

Introduction to Polar Codes

Henry D. Pfister

Error-Correcting Codes
Duke University
Spring 2022

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

- First deterministic construction of:
provably capacity-achieving codes for binary memoryless symmetric (BMS) channels [Arı09]

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

- First deterministic construction of:
provably capacity-achieving codes for binary memoryless symmetric (BMS) channels [Ari09]
- They also have low encoding/decoding complexity (i.e., $O(N \log N)$ operations)

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

Erdal Arıkan, *Senior Member, IEEE*

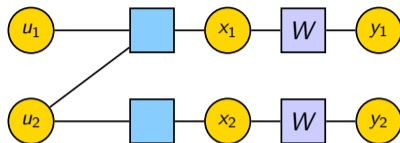
Abstract—A method is proposed, called channel polarization, to construct code sequences that achieve the symmetric capacity $I(W)$ of any given binary-input discrete memoryless channel (B-DMC) W . The symmetric capacity is the highest rate achievable subject to using the input letters of the channel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

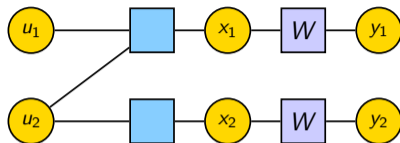
- First deterministic construction of:
provably capacity-achieving codes for binary memoryless symmetric (BMS) channels [Ari09]
- They also have low encoding/decoding complexity (i.e., $O(N \log N)$ operations)
- We'll begin with a simple description of the idea behind channel polarization

The Fundamental Block



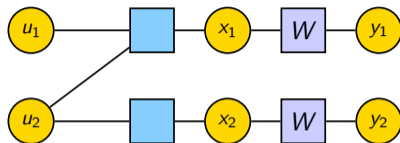
- 1 Given uniform bits $u_1, u_2 \in \{0, 1\}$, encode into $x_1 = u_1 \oplus u_2$ and $x_2 = u_2$

The Fundamental Block



- 1 Given uniform bits $u_1, u_2 \in \{0, 1\}$, encode into $x_1 = u_1 \oplus u_2$ and $x_2 = u_2$
- 2 Transmit x_1, x_2 over independent channels defined by $W(y_1|x_1)$ and $W(y_2|x_2)$

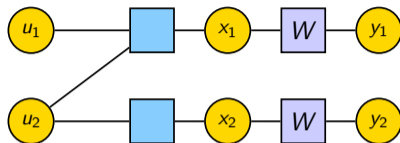
The Fundamental Block



- 1 Given uniform bits $u_1, u_2 \in \{0, 1\}$, encode into $x_1 = u_1 \oplus u_2$ and $x_2 = u_2$
- 2 Transmit x_1, x_2 over independent channels defined by $W(y_1|x_1)$ and $W(y_2|x_2)$
- 3 Using y_1, y_2 , estimate u_1 as \hat{u}_1 based on the effective channel

$$(W \boxtimes W)(y_1, y_2|u_1) \triangleq \frac{1}{2} \sum_{u_2 \in \{0,1\}} W(y_1|u_1 \oplus u_2)W(y_2|u_2)$$

The Fundamental Block



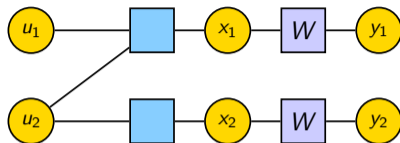
- 1 Given uniform bits $u_1, u_2 \in \{0, 1\}$, encode into $x_1 = u_1 \oplus u_2$ and $x_2 = u_2$
- 2 Transmit x_1, x_2 over independent channels defined by $W(y_1|x_1)$ and $W(y_2|x_2)$
- 3 Using y_1, y_2 , estimate u_1 as \hat{u}_1 based on the effective channel

$$(W \boxtimes W)(y_1, y_2|u_1) \triangleq \frac{1}{2} \sum_{u_2 \in \{0,1\}} W(y_1|u_1 \oplus u_2)W(y_2|u_2)$$

- 4 Decode u_2 from y_1, y_2, \hat{u}_1 based on the effective channel

$$(W \circledast W)(y_1, y_2, u_1|u_2) \triangleq \frac{1}{2} W(y_1|u_1 \oplus u_2)W(y_2|u_2)$$

The Fundamental Block



- 1 Given uniform bits $u_1, u_2 \in \{0, 1\}$, encode into $x_1 = u_1 \oplus u_2$ and $x_2 = u_2$
- 2 Transmit x_1, x_2 over independent channels defined by $W(y_1|x_1)$ and $W(y_2|x_2)$
- 3 Using y_1, y_2 , estimate u_1 as \hat{u}_1 based on the effective channel

$$(W \boxtimes W)(y_1, y_2|u_1) \triangleq \frac{1}{2} \sum_{u_2 \in \{0,1\}} W(y_1|u_1 \oplus u_2)W(y_2|u_2)$$

- 4 Decode u_2 from y_1, y_2, \hat{u}_1 based on the effective channel

$$(W \circledast W)(y_1, y_2, u_1|u_2) \triangleq \frac{1}{2} W(y_1|u_1 \oplus u_2)W(y_2|u_2)$$

- 5 Maps 2 independent channels into 2 coupled channels. How does this help?

Erasure Channel Example: $W = \text{BEC}(\epsilon)$

What is the erasure probability for $\hat{u}_1 = y_1 \oplus y_2$?

u_1	$x_1 x_2 u_2$	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$\mathbb{P}(\hat{u}_1 = ?)$
0	000	00	0	?0	?	0?	?	??	?	$2\epsilon(1 - \epsilon) + \epsilon^2$
	111	11	0	?1	?	1?	?	??	?	
1	100	10	1	?0	?	0?	?	??	?	$2\epsilon(1 - \epsilon) + \epsilon^2$
	011	01	1	?1	?	1?	?	??	?	

Erasure Channel Example: $W = \text{BEC}(\epsilon)$

What is the erasure probability for $\hat{u}_1 = y_1 \oplus y_2$?

u_1	$x_1 x_2 u_2$	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$\mathbb{P}(\hat{u}_1 = ?)$
0	000	00	0	?0	?	0?	?	??	?	$2\epsilon(1 - \epsilon) + \epsilon^2$
	111	11	0	?1	?	1?	?	??	?	
1	100	10	1	?0	?	0?	?	??	?	$2\epsilon(1 - \epsilon) + \epsilon^2$
	011	01	1	?1	?	1?	?	??	?	

What is the erasure probability for $\hat{u}_2 = y_1 \oplus \hat{u}_1$ / $\hat{u}_2 = y_2$ assuming $\hat{u}_1 = u_1$?

u_2	$x_1 x_2 u_1$	$y_1 y_2$	\hat{u}_2	$y_1 y_2$	\hat{u}_2	$y_1 y_2$	\hat{u}_2	$y_1 y_2$	\hat{u}_2	$\mathbb{P}(\hat{u}_2 = ?)$
0	000	00	0	?0	0	0?	0	??	?	ϵ^2
	101	10	0	?0	0	1?	0	??	?	
1	110	11	1	?1	1	1?	1	??	?	ϵ^2
	011	01	1	?1	1	0?	1	??	?	

Erasure Channel Example: $W = \text{BEC}(\epsilon)$

What is the erasure probability for $\hat{u}_1 = y_1 \oplus y_2$?

u_1	$x_1 x_2 u_2$	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$y_1 y_2$	\hat{u}_1	$\mathbb{P}(\hat{u}_1 = ?)$
0	000	00	0	?0	?	0?	?	??	?	$2\epsilon(1 - \epsilon) + \epsilon^2$
	111	11	0	?1	?	1?	?	??	?	
1	100	10	1	?0	?	0?	?	??	?	$2\epsilon(1 - \epsilon) + \epsilon^2$
	011	01	1	?1	?	1?	?	??	?	

What is the erasure probability for $\hat{u}_2 = y_1 \oplus \hat{u}_1 / \hat{u}_2 = y_2$ assuming $\hat{u}_1 = u_1$?

u_2	$x_1 x_2 u_1$	$y_1 y_2$	\hat{u}_2	$y_1 y_2$	\hat{u}_2	$y_1 y_2$	\hat{u}_2	$y_1 y_2$	\hat{u}_2	$\mathbb{P}(\hat{u}_2 = ?)$
0	000	00	0	?0	0	0?	0	??	?	ϵ^2
	101	10	0	?0	0	1?	0	??	?	
1	110	11	1	?1	1	1?	1	??	?	ϵ^2
	011	01	1	?1	1	0?	1	??	?	

Average erasure probability = $\frac{1}{2} (2\epsilon(1 - \epsilon) + \epsilon^2 + \epsilon^2) = \epsilon$ is conserved!

Combining Fundamental Blocks

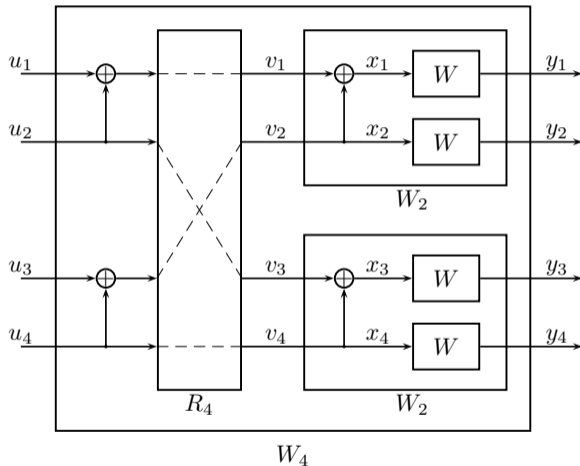


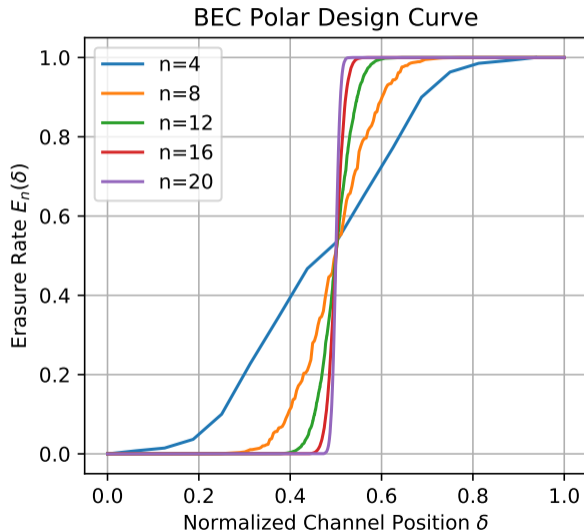
Figure taken from: Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels", IT 2009.

Effective Channels for Polar Codes on the BEC

- For the BEC, all effective channels are BECs with different erasure rates
- Each stage of polarization maps two $\text{BEC}(\epsilon_{\text{in}})$ channels into two new BECs with different erasure rates:

$$\epsilon_{\text{in}}^- = 1 - (1 - \epsilon_{\text{in}})^2 \quad \epsilon_{\text{in}}^+ = \epsilon_{\text{in}}^2.$$

- Consider n polar stages starting from $\text{BEC}(0.5)$ and then sort the channels into increasing order by erasure rate. Let $E_n(\delta)$ be the erasure rate of the $\lceil \delta 2^n \rceil$ -th sorted channel. For $n = 8$, the figure shows roughly 40% of the 256 channels have erasure rate ≤ 0.1 .



Kronecker Products

For matrices $A = \{a_{i,j}\}$ and $B = \{b_{i,j}\}$, let $A \otimes B$ represent the Kronecker product

$$A \otimes B \triangleq \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots \\ a_{2,1}B & a_{2,2}B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Kronecker powers: $A^{\otimes n} \triangleq A \otimes A^{\otimes n-1} = A^{\otimes n-1} \otimes A$. Multiplication identity:

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

Polar codes are based on Kronecker powers of G_2 :

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad G_2^{\otimes n} = \underbrace{(I \otimes \cdots \otimes I \otimes G_2)}_{n-1 \text{ copies}} \underbrace{(I \otimes \cdots \otimes I \otimes G_2 \otimes I)}_{n-2 \text{ copies}} \cdots (G_2 \otimes \underbrace{I \otimes \cdots \otimes I}_{n-1 \text{ copies}})$$

Perfect-Shuffle Permutations

An $N \times N$ permutation matrix P_N satisfies $P_N^T P_N = I_N$, where I_N is the $N \times N$ identity matrix because it preserves the Euclidean distance between pairs of vectors and is therefore orthogonal.

Perfect-Shuffle Permutations

An $N \times N$ permutation matrix P_N satisfies $P_N^T P_N = I_N$, where I_N is the $N \times N$ identity matrix because it preserves the Euclidean distance between pairs of vectors and is therefore orthogonal.

The “mod- p perfect shuffle” $S_{p,q}$ with $N = pq$ is defined by

$$S_{2,N/2}(s_1, s_2, \dots, s_N)^T = (s_1, s_3, \dots, s_{N-1}, s_2, s_4, \dots, s_N)^T$$

$$S_{3,N/3}(s_1, s_2, \dots, s_N)^T = (s_1, s_4, \dots, s_{N-2}, s_2, s_5, \dots, s_{N-1}, s_3, s_6, \dots, s_N)^T.$$

Perfect-Shuffle Permutations

An $N \times N$ permutation matrix P_N satisfies $P_N^T P_N = I_N$, where I_N is the $N \times N$ identity matrix because it preserves the Euclidean distance between pairs of vectors and is therefore orthogonal.

The “mod- p perfect shuffle” $S_{p,q}$ with $N = pq$ is defined by

$$S_{2,N/2}(s_1, s_2, \dots, s_N)^T = (s_1, s_3, \dots, s_{N-1}, s_2, s_4, \dots, s_N)^T$$

$$S_{3,N/3}(s_1, s_2, \dots, s_N)^T = (s_1, s_4, \dots, s_{N-2}, s_2, s_5, \dots, s_{N-1}, s_3, s_6, \dots, s_N)^T.$$

Let A and B be $m_1 \times n_1$ and $m_2 \times n_2$ matrices. Since $A \otimes B$ differs from $B \otimes A$ only in the order of the rows and columns, the mod- p perfect shuffle gives the identity

$$B \otimes A = S_{m_1, m_2}^T (A \otimes B) S_{n_1, n_2}.$$

Perfect-Shuffle Permutations

An $N \times N$ permutation matrix P_N satisfies $P_N^T P_N = I_N$, where I_N is the $N \times N$ identity matrix because it preserves the Euclidean distance between pairs of vectors and is therefore orthogonal.

The “mod- p perfect shuffle” $S_{p,q}$ with $N = pq$ is defined by

$$S_{2,N/2}(s_1, s_2, \dots, s_N)^T = (s_1, s_3, \dots, s_{N-1}, s_2, s_4, \dots, s_N)^T$$

$$S_{3,N/3}(s_1, s_2, \dots, s_N)^T = (s_1, s_4, \dots, s_{N-2}, s_2, s_5, \dots, s_{N-1}, s_3, s_6, \dots, s_N)^T.$$

Let A and B be $m_1 \times n_1$ and $m_2 \times n_2$ matrices. Since $A \otimes B$ differs from $B \otimes A$ only in the order of the rows and columns, the mod- p perfect shuffle gives the identity

$$B \otimes A = S_{m_1, m_2}^T (A \otimes B) S_{n_1, n_2}.$$

In particular, if A is a 2×2 matrix and B is an $(N/2) \times (N/2)$ matrix, then

$$B \otimes A = S_{2, N/2}^T (A \otimes B) S_{2, N/2} = R_N (A \otimes B) R_N^T, \text{ where } R_N \triangleq S_{2, N/2}^T$$

is called the **reverse shuffle** and $(s_1, s_2, \dots, s_N) R_N = (s_1, s_3, \dots, s_{N-1}, s_2, s_4, \dots, s_N)$.

Polar Transform

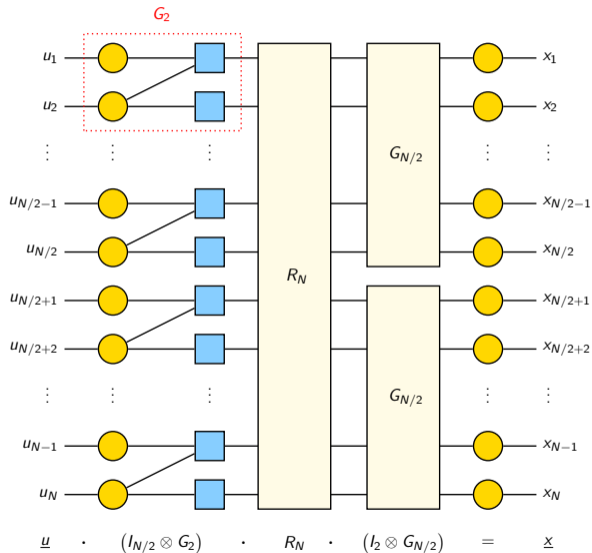
The **polar transform** of size N is defined to be

$$G_N \triangleq B_N \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n} = B_N G_2^{\otimes n},$$

where B_N is the $N \times N$ **bit-reversal permutation matrix**. For $N = 2^n$ and (a_1, \dots, a_n) , this permutation maps the element $i = 1 + \sum_{k=1}^n 2^{k-1} a_k$ to position $j = 1 + \sum_{k=1}^n 2^{n-k} a_k$.

- The matrix G_N is used to define the generator matrix of a polar code
- A key design parameter is the **set of information positions** $\mathcal{A} \subseteq \{1, 2, \dots, N\}$
- The idea is to construct a message vector \underline{u} where the elements u_i with $i \in \mathcal{A}$ carry information and whose other elements u_j with $j \in \mathcal{A}^c$ are **frozen to 0**.
- Generator matrix: submatrix of G_N formed by keeping only the rows with indices in \mathcal{A} .
- Parity-check matrix: submatrix of G_N^T formed by keeping only the rows with indices in \mathcal{A}^c .

Recursive Encoding Perspective



Factor graph associated with $G_N = (I_{N/2} \otimes G_2)R_N(I_2 \otimes G_{N/2})$ decomposition

Pseudocode for Recursive Encoding

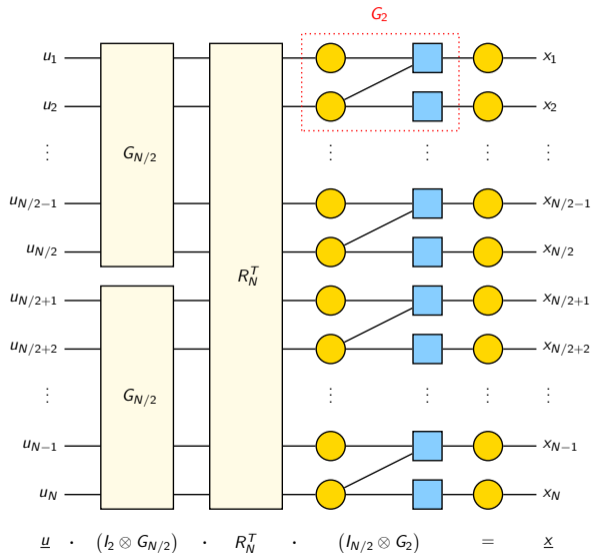
Algorithm 1 Recursive Implementation of Polar Transform in Python

```
# Encode polar information vector u
def polar_transform(u):
    # Recurse down to length 1
    if (len(u)==1):
        return u;

    # R_N maps odd/even indices (i.e., u1u2/u2) to first/second half
    # Compute odd/even outputs of (I_{N/2} \otimes G_2) transform
    x = np.zeros(len(u), dtype=np.int64)
    x[:len(u)//2] = polar_transform((u[::2]+u[1::2])%2)
    x[len(u)//2:] = polar_transform(u[1::2])
    return x
```

Note: The structure of polar codes is naturally suited to recursive algorithms. See [PL17].

Recursive Decoding Perspective



Factor graph associated with $G_N = (I_2 \otimes G_{N/2})R_N^T(I_{N/2} \otimes G_2)$ decomposition

Soft-Decoding Rules

Assume the channel outputs are normalized into “probability of 1” (P1) values that satisfy

$$y_1 = \mathbb{P}(X_1 = 1 | Y_1 = y_1) = \frac{\mathbb{P}(Y_1 = y_1 | X_1 = 1)}{\mathbb{P}(Y_1 = y_1 | X_1 = 0) + \mathbb{P}(Y_1 = y_1 | X_1 = 1)}.$$

Using this, the bit-node and check-node operations used to combine estimates in LDPC decoding are defined, respectively, by

$$y_1 \circledast y_2 = \text{vnode}(y_1, y_2) \triangleq \frac{y_1 y_2}{y_1 y_2 + (1 - y_1)(1 - y_2)},$$

and

$$y_1 \boxtimes y_2 = \text{cnode}(y_1, y_2) \triangleq y_1(1 - y_2) + y_2(1 - y_1).$$

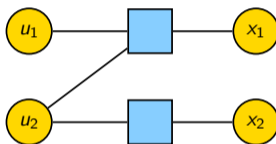
We also need a function that computes hard decisions for P1 variables with randomized rounding in the case of ties:

$$\text{hard_dec_rr}(w) = \begin{cases} \frac{1 + \text{sign}(2w - 1)}{2} & \text{if } w \neq \frac{1}{2} \\ \text{Bernoulli}(\frac{1}{2}) \text{ bit} & \text{if } w = \frac{1}{2}. \end{cases}$$

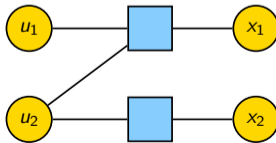
Soft-Decoding of One Layer

Let (U_1, U_2) be uniform bits, $(X_1, X_2) = (U_1 \oplus U_2, U_2)$,
and (y_1, y_2) be a realization of P1 observations of (X_1, X_2) .

Then, $\tilde{u}_1 = y_1 \boxtimes y_2$ is the optimal P1 bit-estimate of U_1 given (Y_1, Y_2) :



And $\tilde{u}_2 = (u_1 \boxtimes y_1) \boxtimes y_2$ is the optimal P1 bit-estimate of U_2 given (Y_1, Y_2, U_1) :



Note: the soft P1 bit-values 0 and 1 naturally include the hard bit-values 0 and 1!

Pseudocode for Recursive Successive Cancellation (SC) Decoding

Algorithm 2 Recursive Implementation of SC Polar Decoder in Python

```
# Recursive Polar SC Decoder (P1 domain)
## y = P1 obs array, f = u prior array
def polar_dec(y,f):
    N = len(y)
    if (N==1):
        x = hard_dec_rr(y)
        if (f[0]==1/2): # If info bit
            return x, x.copy()
        else: # Frozen bit (u,x) = (f,f)
            return x, f.astype(np.int64)

    # Soft mapping back one stage
    u1est = cnop(y[:,2],y[1::2])

    # R_N^T maps u1est to top half
    uhat1,u1hp = polar_dec(u1est,f[::(N//2)])

    # Using u1est and x1hard, estimate u2
    u2est = vnop(cnop(u1hp,y[:,2]),y[1::2])

    # R_N^T maps u2est to bottom half
    uhat2,u2hp = polar_dec(u2est,f[(N//2):])

    # Pass u dec up and interleave x1,x2
    u = np.zeros(N,dtype=np.int64)
    u[::(N//2)] = uhat1
    u[(N//2):] = uhat2
    x1 = cnop(u1hp,u2hp)
    x2 = u2hardprev

    x = np.zeros(N,dtype=np.int64)
    x[:,2] = x1
    x[1::2] = x2
    return u, x
```

A Brief Introduction to Information Theory (1)

Now, we introduce elements of information theory that are important for polar codes.

Definition

The **entropy** (in bits) of a discrete random variable X with pmf $p(x)$ is denoted

$$H(X) \triangleq \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} = \mathbb{E} \left[\log_2 \frac{1}{p(X)} \right],$$

where $0 \log_2 0 = 0$ by continuity. The unit of entropy is determined by the base of the logarithm with base-2 resulting in “bits” and the natural log (i.e., base- e) resulting in “nats”. For binary alphabets, the **binary entropy function** is given by $h(\rho) \triangleq \rho \log_2 \frac{1}{\rho} + (1 - \rho) \log_2 \frac{1}{1 - \rho}$.

A Brief Introduction to Information Theory (1)

Now, we introduce elements of information theory that are important for polar codes.

Definition

The **entropy** (in bits) of a discrete random variable X with pmf $p(x)$ is denoted

$$H(X) \triangleq \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)} = \mathbb{E} \left[\log_2 \frac{1}{p(X)} \right],$$

where $0 \log_2 0 = 0$ by continuity. The unit of entropy is determined by the base of the logarithm with base-2 resulting in “bits” and the natural log (i.e., base- e) resulting in “nats”. For binary alphabets, the **binary entropy function** is given by $h(\rho) \triangleq \rho \log_2 \frac{1}{\rho} + (1 - \rho) \log_2 \frac{1}{1 - \rho}$.

Operational Meaning

The entropy $H(X)$ measures the uncertainty in a random variable X . For compression, it equals the minimum average bit rate (in bits per symbol) to which one can compress long sequences X_1, X_2, \dots drawn i.i.d. according to $p(x)$. It also equals the exponential growth rate, for i.i.d. sequences, of the smallest set that contains almost all the probability.

Definition

The **joint entropy** (in bits) of a pair of r.v. $(X, Y) \sim p_{X,Y}(x, y)$ is denoted

$$H(X, Y) \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p_{XY}(x, y) \log_2 \frac{1}{p_{X,Y}(x, y)} = \mathbb{E} \left[\log_2 \frac{1}{p_{X,Y}(X, Y)} \right].$$

A Brief Introduction to Information Theory (2)

Definition

The **joint entropy** (in bits) of a pair of r.v. $(X, Y) \sim p_{X,Y}(x, y)$ is denoted

$$H(X, Y) \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p_{XY}(x, y) \log_2 \frac{1}{p_{X,Y}(x, y)} = \mathbb{E} \left[\log_2 \frac{1}{p_{X,Y}(X, Y)} \right].$$

Definition

For joint r.v. $(X, Y) \sim p_{X,Y}(x, y)$, the **conditional entropy** of Y given X is denoted

$$H(Y|X) \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x, y) \log_2 \frac{1}{p_{Y|X}(y|x)} = \mathbb{E} \left[\log_2 \frac{1}{p_{Y|X}(Y|X)} \right].$$

A Brief Introduction to Information Theory (2)

Definition

The **joint entropy** (in bits) of a pair of r.v. $(X, Y) \sim p_{X,Y}(x, y)$ is denoted

$$H(X, Y) \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x, y) \log_2 \frac{1}{p_{X,Y}(x, y)} = \mathbb{E} \left[\log_2 \frac{1}{p_{X,Y}(X, Y)} \right].$$

Definition

For joint r.v. $(X, Y) \sim p_{X,Y}(x, y)$, the **conditional entropy** of Y given X is denoted

$$H(Y|X) \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x, y) \log_2 \frac{1}{p_{Y|X}(y|x)} = \mathbb{E} \left[\log_2 \frac{1}{p_{Y|X}(Y|X)} \right].$$

Definition

For joint r.v. $(X, Y) \sim p_{X,Y}(x, y)$, the **mutual information** between X and Y is denoted

$$I(X; Y) \triangleq \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p_{X,Y}(x, y) \log_2 \frac{p_{X,Y}(x, y)}{p_X(x)p_Y(y)} = \mathbb{E} \left[\log_2 \frac{p_{X,Y}(X, Y)}{p_X(X)p_Y(Y)} \right].$$

Lemma

- ① (chain rule) $H(X, Y) = H(X) + H(Y|X)$. If X, Y are indep., $H(X, Y) = H(X) + H(Y)$.

Proof: Take the expectation of $\log_2 \frac{1}{p_{X,Y}(X,Y)} = \log_2 \frac{1}{p_X(X)} + \log_2 \frac{1}{p_{Y|X}(Y|X)}$ and note that $P_{Y|X}(y|x) = p_Y(y)$ for all x, y if X and Y are independent.

- ② (mutual information) The mutual information satisfies $I(X; Y) = I(Y; X)$ and

$$I(X; Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

Proof: Take expectation of $\log_2 \frac{p_{X,Y}(X,Y)}{p_X(X)p_Y(Y)} = \log_2 \frac{1}{p_X(X)} + \log_2 \frac{1}{p_Y(Y)} - \log_2 \frac{1}{p_{X,Y}(X,Y)}$.

- ③ Information preserved by one-to-one mapping: $I(U; Y) = I(f(U), Y)$ if f is one-to-one.

A Brief Introduction to Information Theory (3)

Lemma

- ① (chain rule) $H(X, Y) = H(X) + H(Y|X)$. If X, Y are indep., $H(X, Y) = H(X) + H(Y)$.

Proof: Take the expectation of $\log_2 \frac{1}{p_{X,Y}(X,Y)} = \log_2 \frac{1}{p_X(X)} + \log_2 \frac{1}{p_{Y|X}(Y|X)}$ and note that $P_{Y|X}(y|x) = p_Y(y)$ for all x, y if X and Y are independent.

- ② (mutual information) The mutual information satisfies $I(X; Y) = I(Y; X)$ and

$$I(X; Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

Proof: Take expectation of $\log_2 \frac{p_{X,Y}(X,Y)}{p_X(X)p_Y(Y)} = \log_2 \frac{1}{p_X(X)} + \log_2 \frac{1}{p_Y(Y)} - \log_2 \frac{1}{p_{X,Y}(X,Y)}$.

- ③ Information preserved by one-to-one mapping: $I(U; Y) = I(f(U), Y)$ if f is one-to-one.

Example

Let $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and $p_{X,Y}(x, y) = \rho/2$ if $x \neq y$ and $p_{X,Y}(x, y) = (1 - \rho)/2$ if $x = y$. From $p_X(x) = p_Y(y) = \frac{1}{2}$, we see $H(X) = 1$ and $H(Y) = 1$. From $p_{Y|X}(y|x) \in \{\rho, 1 - \rho\}$, we see that $H(Y|X) = h(\rho)$. Thus, we have $I(X; Y) = H(Y) - H(Y|X) = 1 - h(\rho)$. The conditional $p_{Y|X}$ is called the **binary symmetric channel** with error rate ρ and denoted by $\text{BSC}(\rho)$.

Theorem (Channel Coding Theorem)

For a discrete memoryless channel (DMC) $W(y|x)$, let the channel capacity be defined by

$$C \triangleq \max_{p(x)} I(X; Y) = \max_{p(x)} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x) W(y|x) \log_2 \frac{W(y|x)}{\sum_{x'} p(x') W(y|x')}.$$

For any $R \leq C$, there is a sequence of encoder/decoder pairs with increasing block length where the code rate $R_N \rightarrow R$ and the block error rate converges to 0. Conversely, if a sequence of encoder/decoder pairs has $R_N \rightarrow R$ and block error rate converging to 0, then $R \leq C$.

Theorem (Channel Coding Theorem)

For a discrete memoryless channel (DMC) $W(y|x)$, let the channel capacity be defined by

$$C \triangleq \max_{p(x)} I(X; Y) = \max_{p(x)} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x) W(y|x) \log_2 \frac{W(y|x)}{\sum_{x'} p(x') W(y|x')}.$$

For any $R \leq C$, there is a sequence of encoder/decoder pairs with increasing block length where the code rate $R_N \rightarrow R$ and the block error rate converges to 0. Conversely, if a sequence of encoder/decoder pairs has $R_N \rightarrow R$ and block error rate converging to 0, then $R \leq C$.

Definition

A DMC with binary inputs is called **symmetric** (or binary memoryless symmetric (BMS)) if there is a permutation $\pi : \mathcal{Y} \rightarrow \mathcal{Y}$ satisfying $W(\pi(y)|1) = W(y|0)$ and $\pi(\pi(y)) = y$ for all $y \in \mathcal{Y}$.

Theorem (Channel Coding Theorem)

For a discrete memoryless channel (DMC) $W(y|x)$, let the channel capacity be defined by

$$C \triangleq \max_{p(x)} I(X; Y) = \max_{p(x)} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x) W(y|x) \log_2 \frac{W(y|x)}{\sum_{x'} p(x') W(y|x')}.$$

For any $R \leq C$, there is a sequence of encoder/decoder pairs with increasing block length where the code rate $R_N \rightarrow R$ and the block error rate converges to 0. Conversely, if a sequence of encoder/decoder pairs has $R_N \rightarrow R$ and block error rate converging to 0, then $R \leq C$.

Definition

A DMC with binary inputs is called **symmetric** (or binary memoryless symmetric (BMS)) if there is a permutation $\pi : \mathcal{Y} \rightarrow \mathcal{Y}$ satisfying $W(\pi(y)|1) = W(y|0)$ and $\pi(\pi(y)) = y$ for all $y \in \mathcal{Y}$.

For polar codes, the mapping from (U_1, U_2) to (X_1, X_2) is invertible. Thus, one finds that

$$I(U_1, U_2; Y_1, Y_2) = I(X_1, X_2; Y_1, Y_2) = I(X_1; Y_1) + I(X_2; Y_2) = 2I(X_1, Y_1).$$

For a BMS channel W , the capacity equals $C = I(W) \triangleq I(X_1; Y_1)$ with X_1 uniform.

A Brief Introduction to Information Theory (5)

Lemma

The chain rule of mutual information states that $I(X; Y, Z) = I(X; Y) + I(X; Z|Y)$.

Proof.

This follows from the expectation of the decomposition

$$\begin{aligned}\log_2 \frac{p_{X,Y,Z}(X, Y, Z)}{p_X(X)p_{Y,Z}(Y, Z)} &= \log_2 \frac{p_{X,Y}(X, Y)p_{Z|X,Y}(Z|X, Y)}{p_X(X)p_Y(Y)p_{Z|Y}(Z|Y)} \\ &= \log_2 \frac{p_{X,Y}(X, Y)}{p_X(X)p_Y(Y)} + \log_2 \frac{p_{X,Z|Y}(X, Z|Y)}{p_{Z|Y}(Z|Y)p_{X|Y}(X|Y)}.\end{aligned}$$

□

For polar codes, the chain rule shows that SC decoding preserves information:

$$I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2|U_1) = 2I(W).$$

One can achieve rate $2I(W)$ in two steps: First, U_1 's are sent through $W^- : U_1 \rightarrow (Y_1, Y_2)$ and the coding theorem says decoding possible if rate less than $I(U_1; Y_1, Y_2)$. Since all U_1 's are given by decoding the first stage, one also observes U_2 's through $W^+ : U_2 \rightarrow (Y_1, Y_2, U_1)$. For this channel, coding allows one to achieve any rate up to $I(U_2; Y_1, Y_2|U_1)$.

A Brief Introduction to Information Theory (5)

Lemma

The chain rule of mutual information states that $I(X; Y, Z) = I(X; Y) + I(X; Z|Y)$.

Proof.

This follows from the expectation of the decomposition

$$\begin{aligned}\log_2 \frac{p_{X,Y,Z}(X, Y, Z)}{p_X(X)p_{Y,Z}(Y, Z)} &= \log_2 \frac{p_{X,Y}(X, Y)p_{Z|X,Y}(Z|X, Y)}{p_X(X)p_Y(Y)p_{Z|Y}(Z|Y)} \\ &= \log_2 \frac{p_{X,Y}(X, Y)}{p_X(X)p_Y(Y)} + \log_2 \frac{p_{X,Z|Y}(X, Z|Y)}{p_{Z|Y}(Z|Y)p_{X|Y}(X|Y)}.\end{aligned}$$

□

For polar codes, the chain rule shows that SC decoding preserves information:

$$I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2|U_1) = 2I(W).$$

One can achieve rate $2I(W)$ in two steps: First, U_1 's are sent through $W^- : U_1 \rightarrow (Y_1, Y_2)$ and the coding theorem says decoding possible if rate less than $I(U_1; Y_1, Y_2)$. Since all U_1 's are given by decoding the first stage, one also observes U_2 's through $W^+ : U_2 \rightarrow (Y_1, Y_2, U_1)$. For this channel, coding allows one to achieve any rate up to $I(U_2; Y_1, Y_2|U_1)$.

Successive Cancellation Decoding for a General Transform

- General Setup
 - Suppose one wants to communicate using N uses of a BMS channel

Successive Cancellation Decoding for a General Transform

- General Setup
 - Suppose one wants to communicate using N uses of a BMS channel
 - Input/output vectors are $X^N = (X_1, \dots, X_N)$ and $Y^N = (Y_1, \dots, Y_N)$

Successive Cancellation Decoding for a General Transform

- General Setup

- Suppose one wants to communicate using N uses of a BMS channel
- Input/output vectors are $X^N = (X_1, \dots, X_N)$ and $Y^N = (Y_1, \dots, Y_N)$
- For an $N \times N$ invertible binary matrix G_N , we choose $X^N = U^N G_N$ where $U^N = (U_1, \dots, U_N)$ is a random binary vector. Then,

$$I(X^N; Y^N) = I(U^N; Y^N)$$

Successive Cancellation Decoding for a General Transform

- General Setup

- Suppose one wants to communicate using N uses of a BMS channel
- Input/output vectors are $X^N = (X_1, \dots, X_N)$ and $Y^N = (Y_1, \dots, Y_N)$
- For an $N \times N$ invertible binary matrix G_N , we choose $X^N = U^N G_N$ where $U^N = (U_1, \dots, U_N)$ is a random binary vector. Then,

$$I(X^N; Y^N) = I(U^N; Y^N)$$

- Successive Cancellation (SC) Decoding

- Decode U_1, U_2, \dots, U_N in order where step- i uses optimal decoding assuming all past decisions $\hat{U}_1, \dots, \hat{U}_{i-1}$ are correct. This gives the effective channels

$$W_N^{(i)}(y^N, u^{i-1} | u_i) \triangleq \mathbb{P}(Y^N = y^N, U^{i-1} = u^{i-1} | U_i = u_i)$$

Successive Cancellation Decoding for a General Transform

- General Setup

- Suppose one wants to communicate using N uses of a BMS channel
- Input/output vectors are $X^N = (X_1, \dots, X_N)$ and $Y^N = (Y_1, \dots, Y_N)$
- For an $N \times N$ invertible binary matrix G_N , we choose $X^N = U^N G_N$ where $U^N = (U_1, \dots, U_N)$ is a random binary vector. Then,

$$I(X^N; Y^N) = I(U^N; Y^N)$$

- Successive Cancellation (SC) Decoding

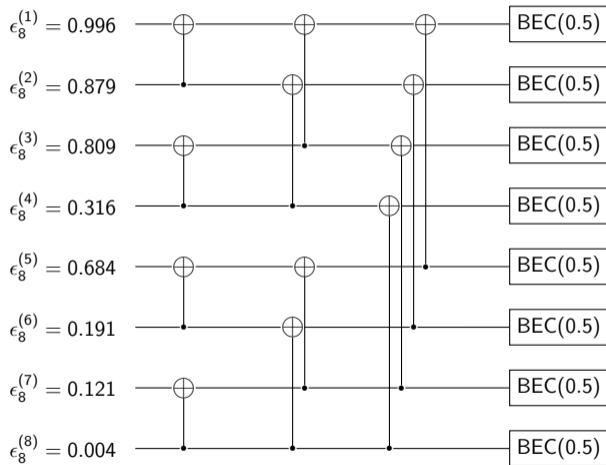
- Decode U_1, U_2, \dots, U_N in order where step- i uses optimal decoding assuming all past decisions $\hat{U}_1, \dots, \hat{U}_{i-1}$ are correct. This gives the effective channels

$$W_N^{(i)}(y^N, u^{i-1} | u_i) \triangleq \mathbb{P}(Y^N = y^N, U^{i-1} = u^{i-1} | U_i = u_i)$$

- Ignoring errors, we can apply the chain rule to see that

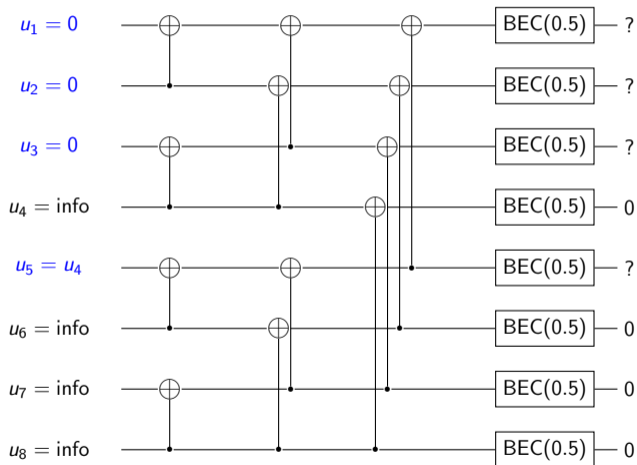
$$I(U^N; Y^N) = \sum_{i=1}^N I(U_i; Y^N | U^{i-1}) = \sum_{i=1}^N I(W_N^{(i)})$$

Polar Code Design (Length-8 Example)



SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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

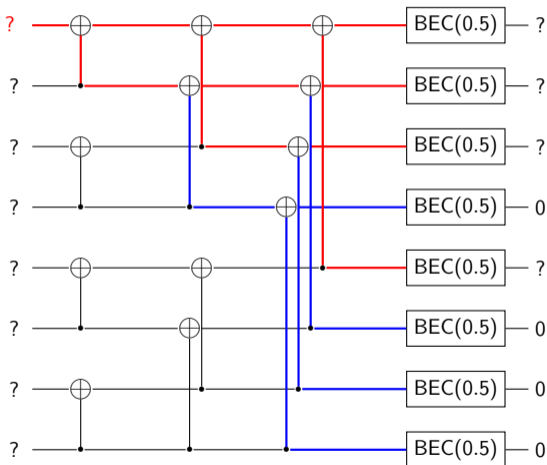


SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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

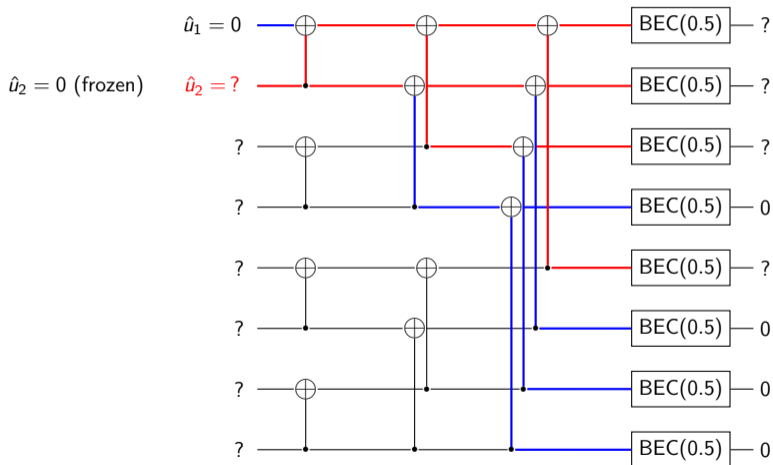
$\hat{u}_1 = 0$ (frozen)

$\hat{u}_1 = ?$



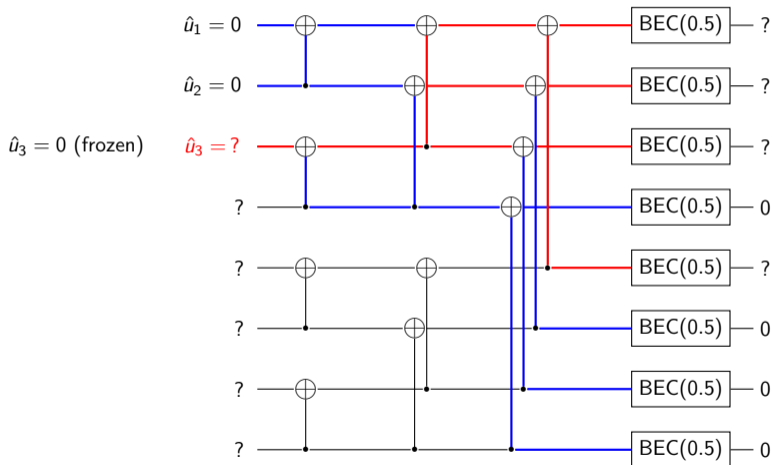
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



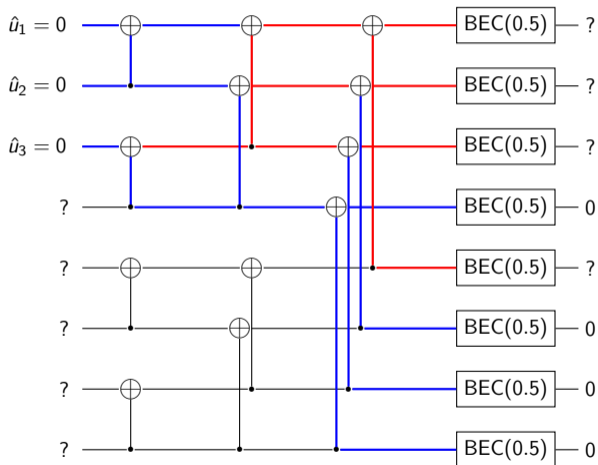
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



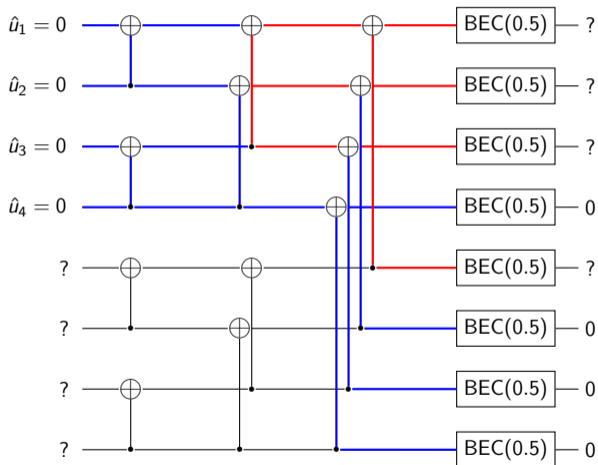
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



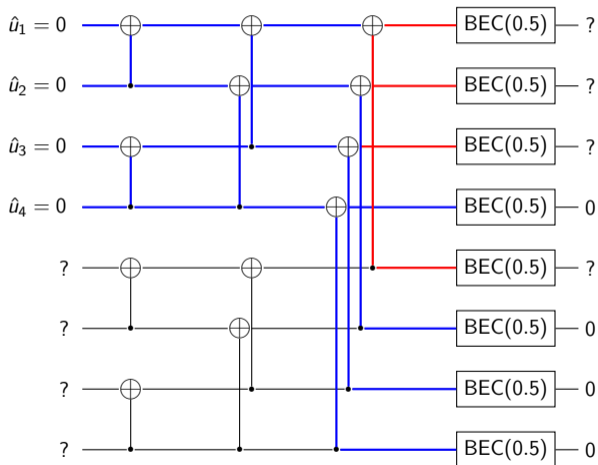
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



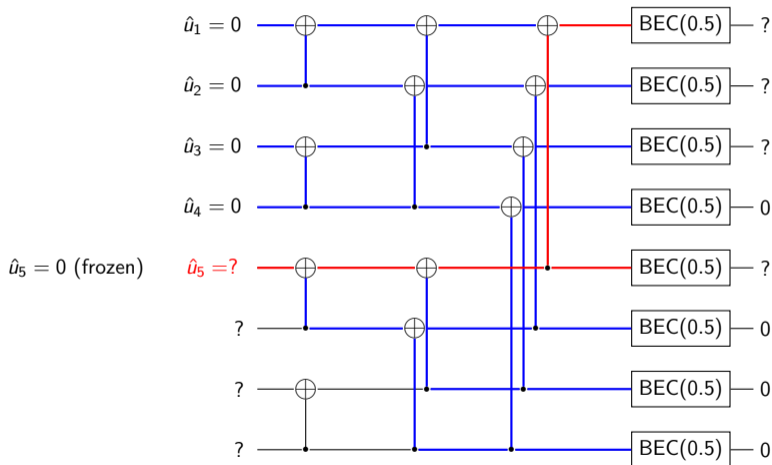
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



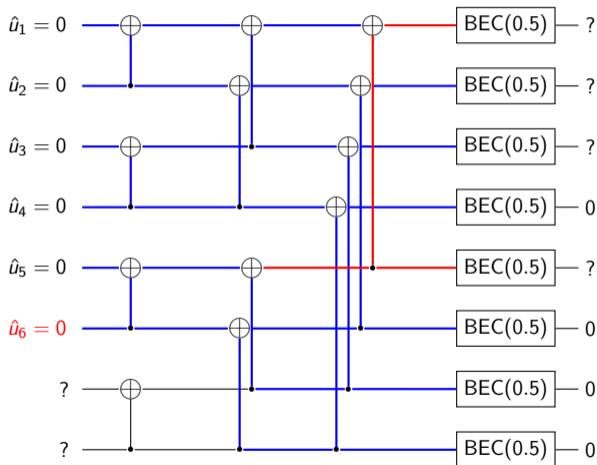
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



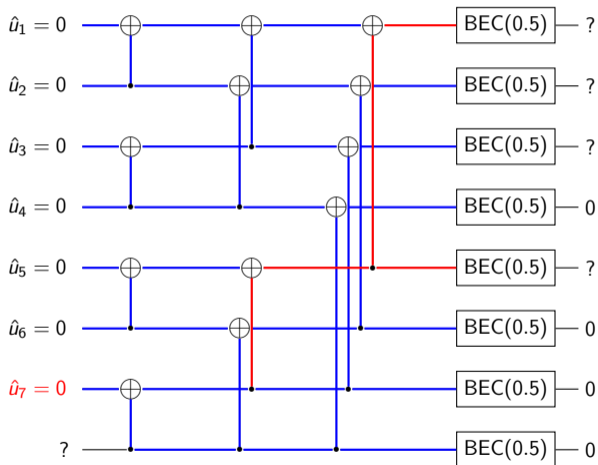
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



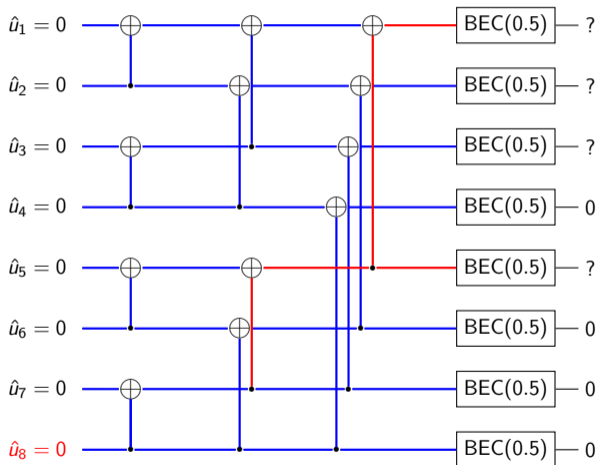
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



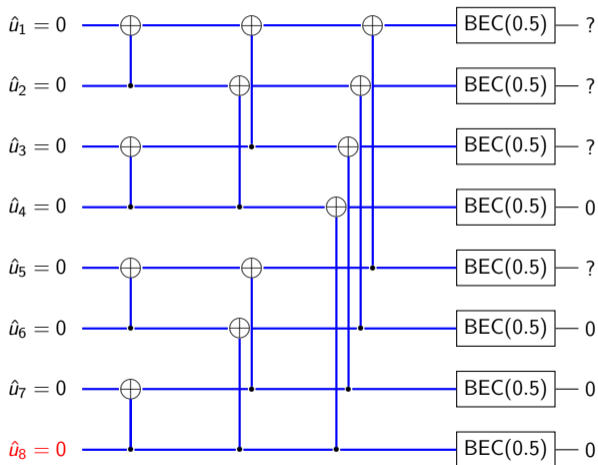
SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



SC Decoding = Sequence of BP Decoders on Trees (Length-8 Example)

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



Design of Polar Codes (1)

Channel combining maps chan W into worse chan $W^- = W \boxtimes W$ and better chan $W^+ = W \boxplus W$
When applied multiple times, the question "How to order them?" is resolved by $W_1^{(1)} = W$ and

$$W_N^{(2i-1)} = W_{N/2}^{(i)-} = W_{N/2}^{(i)} \boxtimes W_{N/2}^{(i)}$$
$$W_N^{(2i)} = W_{N/2}^{(i)+} = W_{N/2}^{(i)} \boxplus W_{N/2}^{(i)}$$

Design of Polar Codes (1)

Channel combining maps chan W into worse chan $W^- = W \boxtimes W$ and better chan $W^+ = W \boxplus W$
When applied multiple times, the question "How to order them?" is resolved by $W_1^{(1)} = W$ and

$$\begin{aligned}W_N^{(2i-1)} &= W_{N/2}^{(i)-} = W_{N/2}^{(i)} \boxtimes W_{N/2}^{(i)} \\W_N^{(2i)} &= W_{N/2}^{(i)+} = W_{N/2}^{(i)} \boxplus W_{N/2}^{(i)}.\end{aligned}$$

At each stage, $N/2$ channels in their current decoding order $W_{N/2}^{(i)}$ are split into $+/-$ channels and the new order is the same except $W_N^{(2i-1)} = W_{N/2}^{(i)-}$ is decoded before $W_N^{(2i)} = W_{N/2}^{(i)+}$.

Design of Polar Codes (1)

Channel combining maps chan W into worse chan $W^- = W \boxtimes W$ and better chan $W^+ = W \boxplus W$
When applied multiple times, the question "How to order them?" is resolved by $W_1^{(1)} = W$ and

$$\begin{aligned}W_N^{(2i-1)} &= W_{N/2}^{(i)-} = W_{N/2}^{(i)} \boxtimes W_{N/2}^{(i)} \\W_N^{(2i)} &= W_{N/2}^{(i)+} = W_{N/2}^{(i)} \boxplus W_{N/2}^{(i)}.\end{aligned}$$

At each stage, $N/2$ channels in their current decoding order $W_{N/2}^{(i)}$ are split into $+/-$ channels and the new order is the same except $W_N^{(2i-1)} = W_{N/2}^{(i)-}$ is decoded before $W_N^{(2i)} = W_{N/2}^{(i)+}$.

For the erasure channel, one can simply track the erasure probability and this gives $\epsilon_1^{(1)} = \epsilon$ and

$$\begin{aligned}\epsilon_N^{(2i-1)} &= 1 - (1 - \epsilon_{N/2}^{(i)})^2 \\ \epsilon_N^{(2i)} &= (\epsilon_{N/2}^{(i)})^2.\end{aligned}$$

Design of Polar Codes (2)

Let \mathcal{A} denote the set of non-frozen indices used to carry information. The successive cancellation (SC) decoder is successful as long as all of the virtual channels in \mathcal{A} is decoded correctly. Let (U_1, \dots, U_N) be the message vector and $(\hat{U}_1, \dots, \hat{U}_N)$ be the vector of hard decisions produced by SC decoder. Define

$$\delta_N^{(i)} = \mathbb{P} \left(\hat{U}_i \neq U_i \mid \hat{U}_1^{i-1} = U_1^{i-1} \right)$$

to be the hard-decision error probability of the virtual channel $W_N^{(i)}$. If we let the r.v.

$$I = \min \left\{ i \in \mathcal{A} \mid \hat{U}_i \neq U_i \right\}$$

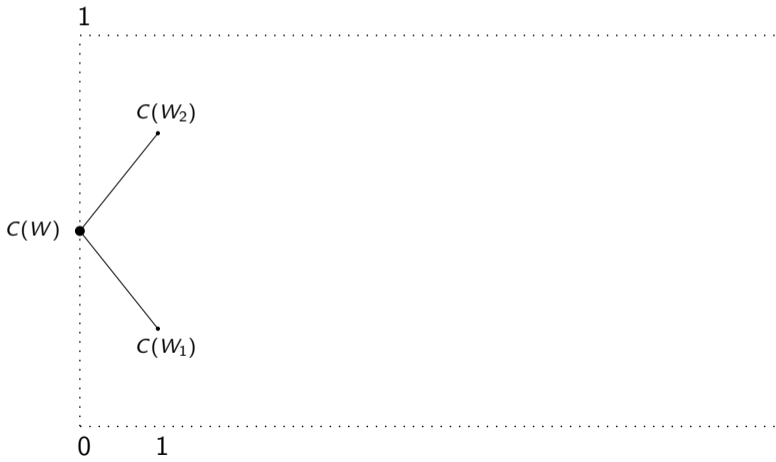
equal the index of the first incorrect information bit, then the prob of block error satisfies

$$\begin{aligned} P_B &= \mathbb{P}(I \in \mathcal{A}) \\ &\leq \sum_{i \in \mathcal{A}} \mathbb{P} \left(\hat{U}_i \neq U_i \mid \hat{U}_1^{i-1} = U_1^{i-1} \right) \\ &\leq \sum_{i \in \mathcal{A}} \delta_N^{(i)}. \end{aligned}$$

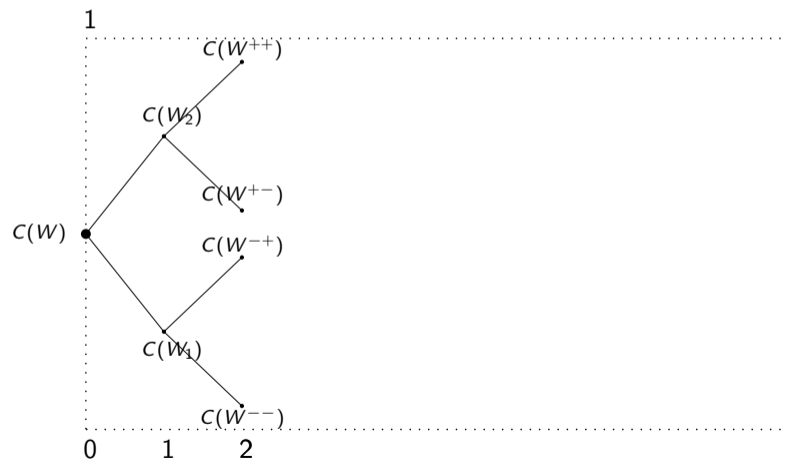
Polarization martingale



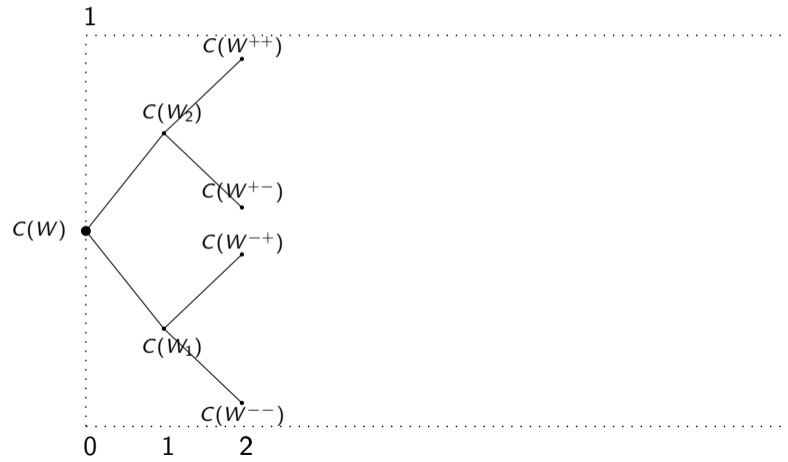
Polarization martingale



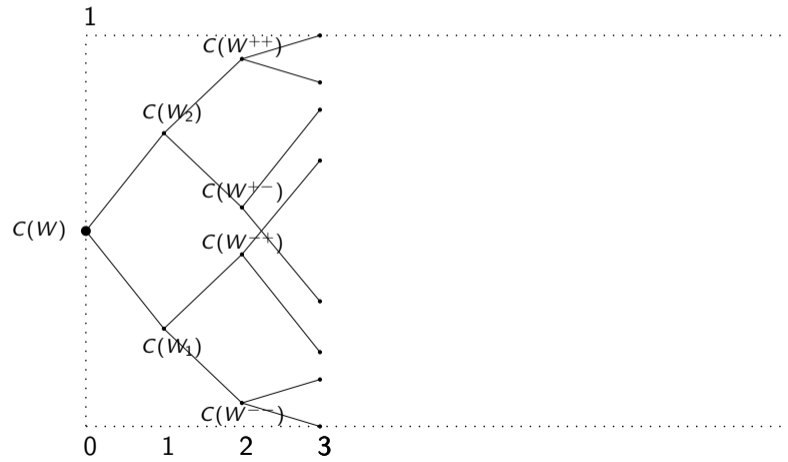
Polarization martingale



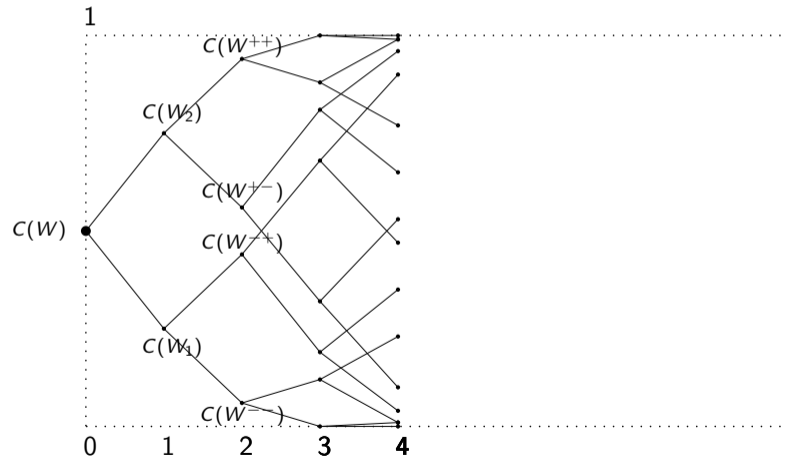
Polarization martingale



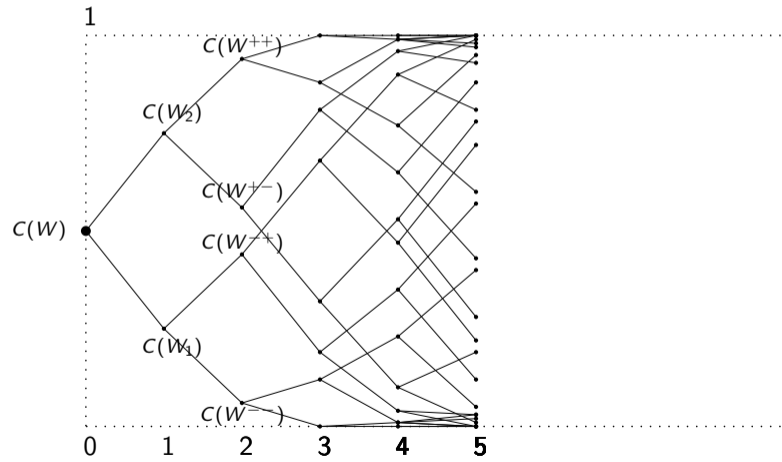
Polarization martingale



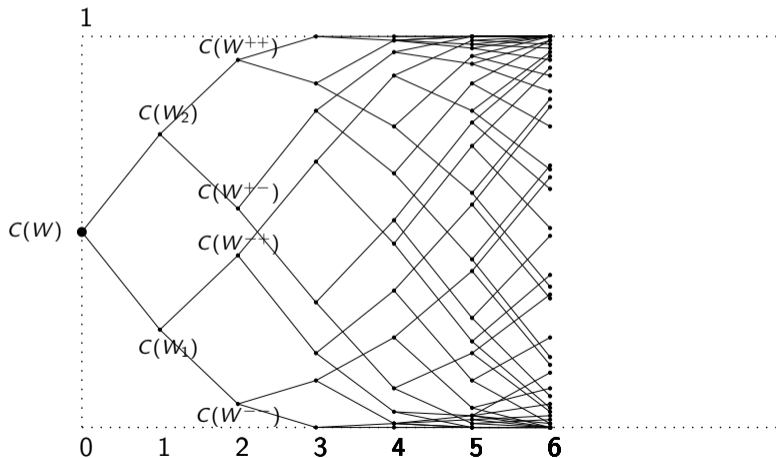
Polarization martingale



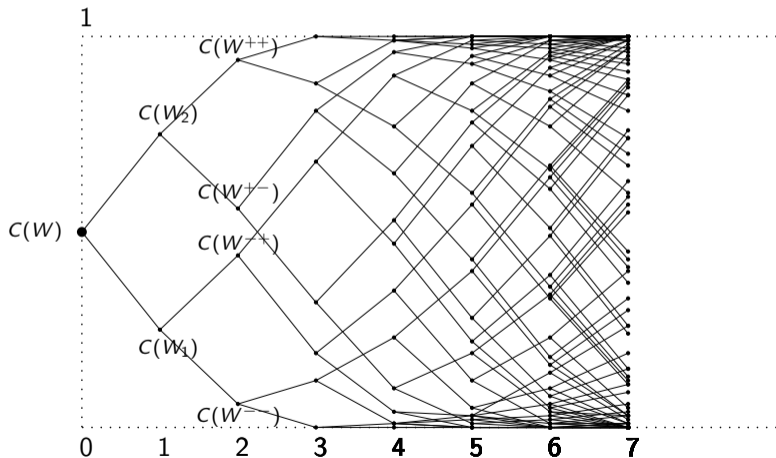
Polarization martingale



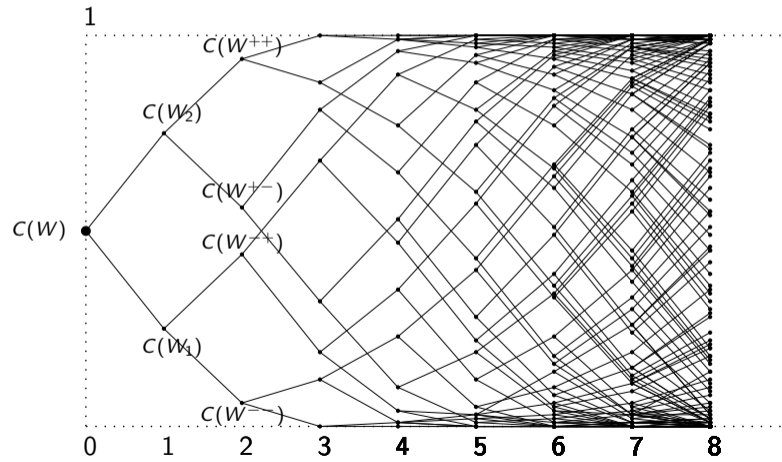
Polarization martingale



Polarization martingale



Polarization martingale



Theorem (Polarization, A. 2007)

The bit-channel capacities $\{C(W_i)\}$ polarize: for any $\delta \in (0, 1)$, as the construction size N grows

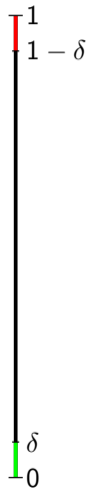
$$\left[\frac{\text{no. channels with } C(W_i) > 1 - \delta}{N} \right] \rightarrow C(W)$$

and

$$\left[\frac{\text{no. channels with } C(W_i) < \delta}{N} \right] \rightarrow 1 - C(W)$$

Theorem (Rate of polarization, A. and Telatar (2008))

Above theorem holds with $\delta \approx 2^{-\sqrt{N}}$.



Theorem (Polarization, A. 2007)

The bit-channel capacities $\{C(W_i)\}$ polarize: for any $\delta \in (0, 1)$, as the construction size N grows

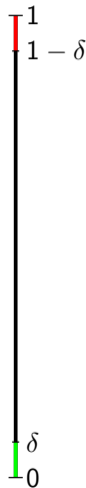
$$\left[\frac{\text{no. channels with } C(W_i) > 1 - \delta}{N} \right] \rightarrow C(W)$$

and

$$\left[\frac{\text{no. channels with } C(W_i) < \delta}{N} \right] \rightarrow 1 - C(W)$$

Theorem (Rate of polarization, A. and Telatar (2008))

Above theorem holds with $\delta \approx 2^{-\sqrt{N}}$.



Proof of Channel Polarization (1)

For any BMS channel W , one stage of polarization preserves mutual information

$$I(W) = \frac{1}{2} (I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2 | U_1)) = \frac{1}{2} (I(W^+) + I(W^-)).$$

Proof of Channel Polarization (1)

For any BMS channel W , one stage of polarization preserves mutual information

$$I(W) = \frac{1}{2} (I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2|U_1)) = \frac{1}{2} (I(W^+) + I(W^-)).$$

Thus, averaging the mutual information over all channels in the n -th stage gives

$$\begin{aligned}\mu_{n+1} &\triangleq \frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} I(W_{2^{n+1}}^{(i)}) \\ &= \frac{1}{2^n} \sum_{i=1}^{2^n} \frac{1}{2} (I(W_{2^{n+1}}^{(2i-1)}) + I(W_{2^{n+1}}^{(2i)})) \\ &= \frac{1}{2^n} \sum_{i=1}^{2^n} \frac{1}{2} (I(W_{2^n}^{(i)-}) + I(W_{2^n}^{(i)+})) \\ &= \frac{1}{2^n} \sum_{i=1}^{2^n} I(W_{2^n}^{(i)}) = \mu_n,\end{aligned}$$

By induction, $\mu_n = \mu_0 = I(W)$.

Proof of Channel Polarization (2)

For any BMS channel W , define $\Delta(W) \triangleq \frac{1}{2} (I(W^+) - I(W^-))$ and observe that

$$I(W)^2 + \Delta(W)^2 = \frac{1}{4} (I(W^+) + I(W^-))^2 + \frac{1}{4} (I(W^+) - I(W^-))^2 = \frac{1}{2} (I(W^+)^2 + I(W^-)^2).$$

Proof of Channel Polarization (2)

For any BMS channel W , define $\Delta(W) \triangleq \frac{1}{2} (I(W^+) - I(W^-))$ and observe that

$$I(W)^2 + \Delta(W)^2 = \frac{1}{4} (I(W^+) + I(W^-))^2 + \frac{1}{4} (I(W^+) - I(W^-))^2 = \frac{1}{2} (I(W^+)^2 + I(W^-)^2).$$

Averaging the square of the mutual information over all channels in the n -th stage gives

$$\begin{aligned}\nu_{n+1} &\triangleq \frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} I(W_{2^{n+1}}^{(i)})^2 \\ &= \frac{1}{2^n} \sum_{i=1}^{2^n} \left(\frac{1}{2} I(W_{2^n}^{(i)+})^2 + \frac{1}{2} I(W_{2^n}^{(i)-})^2 \right) \\ &= \frac{1}{2^n} \sum_{i=1}^{2^n} \left(I(W_{2^n}^{(i)})^2 + \Delta(W_{2^n}^{(i)})^2 \right) \\ &= \nu_n + \frac{1}{2^n} \sum_{i=1}^{2^n} \Delta(W_{2^n}^{(i)})^2\end{aligned}$$

Since $\nu_{n+1} \geq \nu_n$ and $\nu_n \in [0, 1]$, we see ν_n converges and the **last term** vanishes as $n \rightarrow \infty$.

Proof of Channel Polarization (3)

Lemma

For $\delta \in [0, \frac{1}{2}]$ and a BMS channel W with $I(W) \in [\delta, 1 - \delta]$, we have $\Delta(W)^2 \geq \kappa(\delta)$, where

$$\kappa(\delta) \triangleq \min_{h^{-1}(\delta) \leq p \leq h^{-1}(1-\delta)} (h(2p(1-p)) - h(p))^2,$$

$h(p)$ is the binary entropy function, and $\kappa(\delta) > 0$ for $\delta \in (0, \frac{1}{2}]$.

After n steps of polarization, define the fraction of δ -unpolarized channels to be

$$\theta_n(\delta) \triangleq \frac{1}{2^n} \left| \left\{ i \in [2^n] \mid I(W_{2^n}^{(i)}) \in [\delta, 1 - \delta] \right\} \right|.$$

Proof of Channel Polarization (3)

Lemma

For $\delta \in [0, \frac{1}{2}]$ and a BMS channel W with $I(W) \in [\delta, 1 - \delta]$, we have $\Delta(W)^2 \geq \kappa(\delta)$, where

$$\kappa(\delta) \triangleq \min_{h^{-1}(\delta) \leq p \leq h^{-1}(1-\delta)} (h(2p(1-p)) - h(p))^2,$$

$h(p)$ is the binary entropy function, and $\kappa(\delta) > 0$ for $\delta \in (0, \frac{1}{2}]$.

After n steps of polarization, define the fraction of δ -unpolarized channels to be

$$\theta_n(\delta) \triangleq \frac{1}{2^n} \left| \left\{ i \in [2^n] \mid I(W_{2^n}^{(i)}) \in [\delta, 1 - \delta] \right\} \right|.$$

Using the lemma, it follows that

$$\nu_{n+1} - \nu_n = \frac{1}{2^n} \sum_{i=1}^{2^n} \Delta(W_{2^n}^{(i)})^2 \geq \theta_n(\delta) \kappa(\delta).$$

Thus, for any $\delta \in (0, \frac{1}{2}]$, it follows that $\theta_n(\delta) \leq (\nu_{n+1} - \nu_n) / \kappa(\delta) \rightarrow 0$.

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

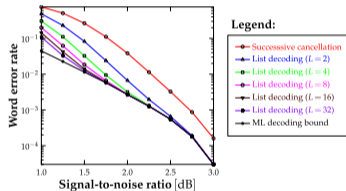


Fig. 1. List-decoding performance for a polar code of length $n = 2048$ and rate $R = 0.5$ on the BPSK-modulated Gaussian channel. The code was constructed using the methods of [15], with optimization for $E_b/N_0 = 2$ dB.

- Successive cancellation (SC) decoding performs poorly for small blocks

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 Arikan. 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

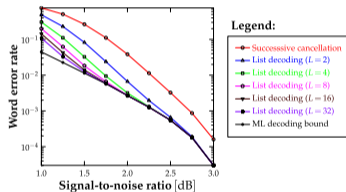


Fig. 1. List-decoding performance for a polar code of length $n = 2048$ and rate $R = 0.5$ on the BPSK-modulated Gaussian channel. The code was constructed using the methods of [15], with optimization for $E_b/N_0 = 2$ dB.

- Successive cancellation (SC) decoding **performs poorly for small blocks**
- But, SC list (SCL) decoding with **CRC and large list-size performs very well** and matches maximum-likelihood (ML) [TV15]

Successive Cancellation List Decoding

- SCL decoding of y^N computes (for $i = 1, 2, \dots$ and $\tilde{u}^i \in \mathcal{U}_i \subseteq \{0, 1\}^i$)

$$Q_i(\tilde{u}^i) \triangleq c_i \cdot \mathbb{P}(U^i = \tilde{u}^i, Y^N = y^N)$$

Successive Cancellation List Decoding

- SCL decoding of y^N computes (for $i = 1, 2, \dots$ and $\tilde{u}^i \in \mathcal{U}_i \subseteq \{0, 1\}^i$)

$$Q_i(\tilde{u}^i) \triangleq c_i \cdot \mathbb{P}(U^i = \tilde{u}^i, Y^N = y^N)$$

- The sets $\mathcal{U}_1, \mathcal{U}_2, \dots$ determine the active paths after each stage
- The constants c_1, c_2, \dots are implicit (unless $\mathcal{U}_i = \{0, 1\}^i$)

Successive Cancellation List Decoding

- SCL decoding of y^N computes (for $i = 1, 2, \dots$ and $\tilde{u}^i \in \mathcal{U}_i \subseteq \{0, 1\}^i$)

$$Q_i(\tilde{u}^i) \triangleq c_i \cdot \mathbb{P}(U^i = \tilde{u}^i, Y^N = y^N)$$

- The sets $\mathcal{U}_1, \mathcal{U}_2, \dots$ determine the active paths after each stage
- The constants c_1, c_2, \dots are implicit (unless $\mathcal{U}_i = \{0, 1\}^i$)
- Recursively compute $Q_i(\tilde{u}^i)$ for $\tilde{u}^{i-1} \in \mathcal{U}_{i-1}$ and $\tilde{u}_i \in \{0, 1\}$

$$Q_i(\tilde{u}^i) = c_i \cdot \mathbb{P}(U^{i-1} = \tilde{u}^{i-1}, Y^N = y^N) \cdot \mathbb{P}(U_i = \tilde{u}_i | Y^N = y^N, U^{i-1} = \tilde{u}^{i-1})$$

Successive Cancellation List Decoding

- SCL decoding of y^N computes (for $i = 1, 2, \dots$ and $\tilde{u}^i \in \mathcal{U}_i \subseteq \{0, 1\}^i$)

$$Q_i(\tilde{u}^i) \triangleq c_i \cdot \mathbb{P}(U^i = \tilde{u}^i, Y^N = y^N)$$

- The sets $\mathcal{U}_1, \mathcal{U}_2, \dots$ determine the active paths after each stage
- The constants c_1, c_2, \dots are implicit (unless $\mathcal{U}_i = \{0, 1\}^i$)
- Recursively compute $Q_i(\tilde{u}^i)$ for $\tilde{u}^{i-1} \in \mathcal{U}_{i-1}$ and $\tilde{u}_i \in \{0, 1\}$

$$\begin{aligned} Q_i(\tilde{u}^i) &= c_i \cdot \mathbb{P}(U^{i-1} = \tilde{u}^{i-1}, Y^N = y^N) \cdot \mathbb{P}(U_i = \tilde{u}_i | Y^N = y^N, U^{i-1} = \tilde{u}^{i-1}) \\ &= c_i \cdot \frac{1}{c_{i-1}} Q_{i-1}(\tilde{u}^{i-1}) \cdot \frac{p(\tilde{u}_i | \tilde{u}^{i-1}) \cdot W_N^{(i)}(y^N, \tilde{u}^{i-1} | \tilde{u}_i)}{p(0 | \tilde{u}^{i-1}) \cdot W_N^{(i)}(y^N, \tilde{u}^{i-1} | 0) + p(1 | \tilde{u}^{i-1}) \cdot W_N^{(i)}(y^N, \tilde{u}^{i-1} | 1)} \end{aligned}$$

Successive Cancellation List Decoding

- SCL decoding of y^N computes (for $i = 1, 2, \dots$ and $\tilde{u}^i \in \mathcal{U}_i \subseteq \{0, 1\}^i$)

$$Q_i(\tilde{u}^i) \triangleq c_i \cdot \mathbb{P}(U^i = \tilde{u}^i, Y^N = y^N)$$

- The sets $\mathcal{U}_1, \mathcal{U}_2, \dots$ determine the active paths after each stage
- The constants c_1, c_2, \dots are implicit (unless $\mathcal{U}_i = \{0, 1\}^i$)
- Recursively compute $Q_i(\tilde{u}^i)$ for $\tilde{u}^{i-1} \in \mathcal{U}_{i-1}$ and $\tilde{u}_i \in \{0, 1\}$

$$\begin{aligned} Q_i(\tilde{u}^i) &= c_i \cdot \mathbb{P}(U^{i-1} = \tilde{u}^{i-1}, Y^N = y^N) \cdot \mathbb{P}(U_i = \tilde{u}_i | Y^N = y^N, U^{i-1} = \tilde{u}^{i-1}) \\ &= c_i \cdot \frac{1}{c_{i-1}} Q_{i-1}(\tilde{u}^{i-1}) \cdot \frac{p(\tilde{u}_i | \tilde{u}^{i-1}) \cdot W_N^{(i)}(y^N, \tilde{u}^{i-1} | \tilde{u}_i)}{p(0 | \tilde{u}^{i-1}) \cdot W_N^{(i)}(y^N, \tilde{u}^{i-1} | 0) + p(1 | \tilde{u}^{i-1}) \cdot W_N^{(i)}(y^N, \tilde{u}^{i-1} | 1)} \end{aligned}$$

- computes $Q_i(\tilde{u}^i)$ values for $\leq 2|\mathcal{U}_{i-1}|$ valid partial input sequences
- $W_N^{(i)}(y^N, \tilde{u}^i | \tilde{u}_{i-1})$ via standard SC decoder, \mathcal{U}_i is path set after pruning

Dynamic Frozen Bits

- A **dynamic frozen bit** is a frozen bit whose value is set to a **linear combination of previous information bits** (rather than a fixed 0 or 1 value) [TM15]

Dynamic Frozen Bits

- A **dynamic frozen bit** is a frozen bit whose value is set to a **linear combination of previous information bits** (rather than a fixed 0 or 1 value) [TM15]
- SC/SCL decoding is easily modified for polar codes with dynamic frozen bits

Dynamic Frozen Bits

- A **dynamic frozen bit** is a frozen bit whose value is set to a **linear combination of previous information bits** (rather than a fixed 0 or 1 value) [TM15]
- SC/SCL decoding is 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!!!
 - Let \mathbf{H} be the parity-check matrix of an (N, K) code and define $\mathbf{x} = \mathbf{u} \mathbf{G}_N$ so

$$\mathbf{0} = \mathbf{H} \mathbf{x}^T = \mathbf{H} \mathbf{G}_N^T \mathbf{u}^T$$

Dynamic Frozen Bits

- A **dynamic frozen bit** is a frozen bit whose value is set to a **linear combination of previous information bits** (rather than a fixed 0 or 1 value) [TM15]
- SC/SCL decoding is 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!!!

- Let \mathbf{H} be the parity-check matrix of an (N, K) code and define $\mathbf{x} = \mathbf{u}\mathbf{G}_N$ so

$$\mathbf{0} = \mathbf{H}\mathbf{x}^T = \mathbf{H}\mathbf{G}_N^T\mathbf{u}^T$$

- Applying row reduction to $\mathbf{H}\mathbf{G}_N^T$ (from the right) gives $\mathbf{V} = \mathbf{Q}\mathbf{H}\mathbf{G}_N^T$ (for invertible \mathbf{Q}) where the last 1 in each row lies in a column with a single 1

Example

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

Dynamic Frozen Bits

- A **dynamic frozen bit** is a frozen bit whose value is set to a **linear combination of previous information bits** (rather than a fixed 0 or 1 value) [TM15]
- SC/SCL decoding is 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!!!

- Let \mathbf{H} be the parity-check matrix of an (N, K) code and define $\mathbf{x} = \mathbf{u}\mathbf{G}_N$ so

$$\mathbf{0} = \mathbf{H}\mathbf{x}^T = \mathbf{H}\mathbf{G}_N^T\mathbf{u}^T$$

- Applying row reduction to $\mathbf{H}\mathbf{G}_N^T$ (from the right) gives $\mathbf{V} = \mathbf{Q}\mathbf{H}\mathbf{G}_N^T$ (for invertible \mathbf{Q}) where the last 1 in each row lies in a column with a single 1
- Let $j_i = \max\{t \in \{1, \dots, N\} | V_{i,t} = 1\}$ be the index of the last 1 in row i . Then, $\mathbf{V}\mathbf{u}^T = \mathbf{0}$ is identical to

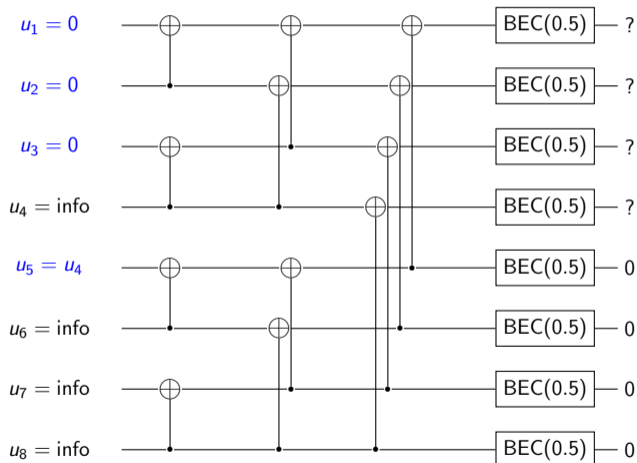
$$u_{j_i} = \sum_{s=0}^{j_i-1} V_{i,s} u_s$$

Example

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

Successive Cancellation List Decoding with Dynamic Frozen Bits

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

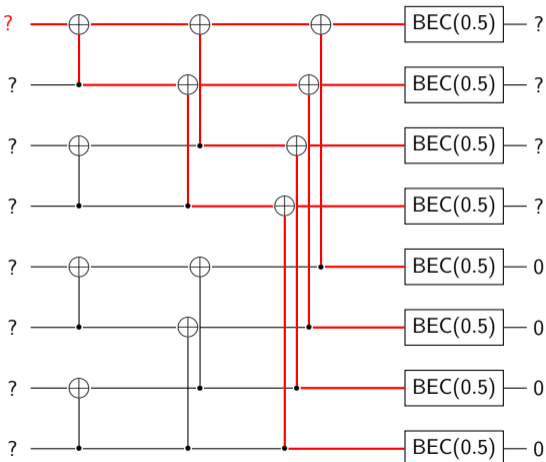


Successive Cancellation List Decoding with Dynamic Frozen Bits

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

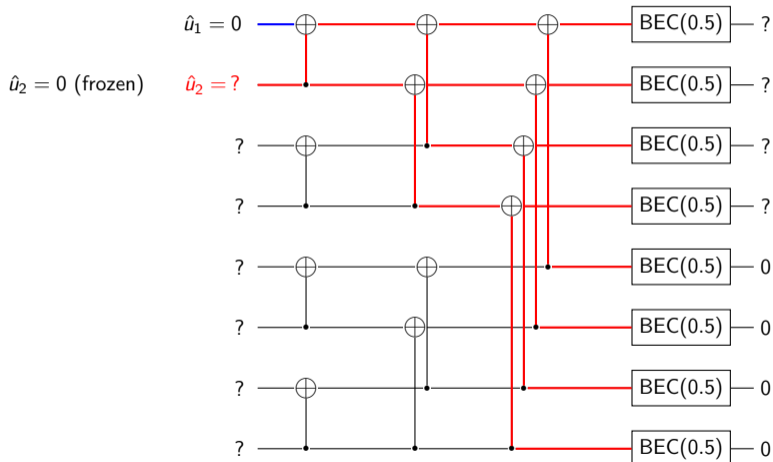
$\hat{u}_1 = 0$ (frozen)

$\hat{u}_1 = ?$



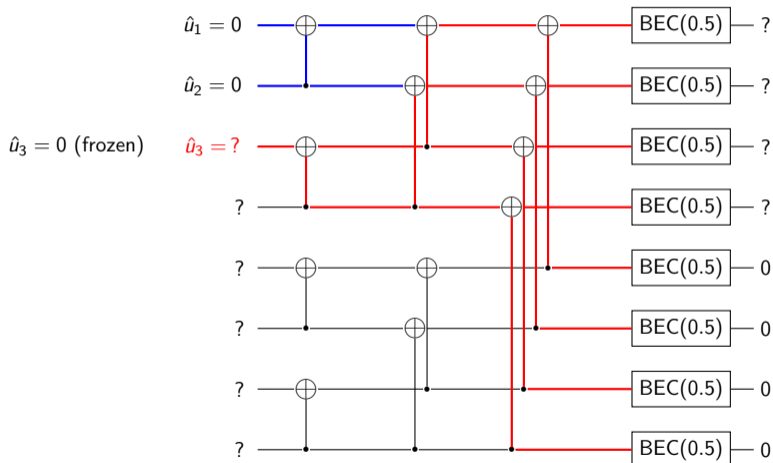
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



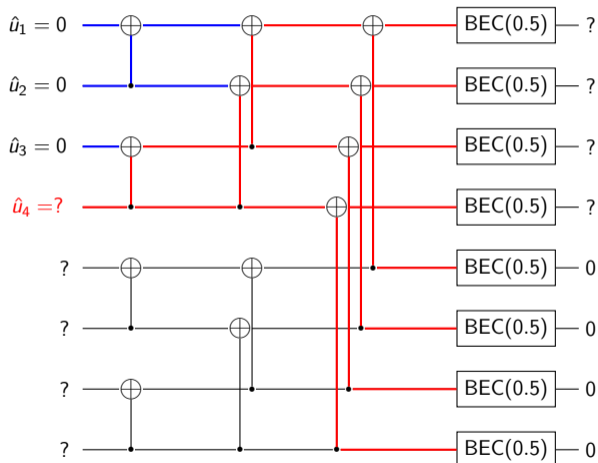
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



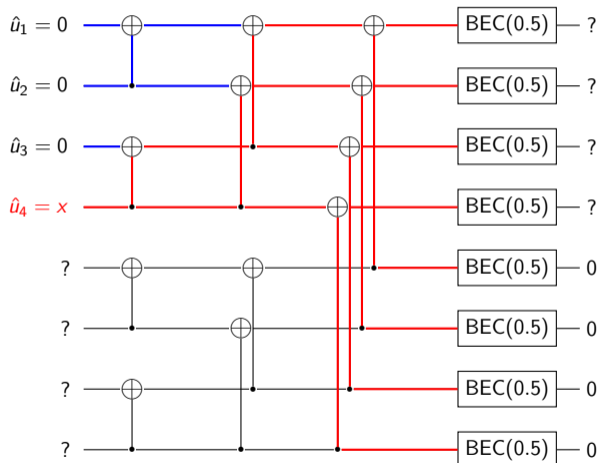
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



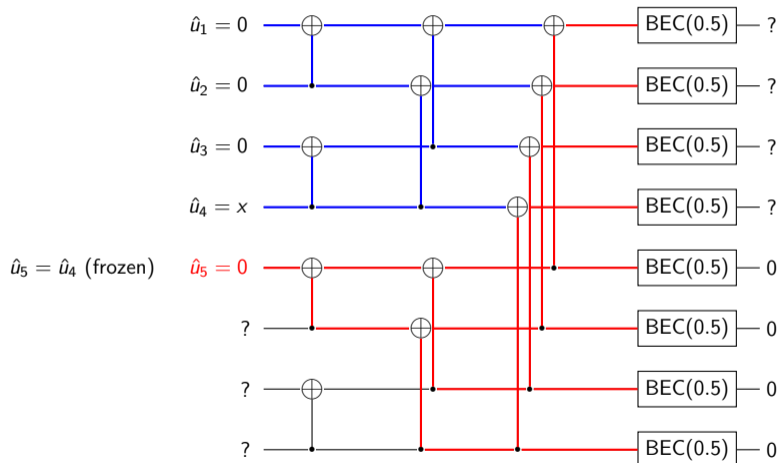
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



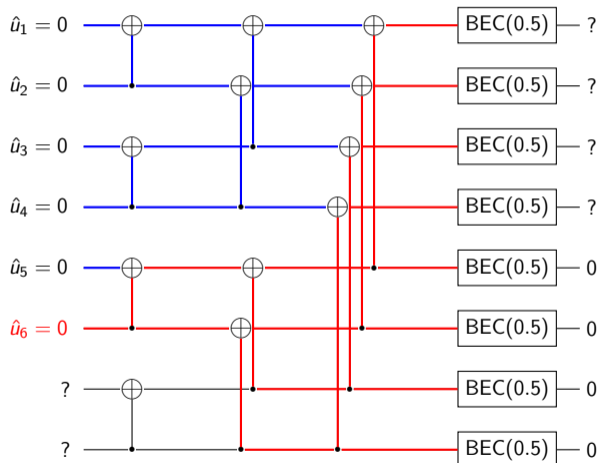
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



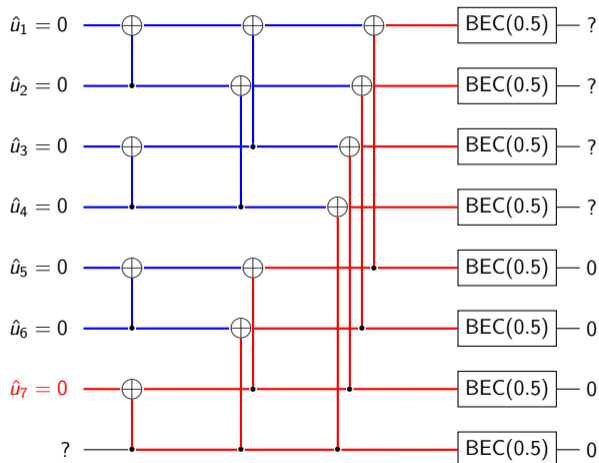
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



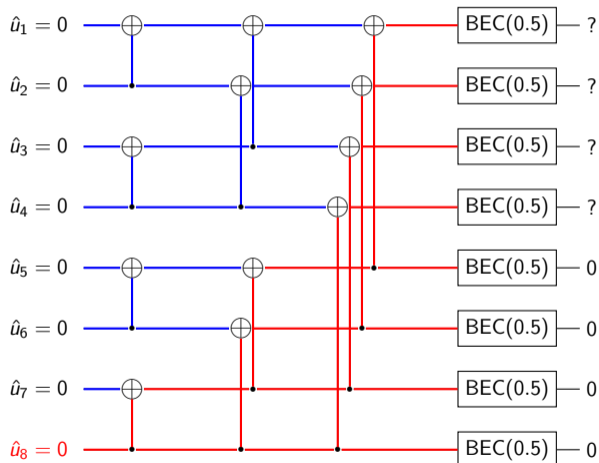
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



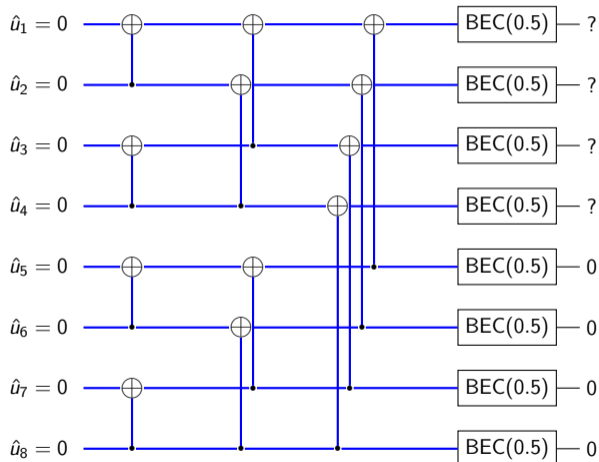
Successive Cancellation List Decoding with Dynamic Frozen Bits

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



Successive Cancellation List Decoding with Dynamic Frozen Bits

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



Polar Codes as Evaluation Codes

- Evaluation Codes

- Codewords formed by evaluating vector space of functions at fixed set of point
- $[n, k, n - k + 1]$ Reed–Solomon codes: Degree- $< k$ polynomials at n distinct points
- Reed–Muller code $RM(r, m)$: Degree- $\leq r$ polynomials in m binary variables at all points

- Kronecker Products

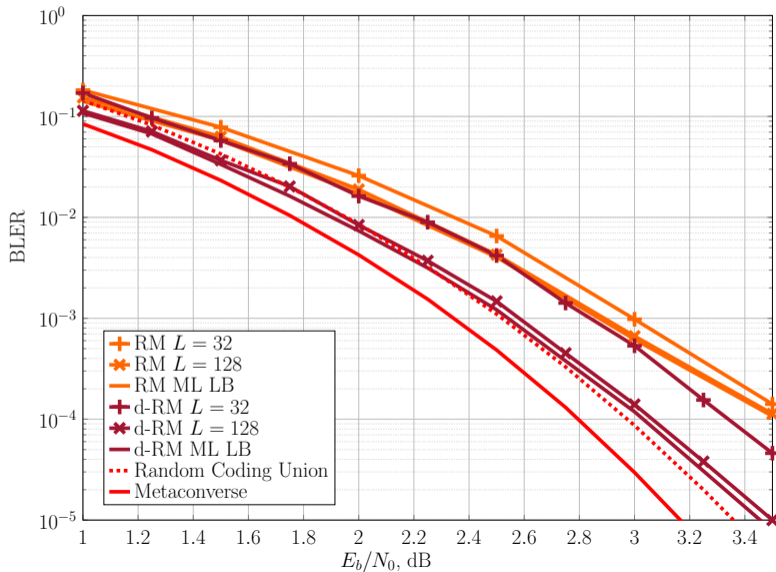
- Let $f_i^{(1)}(x)$ be a basis for a subset of functions mapping \mathcal{X}_1 to \mathcal{X} ($G_{i,j}^{(1)} = f_i^{(1)}(x_j^{(1)})$)
- Let $f_i^{(2)}(x)$ be the basis for functions mapping \mathcal{X}_2 to \mathcal{X} ($G_{i,j}^{(2)} = f_i^{(2)}(x_j^{(2)})$)
- Span of products $f_i^{(1)}(x_j^{(1)})f_{i'}^{(2)}(x_{j'}^{(2)})$ generated by $G^{(1)} \otimes G^{(2)}$

- Binary Monomial Codes

- $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ is the evaluation of $\{1, x\}$ at $\{0, 1\}$
- By induction, $G_2^{\otimes n}$ is the evaluation of all n -variate binary polynomials at all points in $\{0, 1\}^n$

- Reed–Muller Codes
 - Codes by Muller, decoder by Reed, both in 1954
 - Recently shown to be capacity achieving
 - Can be formed as polar code by freezing the right set of bits!

(128, 64) Codes over the AWGN Channel



- [Ar09] E. Arıkan.
Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels.
IEEE Trans. Inform. Theory, 55(7):3051–3073, July 2009.
- [PL17] Noam Presman and Simon Litsyn.
Recursive descriptions of polar codes.
Advances in Mathematics of Communications, 11(1), 2017.
- [TM15] Peter Trifonov and Vera Miloslavskaya.
Polar subcodes.
IEEE J. Select. Areas Commun., 34(2):254–266, 2015.
- [TV15] Ido Tal and Alexander Vardy.
List decoding of polar codes.
IEEE Trans. Inform. Theory, 61(5):2213–2226, 2015.