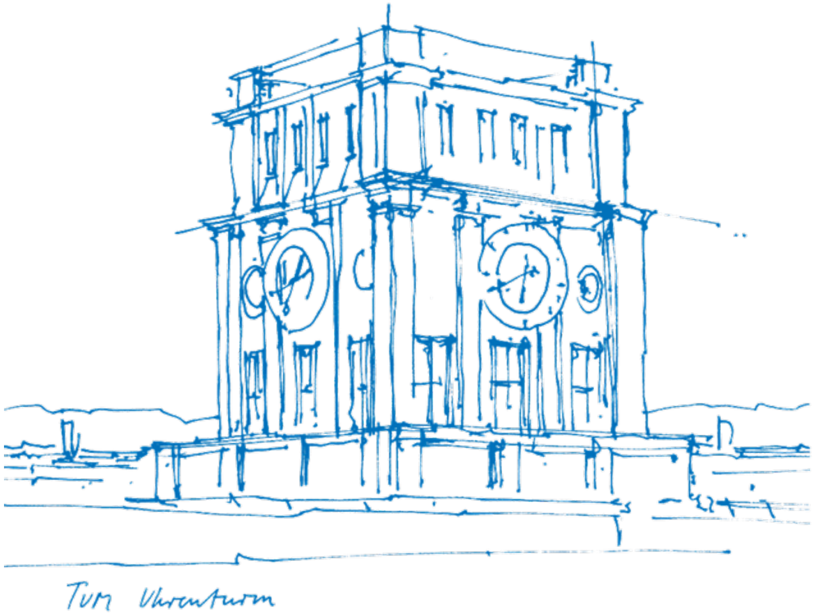
Polar Codes: Basics and Recent Advances

Mustafa Cemil Coşkun, mustafa.coskun@tum.de
Technical University of Munich (TUM)
German Aerospace Center (DLR)

INCOMING School on Massive IoT, Distributed and Decentralized Information Processing
November 5, 2020



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**

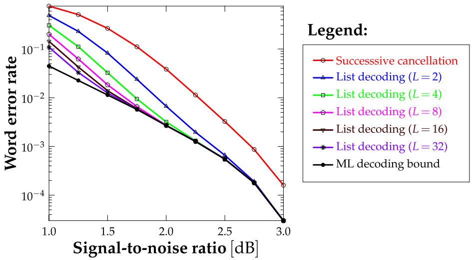


Fig. 1. List-decoding performance for a polar code of length $n = 2048$ and rate $R = 0.5$ on the BPSK-modulated Gaussian channel. The code was constructed using the methods of [15], with optimization for $E_b/N_0 = 2$ dB.

- SC list (SCL) decoding with CRC and large list-size performs very well and matches maximum-likelihood (ML) [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**

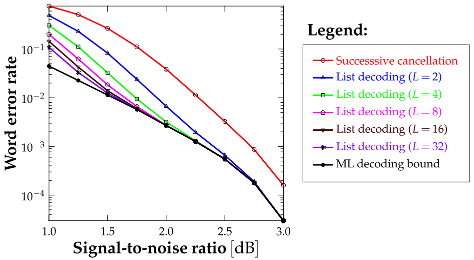


Fig. 1. List-decoding performance for a polar code of length $n = 2048$ and rate $R = 0.5$ on the BPSK-modulated Gaussian channel. The code was constructed using the methods of [15], with optimization for $E_b/N_0 = 2\,\text{dB}$.

- SC list (SCL) decoding with CRC and large list-size performs very well and matches maximum-likelihood (ML) [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*

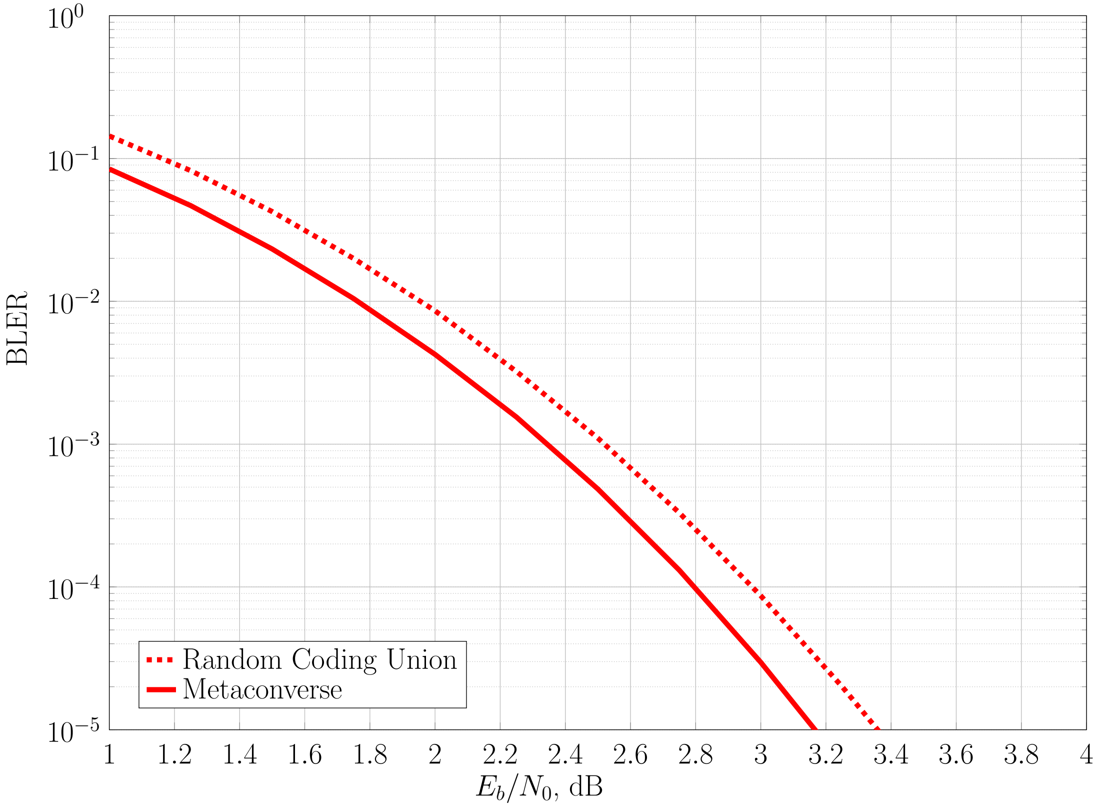
***Abstract*—An extension of polar codes is proposed, which allows some of the frozen symbols, called dynamic frozen symbols, to be data-dependent. A construction of polar codes with dynamic frozen symbols, being subcodes of extended BCH codes, is proposed. The proposed codes have higher minimum distance 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

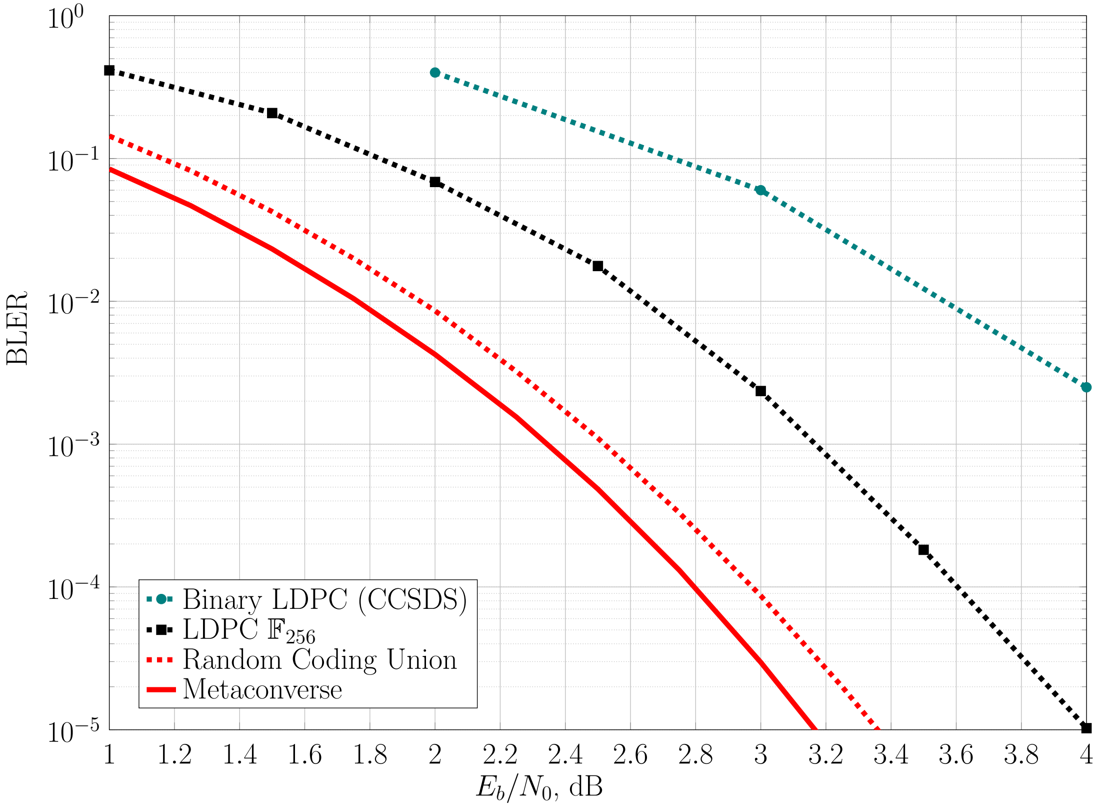
- Later, polar codes were extended with the concept of dynamic frozen bits, which enabled state-of-art designs.
- It is also shown that any code can be decoded using SCL decoding, but some require very large complexity for a good performance.

$n = 128, k = 64$



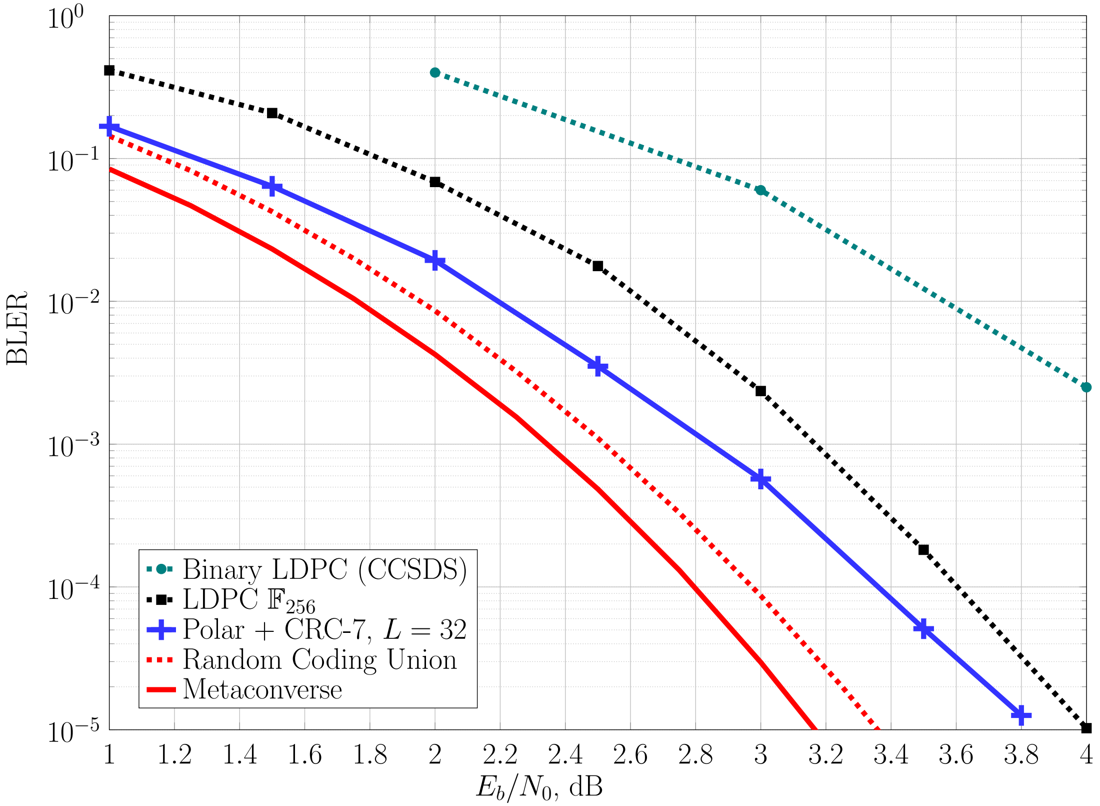
BLER
E_b/N_0, dB
Random Coding Union
Metaconverse
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 3.2 3.4 3.6 3.8 4

$n = 128, k = 64$



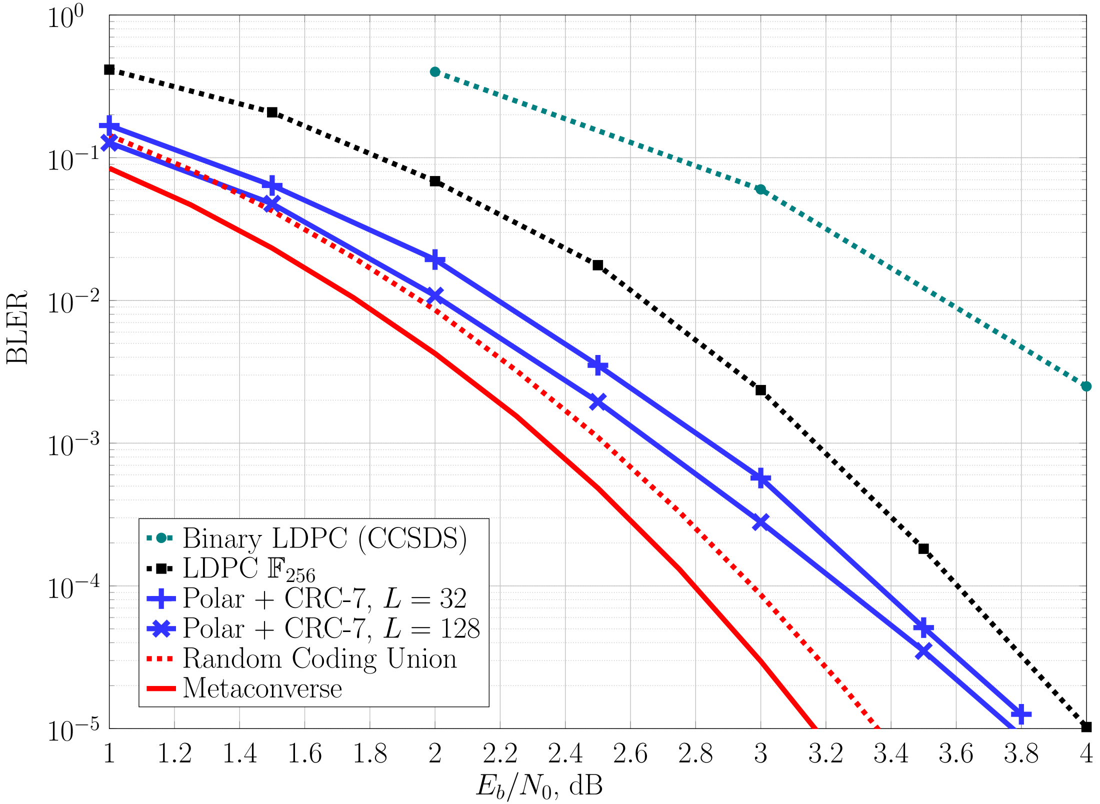
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
E_b/N_0, dB
Binary LDPC (CCSDS)
LDPC $\mathbb{F}_{256}$
Random Coding Union
Metaconverse

$n = 128, k = 64$



BLER
E_b/N_0, dB
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
Binary LDPC (CCSDS)
LDPC $\mathbb{F}_{256}$
Polar + CRC-7, $L = 32$
Random Coding Union
Metaconverse

$n = 128, k = 64$



Most of the curves can be obtained on pretty-good-codes.org. For the rest, send an e-mail.

Outline

Outline

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., $H(X|Y) \approx 1$).

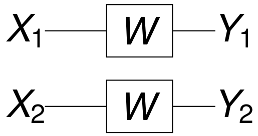
Channel polarization is a technique to convert any BMS channel to a mixture of easy channels, asymptotically in the block length.

- 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.,

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



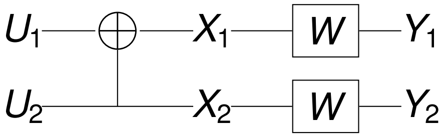
Example: Binary Erasure Channel

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

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



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

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

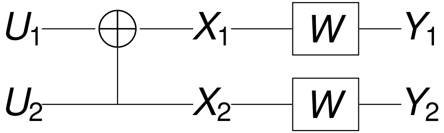
Example: Binary Erasure Channel

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

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



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

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

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

$$(Y_1, Y_2) = \begin{cases} \\ \\ \end{cases}$$

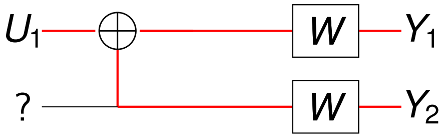
Example: Binary Erasure Channel

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

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



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

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

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

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

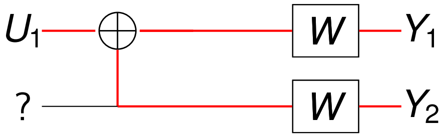
Example: Binary Erasure Channel

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

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

we set

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



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

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

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

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

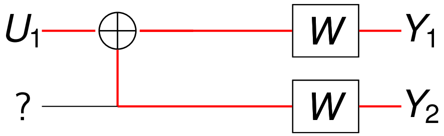
Example: Binary Erasure Channel

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

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



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

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

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

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

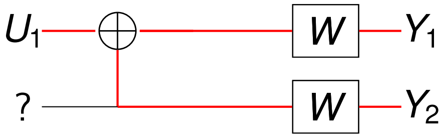
Example: Binary Erasure Channel

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

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



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

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

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

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

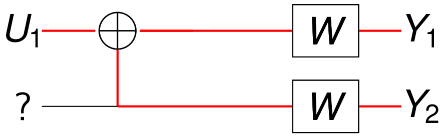
Example: Binary Erasure Channel

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

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



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

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

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

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

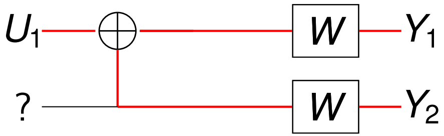
Example: Binary Erasure Channel

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

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



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

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

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

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

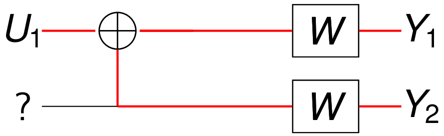
Example: Binary Erasure Channel

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

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



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

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

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

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

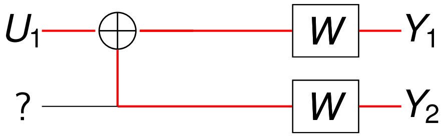
Example: Binary Erasure Channel

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

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



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

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

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

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

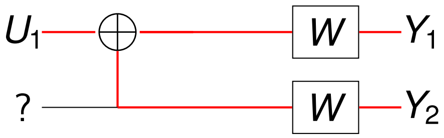
Example: Binary Erasure Channel

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

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



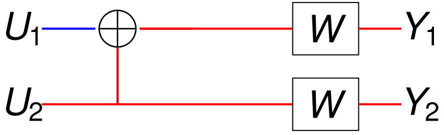
$$X_1^2 = U_1^2 \mathbf{G}_2$$
$$\mathbf{G}_2 \triangleq \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

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

Example: Binary Erasure Channel

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

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



we set

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

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

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

- 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):

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

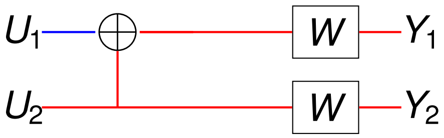
Example: Binary Erasure Channel

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

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

we set

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



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

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

- 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):

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

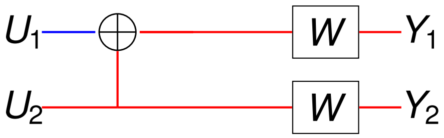
Example: Binary Erasure Channel

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

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



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

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

- 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):

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

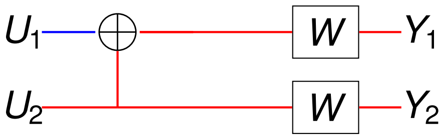
Example: Binary Erasure Channel

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

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



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

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

- 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):

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

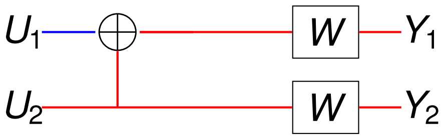
Example: Binary Erasure Channel

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

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



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

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

- 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):

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

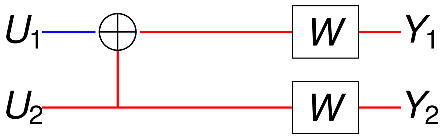
Example: Binary Erasure Channel

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

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



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

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

- 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):

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

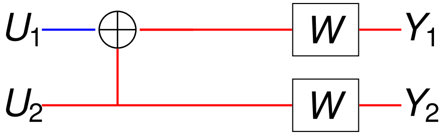
Example: Binary Erasure Channel

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

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

we set

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



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

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

- 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):

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

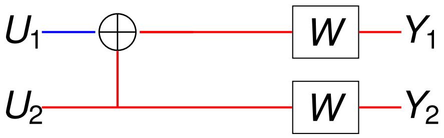
Example: Binary Erasure Channel

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

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



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

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

- 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):

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

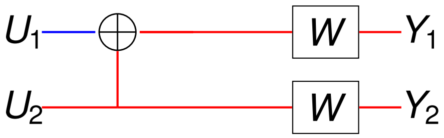
Example: Binary Erasure Channel

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

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

we set

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



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

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

- 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):

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

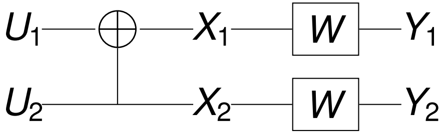
Example: Binary Erasure Channel

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

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



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

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

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

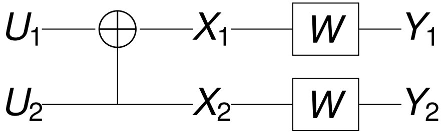
Example: Binary Erasure Channel

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

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



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

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

- 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.

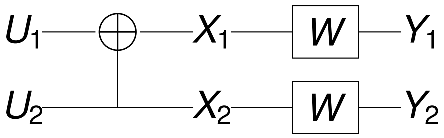
Example: Binary Erasure Channel

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

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



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

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

- 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., $H(U_2|Y_1^2 U_1) = \epsilon^2$.

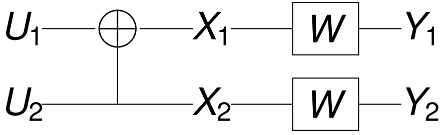
Example: Binary Erasure Channel

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

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

we set

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



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

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

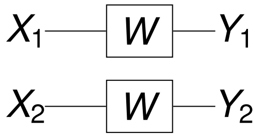
- The input U_1 is erased w.p. $(2\epsilon - \epsilon^2)$, i.e., $H(U_1|Y_1^2) = 2\epsilon - \epsilon^2$.
- Given U_1, the input U_2 is erased w.p. ϵ^2, i.e., $H(U_2|Y_1^2 U_1) = \epsilon^2$.

Hence, we have

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

A Basic Transform: General BMS Channels

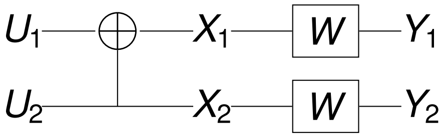
Given two independent copies of a BMS channel $W : \{0, 1\} \to \mathcal{Y}$,



A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0,1\} \to \mathcal{Y}$, set

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



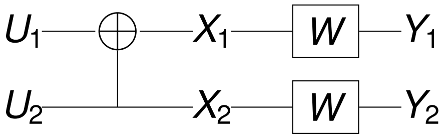
$$X_1^2 = U_1^2 \mathbf{G}_2$$

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

A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0,1\} \to \mathcal{Y}$, set

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



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

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

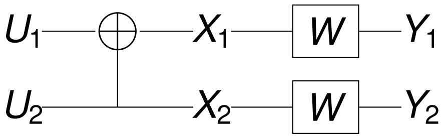
Let $H(W) \triangleq H(X_1|Y_1)$. As (X_1, Y_1) is independent from (X_2, Y_2), we write

$$H(X_1|Y_1) + H(X_2|Y_2) = 2H(W)$$

A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0,1\} \to \mathcal{Y}$, set

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



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

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

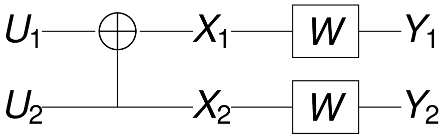
Let $H(W) \triangleq H(X_1|Y_1)$. As (X_1, Y_1) is independent from (X_2, Y_2), we write

$$H(X_1|Y_1) + H(X_2|Y_2) = 2H(W) = H(U_1^2|Y_1^2)$$

A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0,1\} \to \mathcal{Y}$, set

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



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

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

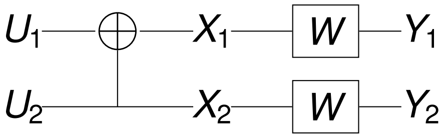
Let $H(W) \triangleq H(X_1|Y_1)$. As (X_1, Y_1) is independent from (X_2, Y_2), we write

$$\begin{aligned} H(X_1|Y_1) + H(X_2|Y_2) = 2H(W) &= H(U_1^2|Y_1^2) \\ &= H(U_1|Y_1^2) + H(U_2|Y_1^2 U_1) \end{aligned}$$

A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0,1\} \to \mathcal{Y}$, set

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



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

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

Let $H(W) \triangleq H(X_1|Y_1)$. As (X_1, Y_1) is independent from (X_2, Y_2), we write

$$\begin{aligned} H(X_1|Y_1) + H(X_2|Y_2) = 2H(W) &= H(U_1^2|Y_1^2) \\ &= H(U_1|Y_1^2) + H(U_2|Y_1^2 U_1) \end{aligned}$$

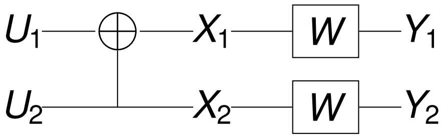
Now consider the second term at the RHS:

$$H(U_2|Y_1 Y_2 U_1) \leq H(U_2|Y_2)$$

A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0,1\} \to \mathcal{Y}$, set

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



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

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

Let $H(W) \triangleq H(X_1|Y_1)$. As (X_1, Y_1) is independent from (X_2, Y_2), we write

$$\begin{aligned} H(X_1|Y_1) + H(X_2|Y_2) = 2H(W) &= H(U_1^2|Y_1^2) \\ &= H(U_1|Y_1^2) + H(U_2|Y_1^2 U_1) \end{aligned}$$

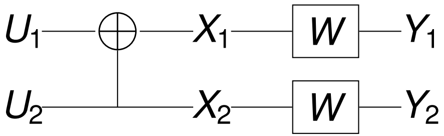
Now consider the second term at the RHS:

$$H(U_2|Y_1 Y_2 U_1) \leq H(U_2|Y_2)$$

A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0, 1\} \to \mathcal{Y}$, set

$$
\begin{aligned}
X_1 &= U_1 \oplus U_2 \\
X_2 &= U_2
\end{aligned}
$$



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

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

Let $H(W) \triangleq H(X_1|Y_1)$. As (X_1, Y_1) is independent from (X_2, Y_2), we write

$$
\begin{aligned}
H(X_1|Y_1) + H(X_2|Y_2) = 2H(W) &= H(U_1^2|Y_1^2) \\
&= H(U_1|Y_1^2) + H(U_2|Y_1^2 U_1)
\end{aligned}
$$

Now consider the second term at the RHS:

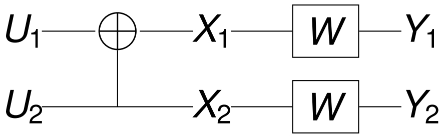
$$H(U_2|Y_1 Y_2 U_1) \leq H(U_2|Y_2) = H(X_2|Y_2) = H(W)$$

Combining these, we conclude $H(U_2|Y_1^2 U_1) \leq H(W) \leq H(U_1|Y_1^2)$.

A Basic Transform: General BMS Channels

Given two independent copies of a BMS channel $W : \{0,1\} \to \mathcal{Y}$, set

$$\begin{aligned} X_1 &= U_1 \oplus U_2 \\ X_2 &= U_2 \end{aligned}$$



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

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

Let $H(W) \triangleq H(X_1|Y_1)$. As (X_1, Y_1) is independent from (X_2, Y_2), we write

$$\begin{aligned} H(X_1|Y_1) + H(X_2|Y_2) = 2H(W) &= H(U_1^2|Y_1^2) \\ &= H(U_1|Y_1^2) + H(U_2|Y_1^2 U_1) \end{aligned}$$

Now consider the second term at the RHS:

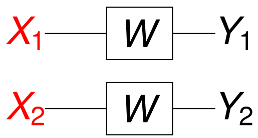
$$H(U_2|Y_1 Y_2 U_1) \leq H(U_2|Y_2) = H(X_2|Y_2) = H(W)$$

Combining these, we conclude $H(U_2|Y_1^2 U_1) \leq H(W) \leq H(U_1|Y_1^2)$. Indeed, the polarization is strict [Arı09], i.e., if $H(W) \notin \{0,1\}$, then

$$H(U_2|Y_1^2 U_1) < H(W) < H(U_1|Y_1^2)$$

Polarized Synthetic Channels

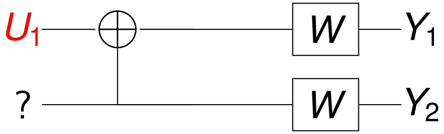
Given two independent copies of $W : \{0, 1\} \to \mathcal{Y}$ with a capacity of $C(W)$, we obtain two synthetic channels:



Polarized Synthetic Channels

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

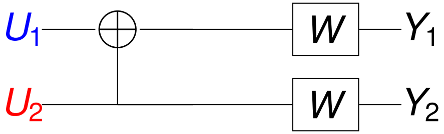
- A downgraded channel $W_2^{(1)} : \{0, 1\} \to \mathcal{Y}^2$ having input U_1 and output Y_1^2 with $C(W_2^{(1)}) < C(W)$



Polarized Synthetic Channels

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

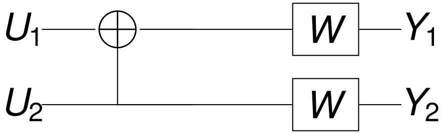
- A downgraded channel $W_2^{(1)} : \{0,1\} \to \mathcal{Y}^2$ having input U_1 and output Y_1^2 with $C(W_2^{(1)}) < C(W)$
- An upgraded channel $W_2^{(2)} : \{0,1\} \to \mathcal{Y}^2 \times \{0,1\}$ having input U_2 and output (Y_1^2, U_1) with $C(W_2^{(2)}) > C(W)$



Polarized Synthetic Channels

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 $C(W_2^{(2)}) > C(W)$

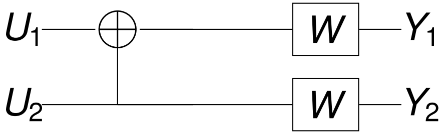


This suggests that a successive decoding can be employed to achieve $C(W)$ [Arı09]:

Polarized Synthetic Channels

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 $C(W_2^{(2)}) > C(W)$



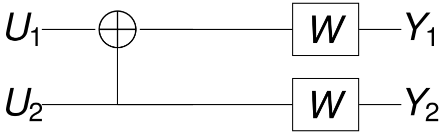
This suggests that a successive decoding can be employed to achieve $C(W)$ [Arı09]:

- 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

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 $C(W_2^{(2)}) > C(W)$

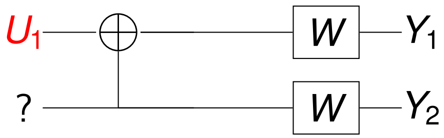


This suggests that a successive decoding can be employed to achieve $C(W)$ [Arı09]:

- 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 $\hat{U}_2$.

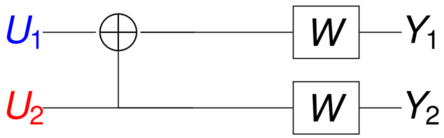
Genie-Aided vs. Real Successive Decoder

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



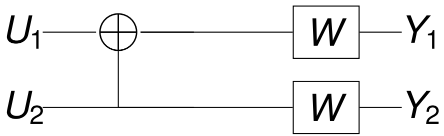
Genie-Aided vs. Real Successive Decoder

- 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

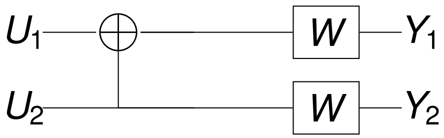
- The channel $W_2^{(1)}$ has the input U_1 and output Y_1^2 ✓
- The channel $W_2^{(2)}$ has the input U_2 and output (Y_1^2, U_1)!



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)!



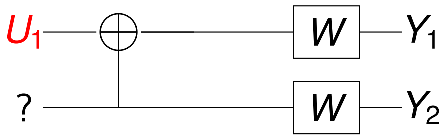
It is possible to obtain $\hat{U}_1$ by first decoding $W_2^{(1)}$. What is the effect of using $\hat{U}_1$ instead of U_1 on the block error events?

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:

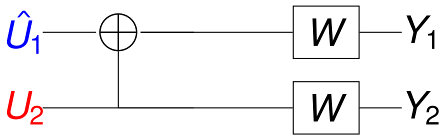
$$\tilde{U}_1 = f_1(Y_1^2)$$

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:

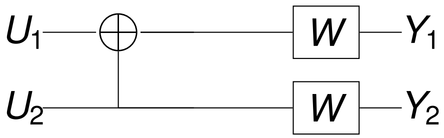
$$\tilde{U}_1 = f_1(Y_1^2)$$
$$\tilde{U}_2 = f_2(Y_1^2 U_1)$$

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

We can apply the basic transform recursively to the independent copies of (W), $(W_2^{(1)}, W_2^{(2)})$, $(W_4^{(1)}, W_4^{(2)}, W_4^{(3)}, W_4^{(4)})$, etc., as many times as needed.

Polar Transform

We can apply the basic transform recursively to the independent copies of (W), $(W_2^{(1)}, W_2^{(2)})$, $(W_4^{(1)}, W_4^{(2)}, W_4^{(3)}, W_4^{(4)})$, etc., as many times as needed.

Definition

The Kronecker product of two matrices $\mathbf{X}$ and $\mathbf{Y}$ is

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

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

Polar Transform

We can apply the basic transform recursively to the independent copies of (W), $(W_2^{(1)}, W_2^{(2)})$, $(W_4^{(1)}, W_4^{(2)}, W_4^{(3)}, W_4^{(4)})$, etc., as many times as needed.

Definition

The Kronecker product of two matrices $\mathbf{X}$ and $\mathbf{Y}$ is

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

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

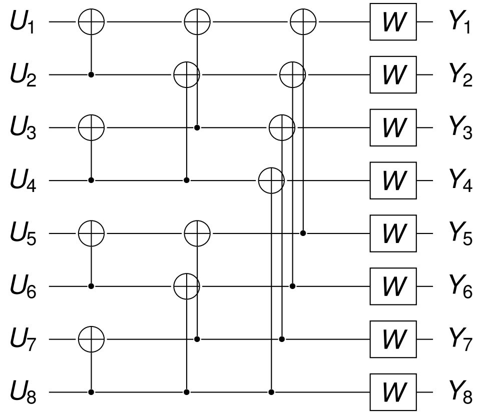
Example

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

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

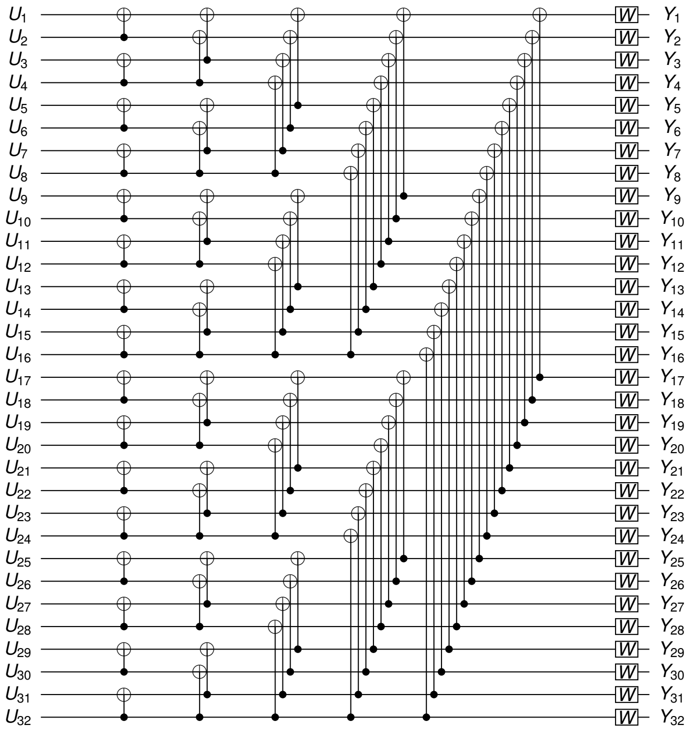
Polar Transform (N=8)

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



Polar Transform (N=32)

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



Channel Polarization

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

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

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

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

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

Channel Polarization

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

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

Since the transform is information-lossless, we can write

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

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

A capacity-achieving scheme:

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

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

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

A capacity-achieving scheme:

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

Channel Polarization

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

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

Since the transform is information-lossless, we can write

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

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

A capacity-achieving scheme:

- 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.

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

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

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

A capacity-achieving scheme:

- 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.

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

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

A capacity-achieving scheme:

- 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.
- $P_B \leq \sum_{i\in A} \delta$

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:

- 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.
- $P_B \le \sum_{i\in A} \delta = N \cdot C(W) \cdot \delta$

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

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

A capacity-achieving scheme:

- 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 $1/N$).

Channel Polarization

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

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

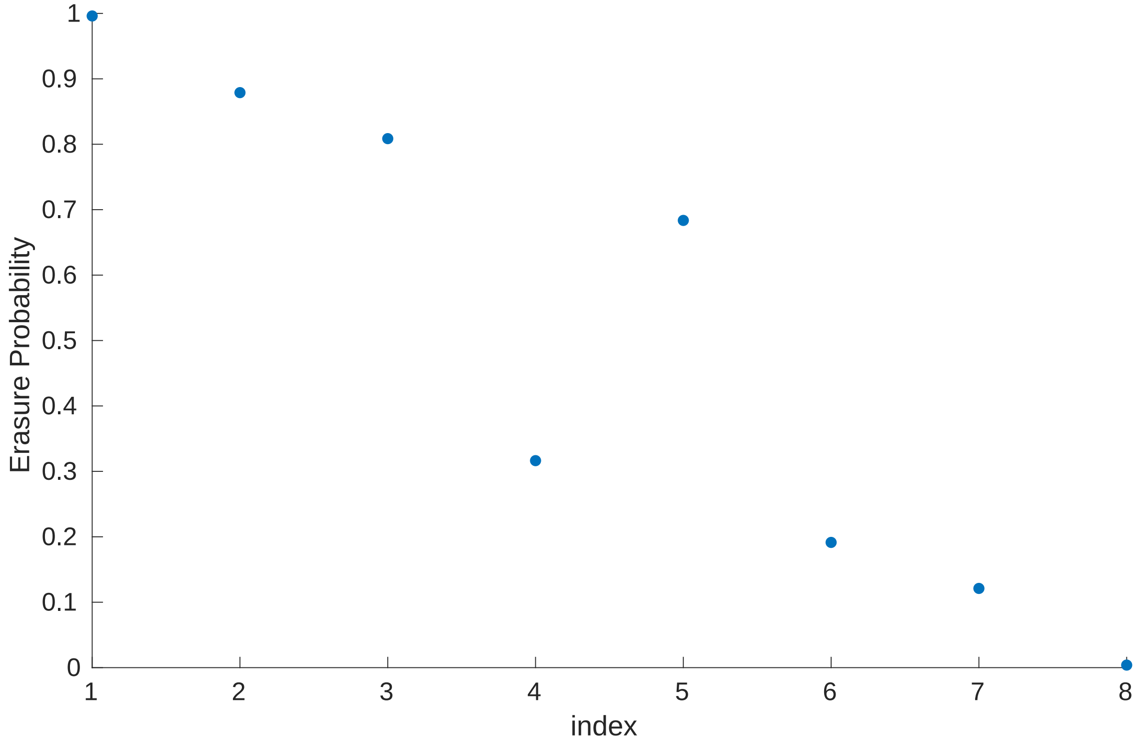
Since the transform is information-lossless, we can write

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

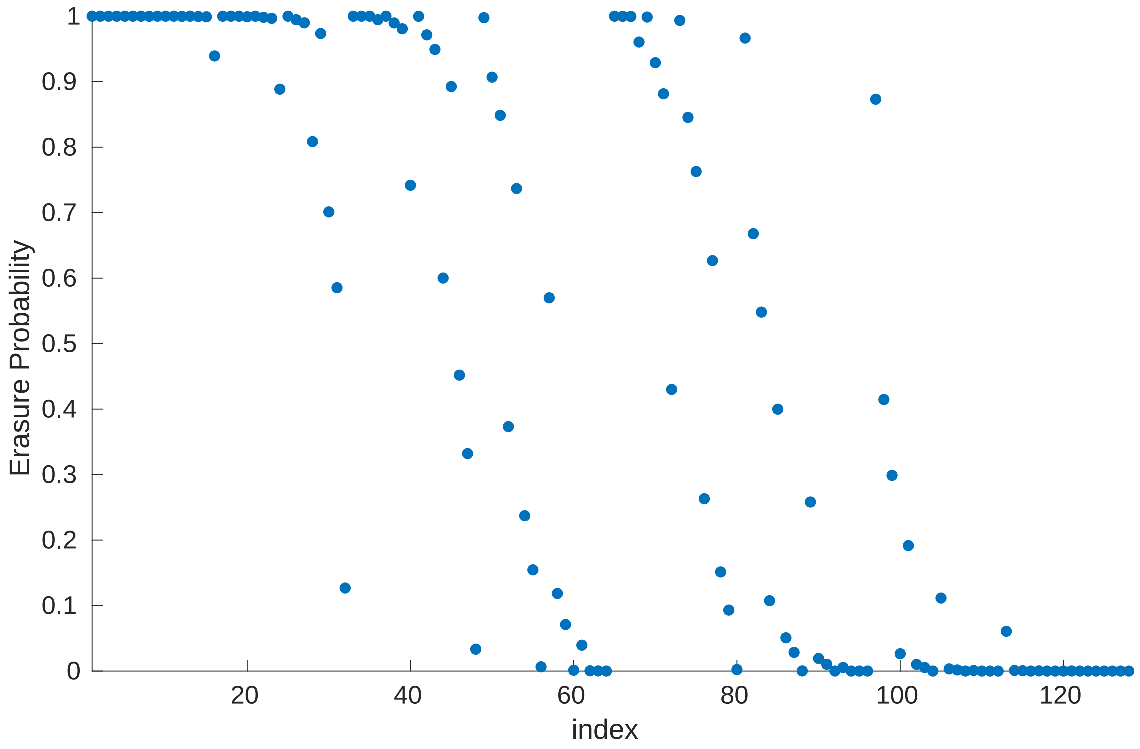
A capacity-achieving scheme:

- 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.
- $P_B \le \sum_{i\in A} \delta = N \cdot C(W) \cdot \delta \le N \cdot C(W) \cdot 2^{-\sqrt{N}}$, resulting in $P_B \to 0$.
- Indeed, polarization holds for $\delta = \mathcal{O}(2^{-\sqrt{N}})$ [AT09] (i.e., faster than $1/N$).

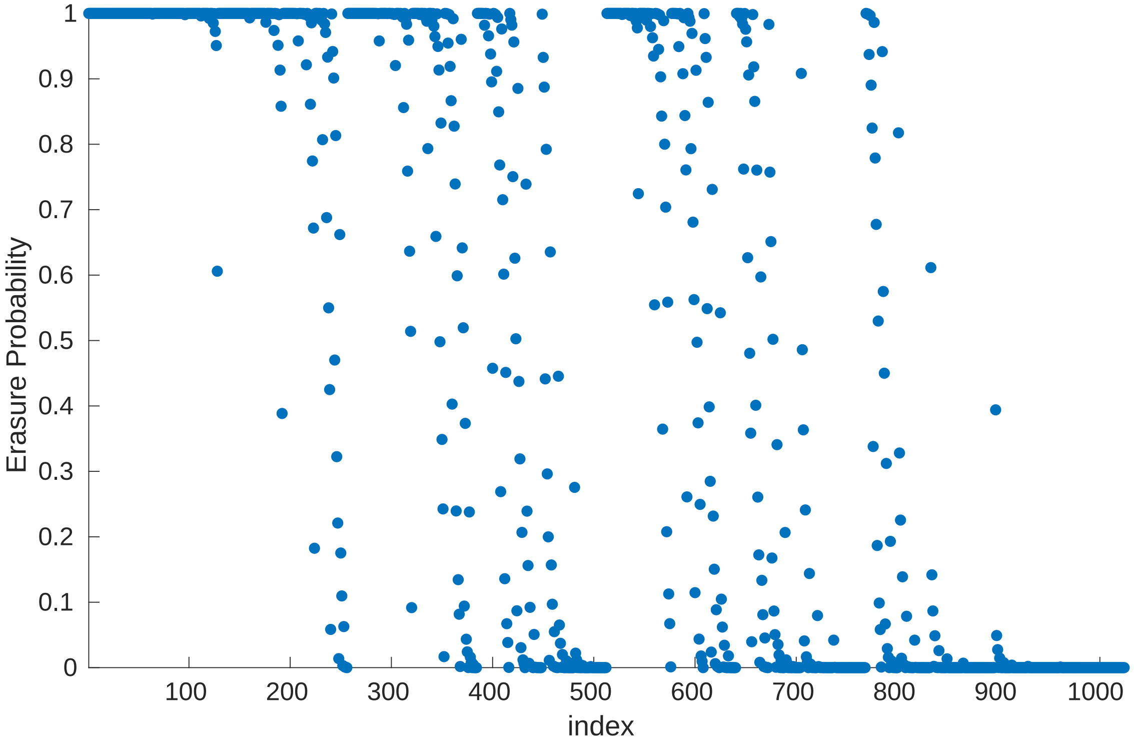
Channel Polarization - Numerical ($N = 2^3$, BEC(0.5))



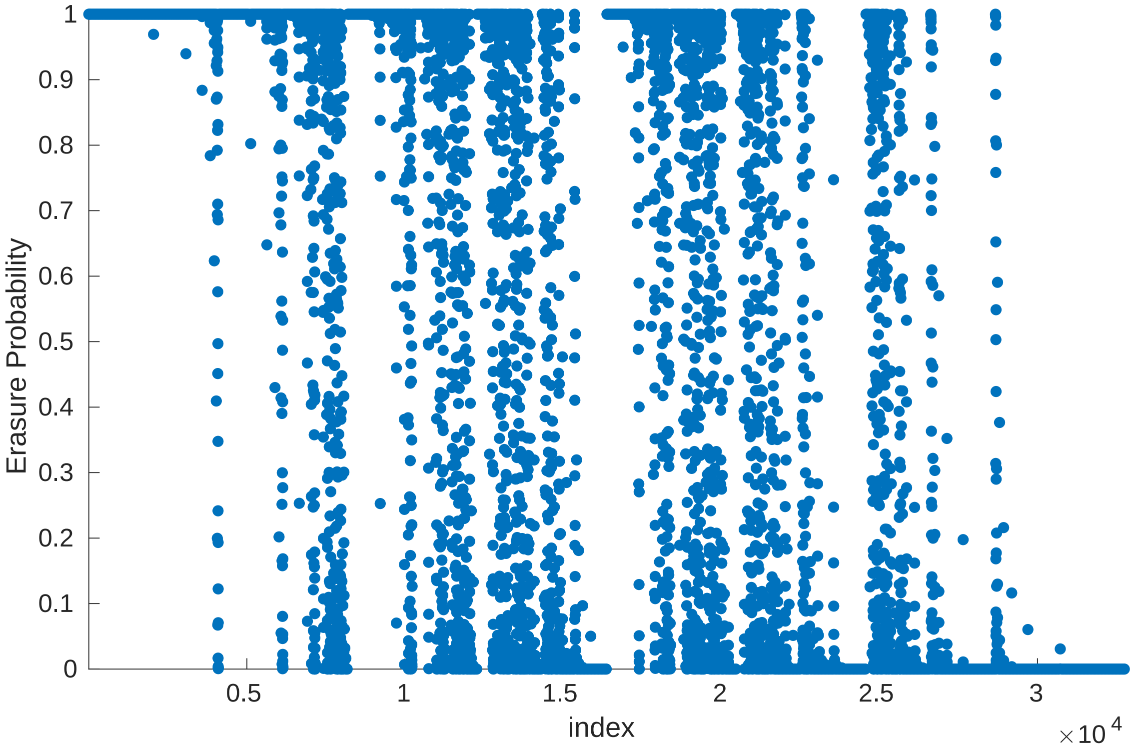
Channel Polarization - Numerical ($N = 2^7$, BEC(0.5))



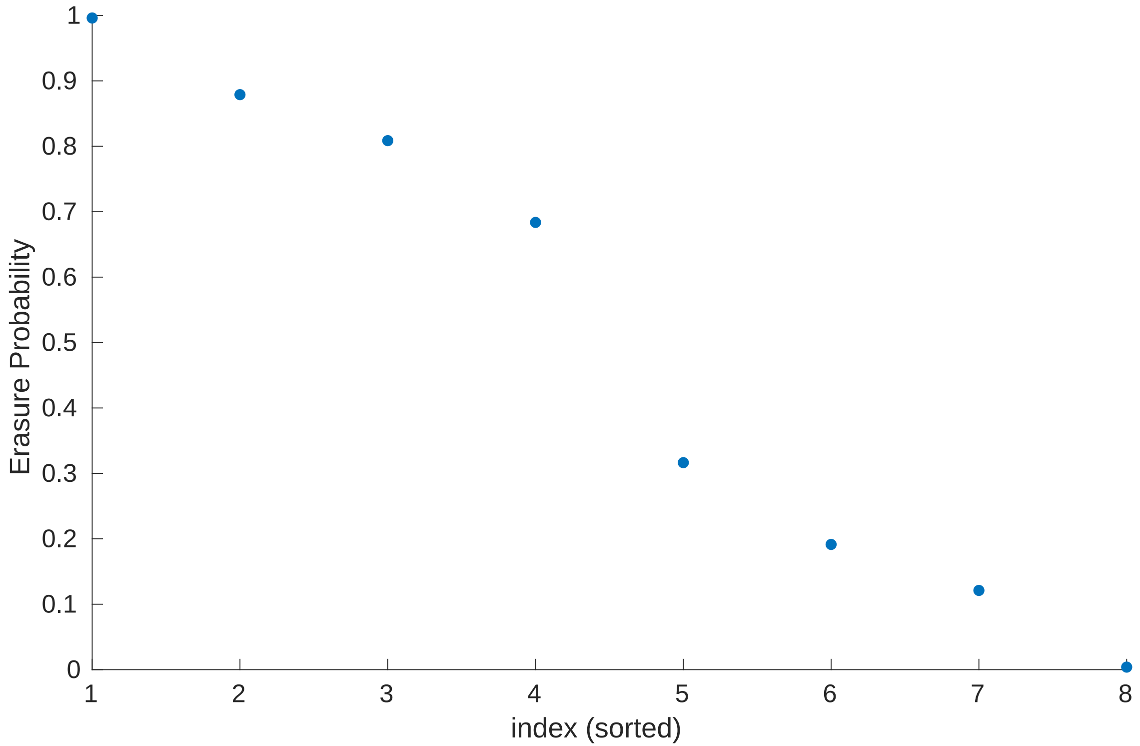
Channel Polarization - Numerical ($N = 2^{10}$, BEC(0.5))



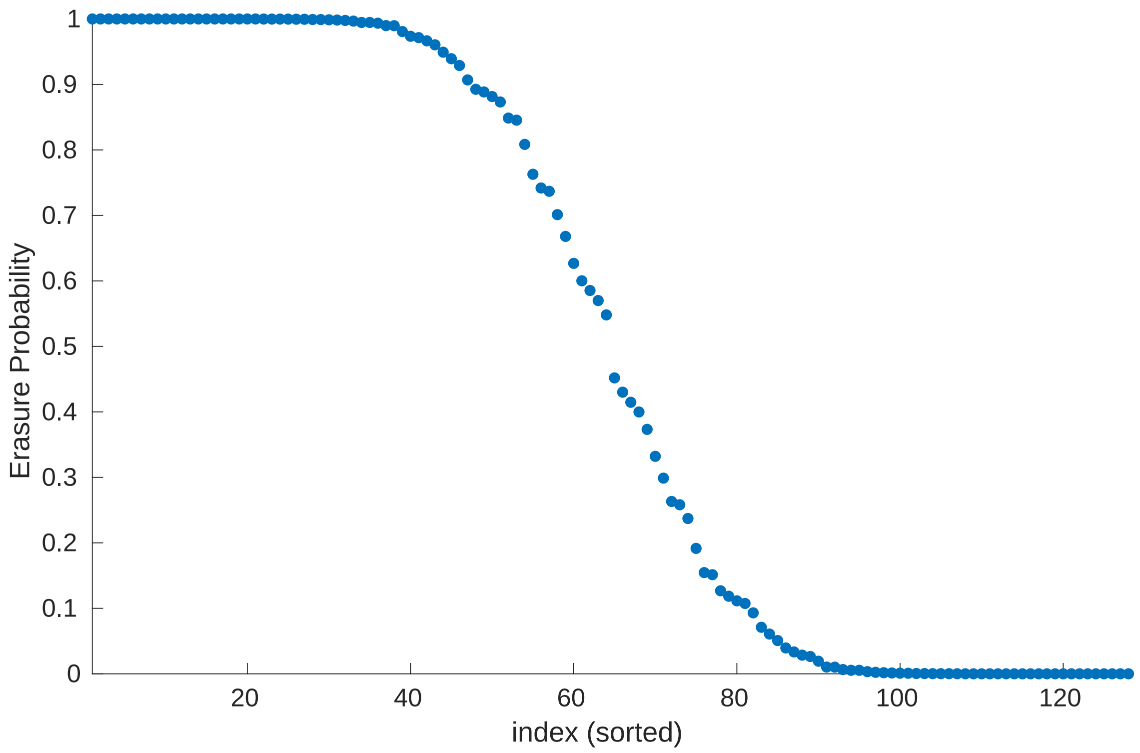
Channel Polarization - Numerical ($N = 2^{15}$, BEC(0.5))



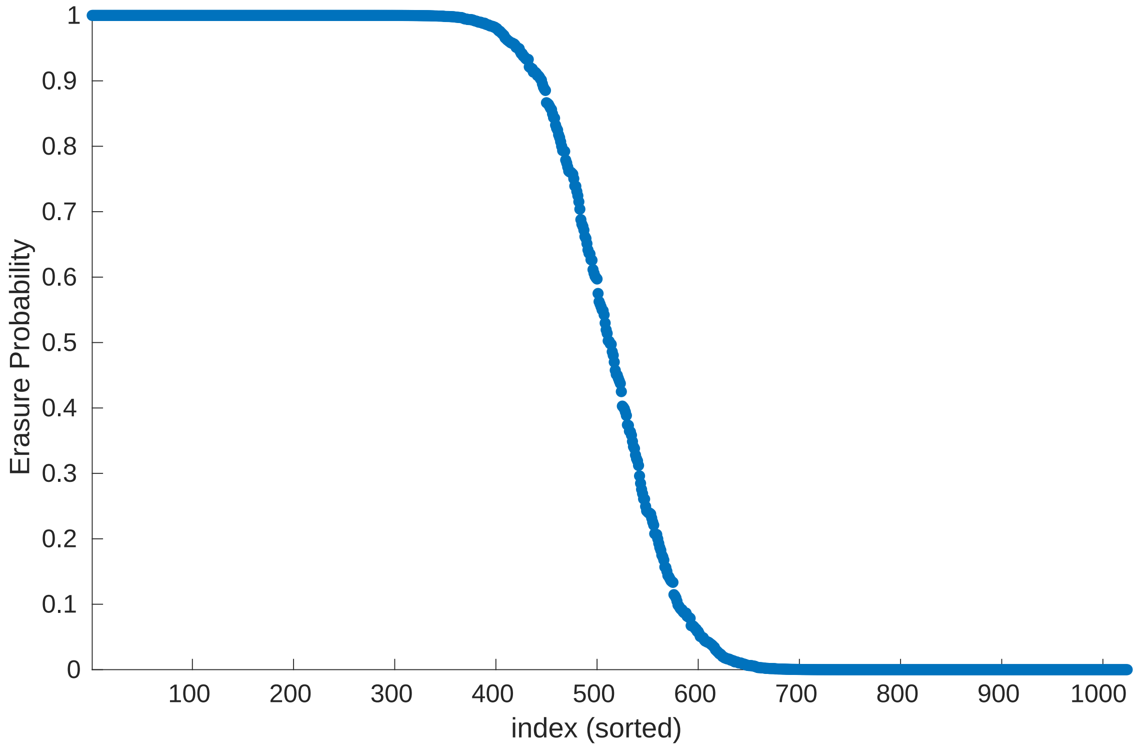
Channel Polarization - Numerical (Sorted, $N = 2^3$, BEC(0.5))



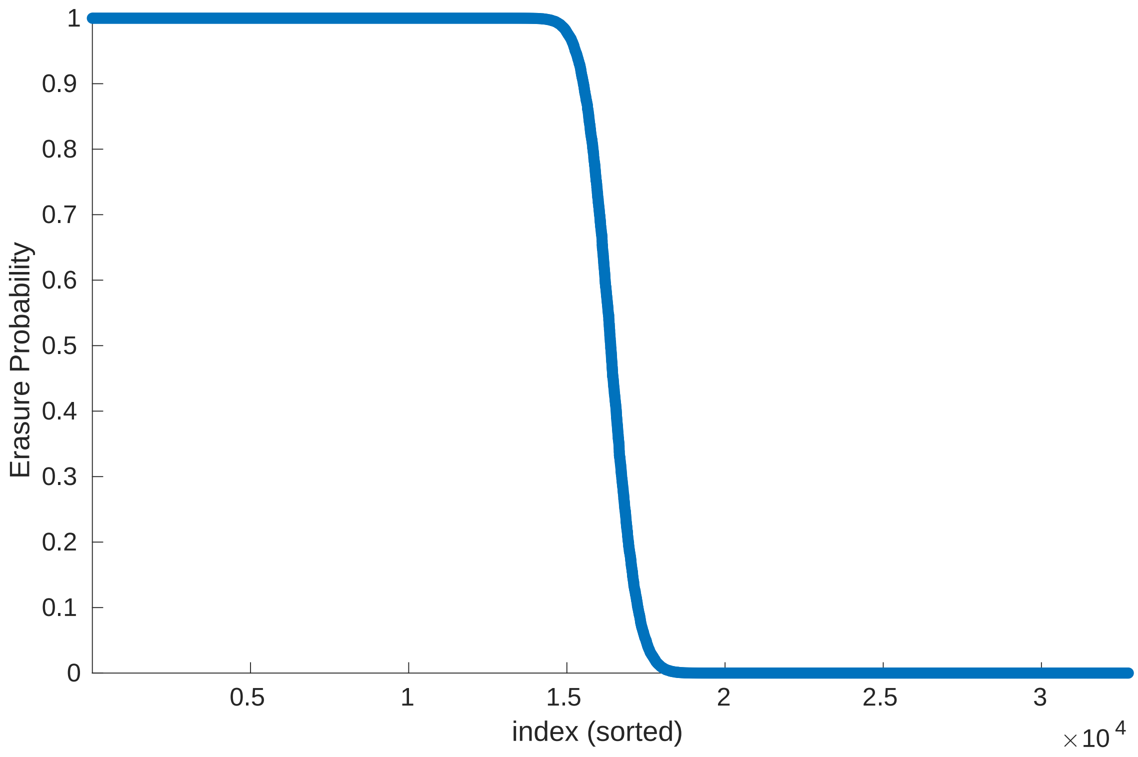
Channel Polarization - Numerical (Sorted, $N = 2^7$, BEC(0.5))



Channel Polarization - Numerical (Sorted, $N = 2^{10}$, BEC(0.5))

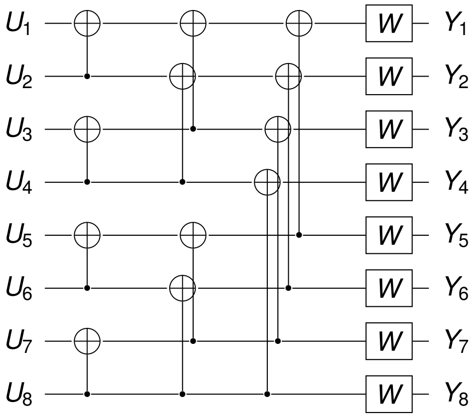


Channel Polarization - Numerical (Sorted, $N = 2^{15}$, BEC(0.5))



Code Design

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



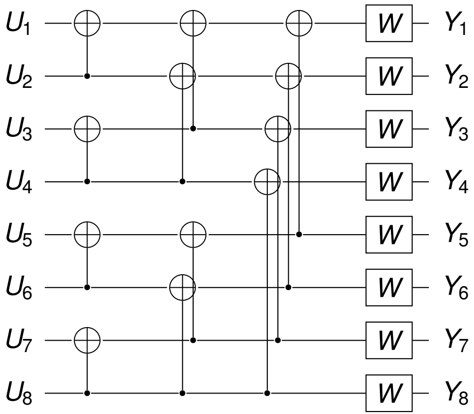
.

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.

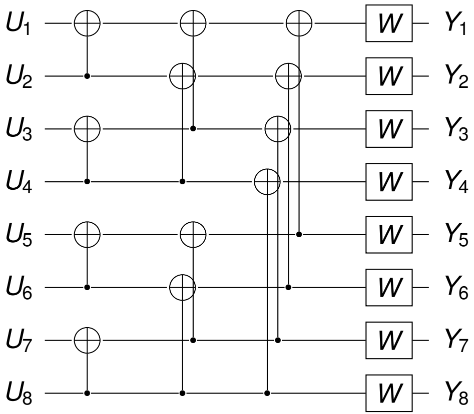
.



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 $\mathbf{G}_2^{\otimes n}$.



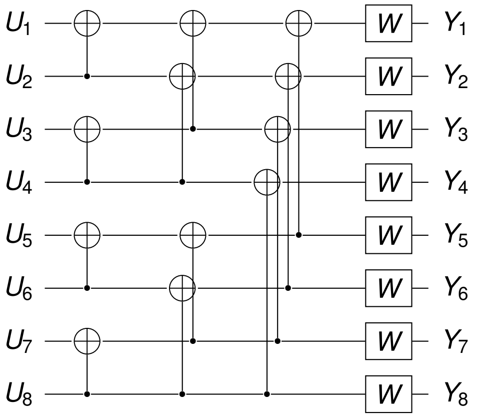
.

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 $\mathbf{G}_2^{\otimes n}$. Note that there is not an RM code for every k.

.

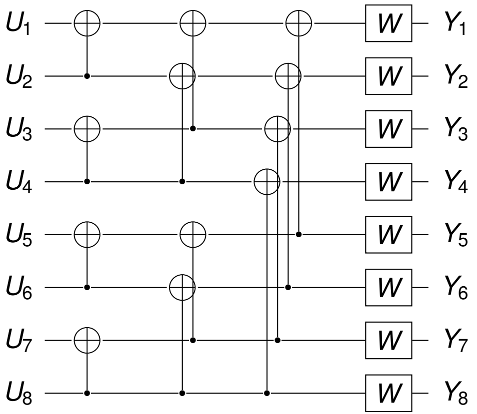


Code Design

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 $\mathbf{G}_2^{\otimes n}$. 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

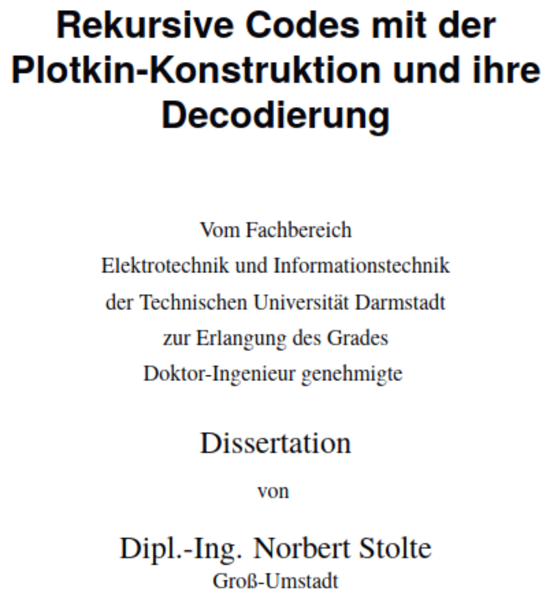
Dissertation

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.

Generator Matrix

After defining a set $\mathcal{A}$, the generator matrix of the code is obtained by removing the rows in $\mathcal{F} \triangleq \{1, \ldots, N\} \setminus \mathcal{A}$ (frozen set) from $\mathbf{G}_2^{\otimes n}$:

Generator Matrix

After defining a set $\mathcal{A}$, the generator matrix of the code is obtained by removing the rows in $\mathcal{F} \triangleq \{1, \ldots, N\} \setminus \mathcal{A}$ (frozen set) from $\mathbf{G}_2^{\otimes n}$:

- $(8, 4)$ polar code: $\mathcal{A} \triangleq \{4, 6, 7, 8\}$ and $\mathcal{F} \triangleq \{1, 2, 3, 5\}$

$$\mathbf{G}_2^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Generator Matrix

After defining a set $\mathcal{A}$, the generator matrix of the code is obtained by removing the rows in $\mathcal{F} \triangleq \{1, \ldots, N\} \setminus \mathcal{A}$ (frozen set) from $\mathbf{G}_2^{\otimes n}$:

- $(8, 4)$ polar code: $\mathcal{A} \triangleq \{4, 6, 7, 8\}$ and $\mathcal{F} \triangleq \{1, 2, 3, 5\}$

$$\mathbf{G}_2^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Generator Matrix

After defining a set $\mathcal{A}$, the generator matrix of the code is obtained by removing the rows in $\mathcal{F} \triangleq \{1, \ldots, N\} \setminus \mathcal{A}$ (frozen set) from $\mathbf{G}_2^{\otimes n}$:

- $(8, 4)$ polar code: $\mathcal{A} \triangleq \{4, 6, 7, 8\}$ and $\mathcal{F} \triangleq \{1, 2, 3, 5\}$

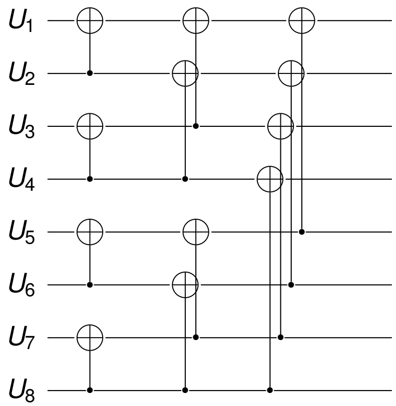
$$\mathbf{G}_2^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$X_1^8 = V_1^4 \mathbf{G}$ for random information bits V_1^4

Encoding

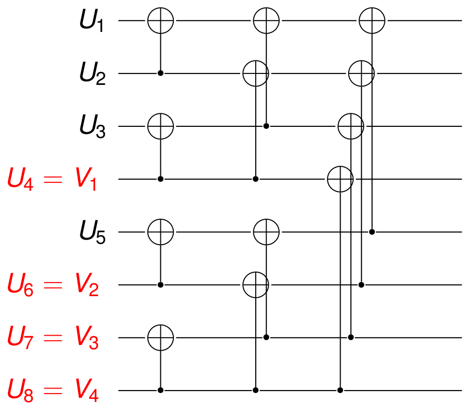
Let V_1^k denote the random information bits to be encoded:



Encoding

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

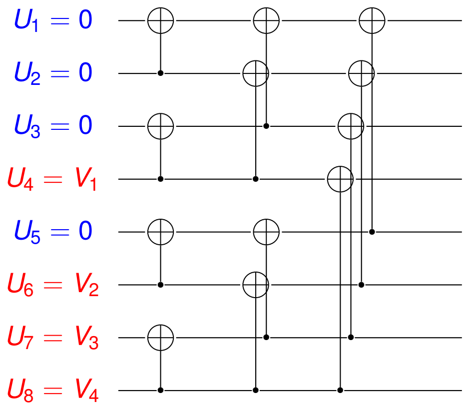
1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.



Encoding

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

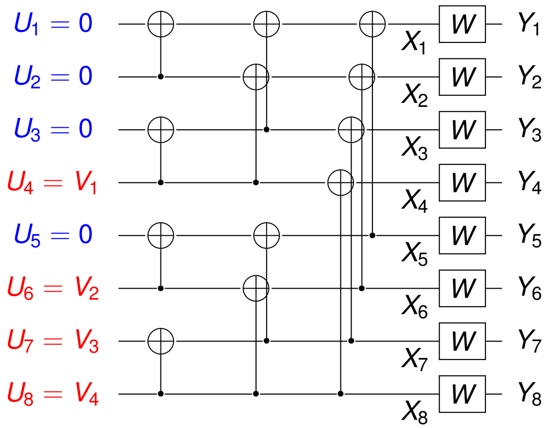
1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.
2. Set the remaining elements of U_1^N to 0 (frozen bits), i.e., $U_{\mathcal{F}} = 0^{n-k}$.



Encoding

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

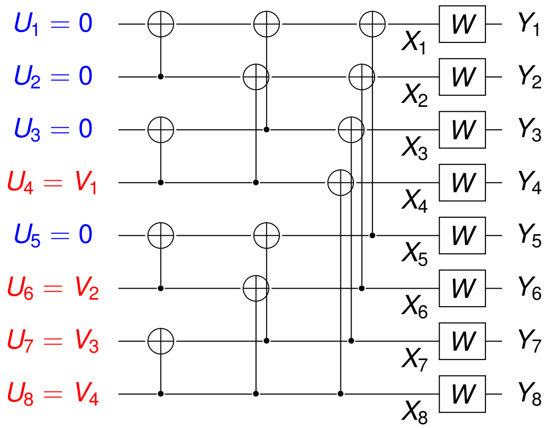
1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.
2. Set the remaining elements of U_1^N to 0 (frozen bits), i.e., $U_{\mathcal{F}} = 0^{n-k}$.
3. Apply polar transform of length$-N$, i.e., $X_1^N = U_1^N \mathbf{G}_2^{\otimes n}$.



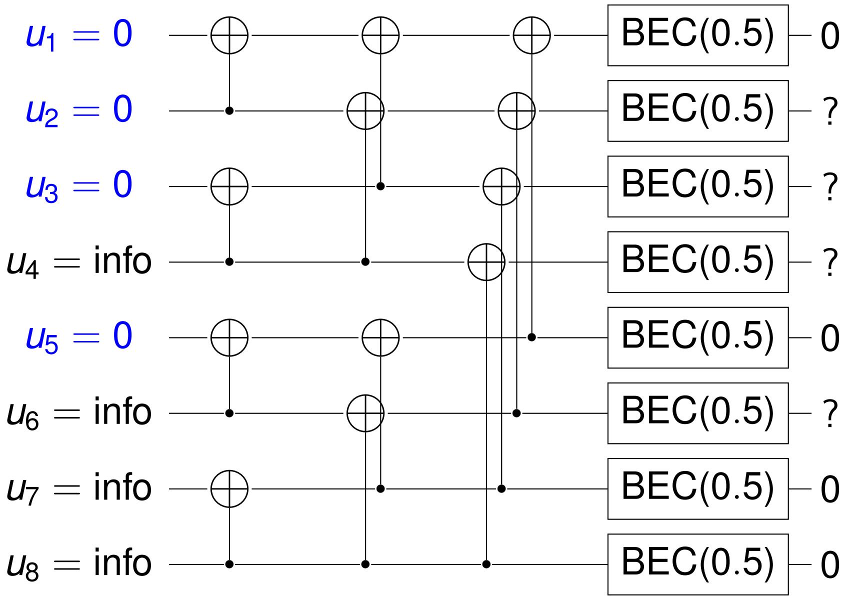
Encoding

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

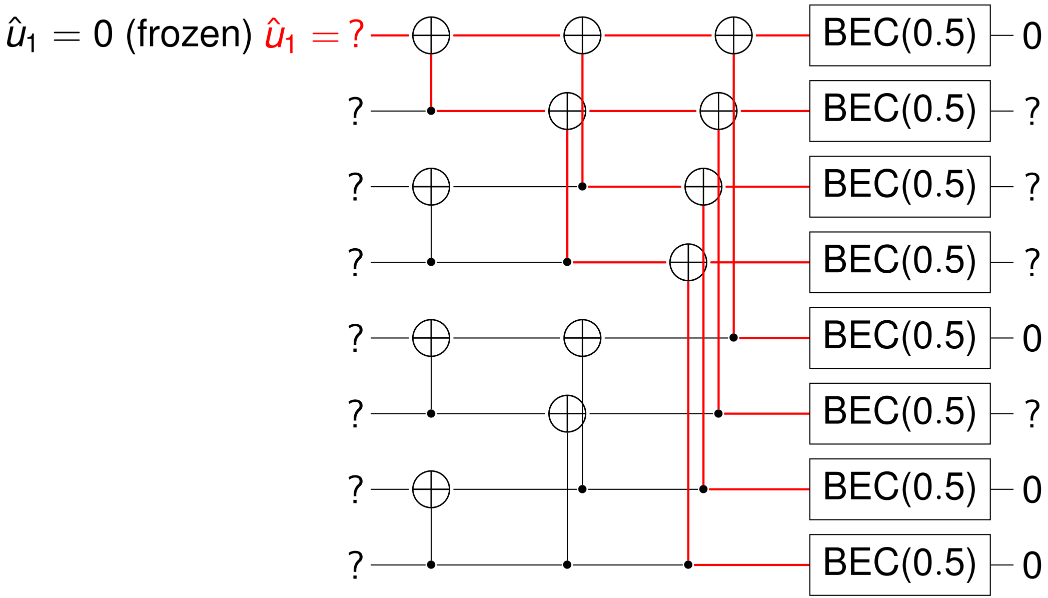
1. For a given set $\mathcal{A}$, map V_1^k onto $U_{\mathcal{A}}$.
2. Set the remaining elements of U_1^N to 0 (frozen bits), i.e., $U_{\mathcal{F}} = 0^{n-k}$.
3. Apply polar transform of length$-N$, i.e., $X_1^N = U_1^N \mathbf{G}_2^{\otimes n}$.
4. This can be done with a complexity of $\mathcal{O}(N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



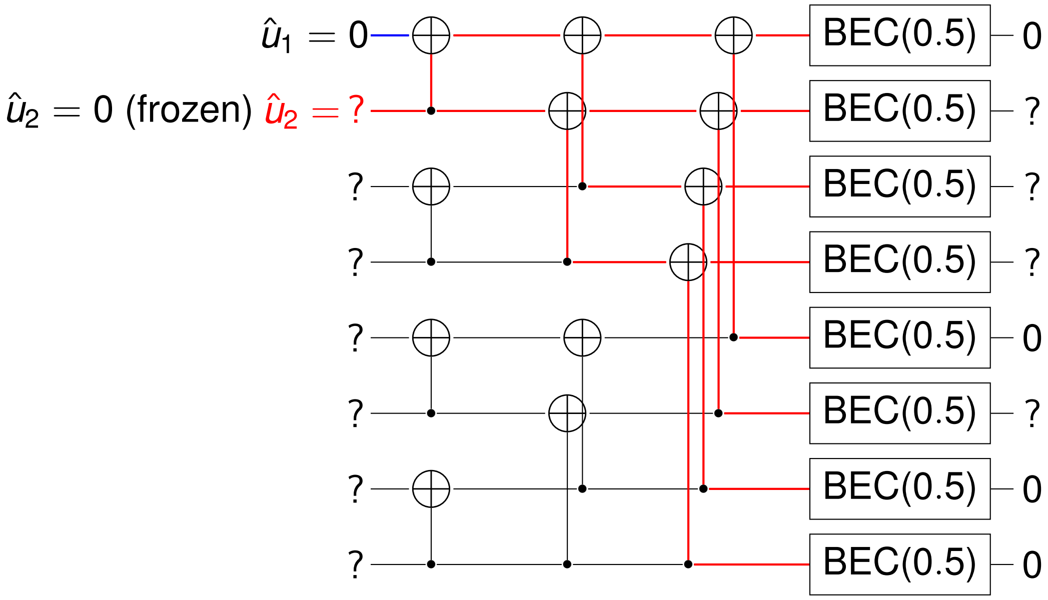
SC Decoding: BEC Example



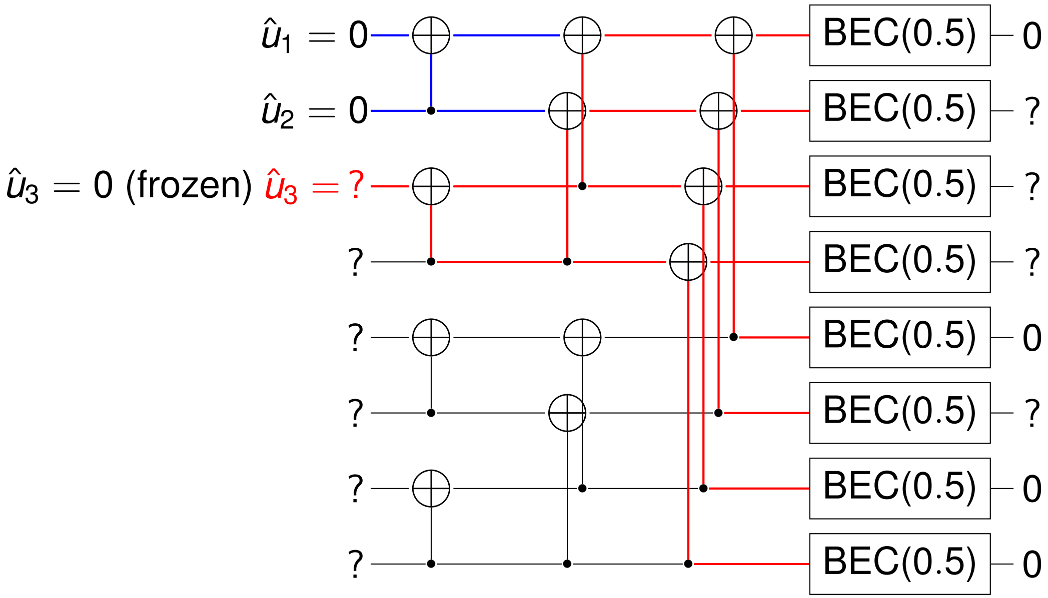
SC Decoding: BEC Example



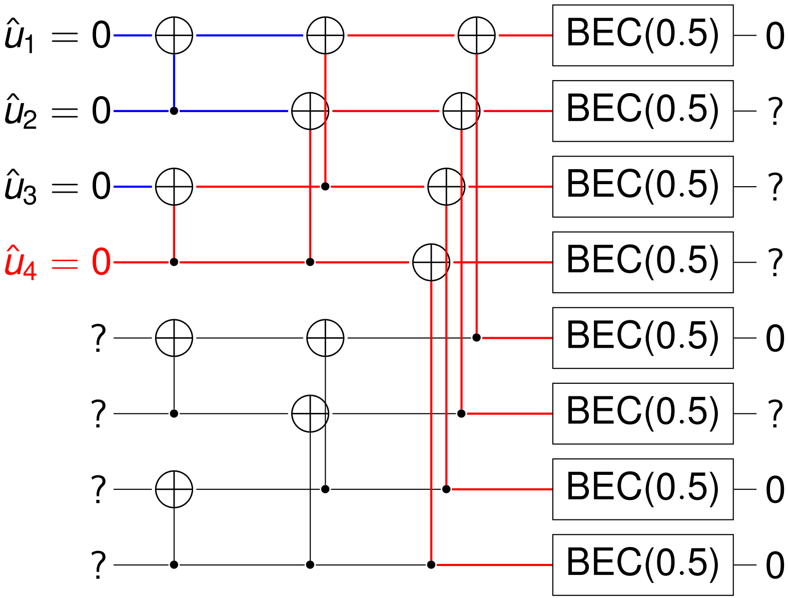
SC Decoding: BEC Example



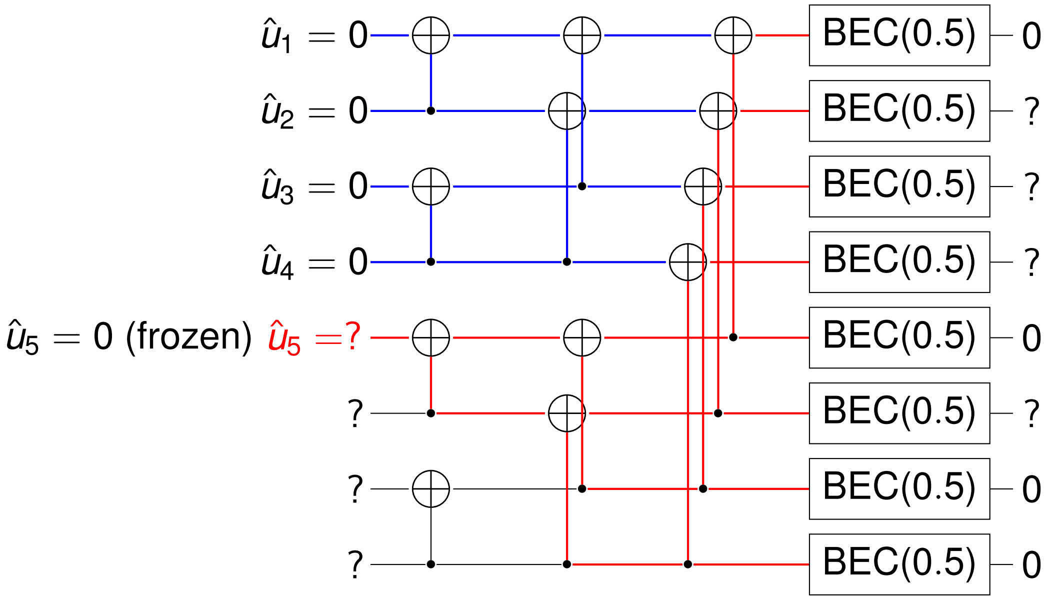
SC Decoding: BEC Example



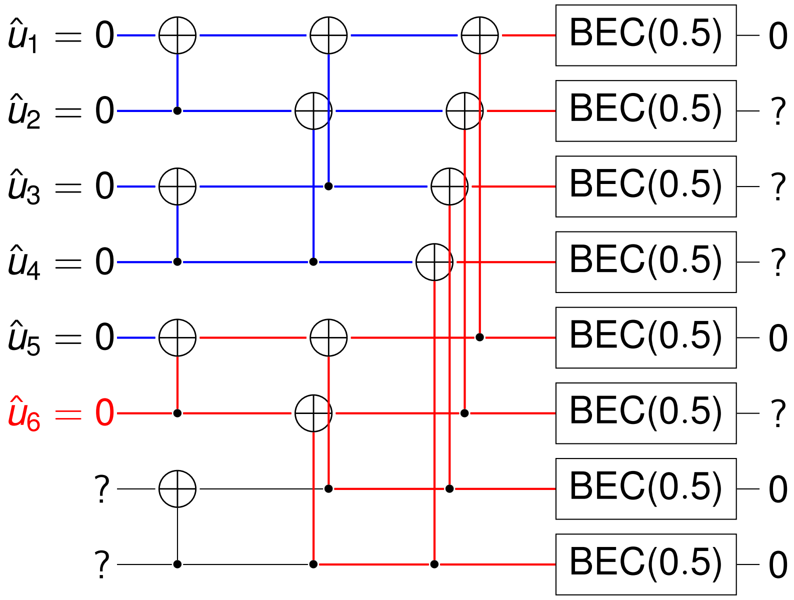
SC Decoding: BEC Example



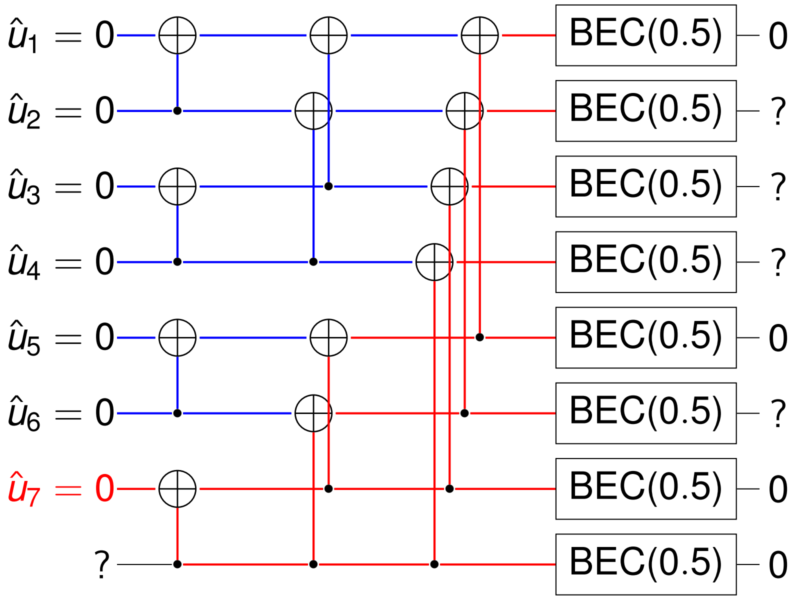
SC Decoding: BEC Example



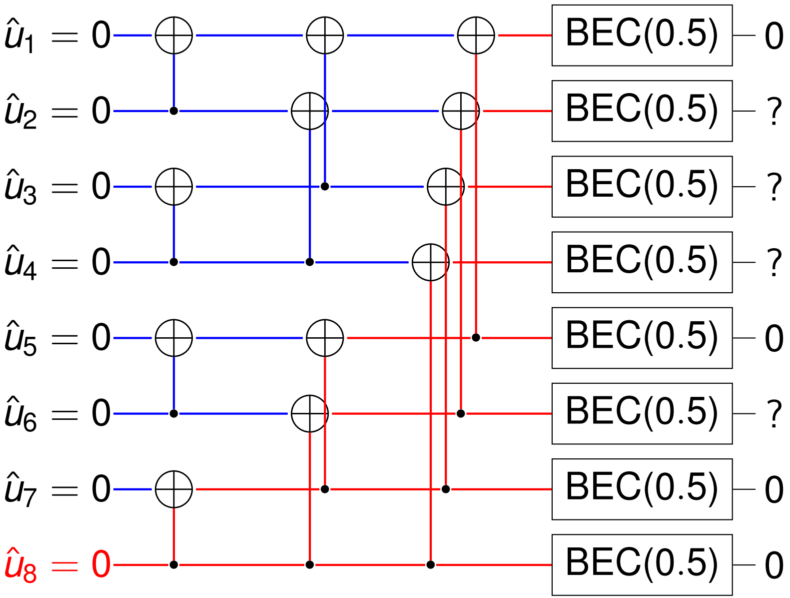
SC Decoding: BEC Example



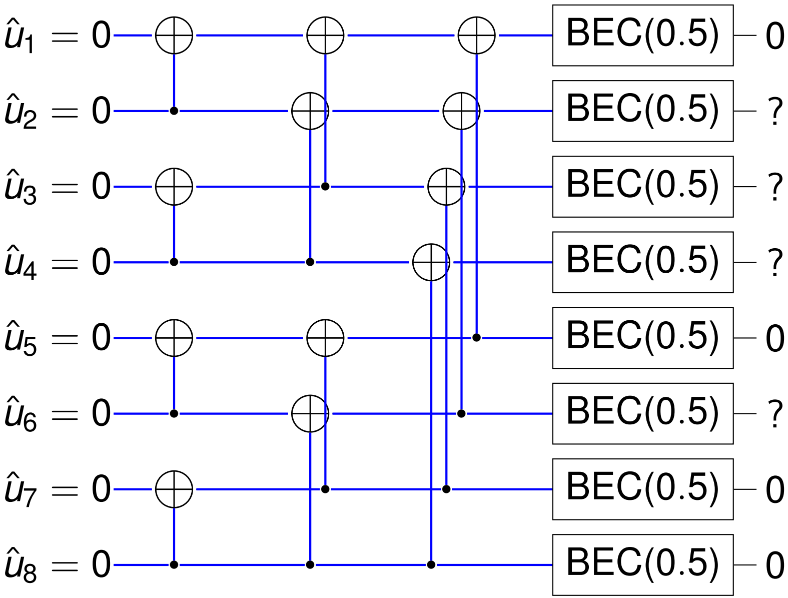
SC Decoding: BEC Example



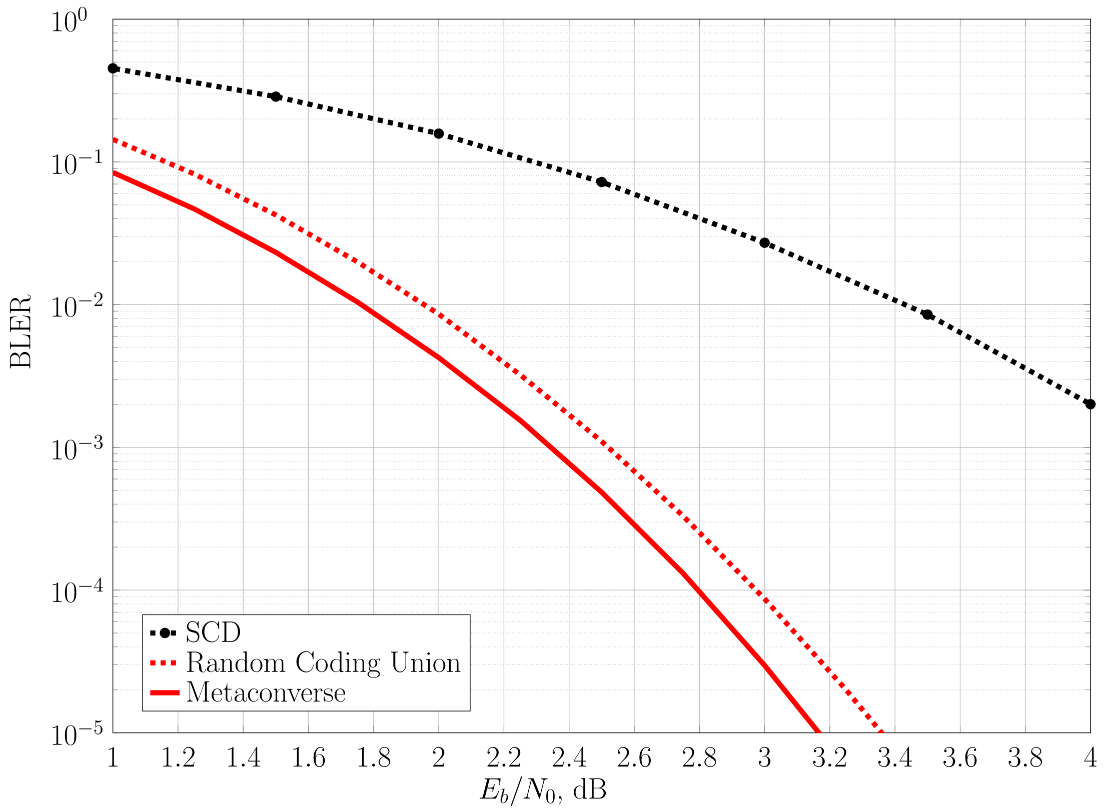
SC Decoding: BEC Example



SC Decoding: BEC Example



$n = 128, k = 64$



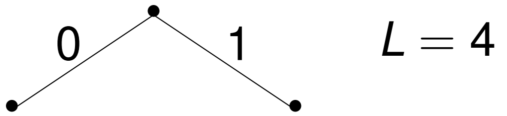
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 3.2 3.4 3.6 3.8 4
E_b/N_0, dB
SCD
Random Coding Union
Metaconverse

Successive Cancellation List Decoding

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

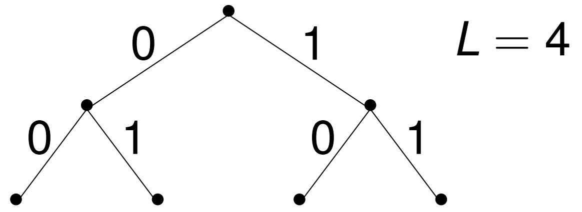
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



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

Successive Cancellation List Decoding

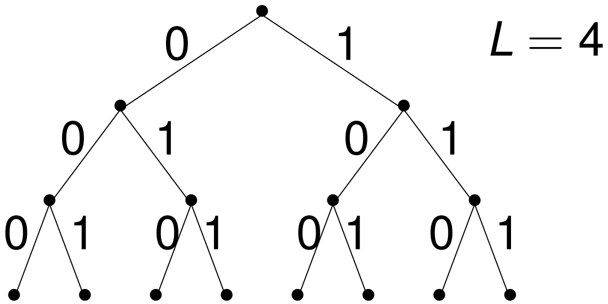
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



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

Successive Cancellation List Decoding

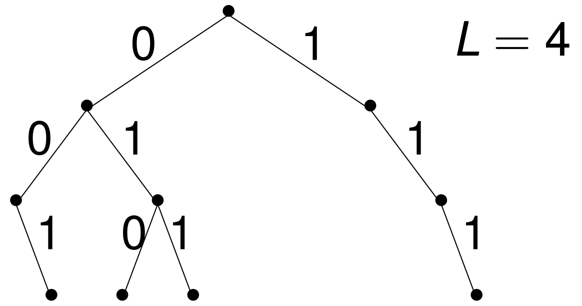
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



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

Successive Cancellation List Decoding

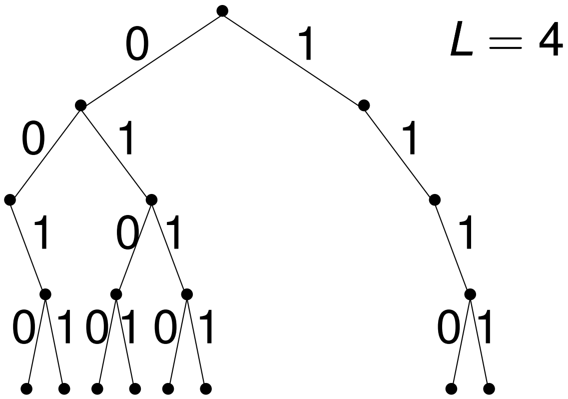
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



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

Successive Cancellation List Decoding

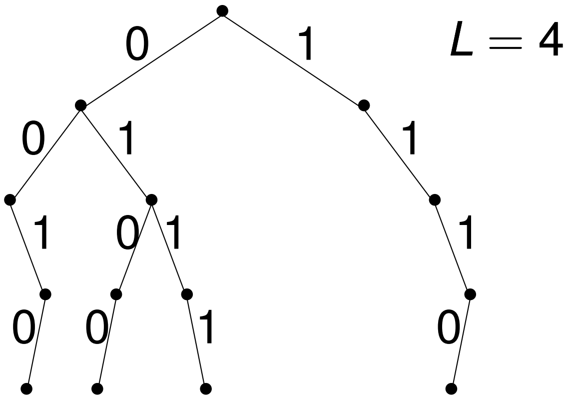
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



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

Successive Cancellation List Decoding

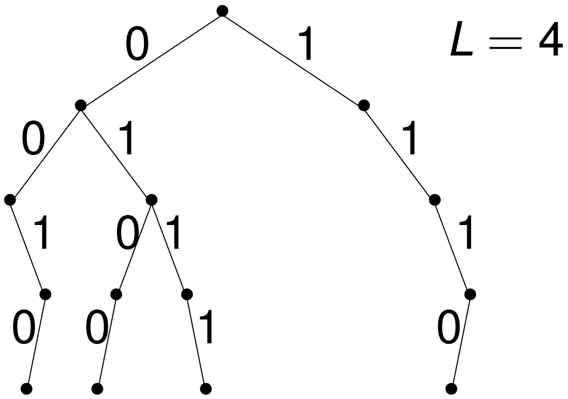
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



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

Successive Cancellation List Decoding

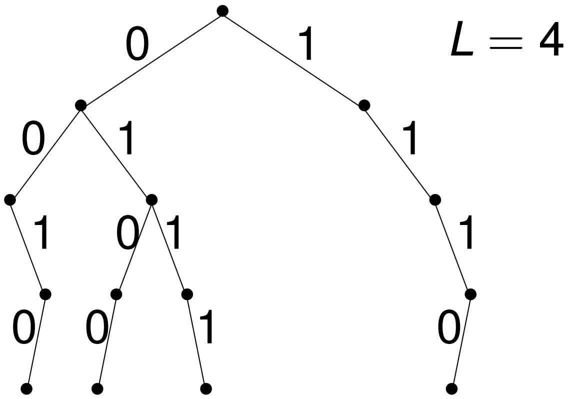
Key idea: Each time a decision is needed on $\hat{u}_i$, both options, i.e., $\hat{u}_i = 0$ and $\hat{u}_i = 1$, are stored. This doubles the number of partial input sequences (paths) at each decoding stage.



- When the number of paths exceeds a predefined list size L, discard the least likely paths.
- After N-th stage, estimate $\hat{u}_1^N$ chosen as $\hat{u}_1^N = \arg\max_{u_1^N \in \mathcal{L}} \mathbb{P}(u_1^N | y_1^N)$.

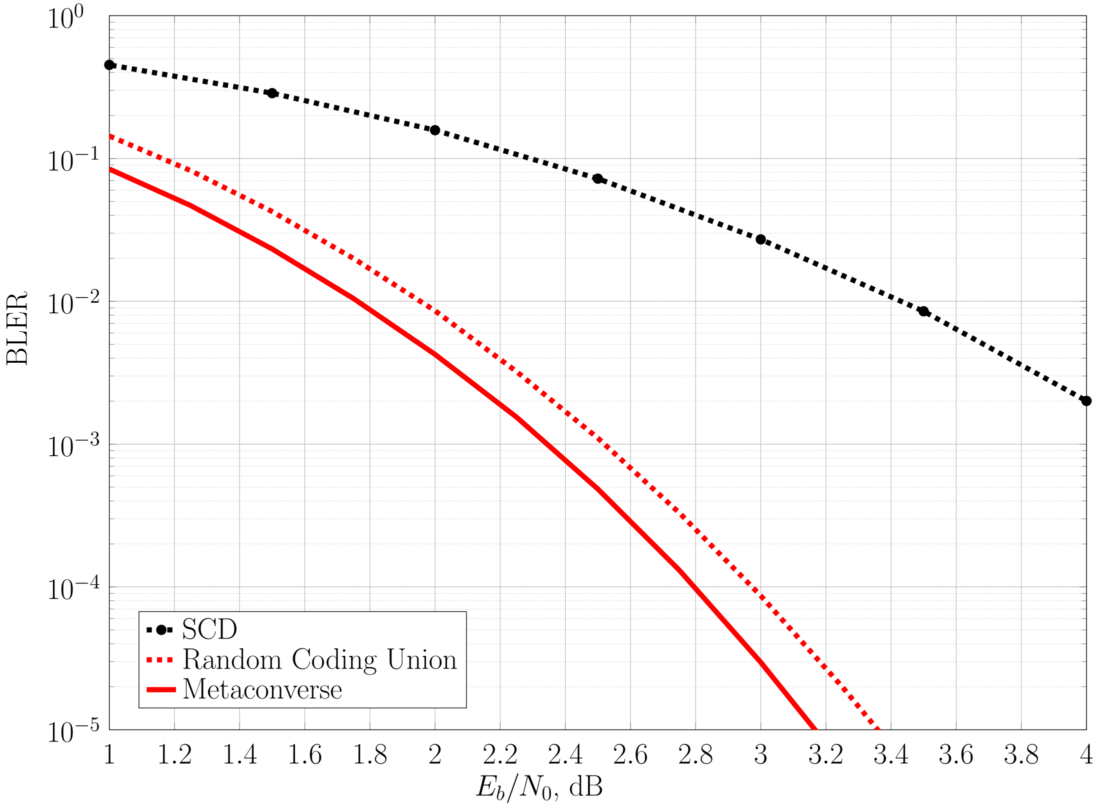
Successive Cancellation List Decoding

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.



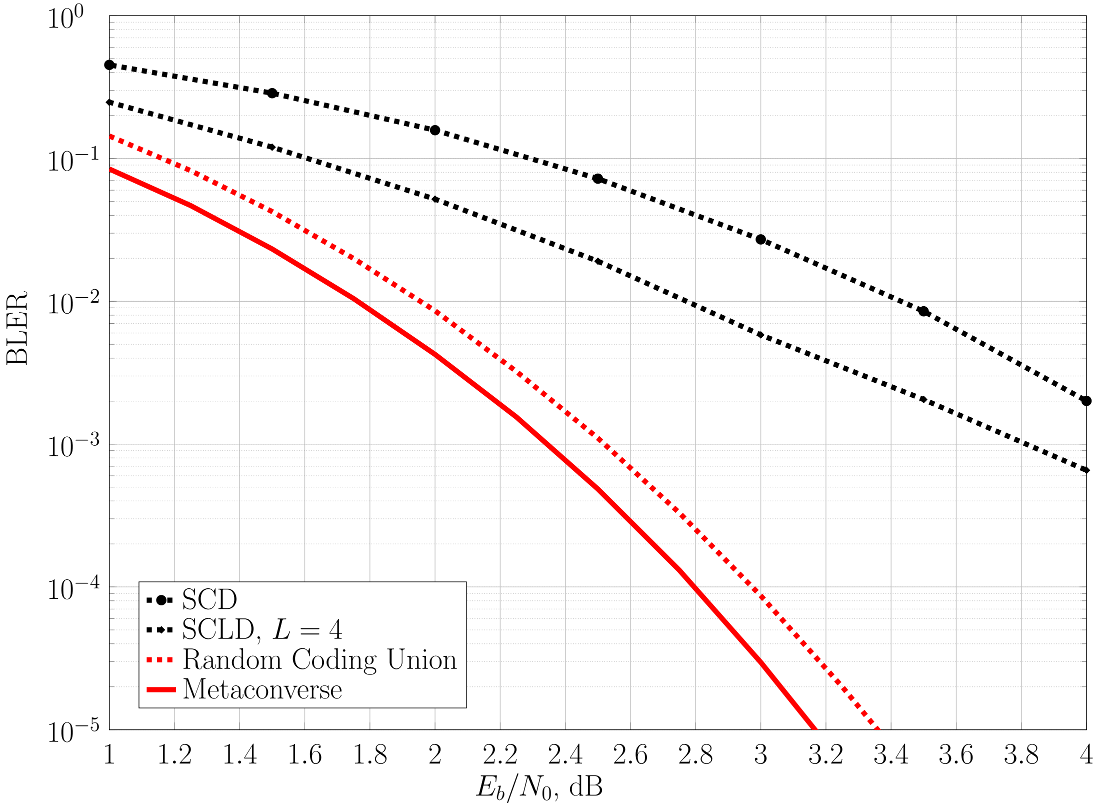
- When the number of paths exceeds a predefined list size L, discard the least likely paths.
- After N-th stage, estimate $\hat{u}_1^N$ chosen as $\hat{u}_1^N = \arg\max_{u_1^N \in \mathcal{L}} \mathbb{P}(u_1^N | y_1^N)$.
- The decoder has been applied to RM codes previously (see, e.g., [Sto02, DS06]).

$N = 128, k = 64$



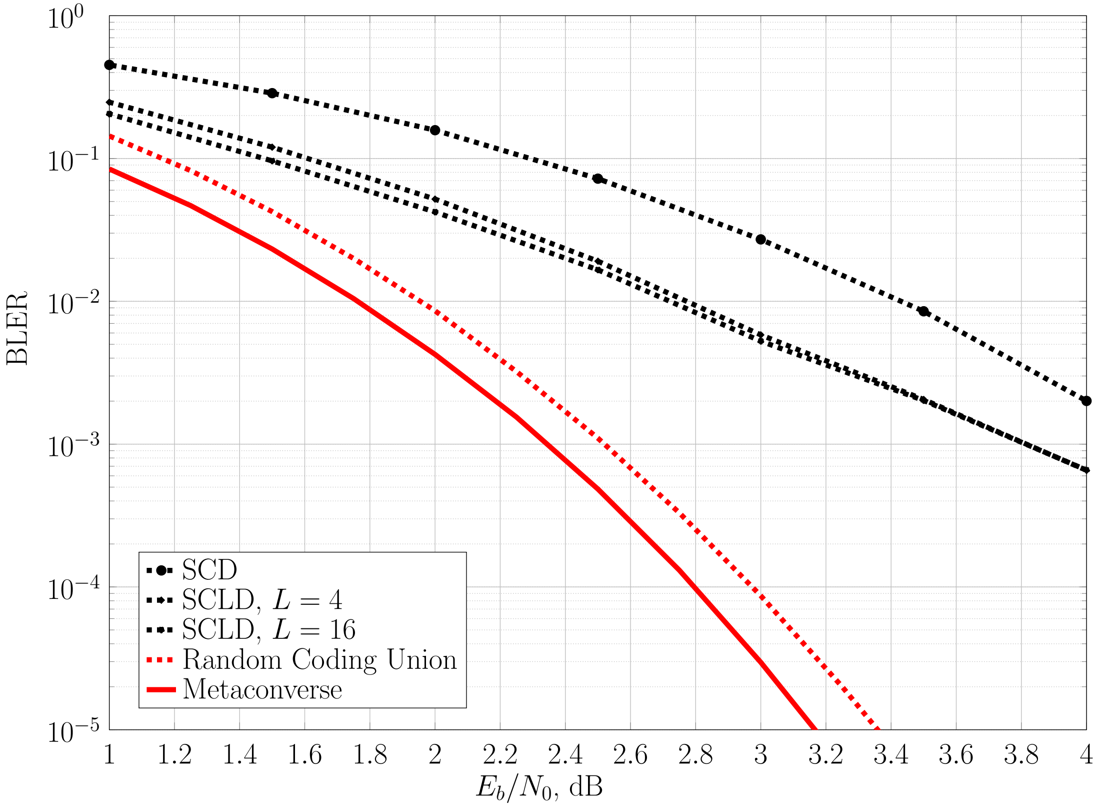
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
E_b/N_0, dB
SCD
Random Coding Union
Metaconverse

$N = 128, k = 64$



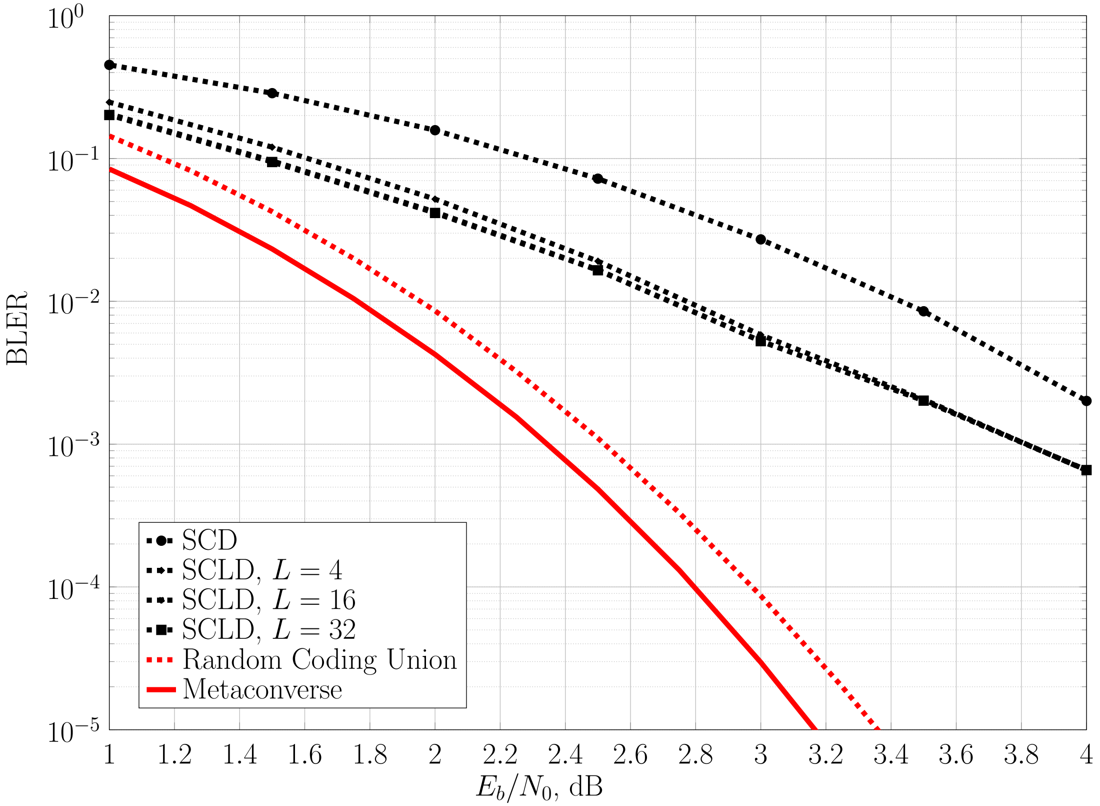
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
E_b/N_0, dB
SCD
SCLD, $L = 4$
Random Coding Union
Metaconverse

$N = 128, k = 64$



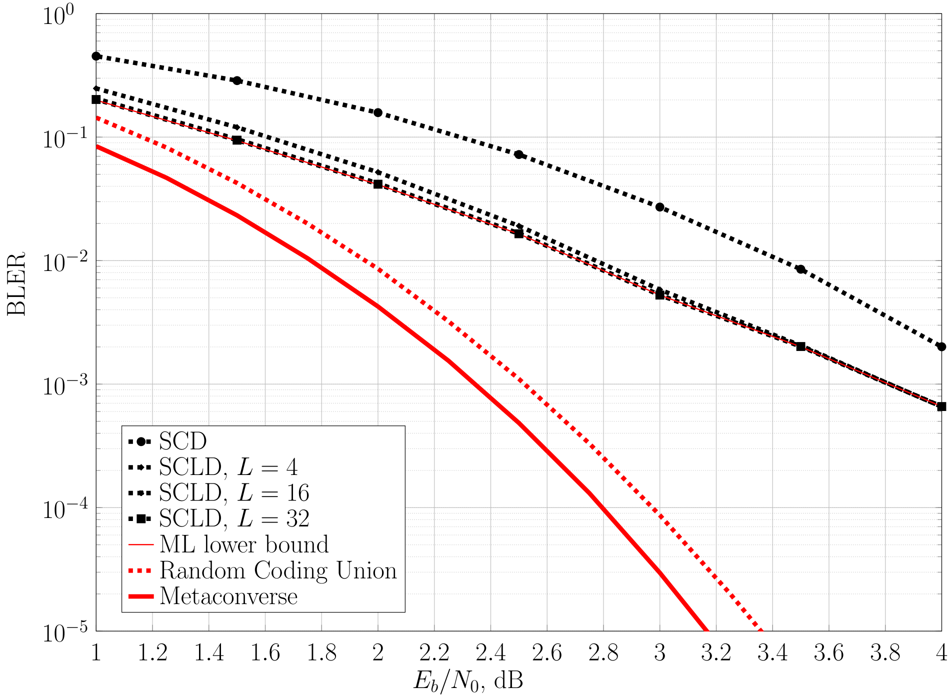
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
E_b/N_0, dB
SCD
SCLD, $L = 4$
SCLD, $L = 16$
Random Coding Union
Metaconverse

$N = 128, k = 64$



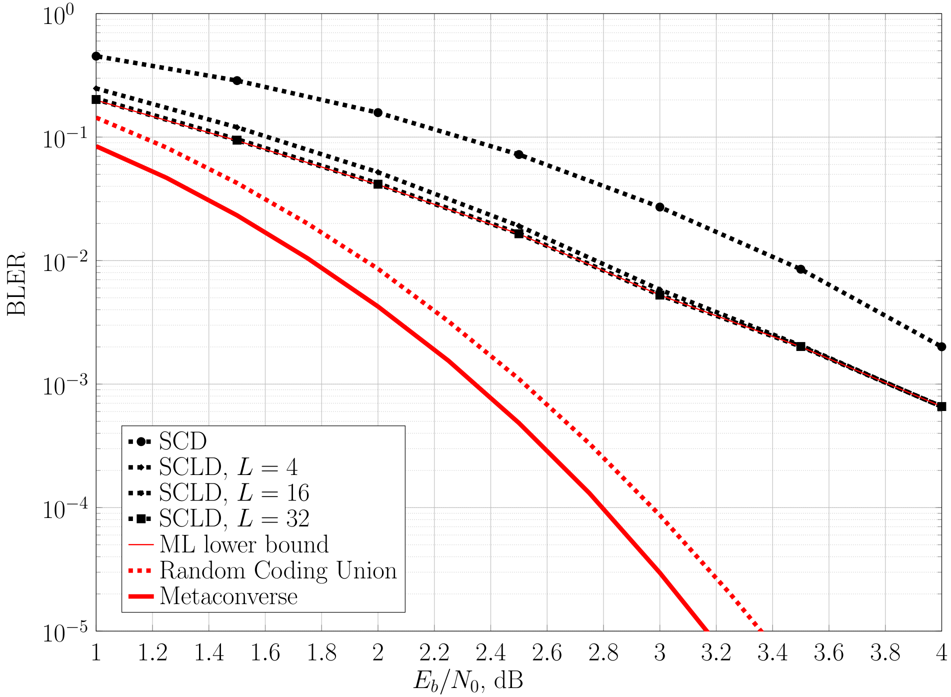
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
E_b/N_0, dB
SCD
SCLD, $L = 4$
SCLD, $L = 16$
SCLD, $L = 32$
Random Coding Union
Metaconverse

$N = 128, k = 64$



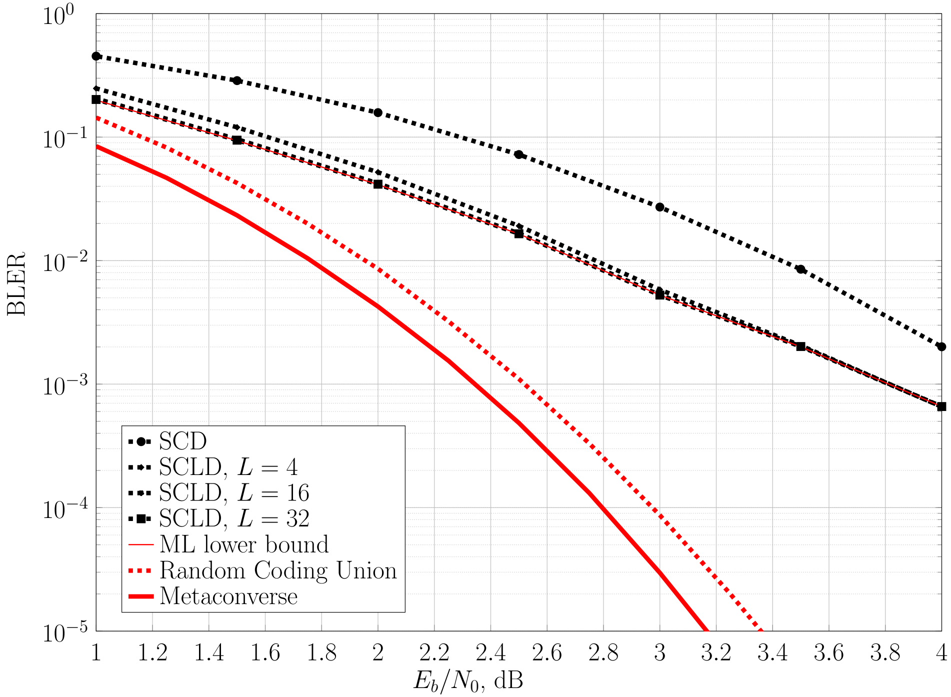
- Gets close to ML for relatively small L ☺

$N = 128, k = 64$



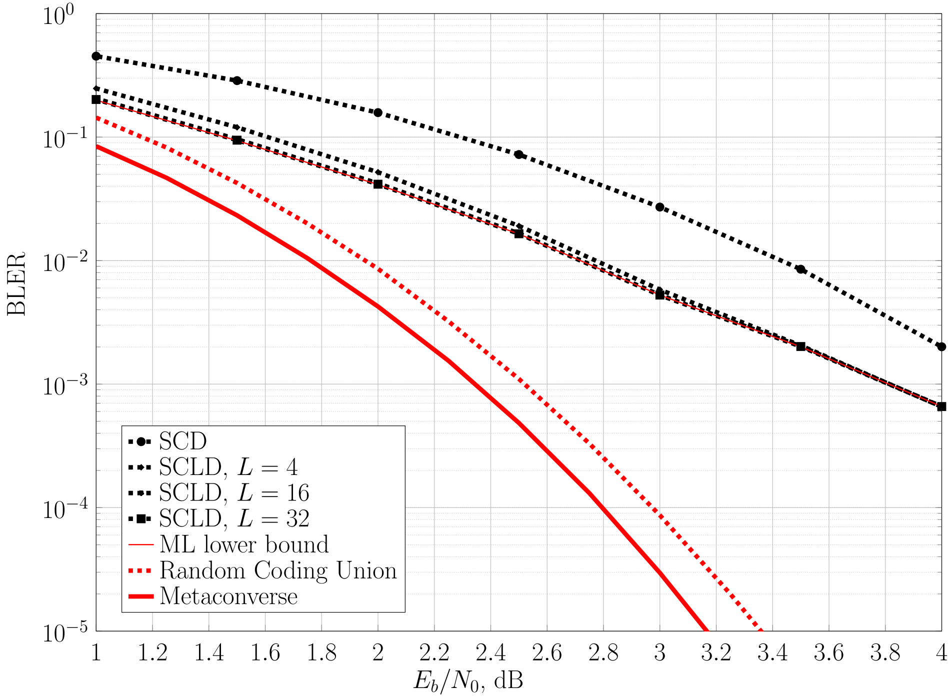
- Gets close to ML for relatively small L ☺
- Not competitive for short blocks ☹

$N = 128, k = 64$



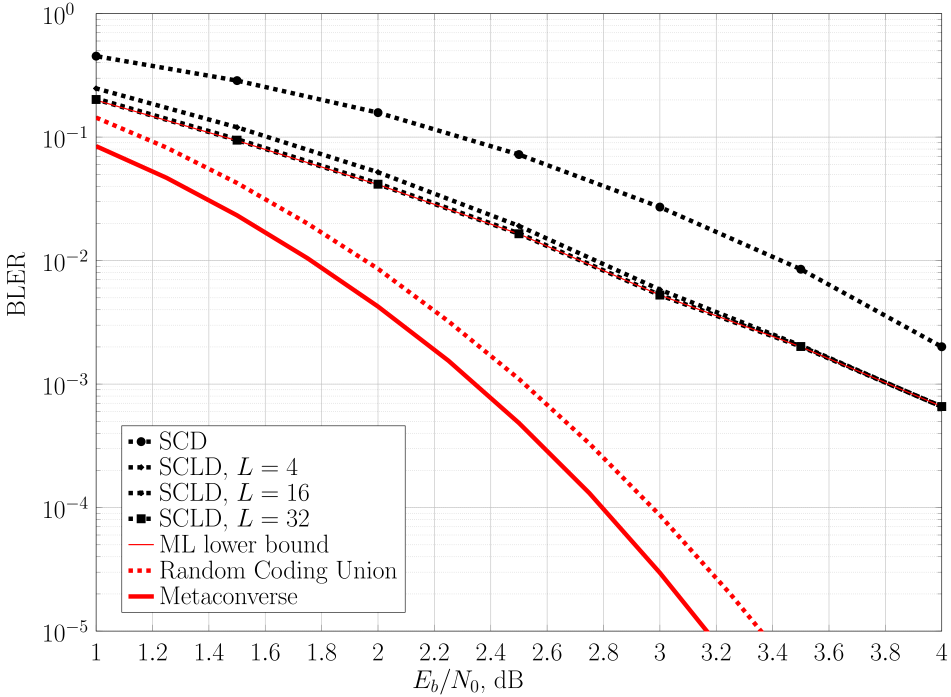
- Gets close to ML for relatively small L ☺
- Not competitive for short blocks ☹
- When error happens, the transmitted codeword, very often, is still a member of the final list.

$N = 128, k = 64$



- Gets close to ML for relatively small L ☺
- Not competitive for short blocks ☹
- When error happens, the transmitted codeword, very often, is still a member of the final list.
- We need a to find a way to pick the correct word.

$N = 128, k = 64$



- Gets close to ML for relatively small L ☺
- Not competitive for short blocks ☹
- When error happens, the transmitted codeword, very often, is still a member of the final list.
- We need a to find a way to pick the correct word.
- Easy to fix by concatenating an outer CRC code. ☺

Polar Codes with Outer Code

Concatenate an $(N, k+\ell)$ inner polar code, with an outer CRC-ℓ code to improve distance spectrum, where at the transmitter:

Polar Codes with Outer Code

Concatenate an $(N, k + \ell)$ inner polar code, with an outer CRC-ℓ code to improve distance spectrum, where at the transmitter:

- The first k positions in $\mathcal{A}$ of the inner polar code is used to encode the information bits.

Polar Codes with Outer Code

Concatenate an $(N, k + \ell)$ inner polar code, with an outer CRC-ℓ code to improve distance spectrum, where at the transmitter:

- The first k positions in $\mathcal{A}$ of the inner polar code is used to encode the information bits.
- Other ℓ positions in $\mathcal{A}$ to encode the CRC bits generated by k information bits.

Polar Codes with Outer Code

Concatenate an $(N, k+\ell)$ inner polar code, with an outer CRC-ℓ code to improve distance spectrum, where at the transmitter:

- The first k positions in $\mathcal{A}$ of the inner polar code is used to encode the information bits.
- Other ℓ positions in $\mathcal{A}$ to encode the CRC bits generated by k information bits.
- Any systematic $(k+\ell, k)$ code would work!

Polar Codes with Outer Code

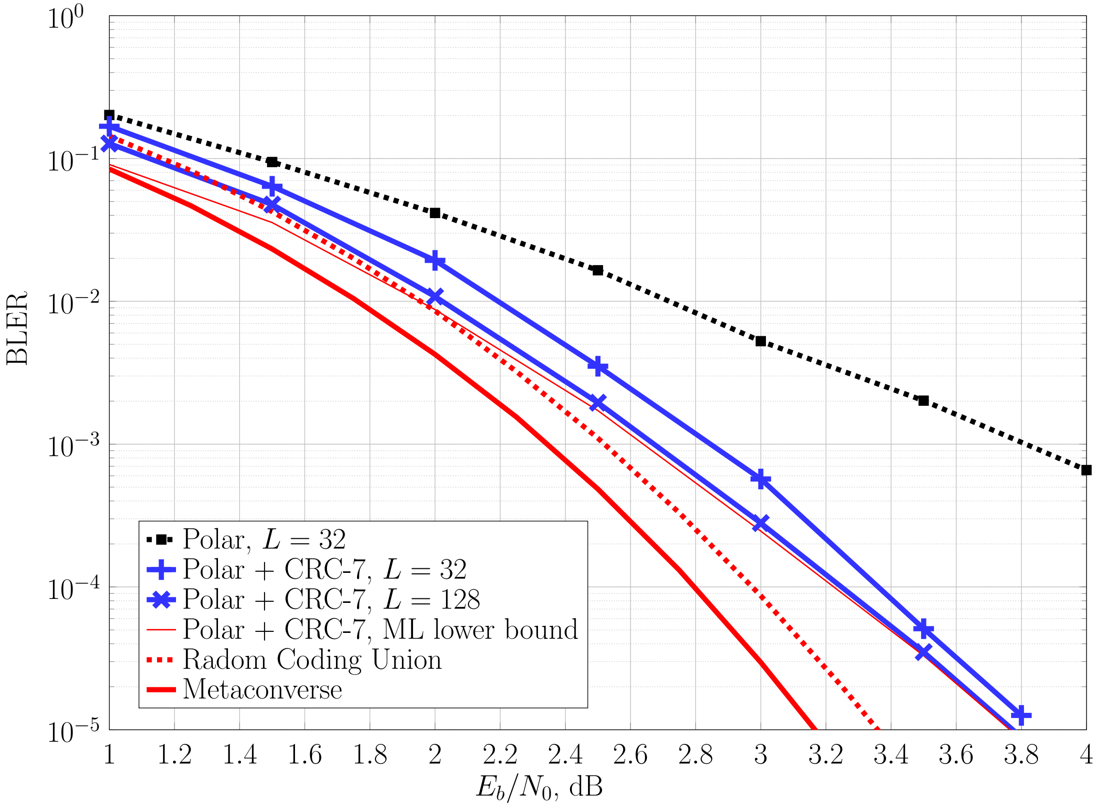
Concatenate an $(N, k+\ell)$ inner polar code, with an outer CRC-ℓ code to improve distance spectrum, where at the transmitter:

- The first k positions in $\mathcal{A}$ of the inner polar code is used to encode the information bits.
- Other ℓ positions in $\mathcal{A}$ to encode the CRC bits generated by k information bits.
- Any systematic $(k+\ell, k)$ code would work!

At the receiver:

- SCL decoding (inner code), followed by syndrome check with outer code: pick the most probably codeword on the list fulfilling the CRC.

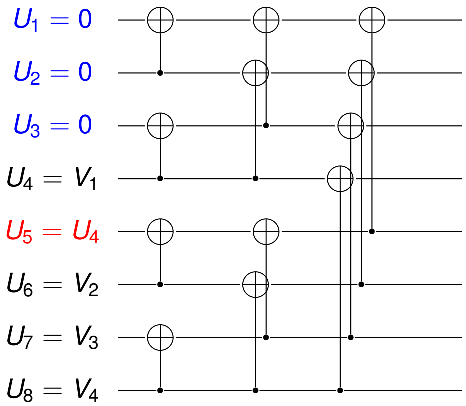
$N = 128, k = 64$



Polar, $L = 32$
Polar + CRC-7, $L = 32$
Polar + CRC-7, $L = 128$
Polar + CRC-7, ML lower bound
Radom Coding Union
Metaconverse
BLER
E_b/N_0, dB
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

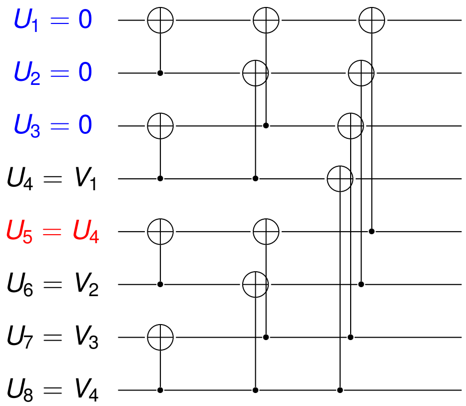
Dynamic Frozen Bits

- 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]



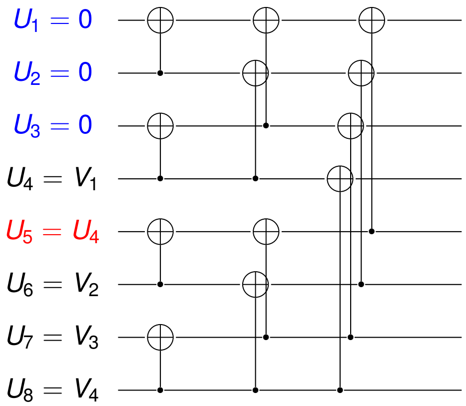
Dynamic Frozen Bits

- 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.



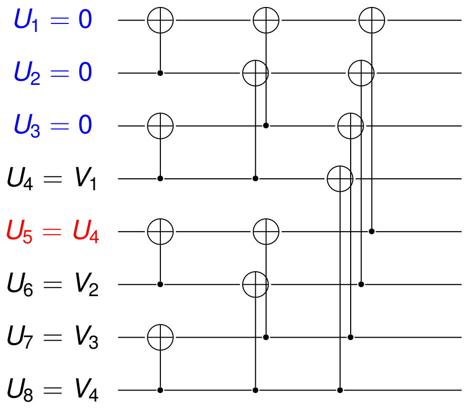
Dynamic Frozen Bits

- 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!



Outline

1 Overview of Polar Codes

2 Recent Advances in Polar Codes

- Binary Erasure Channel

3 Conclusions

Motivating Question

- What list size is sufficient to approach ML decoding performance for a given polar code and channel?

Motivating Question

- What list size is sufficient to approach ML decoding performance for a given polar code and channel?
  - To avoid losing true codeword, its rank must not be larger than list size.

Motivating Question

- What list size is sufficient to approach ML decoding performance for a given polar code and channel?
  - To avoid losing true codeword, its rank must not be larger than list size.
  - The expected log-rank of correct codeword is upper bounded by an entropy.

Motivating Question

- What list size is sufficient to approach ML decoding performance for a given polar code and channel?
  - To avoid losing true codeword, its rank must not be larger than list size.
  - The expected log-rank of correct codeword is upper bounded by an entropy.
- For the BEC,
  - This entropy equals the dimension of an affine subspace.

Motivating Question

- What list size is sufficient to approach ML decoding performance for a given polar code and channel?
  - To avoid losing true codeword, its rank must not be larger than list size.
  - The expected log-rank of correct codeword is upper bounded by an entropy.
- For the BEC,
  - This entropy equals the dimension of an affine subspace.

Based on joint works with Henry D. Pfister [CP20, CP21]

An Information-Theoretic Perspective (1)

Basic Idea: After m steps, consider the conditional entropy $H\left(U_1^m \middle| Y_1^N\right)$.

An Information-Theoretic Perspective (1)

Basic Idea: After m steps, consider the conditional entropy $H\left(U_1^m | Y_1^N\right)$.

The chain rule of entropy implies:

$$\begin{aligned} H\left(U_1^m | Y_1^N\right) &= \sum_{i=1}^{m} H\left(U_i \middle| U_1^{i-1}, Y_1^N\right) \\ &= \sum_{i=1}^{m} H\left(W_N^{(i)}\right) \end{aligned}$$

An Information-Theoretic Perspective (1)

Basic Idea: After *m* steps, consider the conditional entropy $H\left(U_1^m \middle| Y_1^N\right)$.

The chain rule of entropy implies:

$$H\left(U_1^m \middle| Y_1^N\right) = \sum_{i=1}^{m} H\left(U_i \middle| U_1^{i-1}, Y_1^N\right)$$
$$= \sum_{i=1}^{m} H\left(W_N^{(i)}\right)$$

Note: this ignores frozen bits and will be modified soon!

An Information-Theoretic Perspective (2)

- For the first m input bits, the information/frozen sets are denoted as

$$\mathcal{A}^{(m)} \triangleq \mathcal{A} \cap [m] \quad \text{and} \quad \mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [m]$$

An Information-Theoretic Perspective (2)

- For the first m input bits, the information/frozen sets are denoted as

$$\mathcal{A}^{(m)} \triangleq \mathcal{A} \cap [m] \text{ and } \mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [m]$$

- **Key Idea:** information entropy given frozen bits and difference sequence

$$\bar{D}_m \triangleq H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N, U_{\mathcal{F}^{(m)}}\right) \quad \text{and} \quad \Delta_m \triangleq \bar{D}_m - \bar{D}_{m-1}$$

An Information-Theoretic Perspective (2)

- For the first m input bits, the information/frozen sets are denoted as

$$\mathcal{A}^{(m)} \triangleq \mathcal{A} \cap [m] \quad \text{and} \quad \mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [m]$$

- **Key Idea:** information entropy given frozen bits and difference sequence

$$\bar{D}_m \triangleq H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N, U_{\mathcal{F}^{(m)}}\right) \quad \text{and} \quad \Delta_m \triangleq \bar{D}_m - \bar{D}_{m-1}$$

- Experiment: assume U_1^N is uniform and Rx learns frozen bits causally.

An Information-Theoretic Perspective (2)

- For the first m input bits, the information/frozen sets are denoted as

$$\mathcal{A}^{(m)} \triangleq \mathcal{A} \cap [m] \text{ and } \mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [m]$$

- **Key Idea:** information entropy given frozen bits and difference sequence

$$\bar{D}_m \triangleq H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N, U_{\mathcal{F}^{(m)}}\right) \quad \text{and} \quad \Delta_m \triangleq \bar{D}_m - \bar{D}_{m-1}$$

- Experiment: assume U_1^N is uniform and Rx learns frozen bits causally.

- If U_m is an information bit, then

  $\Delta_m = H(U_m, Y_1^N | U_1^{m-1})$

An Information-Theoretic Perspective (2)

- For the first m input bits, the information/frozen sets are denoted as

$$\mathcal{A}^{(m)} \triangleq \mathcal{A} \cap [m] \text{ and } \mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [m]$$

- **Key Idea:** information entropy given frozen bits and difference sequence

$$\bar{D}_m \triangleq H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N, U_{\mathcal{F}^{(m)}}\right) \quad \text{and} \quad \Delta_m \triangleq \bar{D}_m - \bar{D}_{m-1}$$

- Experiment: assume U_1^N is uniform and Rx learns frozen bits causally.

- If U_m is an information bit, then
  $\Delta_m = H(U_m, Y_1^N | U_1^{m-1})$

- If U_m is a frozen bit, then
  $0 \geq \Delta_m \geq H\left(U_m \middle| Y_1^N, U_1^{m-1}\right) - 1$

An Information-Theoretic Perspective (2)

- For the first m input bits, the information/frozen sets are denoted as

$$\mathcal{A}^{(m)} \triangleq \mathcal{A} \cap [m] \text{ and } \mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [m]$$

- **Key Idea:** information entropy given frozen bits and difference sequence

$$\bar{D}_m \triangleq H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N, U_{\mathcal{F}^{(m)}}\right) \quad \text{and} \quad \Delta_m \triangleq \bar{D}_m - \bar{D}_{m-1}$$

- Experiment: assume U_1^N is uniform and Rx learns frozen bits causally.

- If U_m is an information bit, then
  $\Delta_m = H(U_m, Y_1^N | U_1^{m-1})$
- If U_m is a frozen bit, then
  $0 \geq \Delta_m \geq H\left(U_m \middle| Y_1^N, U_1^{m-1}\right) - 1$

$$\sum_{i \in \mathcal{A}^{(m)}} H\left(U_i \middle| Y_1^N, U_1^{i-1}\right) - \sum_{i \in \mathcal{F}^{(m)}} \left(1 - H\left(U_i \middle| Y_1^N, U_1^{i-1}\right)\right) \leq \bar{D}_m \leq \sum_{i \in \mathcal{A}^{(m)}} H\left(U_i \middle| Y_1^N, U_1^{i-1}\right)$$

Bounding the List Size

Theorem

Upon observing y_1^N when u_1^N is sent, we define the set (for $\alpha \in (0,1]$) $\mathcal{S}_\alpha^{(m)}\left(u_1^m, y_1^N\right) \triangleq \left\{\tilde{u}_1^m : \mathbb{P}\left(\tilde{u}_{\mathcal{A}^{(m)}} | y_1^N, \tilde{u}_{\mathcal{F}^{(m)}}\right) \geq \alpha \mathbb{P}\left(u_{\mathcal{A}^{(m)}} | y_1^N, u_{\mathcal{F}^{(m)}}\right)\right\}$. *Then,*

$$\mathrm{E}\left[\log_2 \left|\mathcal{S}_\alpha^{(m)}\right|\right] \leq \bar{D}_m + \log_2 \frac{1}{\alpha} = H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N, U_{\mathcal{F}^{(m)}}\right) + \log_2 \frac{1}{\alpha}$$

Bounding the List Size

Theorem

Upon observing y_1^N when u_1^N is sent, we define the set (for $\alpha \in (0,1]$)
$\mathcal{S}_\alpha^{(m)}\left(u_1^m, y_1^N\right) \triangleq \left\{\tilde{u}_1^m : \mathbb{P}\left(\tilde{u}_{\mathcal{A}^{(m)}} | y_1^N, \tilde{u}_{\mathcal{F}^{(m)}}\right) \geq \alpha \mathbb{P}\left(u_{\mathcal{A}^{(m)}} | y_1^N, u_{\mathcal{F}^{(m)}}\right)\right\}$. *Then,*

$$\mathrm{E}\left[\log_2 \left|\mathcal{S}_\alpha^{(m)}\right|\right] \leq \bar{D}_m + \log_2 \frac{1}{\alpha} = H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N, U_{\mathcal{F}^{(m)}}\right) + \log_2 \frac{1}{\alpha}$$

Proof.

$$\log_2 \left|\mathcal{S}_\alpha^{(m)}\right| = \log_2 \sum_{\tilde{u}_1^m} \mathbb{1}\left\{\mathbb{P}\left(\tilde{u}_{\mathcal{A}^{(m)}} | y_1^N, \tilde{u}_{\mathcal{F}^{(m)}}\right) \geq \underbrace{\alpha \mathbb{P}\left(u_{\mathcal{A}^{(m)}} \middle| y_1^N, u_{\mathcal{F}^{(m)}}\right)}_{q}\right\}$$
$$\leq \log_2 1/\left(\alpha \mathbb{P}\left(u_{\mathcal{A}^{(m)}} \middle| y_1^N, u_{\mathcal{F}^{(m)}}\right)\right)$$

Valid for all u_1^N and y_1^N; thus, we take expectation over all u_1^m and y_1^N □

Bounding the List Size

Theorem

Upon observing y_1^N when u_1^N is sent, we define the set (for $\alpha \in (0, 1]$)
$\mathcal{S}_\alpha^{(m)}\left(u_1^m, y_1^N\right) \triangleq \{\tilde{u}_1^m : \mathbb{P}\left(\tilde{u}_{\mathcal{A}^{(m)}}|y_1^N, \tilde{u}_{\mathcal{F}^{(m)}}\right) \geq \alpha\mathbb{P}\left(u_{\mathcal{A}^{(m)}}|y_1^N, u_{\mathcal{F}^{(m)}}\right)\}$*. Then,*

$$\mathrm{E}\left[\log_2 |\mathcal{S}_\alpha^{(m)}|\right] \leq \bar{D}_m + \log_2 \frac{1}{\alpha} = H\left(U_{\mathcal{A}^{(m)}}\middle|Y_1^N, U_{\mathcal{F}^{(m)}}\right) + \log_2 \frac{1}{\alpha}$$

- For an SCL decoder with max list size L_m during the m-th decoding step,
  - the decoder needs $L_m \geq |\mathcal{S}_1^{(m)}|$ for the true u_1^m to stay on the list
  - Choosing $\alpha < 1$ (say 0.94) captures near misses and matches entropy better.

A Few Remarks

- Our approach currently has two weaknesses:

A Few Remarks

- Our approach currently has two weaknesses:
  - Entropy mainly characterizes typical events but we care about rare events.

A Few Remarks

- Our approach currently has two weaknesses:
  - Entropy mainly characterizes typical events but we care about rare events.
  - The sequence $\bar{D}_m$ is averaged over Y_1^N, i.e., $\bar{D}_m = \sum_{y_1^N} \mathbb{P}(y_1^N) H\left(U_{\mathcal{A}(m)} \middle| Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$.

A Few Remarks

- Our approach currently has two weaknesses:
  - Entropy mainly characterizes typical events but we care about rare events.
  - The sequence $\bar{D}_m$ is averaged over Y_1^N, i.e., $\bar{D}_m = \sum_{y_1^N} \mathbb{P}(y_1^N) H\left(U_{\mathcal{A}(m)} | Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$.

    But the actual decoder sees a realization $d_m(y_1^N) \triangleq H\left(U_{\mathcal{A}(m)} | Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$

A Few Remarks

- Our approach currently has two weaknesses:
  - Entropy mainly characterizes typical events but we care about rare events.
  - The sequence $\bar{D}_m$ is averaged over Y_1^N, i.e., $\bar{D}_m = \sum_{y_1^N} \mathbb{P}(y_1^N) H\left(U_{\mathcal{A}(m)} | Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$.

    But the actual decoder sees a realization $d_m(y_1^N) \triangleq H\left(U_{\mathcal{A}(m)} | Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$
- Significance for code design:

A Few Remarks

- Our approach currently has two weaknesses:
  - Entropy mainly characterizes typical events but we care about rare events.
  - The sequence $\bar{D}_m$ is averaged over Y_1^N, i.e., $\bar{D}_m = \sum_{y_1^N} \mathbb{P}(y_1^N) H\left(U_{\mathcal{A}(m)} | Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$.

    But the actual decoder sees a realization $d_m(y_1^N) \triangleq H\left(U_{\mathcal{A}(m)} | Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$
- Significance for code design:
  - A first-order code design criterion can be seen as $\log_2 L_m \geq d_m$.

A Few Remarks

- Our approach currently has two weaknesses:
  - Entropy mainly characterizes typical events but we care about rare events.
  - The sequence $\bar{D}_m$ is averaged over Y_1^N, i.e., $\bar{D}_m = \sum_{y_1^N} \mathbb{P}(y_1^N) H\left(U_{\mathcal{A}(m)} \middle| Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$.
    But the actual decoder sees a realization $d_m(y_1^N) \triangleq H\left(U_{\mathcal{A}(m)} \middle| Y_1^N = y_1^N, U_{\mathcal{F}(m)}\right)$
- Significance for code design:
  - A first-order code design criterion can be seen as $\log_2 L_m \geq d_m$.
  - Based on this, a small code improvement will be introduced.

Dynamic Reed-Muller Codes

- d-RM code ensemble [CNP20]:
  - Let $\mathcal{A}$ be the information indices of an RM code.
  - u_i is an information bit if $i \in \mathcal{A}$.
  - $u_i = \sum_{j \in \mathcal{A}^{(i)}} A_{ij} u_j$ if $i \in \mathcal{F}$ where A_{ij} iid $\sim$ Bernoulli(0.5)

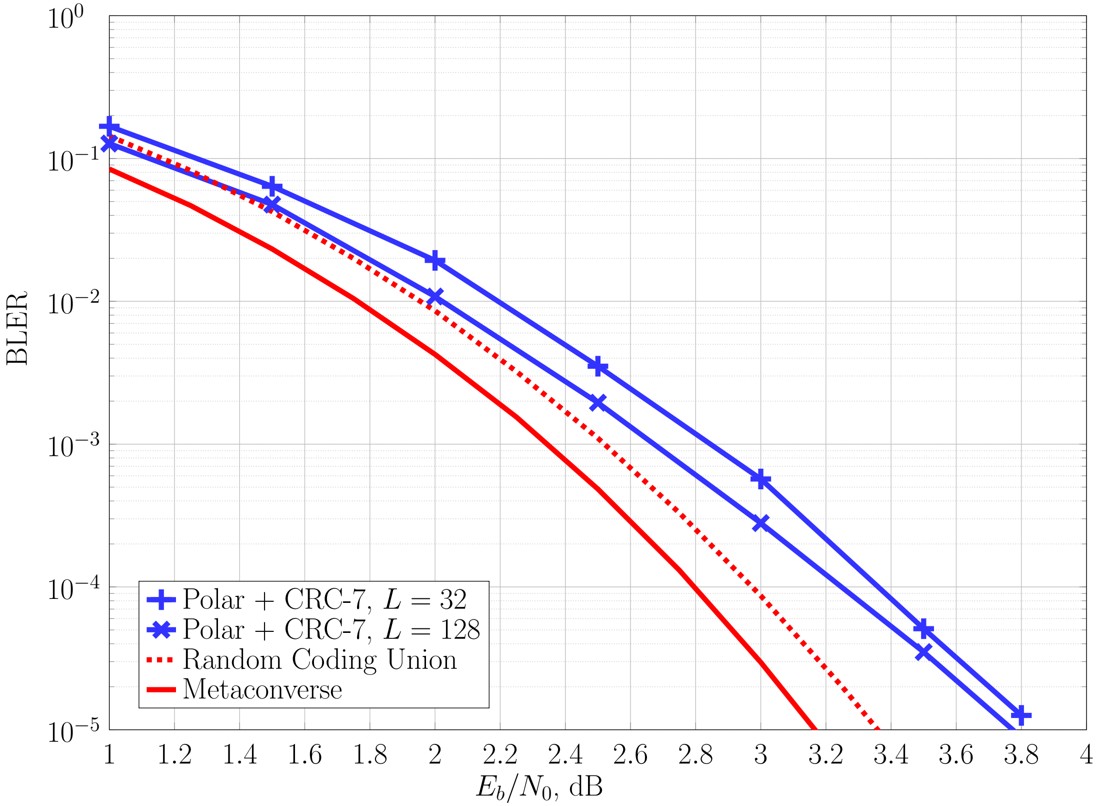
Dynamic Reed-Muller Codes

- d-RM code ensemble [CNP20]:
  - Let $\mathcal{A}$ be the information indices of an RM code.
  - u_i is an information bit if $i \in \mathcal{A}$.
  - $u_i = \sum_{j \in \mathcal{A}^{(i)}} A_{ij} u_j$ if $i \in \mathcal{F}$ where A_{ij} iid $\sim$ Bernoulli(0.5)
- Closely related to polarization-adjusted convolutional (PAC) codes [Arı19].

Dynamic Reed-Muller Codes

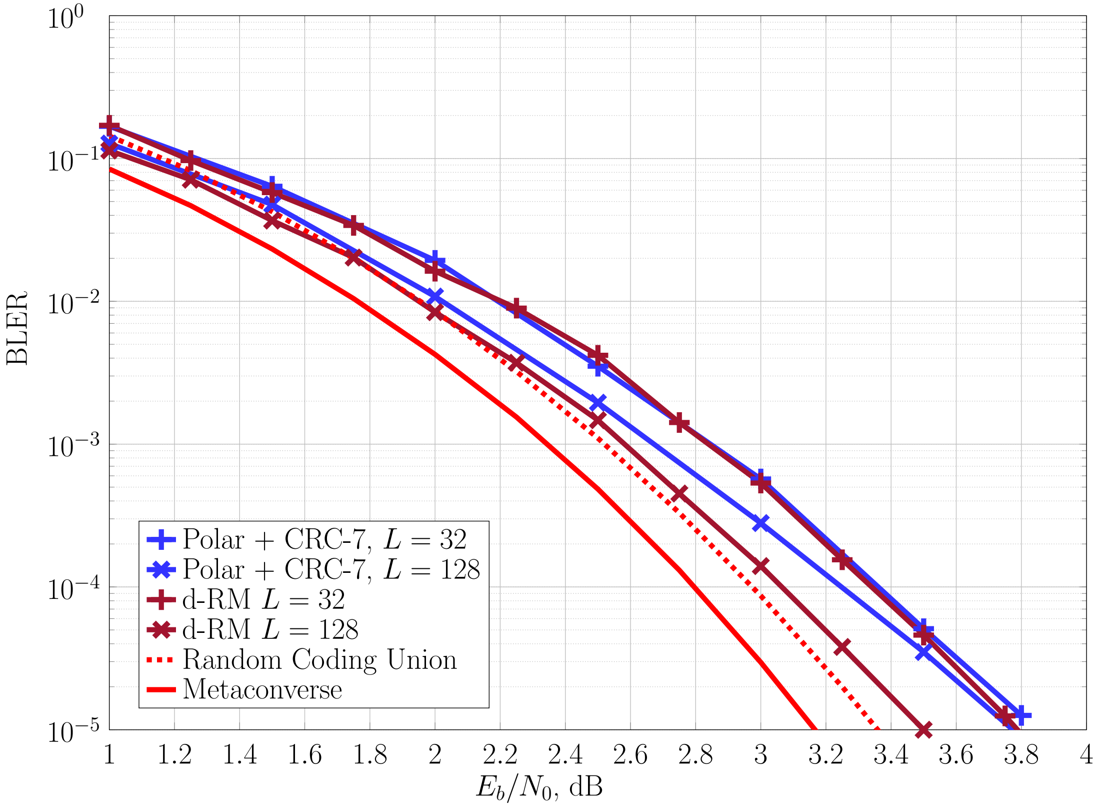
- d-RM code ensemble [CNP20]:
  - Let $\mathcal{A}$ be the information indices of an RM code.
  - u_i is an information bit if $i \in \mathcal{A}$.
  - $u_i = \sum_{j \in \mathcal{A}^{(i)}} A_{ij} u_j$ if $i \in \mathcal{F}$ where A_{ij} iid $\sim$ Bernoulli(0.5)
- Closely related to polarization-adjusted convolutional (PAC) codes [Arı19].
- PAC and (random instances of) d-RM code perform very similar under SCL decoding with the same list sizes.

$n = 128, k = 64$



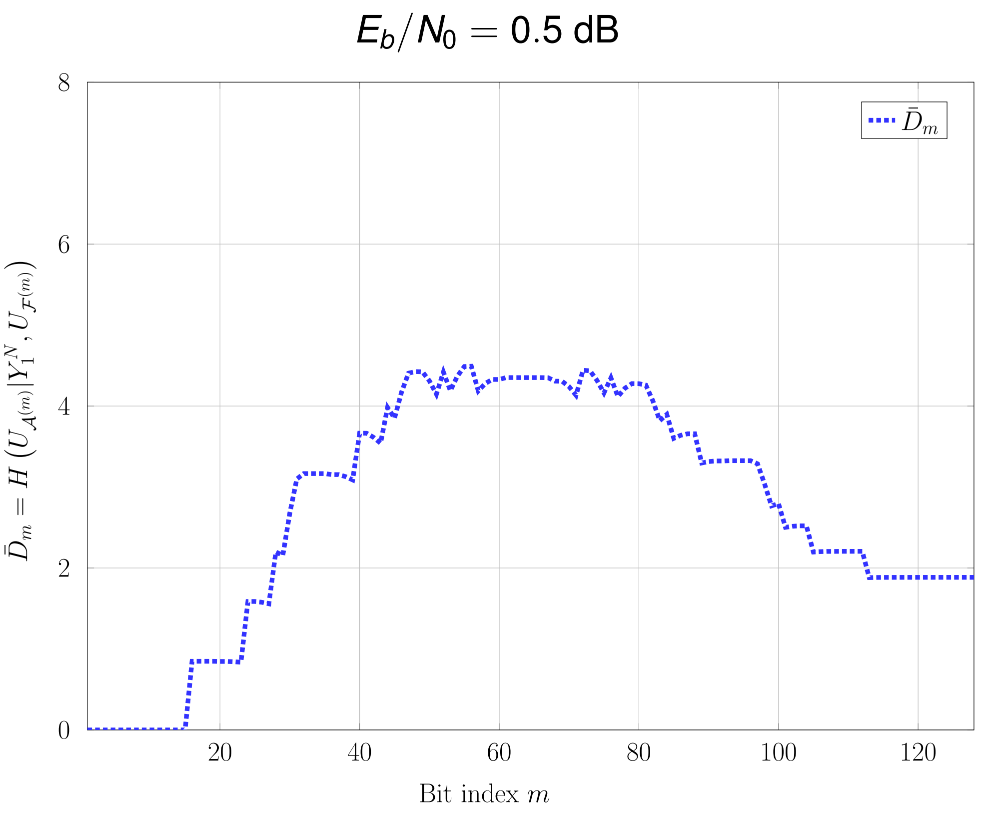
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1 1.2 1.4 1.6 1.8 2 2.2 2.4 2.6 2.8 3 3.2 3.4 3.6 3.8 4
E_b/N_0, dB
Polar + CRC-7, $L = 32$
Polar + CRC-7, $L = 128$
Random Coding Union
Metaconverse

$n = 128, k = 64$



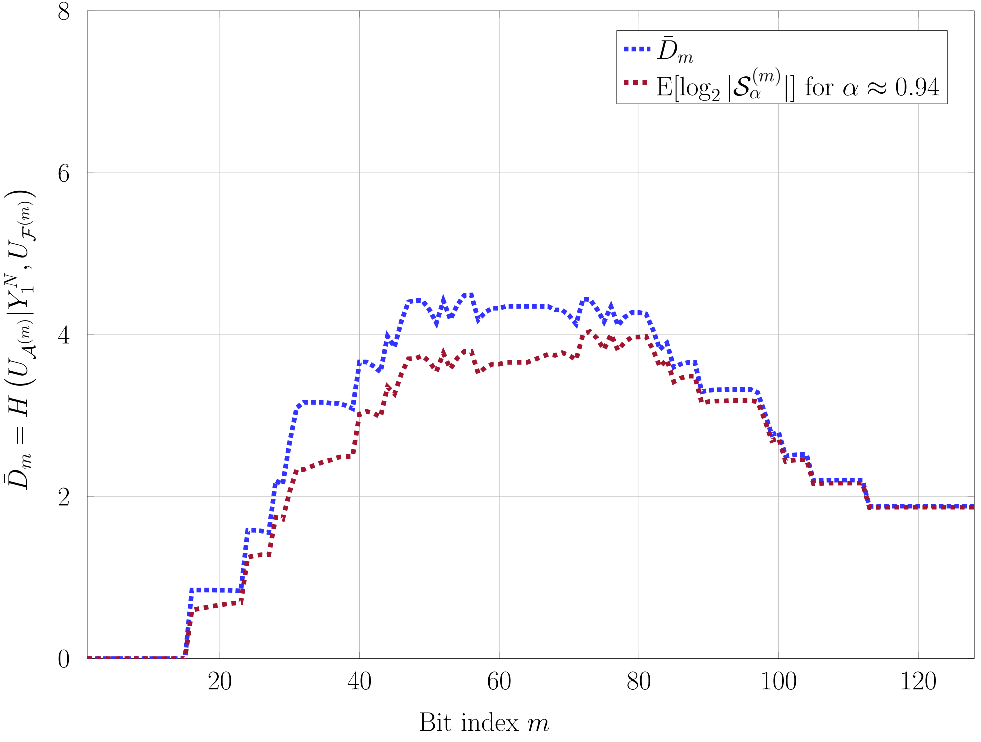
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
BLER
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4
E_b/N_0, dB
Polar + CRC-7, $L = 32$
Polar + CRC-7, $L = 128$
d-RM $L = 32$
d-RM $L = 128$
Random Coding Union
Metaconverse

$(128, 64)$ d-RM Code over the AWGN Channel



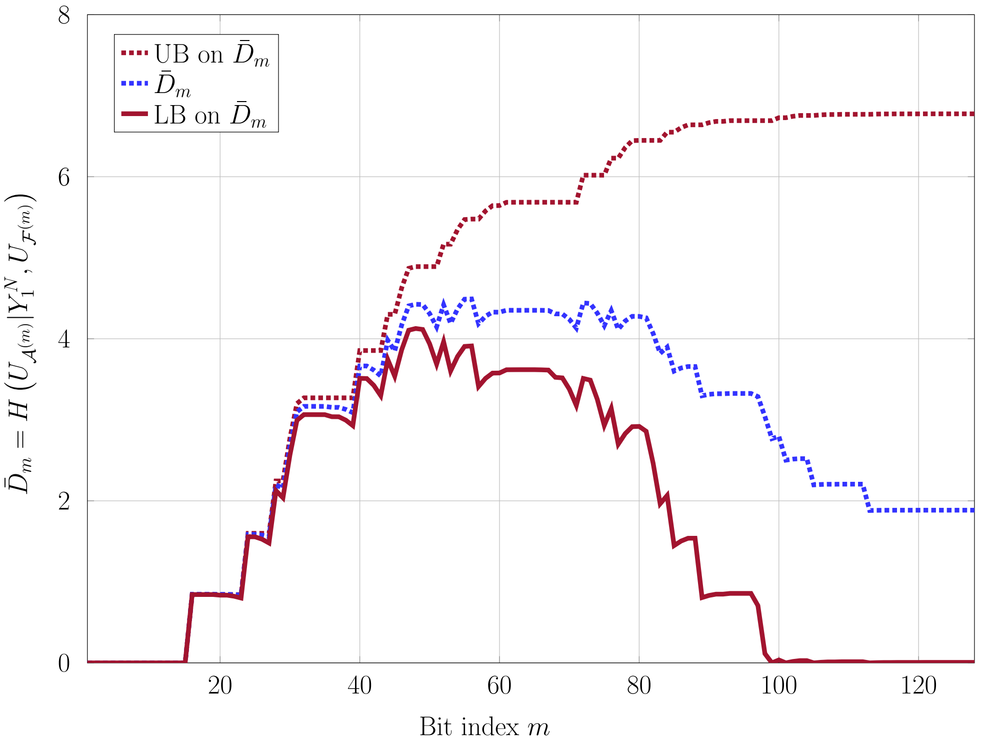
$(128, 64)$ d-RM Code over the AWGN Channel

$E_b/N_0 = 0.5$ dB



$(128, 64)$ d-RM Code over the AWGN Channel

$E_b/N_0 = 0.5$ dB

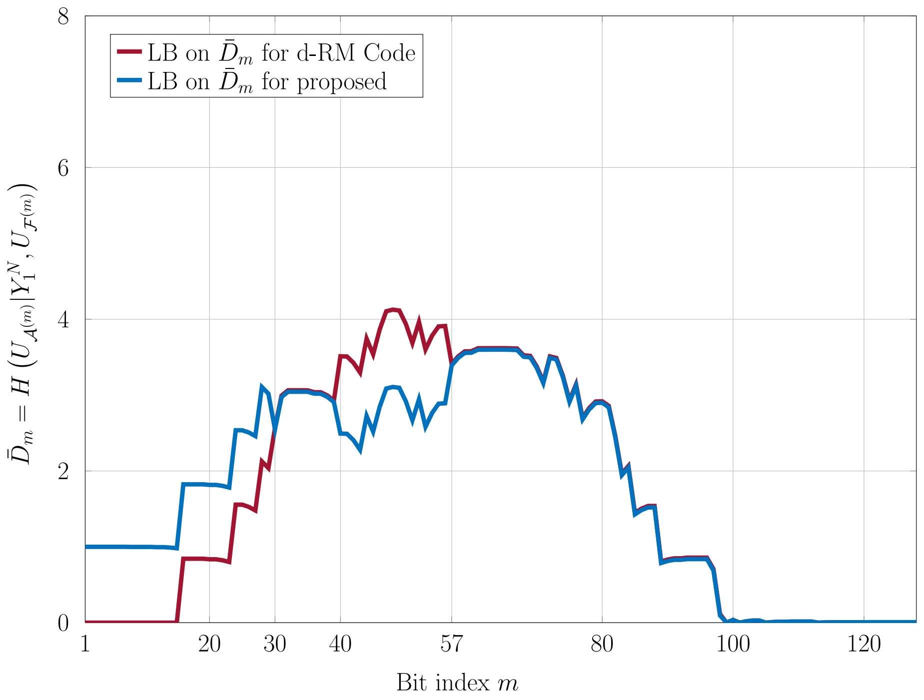


$(128, 64)$ Proposed vs d-RM Code over the AWGN Channel

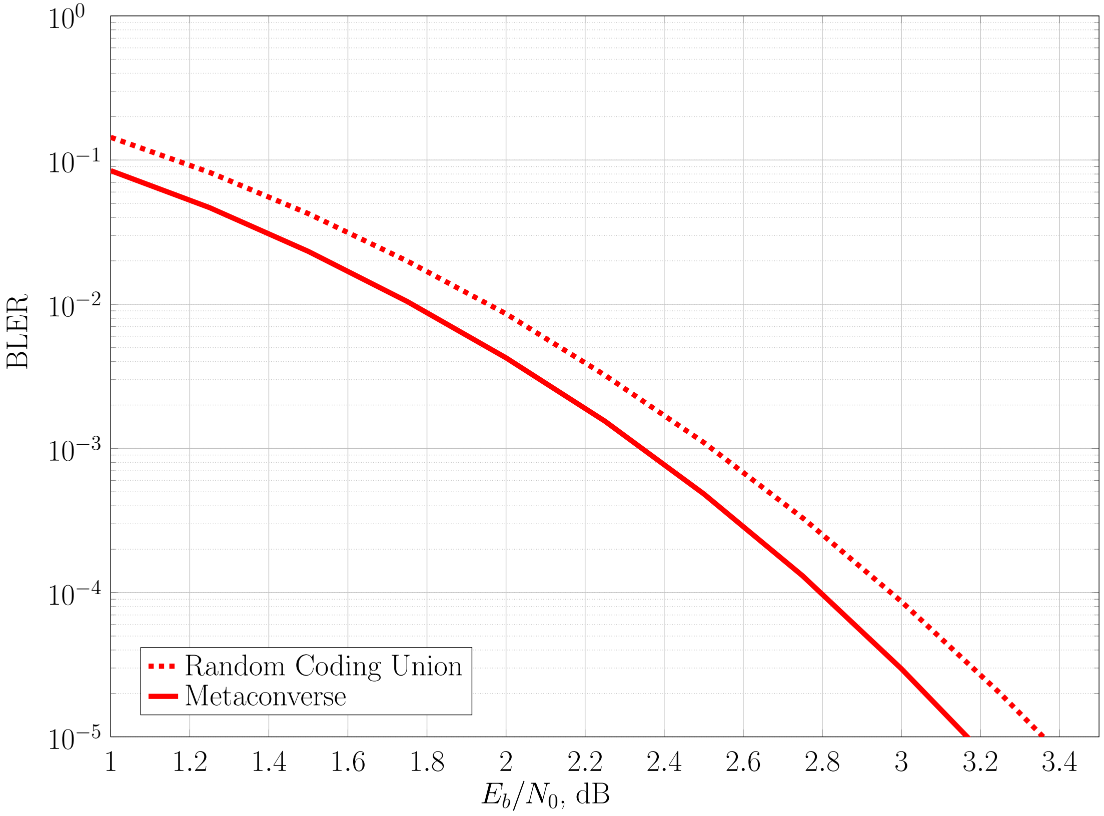
Proposed Code

- $u_{\{30,40\}}$ dynamic frozen bits
- $u_{\{1,57\}}$ info. bits

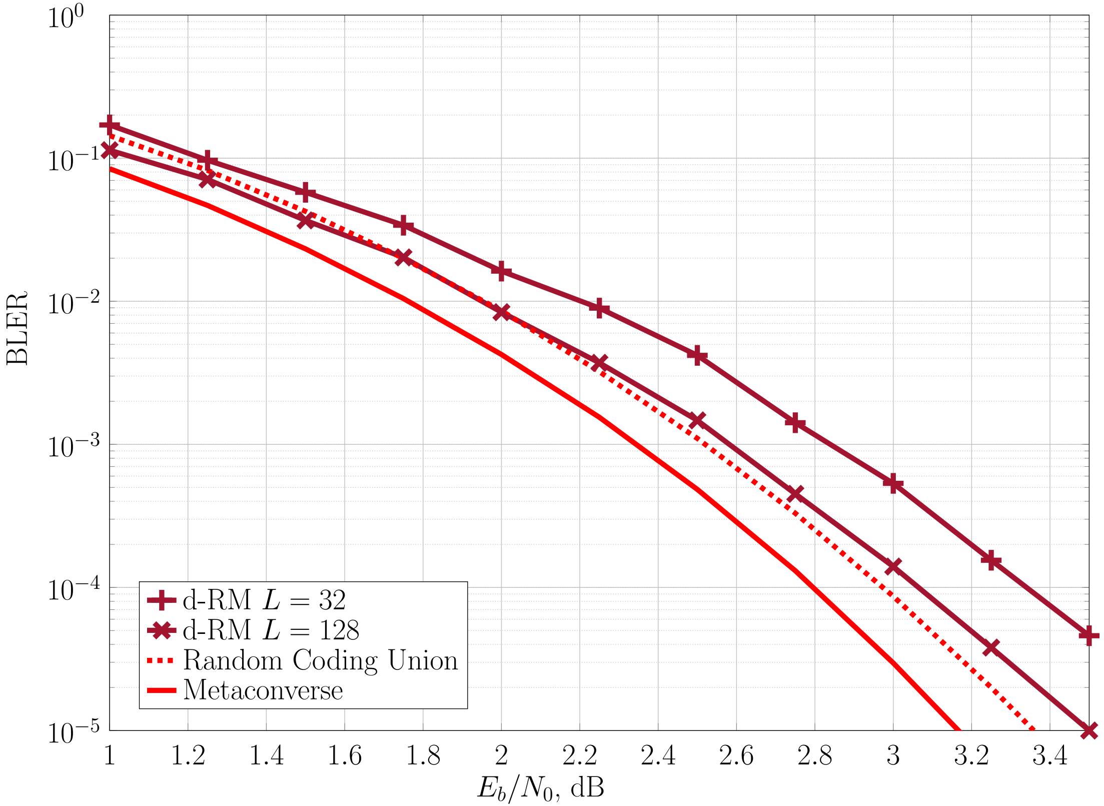
$E_b/N_0 = 0.5$ dB



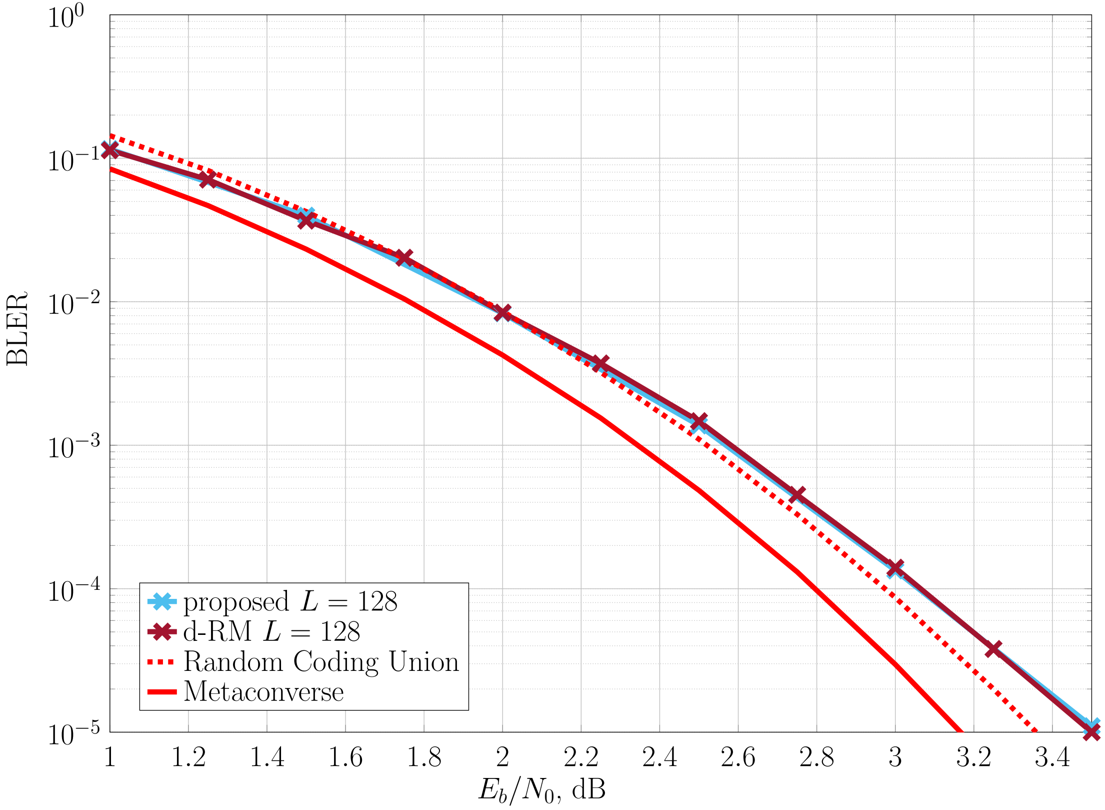
$(128, 64)$ Codes over the AWGN Channel



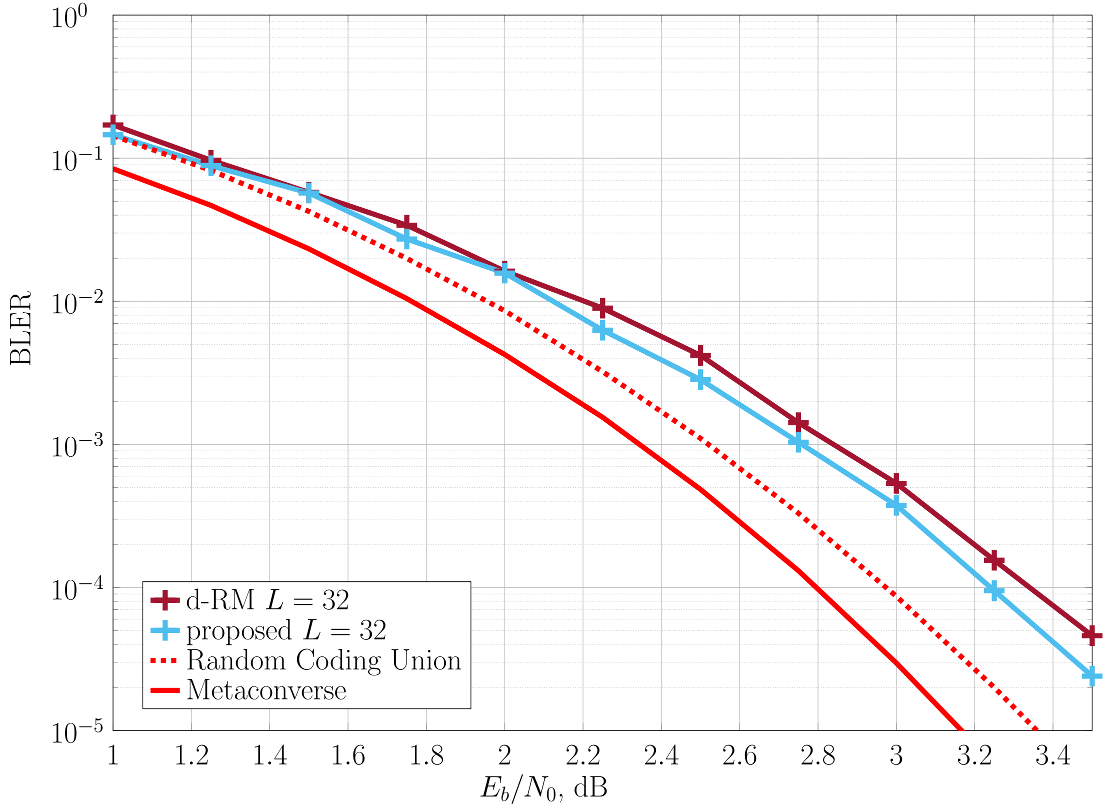
$(128, 64)$ Codes over the AWGN Channel



$(128, 64)$ Codes over the AWGN Channel



$(128, 64)$ Codes over the AWGN Channel



Recent Related Works

- Among many others, there are some recent works to be checked:
  - Works by E. Viterbo and his group: [RV19, RBV20]
  - A paper by A. Vardy and his group: [YFV20]
  - A paper by S. ten Brink and his group: [GEE$^+$20]

Outline

Successive Cancellation Inactivation Decoding

- The SC inactivation decoder has the same message passing schedule as the SC decoder.

Successive Cancellation Inactivation Decoding

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

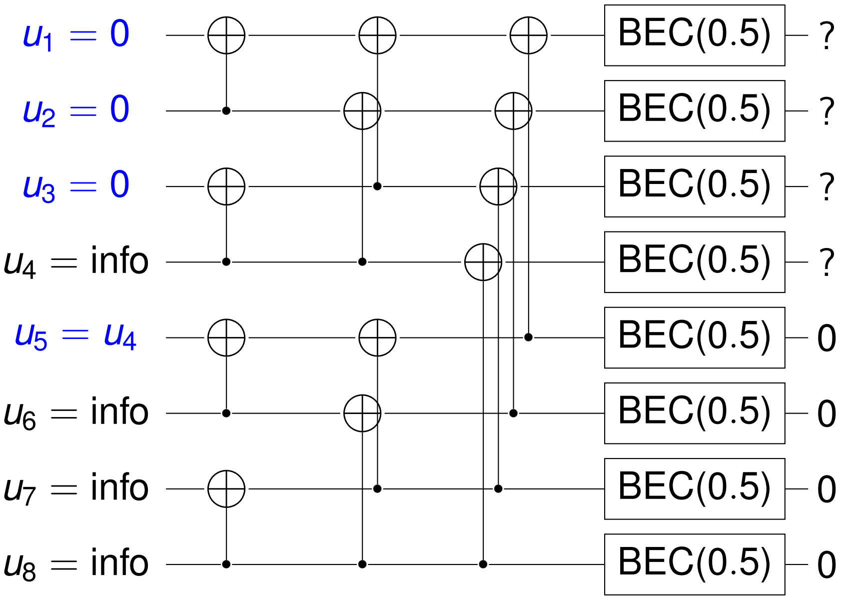
Successive Cancellation Inactivation Decoding

- The SC inactivation decoder has the same message passing schedule as the SC decoder.
- Whenever an information bit is decoded as erased, it is replaced by indeterminate 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.

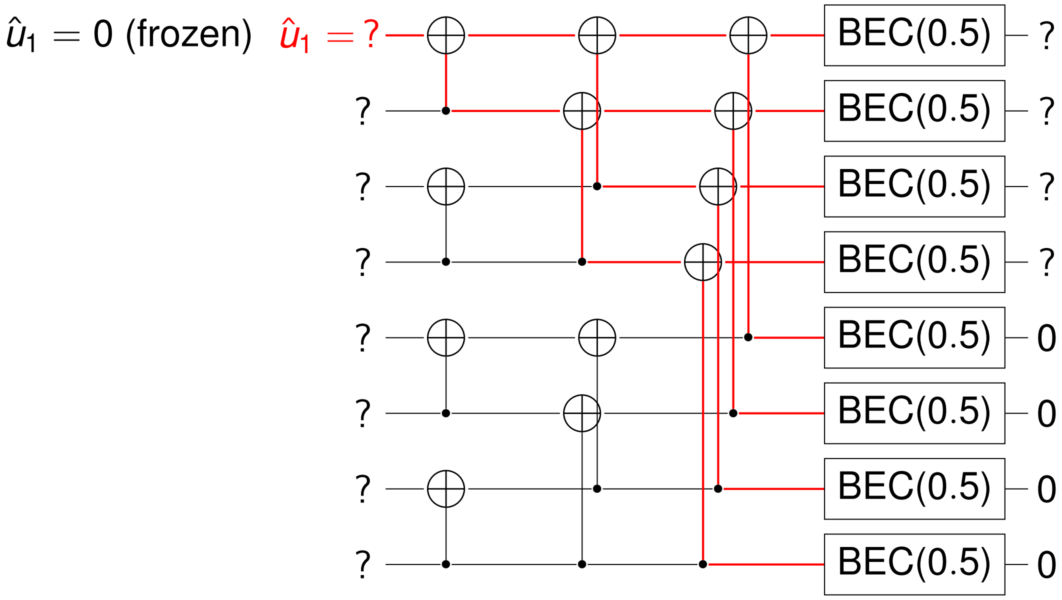
Successive Cancellation Inactivation Decoding

- The SC inactivation decoder has the same message passing schedule as the SC decoder.
- Whenever an information bit is decoded as erased, it is replaced by indeterminate 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.
- Previously inactivated bits may be resolved using linear equations derived from decoding frozen bits.

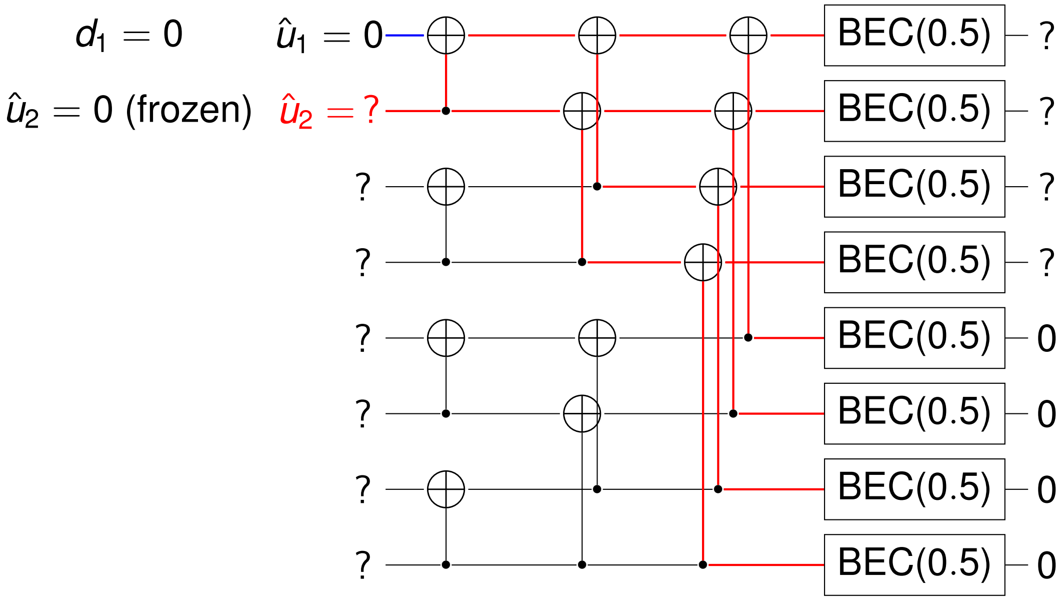
Example: SC Inactivation Decoding



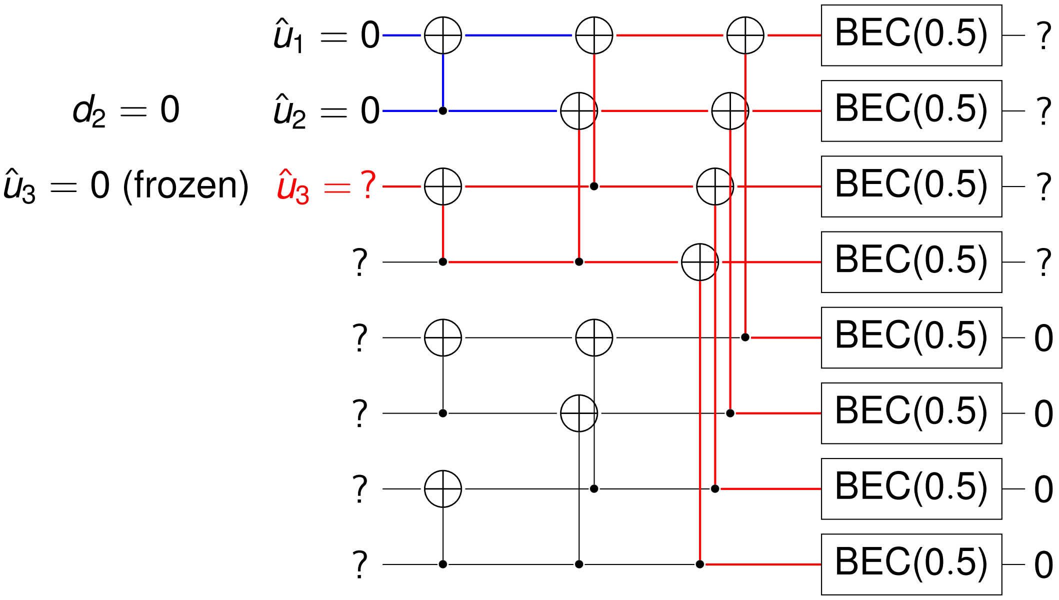
Example: SC Inactivation Decoding



Example: SC Inactivation Decoding

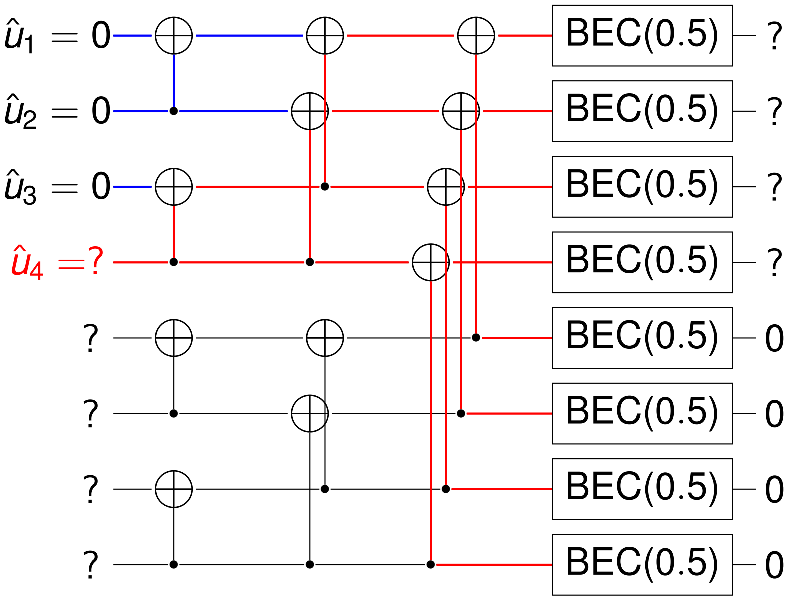


Example: SC Inactivation Decoding



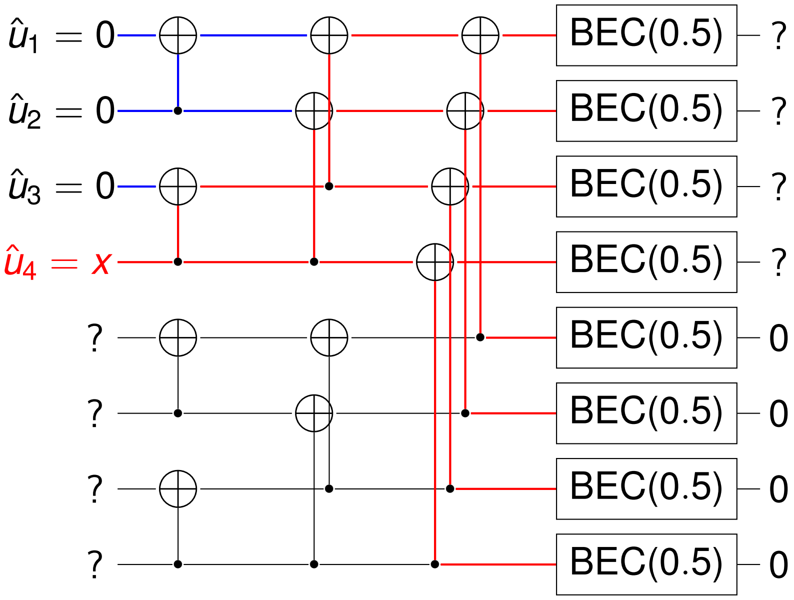
Example: SC Inactivation Decoding

$d_3 = 0$



Example: SC Inactivation Decoding

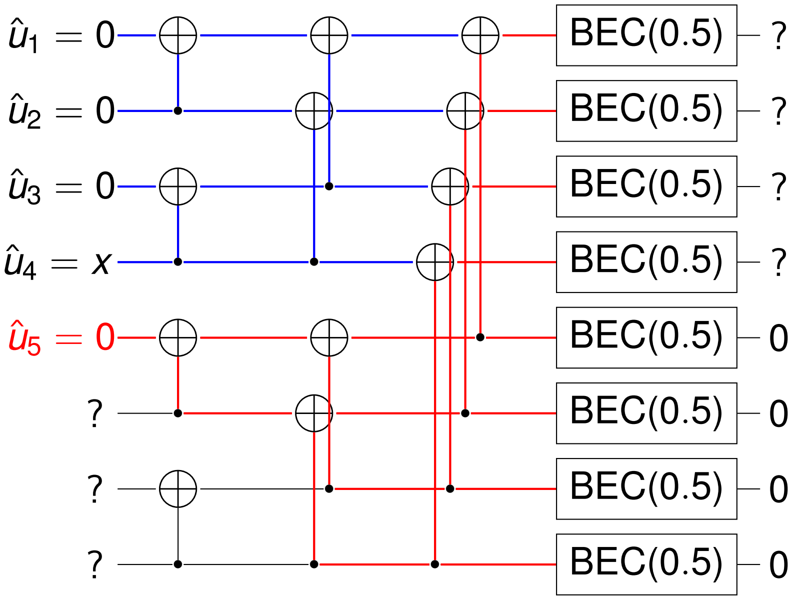
$d_3 = 0$



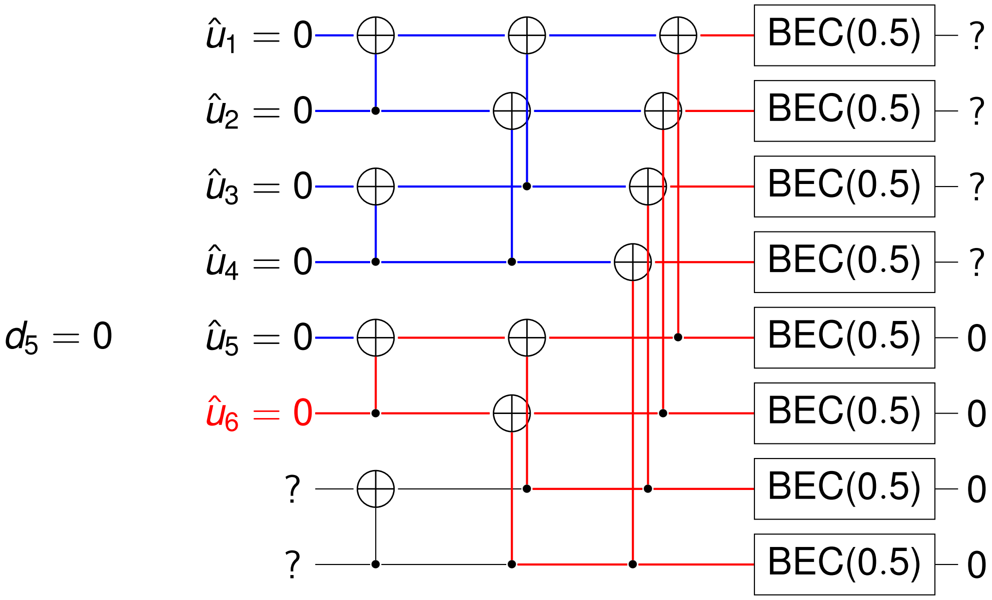
Example: SC Inactivation Decoding

$d_4 = 1$

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

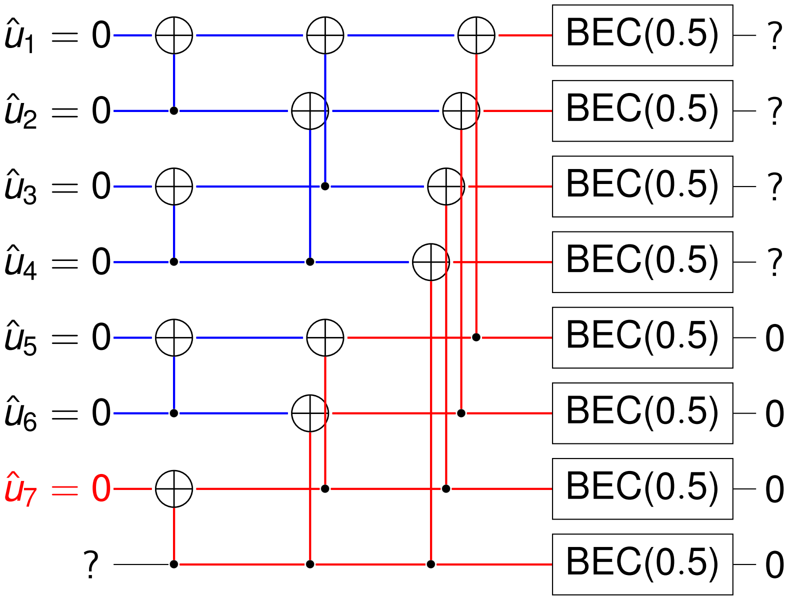


Example: SC Inactivation Decoding



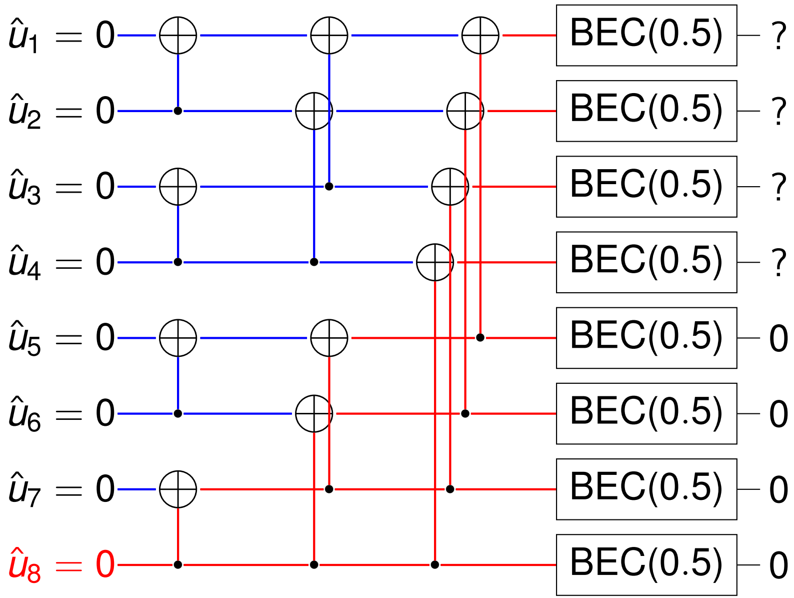
Example: SC Inactivation Decoding

$d_6 = 0$

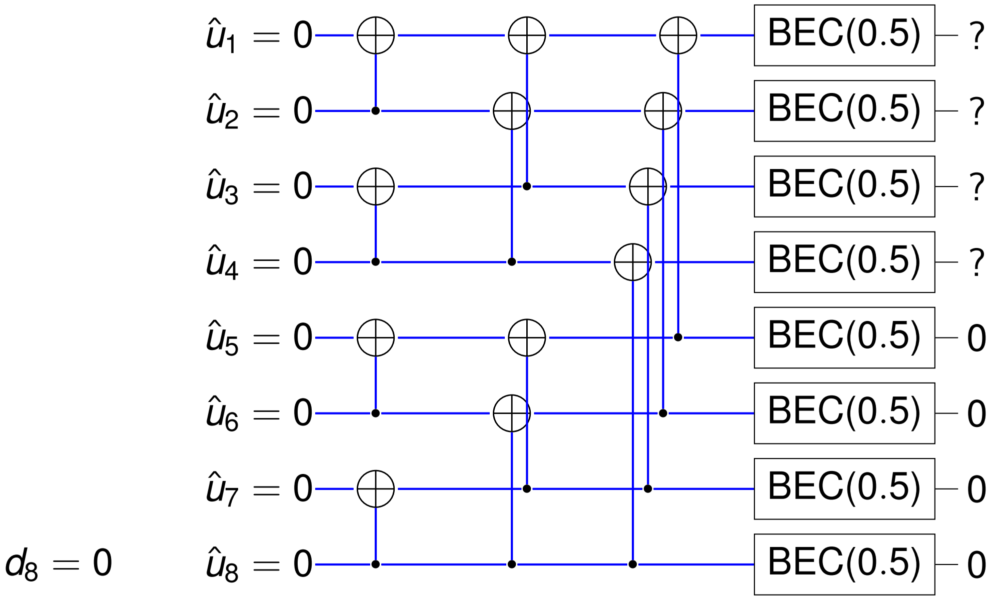


Example: SC Inactivation Decoding

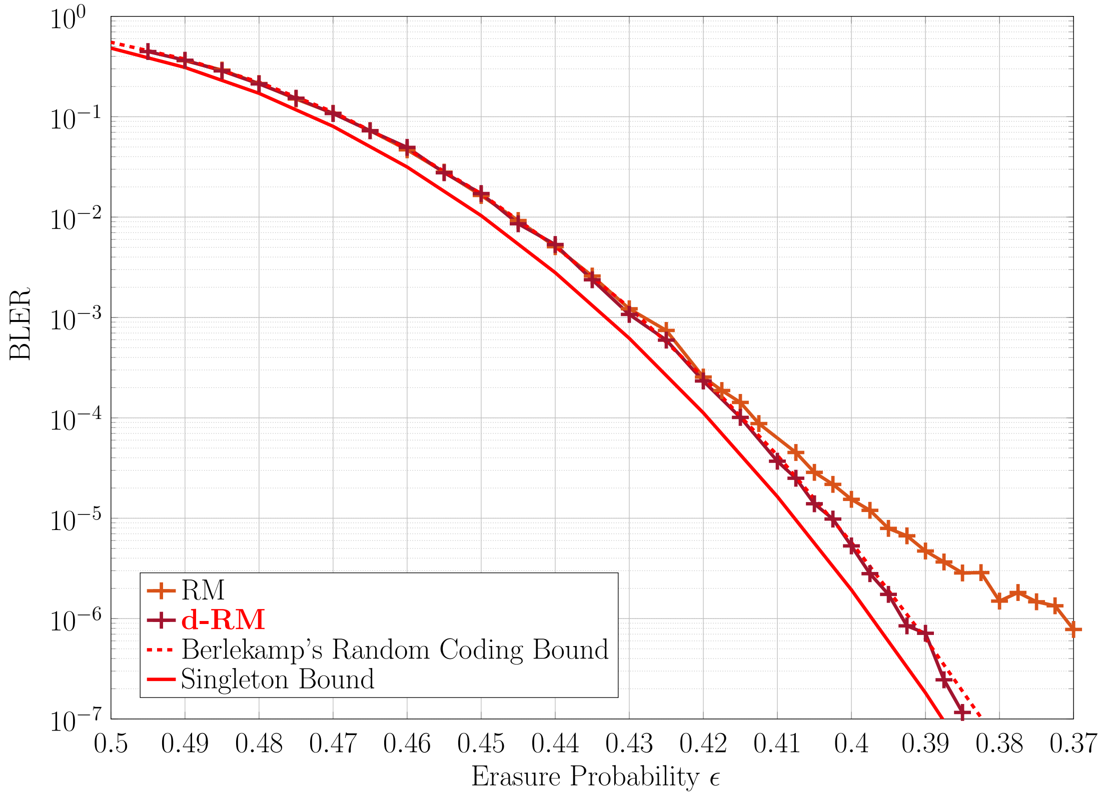
$d_7 = 0$



Example: SC Inactivation Decoding



(512, 256) Codes over the BEC



The Subspace Dimension

- For a fixed y_1^N, the subspace dimension is

$$d_m(y_1^N) = H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N = y_1^N, U_{\mathcal{F}^{(m)}}\right)$$

The Subspace Dimension

- For a fixed y_1^N, the subspace dimension is

$$d_m(y_1^N) = H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N = y_1^N, U_{\mathcal{F}^{(m)}}\right)$$

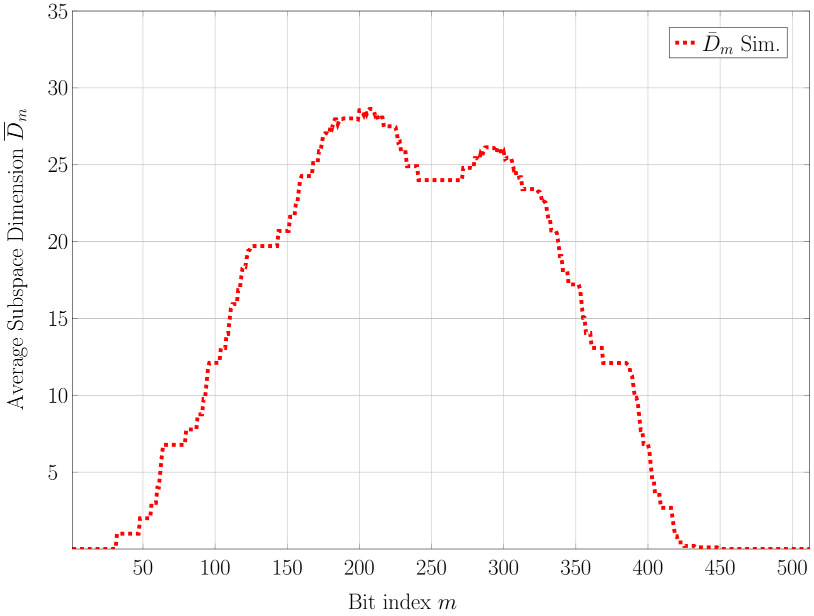
- Let $D_m = d_m(Y_1^N)$ denote corresponding random value at step m.

The Subspace Dimension

- For a fixed y_1^N, the subspace dimension is

$$d_m(y_1^N) = H\left(U_{\mathcal{A}^{(m)}} \middle| Y_1^N = y_1^N, U_{\mathcal{F}^{(m)}}\right)$$

- Let $D_m = d_m(Y_1^N)$ denote corresponding random value at step m.



Evolution of the Subspace Dimension

- If U_m is an information bit, then
  - If decoder outputs an erasure, then $d_m(y_1^N) = d_{m-1}(y_1^N) + 1$
  - Else, it outputs affine function and $d_m(y_1^N) = d_{m-1}(y_1^N)$

Evolution of the Subspace Dimension

- If U_m is an information bit, then
  - If decoder outputs an erasure, then $d_m(y_1^N) = d_{m-1}(y_1^N) + 1$
  - Else, it outputs affine function and $d_m(y_1^N) = d_{m-1}(y_1^N)$

- If U_m is a frozen bit, then
  - If decoder outputs an erasure, then $d_m(y_1^N) = d_{m-1}(y_1^N)$
  - Else, it outputs affine function:

Evolution of the Subspace Dimension

- If U_m is an information bit, then
  - If decoder outputs an erasure, then $d_m(y_1^N) = d_{m-1}(y_1^N) + 1$
  - Else, it outputs affine function and $d_m(y_1^N) = d_{m-1}(y_1^N)$
- If U_m is a frozen bit, then
  - If decoder outputs an erasure, then $d_m(y_1^N) = d_{m-1}(y_1^N)$
  - Else, it outputs affine function:
    i) If consolidation: $d_m(y_1^N) = d_{m-1}(y_1^N) - 1$
    ii) Else, no consolidation: $d_m(y_1^N) = d_{m-1}(y_1^N)$

Evolution of the Subspace Dimension

- If U_m is an information bit, then
  - If decoder outputs an erasure, then $d_m(y_1^N) = d_{m-1}(y_1^N) + 1$
  - Else, it outputs affine function and $d_m(y_1^N) = d_{m-1}(y_1^N)$
- If U_m is a frozen bit, then
  - If decoder outputs an erasure, then $d_m(y_1^N) = d_{m-1}(y_1^N)$
  - Else, it outputs affine function:
    - i) If consolidation: $d_m(y_1^N) = d_{m-1}(y_1^N) - 1$
    - ii) Else, no consolidation: $d_m(y_1^N) = d_{m-1}(y_1^N)$

Averaged over all y_1^N, the erasure probabilities are obtained via density evolution.
Must approximate consolidation probabilities.

The Markov Chain Approximation

- The random sequence $D_1, \ldots, D_N$ can be approximated by an inhomogeneous Markov chain with transition probabilities $P_{i,j}^{(m)} \approx \mathbb{P}\left(D_m = j \,|\, D_{m-1} = i\right)$ where

$$P_{i,j}^{(m)} = \begin{cases} \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i+1 \\ 1 - \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i \\ \epsilon_N^{(m)} + \left(1 - \epsilon_N^{(m)}\right) 2^{-D_{m-1}} & \text{if } m \in \mathcal{F},\ j = i \\ \left(1 - \epsilon_N^{(m)}\right)\left(1 - 2^{-D_{m-1}}\right) & \text{if } m \in \mathcal{F},\ j = i-1 \end{cases}$$

The Markov Chain Approximation

- The random sequence $D_1, \ldots, D_N$ can be approximated by an inhomogeneous Markov chain with transition probabilities $P_{i,j}^{(m)} \approx \mathbb{P}\left(D_m = j \mid D_{m-1} = i\right)$ where

$$P_{i,j}^{(m)} = \begin{cases} \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i+1 \\ 1 - \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i \\ \epsilon_N^{(m)} + \left(1 - \epsilon_N^{(m)}\right) 2^{-D_{m-1}} & \text{if } m \in \mathcal{F},\ j = i \\ \left(1 - \epsilon_N^{(m)}\right)\left(1 - 2^{-D_{m-1}}\right) & \text{if } m \in \mathcal{F},\ j = i-1 \end{cases}$$

- $\epsilon_N^{(m)}$ is the DE erasure probability of m-th effective channel

The Markov Chain Approximation

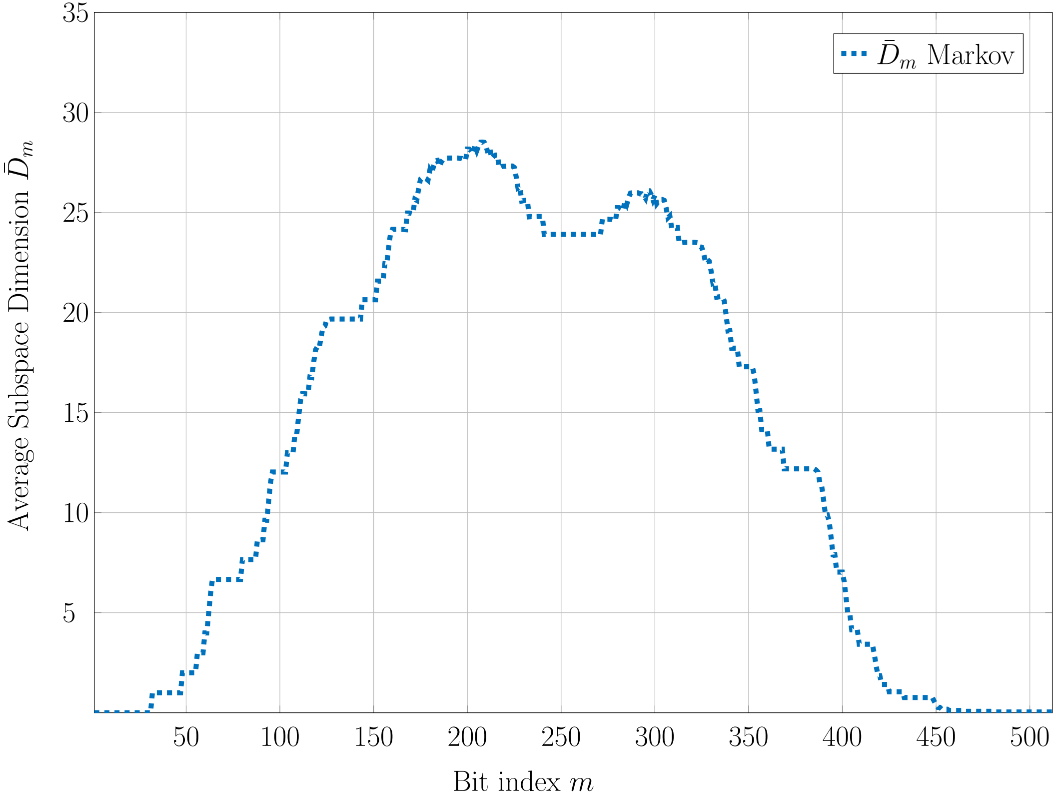
- The random sequence $D_1, \ldots, D_N$ can be approximated by an inhomogeneous Markov chain with transition probabilities $P_{i,j}^{(m)} \approx \mathbb{P}\left(D_m = j \,|\, D_{m-1} = i\right)$ where

$$P_{i,j}^{(m)} = \begin{cases} \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i+1 \\ 1 - \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i \\ \epsilon_N^{(m)} + \left(1 - \epsilon_N^{(m)}\right) 2^{-D_{m-1}} & \text{if } m \in \mathcal{F},\ j = i \\ \left(1 - \epsilon_N^{(m)}\right)\left(1 - 2^{-D_{m-1}}\right) & \text{if } m \in \mathcal{F},\ j = i-1 \end{cases}$$

- $\epsilon_N^{(m)}$ is the DE erasure probability of m-th effective channel
- 2^{-D} is probability a random D-variable equation has all zero coefficients

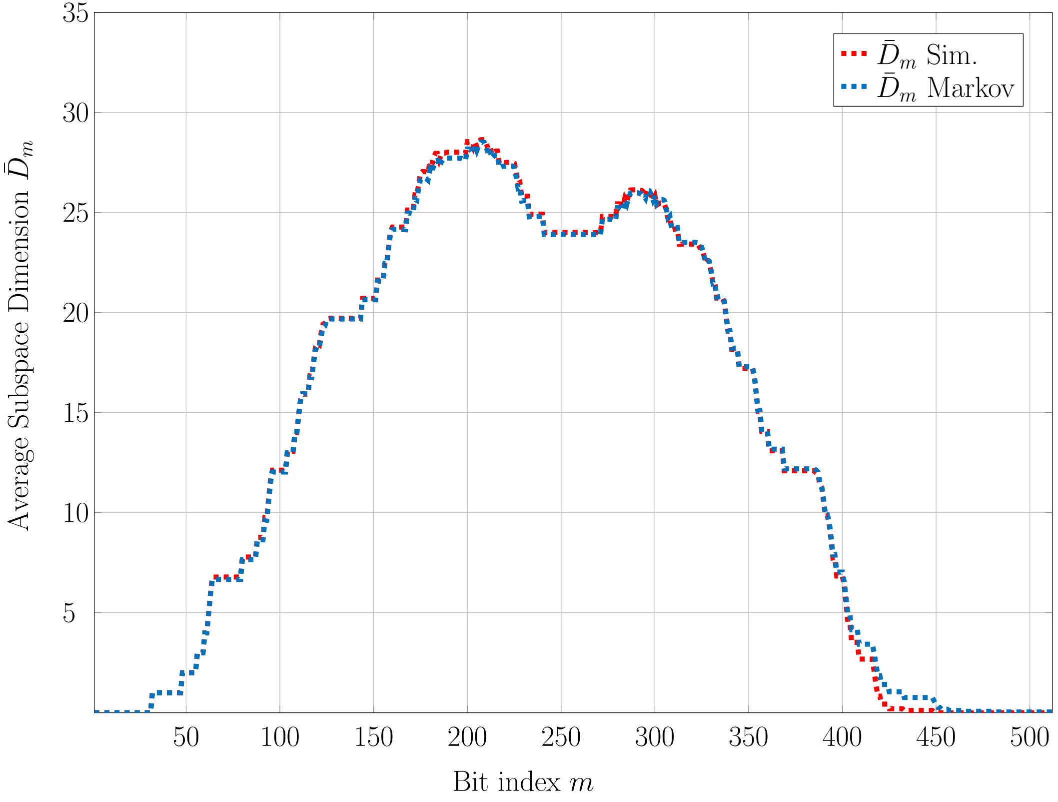
(512, 256) d-RM Code

A fixed-weight BEC with exactly $\texttt{round}(512 \times 0.48) = 246$ erasures



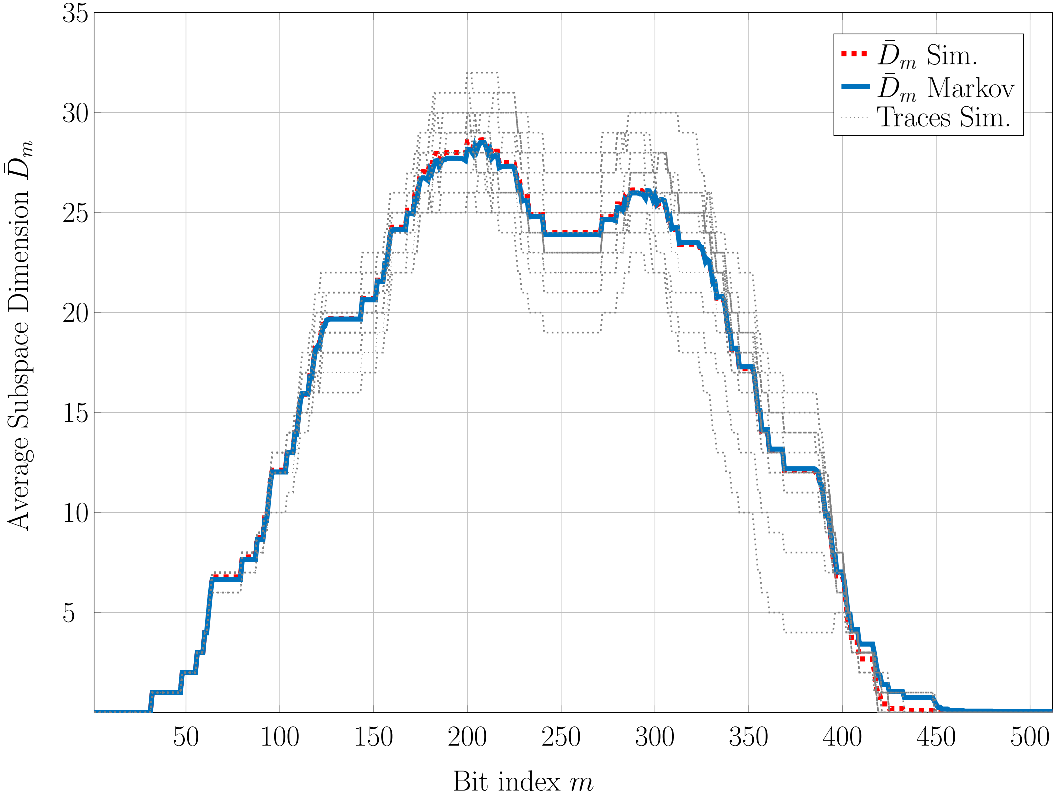
$(512, 256)$ d-RM Code

A fixed-weight BEC with exactly $\texttt{round}(512 \times 0.48) = 246$ erasures



$(512, 256)$ d-RM Code

A fixed-weight BEC with exactly $\texttt{round}(512 \times 0.48) = 246$ erasures



Concentration of the Subspace Dimension

Theorem

The subspace dimension D_m for a particular random realization Y_1^N concentrates around the mean $\bar{D}_m$ for sufficiently large block lengths [CP21], i.e., for any $\beta > 0$, we have

$$\mathbb{P}\left\{\frac{1}{N}|D_m - \bar{D}_m| > \beta\right\} \leq 2\exp\left(-\frac{\beta^2}{2}N\right). \qquad (1)$$

Concentration of the Subspace Dimension

Theorem

The subspace dimension D_m for a particular random realization Y_1^N concentrates around the mean $\bar{D}_m$ for sufficiently large block lengths [CP21], i.e., for any $\beta > 0$, we have

$$\mathbb{P}\left\{\frac{1}{N}|D_m - \bar{D}_m| > \beta\right\} \leq 2\exp\left(-\frac{\beta^2}{2}N\right). \qquad (1)$$

Proof.

Key observation: at any decoding stage, the subspace dimension satisfies Lipschitz-1 condition: For all $i \in [N]$ and all values y_1^N and $\tilde{y}_i$, we have

$$|d_m(y_1^N) - d_m(y_1^{i-1}, \tilde{y}_i, y_{i+1}^N)| \leq 1.$$

Then, use Azuma-Hoeffding inequality by forming a Doob's Martingale. □

Concentration of the Subspace Dimension

Theorem

The subspace dimension D_m for a particular random realization Y_1^N concentrates around the mean $\bar{D}_m$ for sufficiently large block lengths [CP21], i.e., for any $\beta > 0$, we have

$$\mathbb{P}\left\{\frac{1}{N}|D_m - \bar{D}_m| > \beta\right\} \leq 2\exp\left(-\frac{\beta^2}{2}N\right). \tag{1}$$

- We use the theorem above to give bounds on the average complexity of ML decoding of a given code implemented via SCI decoding.
- Extension to general BMS channels is possible (the case of continuous output channels should be tackled with more care).

Outline

Summary

- Recent advances (dynamic frozen bits + SCL) in polar codes allow performance near random coding union bound for $(128, 64)$ with moderate complexity.

Summary

- Recent advances (dynamic frozen bits + SCL) in polar codes allow performance near random coding union bound for $(128, 64)$ with moderate complexity.
- “What list size is sufficient to approach maximum-likelihood (ML) decoding performance under an SCL decoder?”
  - Information theory provides some useful measures.

Summary

- Recent advances (dynamic frozen bits + SCL) in polar codes allow performance near random coding union bound for $(128, 64)$ with moderate complexity.
- “What list size is sufficient to approach maximum-likelihood (ML) decoding performance under an SCL decoder?”
  - Information theory provides some useful measures.
- The analysis leads to an improved code design (in comparison with the PAC code [Arı19]) under SCL decoding with list size 32.

Summary

- Recent advances (dynamic frozen bits + SCL) in polar codes allow performance near random coding union bound for $(128, 64)$ with moderate complexity.
- “What list size is sufficient to approach maximum-likelihood (ML) decoding performance under an SCL decoder?”
  - Information theory provides some useful measures.
- The analysis leads to an improved code design (in comparison with the PAC code [Arı19]) under SCL decoding with list size 32.
- An efficient ML decoding of polar (and RM codes) are introduced and d-RM codes were shown to perform very close to BRCB even for small lengths (e.g., 512 bits)

Summary

- Recent advances (dynamic frozen bits + SCL) in polar codes allow performance near random coding union bound for $(128, 64)$ with moderate complexity.
- "What list size is sufficient to approach maximum-likelihood (ML) decoding performance under an SCL decoder?"
  - Information theory provides some useful measures.
- The analysis leads to an improved code design (in comparison with the PAC code [Arı19]) under SCL decoding with list size 32.
- An efficient ML decoding of polar (and RM codes) are introduced and d-RM codes were shown to perform very close to BRCB even for small lengths (e.g., 512 bits)
- The concentration of the random subspace dimension makes the average analysis meaningful; hence, we can upper bound the average complexity of SC inactivation decoding.

Summary

- Recent advances (dynamic frozen bits + SCL) in polar codes allow performance near random coding union bound for $(128, 64)$ with moderate complexity.
- "What list size is sufficient to approach maximum-likelihood (ML) decoding performance under an SCL decoder?"
  - Information theory provides some useful measures.
- The analysis leads to an improved code design (in comparison with the PAC code [Arı19]) under SCL decoding with list size 32.
- An efficient ML decoding of polar (and RM codes) are introduced and d-RM codes were shown to perform very close to BRCB even for small lengths (e.g., 512 bits)
- The concentration of the random subspace dimension makes the average analysis meaningful; hence, we can upper bound the average complexity of SC inactivation decoding.
- Outlook and Future Work
  - Apply this technique to design longer codes with good SCL performance

Thanks

References I

[Arı09] 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.

[Arı19] E. Arıkan.
From sequential decoding to channel polarization and back again.
CoRR, abs/1908.09594, 2019.

[AT09] E. Arıkan and E. Telatar.
On the rate of channel polarization.
In *IEEE Int. Symp. Inf. Theory*, pages 1493–1495, 2009.

[CNP20] 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*, 2020.

[CP20] M. C. Coşkun and H. D. Pfister.
Bounds on the list size of successive cancellation list decoding.
In *2020 Int. Conf. on Signal Process. and Commun. (SPCOM)*, pages 1–5, 2020.

[CP21] M. C. Coşkun and H. D. Pfister.
An information-theoretic perspective on successive cancellation list decoding and polar code design (in prep.).
2021.

[DS06] I. Dumer and K. Shabunov.
Soft-decision decoding of reed-muller codes: recursive lists.
IEEE Trans. Inf. Theory, 52(3):1260–1266, 2006.

References II

[GEE+20] M. Geiselhart, A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink.
Crc-aided belief propagation list decoding of polar codes.
In *IEEE Int. Symp. Inf. Theory*, pages 395–400, 2020.

[RBV20] Mohammad Rowshan, Andreas Burg, and Emanuele Viterbo.
Polarization-adjusted convolutional (PAC) codes: Fano decoding vs list decoding.
CoRR, abs/2002.06805, 2020.

[RV19] Mohammad Rowshan and Emanuele Viterbo.
How to modify polar codes for list decoding.
In *IEEE International Symposium on Information Theory*, pages 1772–1776, 2019.

[Sto02] Norbert Stolte.
Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung.
PhD thesis, TU Darmstadt, 2002.

[TM16] P. Trifonov and V. Miloslavskaya.
Polar subcodes.
IEEE J. Sel. Areas Commun., 34(2):254–266, Feb. 2016.

[TV15] I. Tal and A. Vardy.
List decoding of polar codes.
IEEE Trans. Inf. Theory, 61(5):2213–2226, May 2015.

[YFV20] Hanwen Yao, Arman Fazeli, and Alexander Vardy.
List decoding of arıkan's PAC codes.
CoRR, abs/2005.13711, 2020.