Polar Coding Tutorial

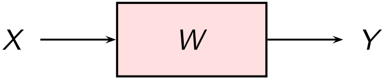
Erdal Arıkan

Electrical-Electronics Engineering Department
Bilkent University
Ankara, Turkey

Jan. 15, 2015
Simons Institute
UC Berkeley

The channel

Let $W : X \to Y$ be a binary-input discrete memoryless channel

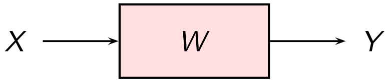


- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:

$$W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y}$$

The channel

Let $W : X \to Y$ be a binary-input discrete memoryless channel

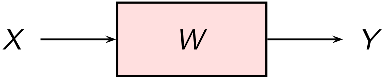


- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:

$$W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y}$$

The channel

Let $W : X \rightarrow Y$ be a binary-input discrete memoryless channel

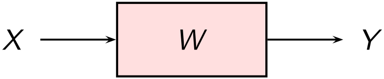


- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:

$$W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y}$$

The channel

Let $W : X \to Y$ be a binary-input discrete memoryless channel



- input alphabet: $\mathcal{X} = \{0, 1\}$,
- output alphabet: $\mathcal{Y}$,
- transition probabilities:

$$W(y|x), \quad x \in \mathcal{X}, y \in \mathcal{Y}$$

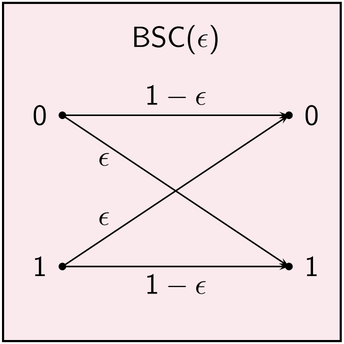
Symmetry assumption

Assume that the channel has "input-output symmetry."

Symmetry assumption

Assume that the channel has "input-output symmetry."

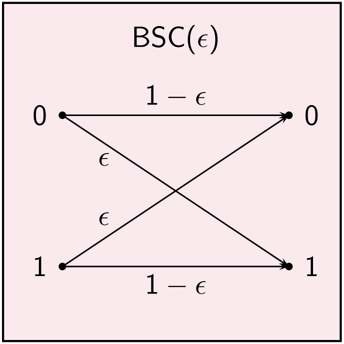
Examples:

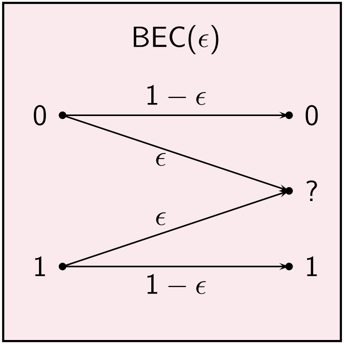


Symmetry assumption

Assume that the channel has "input-output symmetry."

Examples:





Capacity

For channels with input-output symmetry, the capacity is given by

$$C(W) \stackrel{\Delta}{=} I(X;Y), \quad \text{with } X \sim \text{unif. } \{0,1\}$$

Capacity

For channels with input-output symmetry, the capacity is given by

$$C(W) \overset{\Delta}{=} I(X;Y), \quad \text{with } X \sim \text{unif. } \{0,1\}$$

Use base-2 logarithms:

$$0 \le C(W) \le 1$$

The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary W into such extreme channels

The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary W into such extreme channels

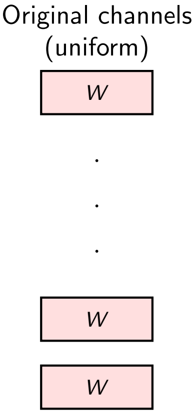
The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary W into such extreme channels

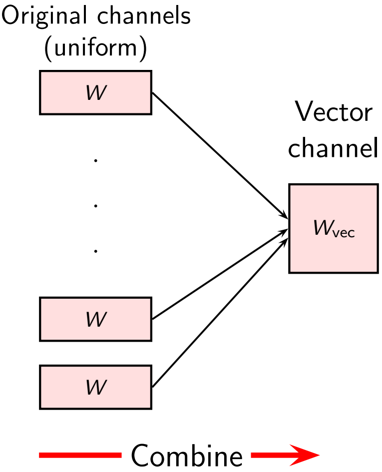
The main idea

- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary W into such extreme channels

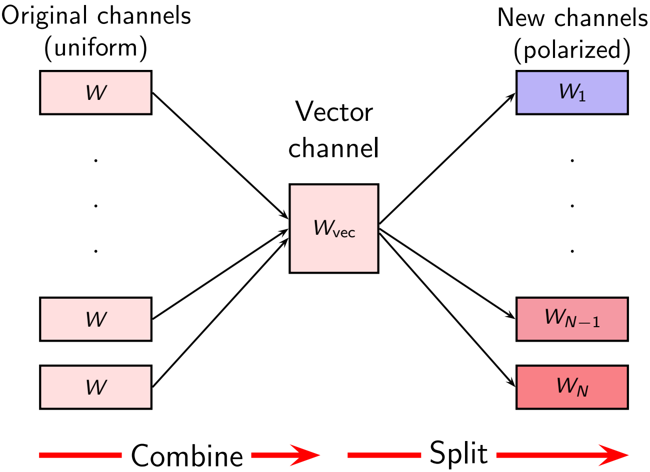
The method: aggregate and redistribute capacity



The method: aggregate and redistribute capacity



The method: aggregate and redistribute capacity



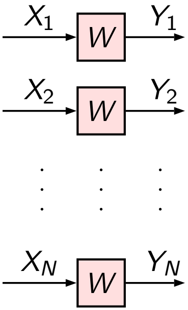
Combining

- Begin with N copies of W,
- use a 1-1 mapping

$$G_N : \{0,1\}^N \to \{0,1\}^N$$

- to create a vector channel

$$W_{\text{vec}} : U^N \to Y^N$$



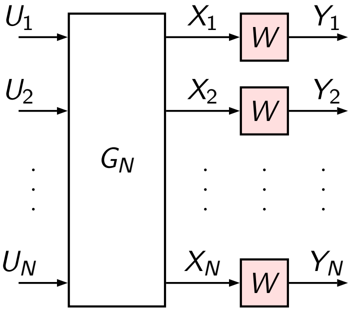
Combining

- Begin with N copies of W,
- use a 1-1 mapping

$$G_N : \{0,1\}^N \to \{0,1\}^N$$

- to create a vector channel

$$W_{\text{vec}} : U^N \to Y^N$$



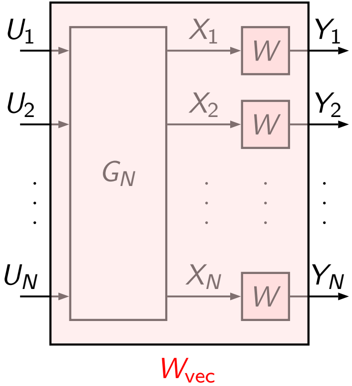
Combining

- Begin with N copies of W,
- use a 1-1 mapping

$$G_N : \{0,1\}^N \to \{0,1\}^N$$

- to create a vector channel

$$W_{\text{vec}} : U^N \to Y^N$$

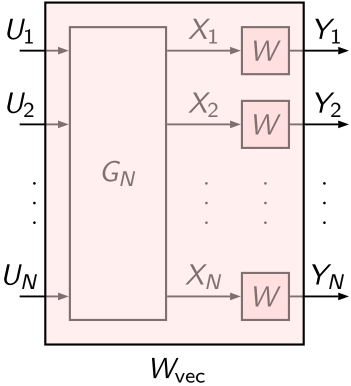


Conservation of capacity

Combining operation is lossless:

- Take $U_1, \ldots, U_N$ i.i.d. unif. $\{0,1\}$
- then, $X_1, \ldots, X_N$ i.i.d. unif. $\{0,1\}$
- and

$$C(W_{\text{vec}}) = I(U^N; Y^N) = I(X^N; Y^N) = NC(W)$$

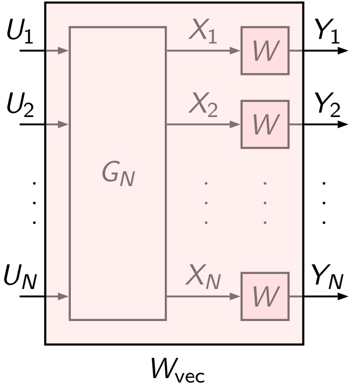


Conservation of capacity

Combining operation is lossless:

- Take $U_1, \ldots, U_N$ i.i.d. unif. $\{0,1\}$
- then, $X_1, \ldots, X_N$ i.i.d. unif. $\{0,1\}$
- and

$$C(W_{\text{vec}}) = I(U^N; Y^N) = I(X^N; Y^N) = NC(W)$$

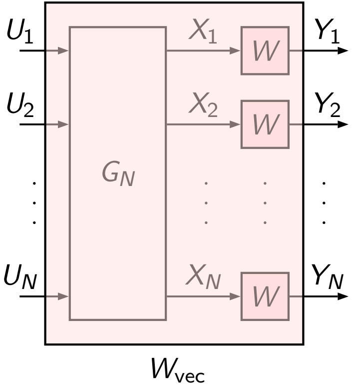


Conservation of capacity

Combining operation is lossless:

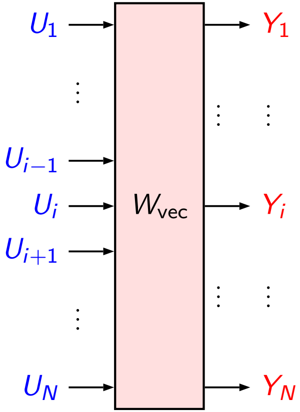
- Take $U_1, \ldots, U_N$ i.i.d. unif. $\{0,1\}$
- then, $X_1, \ldots, X_N$ i.i.d. unif. $\{0,1\}$
- and

$$\begin{aligned} C(W_{\text{vec}}) &= I(U^N; Y^N) \\ &= I(X^N; Y^N) \\ &= NC(W) \end{aligned}$$



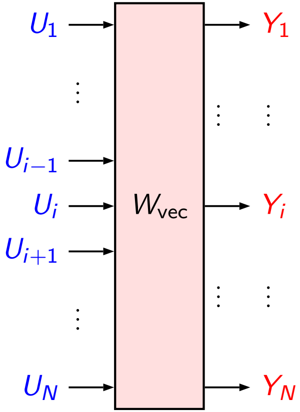
Splitting

$$C(W_{\text{vec}}) = I(U^N; Y^N)$$



Splitting

$$C(W_{\text{vec}}) = I(U^N; Y^N)$$
$$= \sum_{i=1}^{N} I(U_i; Y^N, U^{i-1})$$

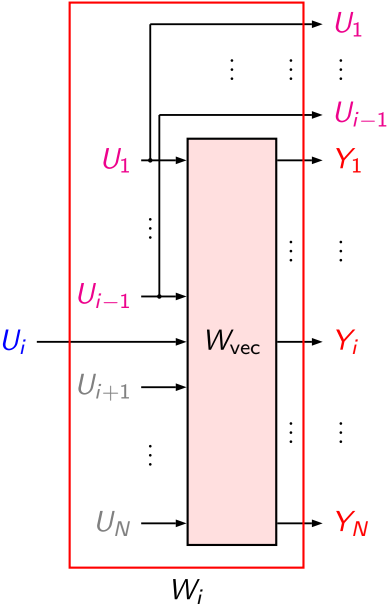


Splitting

$$C(W_{\text{vec}}) = I(U^N; Y^N) = \sum_{i=1}^{N} I(U_i; Y^N, U^{i-1})$$

Define bit-channels

$$W_i : U_i \to (Y^N, U^{i-1})$$

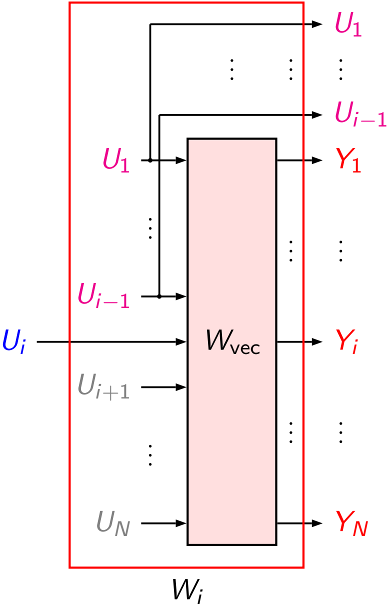


Splitting

$$C(W_{\text{vec}}) = I(U^N; Y^N) = \sum_{i=1}^{N} I(U_i; Y^N, U^{i-1}) = \sum_{i=1}^{N} C(W_i)$$

Define bit-channels

$$W_i : U_i \to (Y^N, U^{i-1})$$

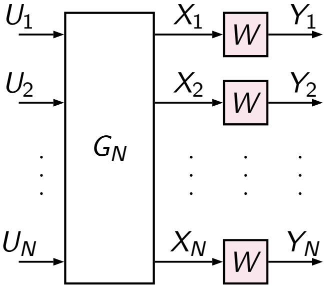


Polarization is commonplace

- Polarization is the rule not the exception
- A random permutation

$$G_N : \{0,1\}^N \to \{0,1\}^N$$

is a good polarizer with high probability
- Equivalent to Shannon's random coding approach

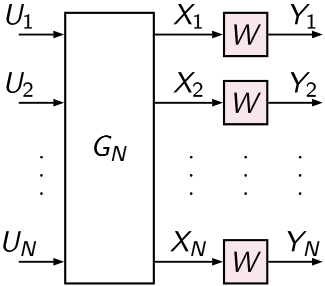


Polarization is commonplace

- Polarization is the rule not the exception
- A random permutation

$$G_N : \{0,1\}^N \to \{0,1\}^N$$

is a good polarizer with high probability
- Equivalent to Shannon's random coding approach



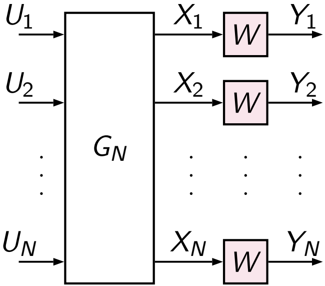
Polarization is commonplace

- Polarization is the rule not the exception
- A random permutation

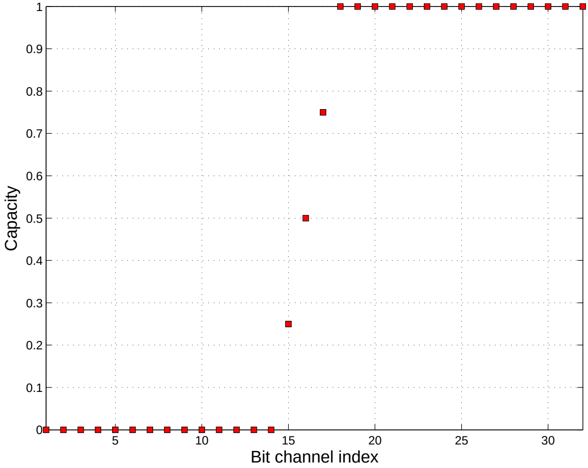
$$G_N : \{0,1\}^N \to \{0,1\}^N$$

is a good polarizer with high probability

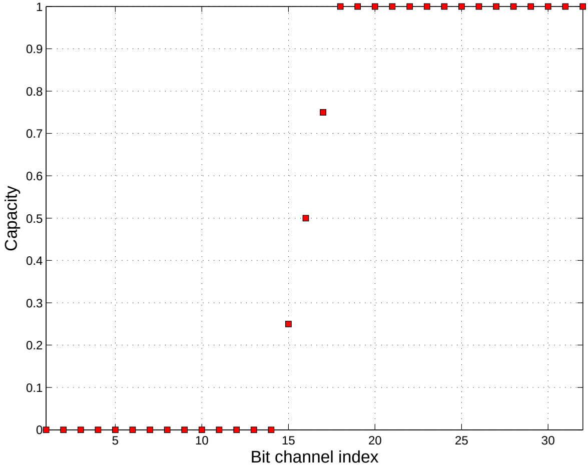
- Equivalent to Shannon's random coding approach



Random polarizers: stepwise, isotropic



Random polarizers: stepwise, isotropic



Isotropy: any redistribution order is as good as any other.

The complexity issue

- Random polarizers lack structure, too complex to implement
- Need a low-complexity polarizer
- May sacrifice stepwise, isotropic properties of random polarizers in return for less complexity

The complexity issue

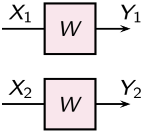
- Random polarizers lack structure, too complex to implement
- Need a low-complexity polarizer
- May sacrifice stepwise, isotropic properties of random polarizers in return for less complexity

The complexity issue

- Random polarizers lack structure, too complex to implement
- Need a low-complexity polarizer
- May sacrifice stepwise, isotropic properties of random polarizers in return for less complexity

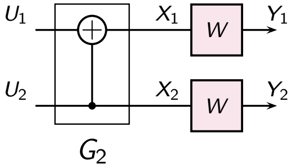
Basic module for a low-complexity scheme

Combine two copies of W



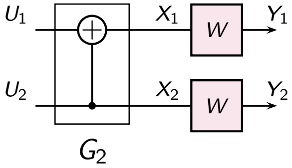
Basic module for a low-complexity scheme

Combine two copies of W



Basic module for a low-complexity scheme

Combine two copies of W

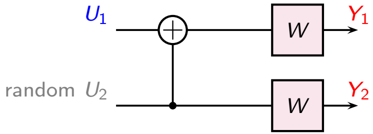


and split to create two bit-channels

$$W_1 : U_1 \to (Y_1, Y_2)$$
$$W_2 : U_2 \to (Y_1, Y_2, U_1)$$

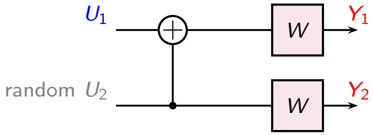
The first bit-channel W_1

$$W_1 : U_1 \rightarrow (Y_1, Y_2)$$



The first bit-channel W_1

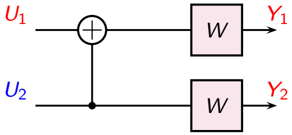
$$W_1 : U_1 \to (Y_1, Y_2)$$



$$C(W_1) = I(U_1; Y_1, Y_2)$$

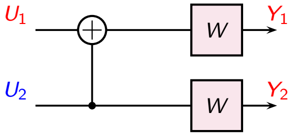
The second bit-channel W_2

$$W_2 : U_2 \to (Y_1, Y_2, U_1)$$



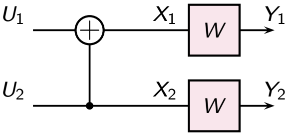
The second bit-channel W_2

$$W_2 : U_2 \to (Y_1, Y_2, U_1)$$



$$C(W_2) = I(U_2; Y_1, Y_2, U_1)$$

Capacity conserved but redistributed unevenly



- Conservation:

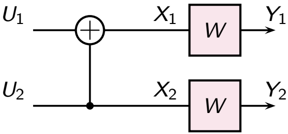
$$C(W_1) + C(W_2) = 2C(W)$$

- Extremization:

$$C(W_1) \leq C(W) \leq C(W_2)$$

with equality iff $C(W)$ equals 0 or 1.

Capacity conserved but redistributed unevenly



- Conservation:

$$C(W_1) + C(W_2) = 2C(W)$$

- Extremization:

$$C(W_1) \leq C(W) \leq C(W_2)$$

with equality iff $C(W)$ equals 0 or 1.

Notation

The two channels created by the basic transform

$$(W, W) \to (W_1, W_2)$$

will be denoted also as

$$W^- = W_1 \quad \text{and} \quad W^+ = W_2$$

Notation

The two channels created by the basic transform

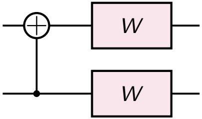
$$(W, W) \to (W_1, W_2)$$

will be denoted also as

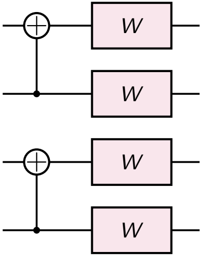
$$W^- = W_1 \quad \text{and} \quad W^+ = W_2$$

Likewise, we write W^{--}, W^{-+} for descendants of W^-; and W^{+-}, W^{++} for descendants of W^+.

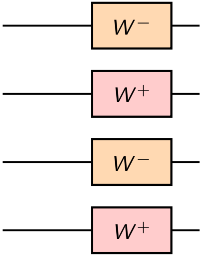
For the size-4 construction



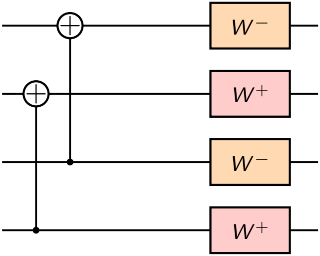
... duplicate the basic transform



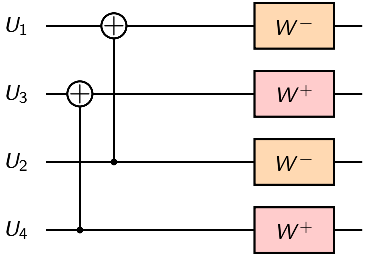
… obtain a pair of W^- and W^+ each



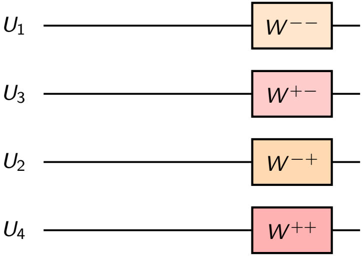
… apply basic transform on each pair



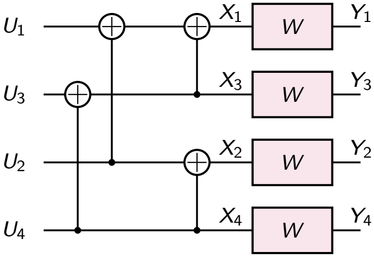
… decode in the indicated order



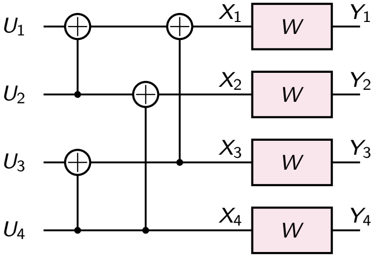
… obtain the four new bit-channels



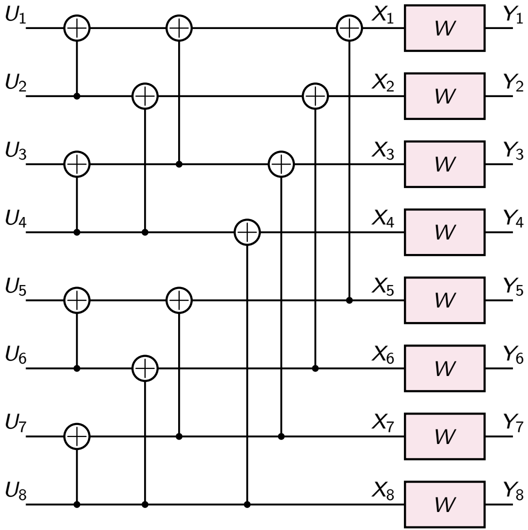
Overall size-4 construction



“Rewire” for standard-form size-4 construction



Size 8 construction



Demonstration of polarization

Polarization is easy to analyze when W is a BEC.

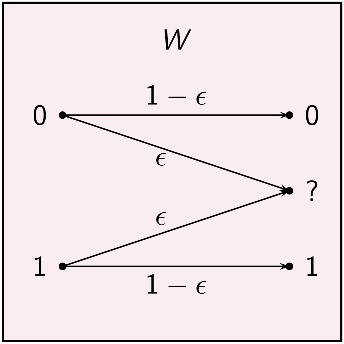
If W is a BEC(ϵ), then so are W^- and W^+, with erasure probabilities

$$\epsilon^- \triangleq 2\epsilon - \epsilon^2$$

and

$$\epsilon^+ \triangleq \epsilon^2$$

respectively.



Demonstration of polarization

Polarization is easy to analyze when W is a BEC.

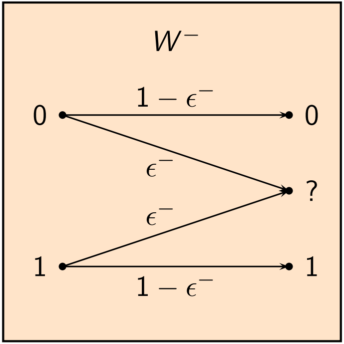
If W is a BEC(ϵ), then so are W^- and W^+, with erasure probabilities

$$\epsilon^- \triangleq 2\epsilon - \epsilon^2$$

and

$$\epsilon^+ \triangleq \epsilon^2$$

respectively.



Demonstration of polarization

Polarization is easy to analyze when W is a BEC.

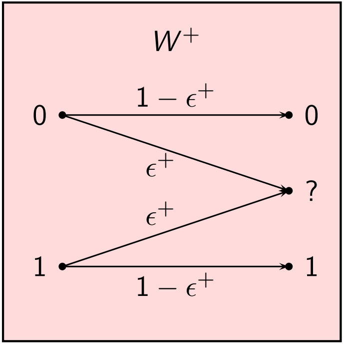
If W is a BEC(ϵ), then so are W^- and W^+, with erasure probabilities

$$\epsilon^- \triangleq 2\epsilon - \epsilon^2$$

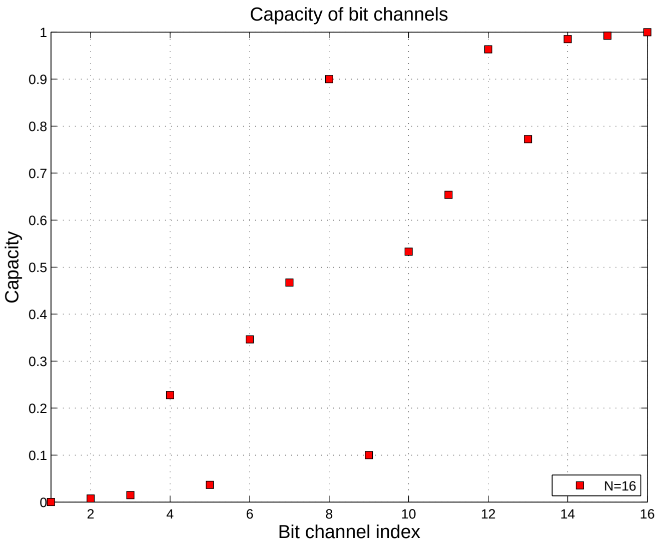
and

$$\epsilon^+ \triangleq \epsilon^2$$

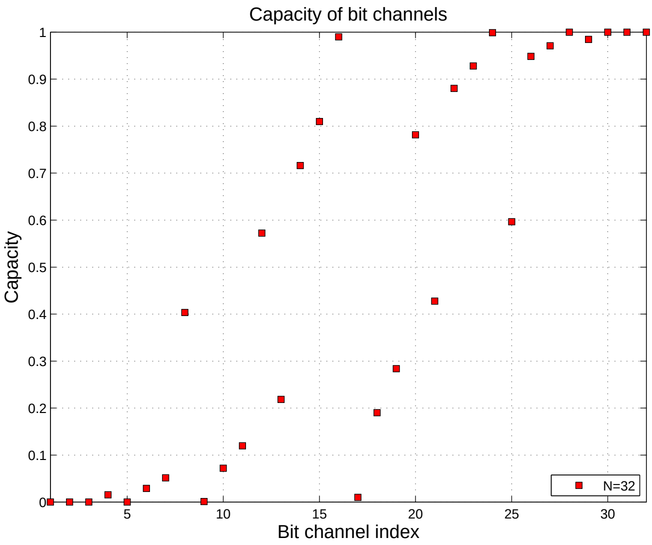
respectively.



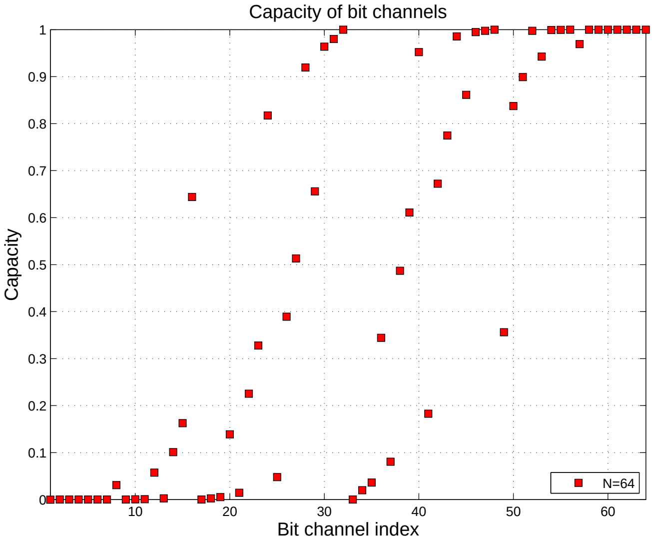
Polarization for BEC($\frac{1}{2}$): $N = 16$



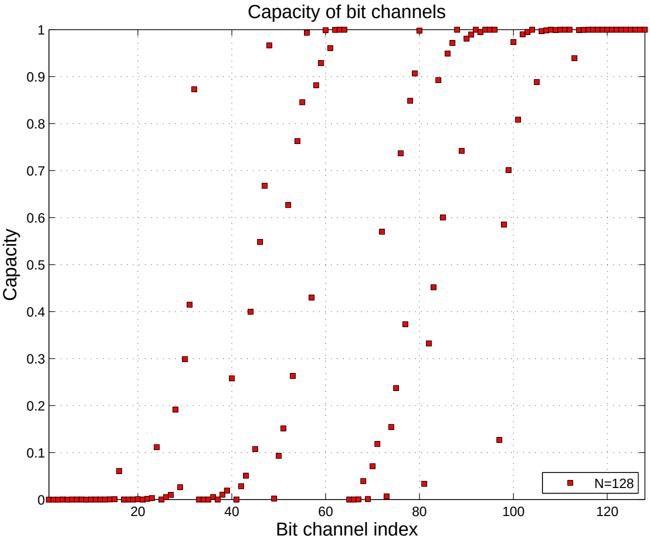
Polarization for BEC($\frac{1}{2}$): $N = 32$



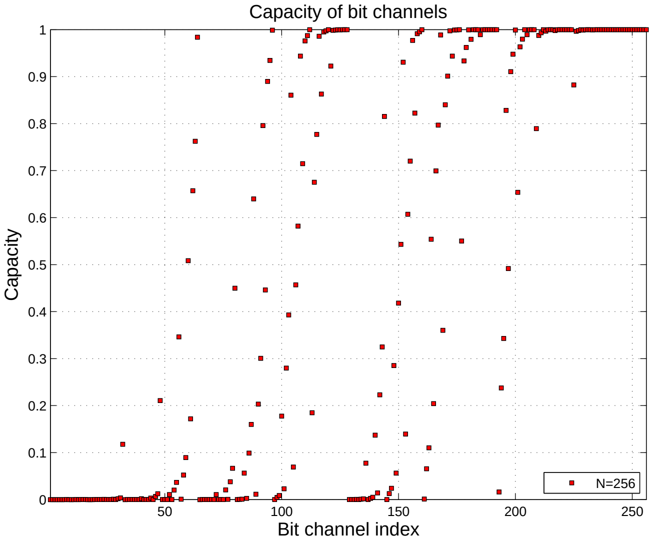
Polarization for BEC($\frac{1}{2}$): $N = 64$



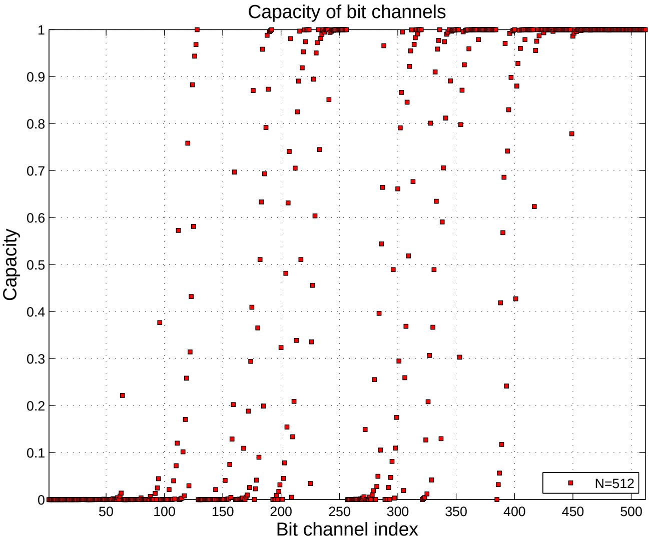
Polarization for BEC($\frac{1}{2}$): $N = 128$



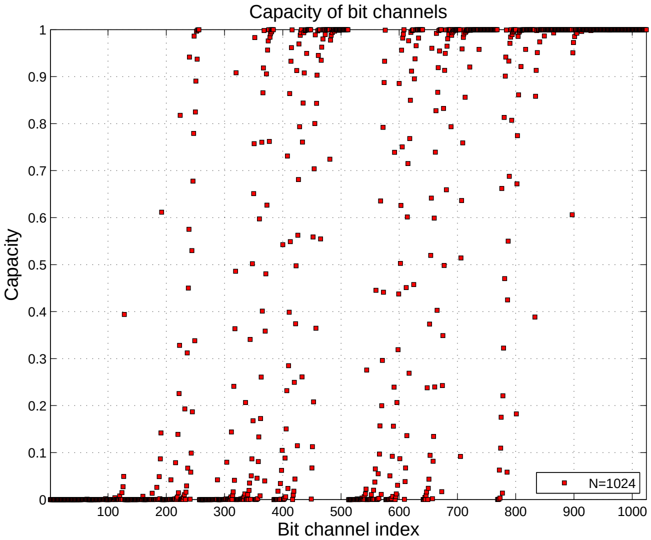
Polarization for BEC($\frac{1}{2}$): $N = 256$



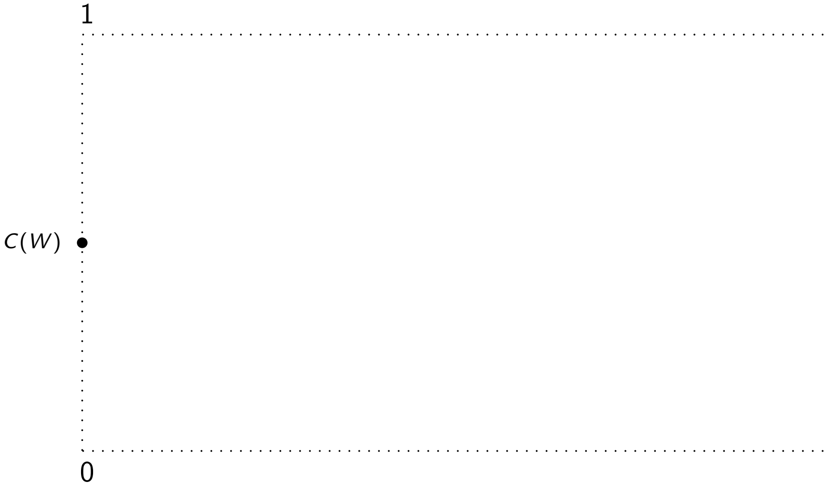
Polarization for BEC($\frac{1}{2}$): $N = 512$



Polarization for BEC($\frac{1}{2}$): $N = 1024$

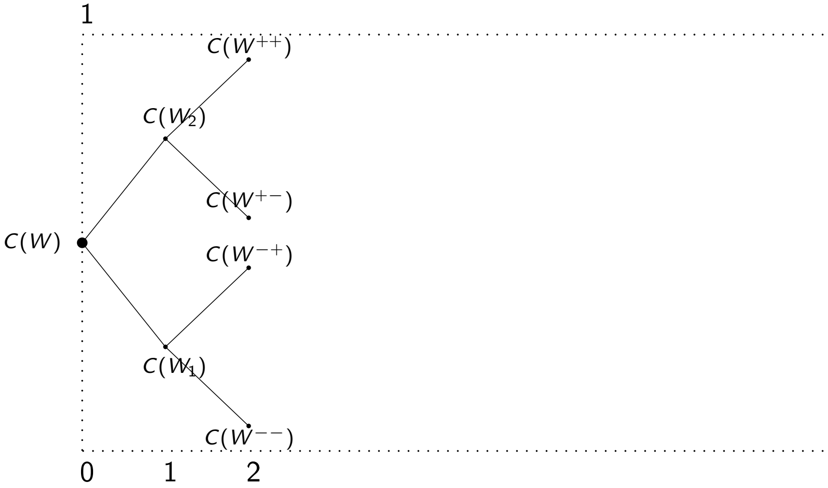


Polarization martingale

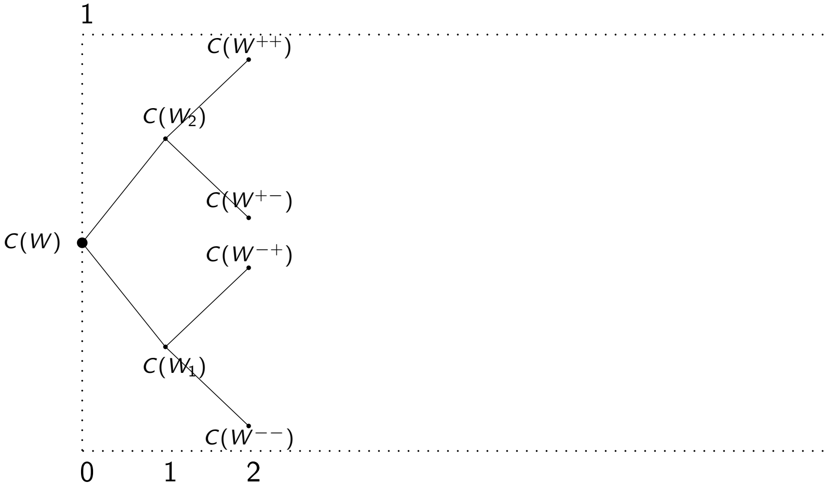


Polarization martingale

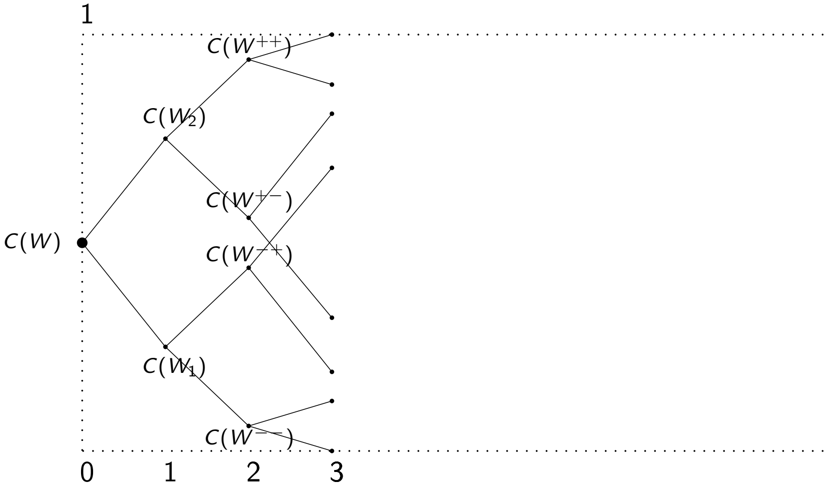
Polarization martingale



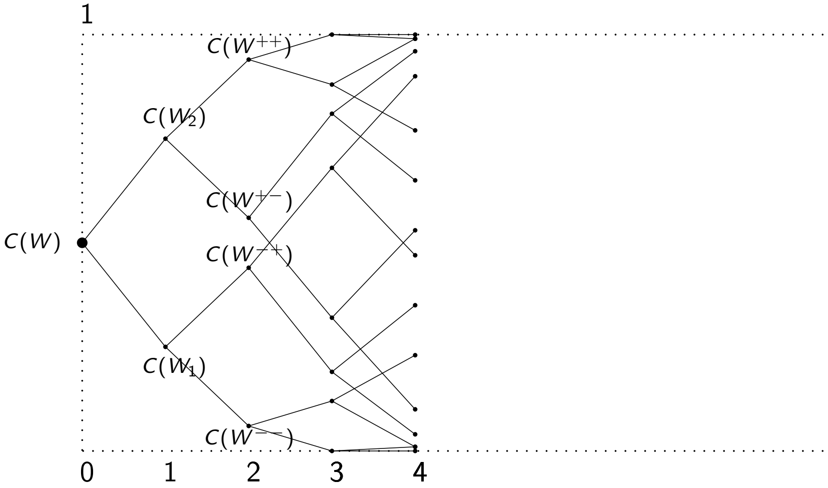
Polarization martingale



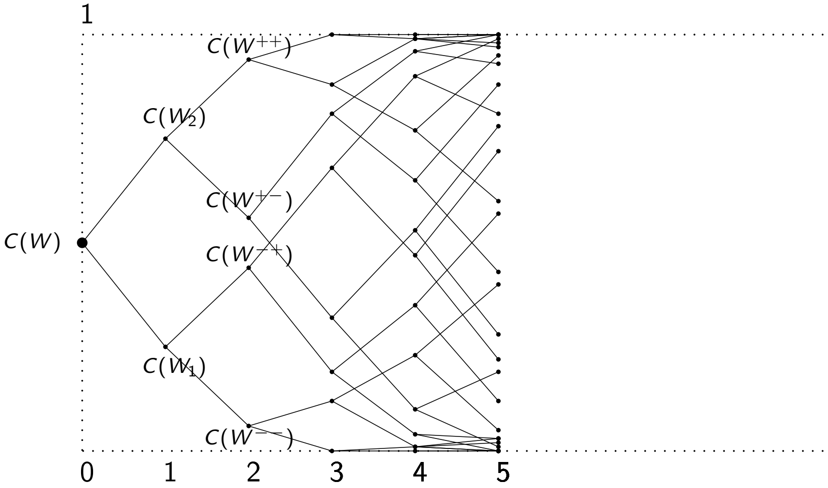
Polarization martingale



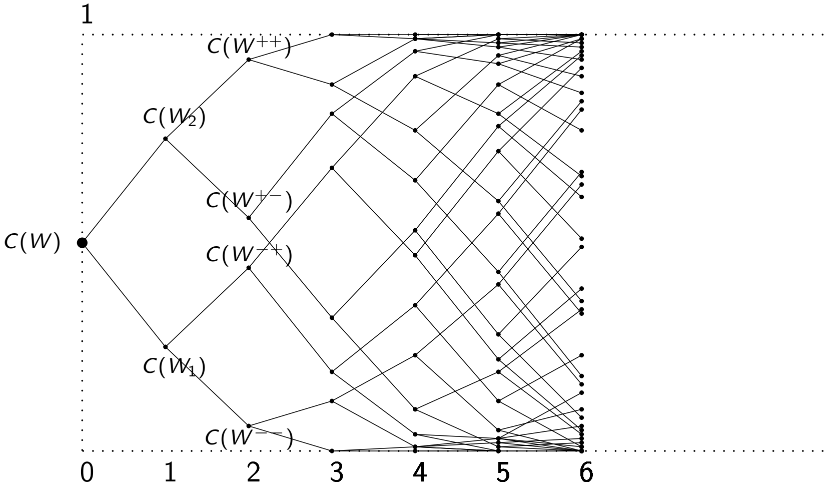
Polarization martingale



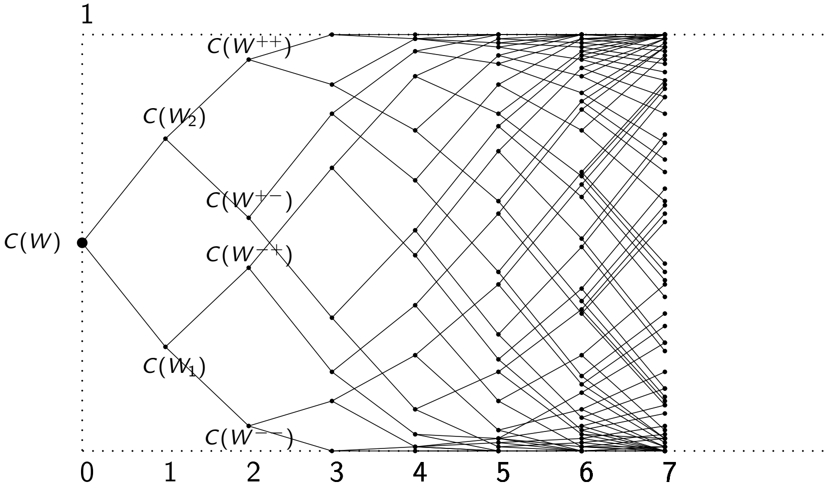
Polarization martingale



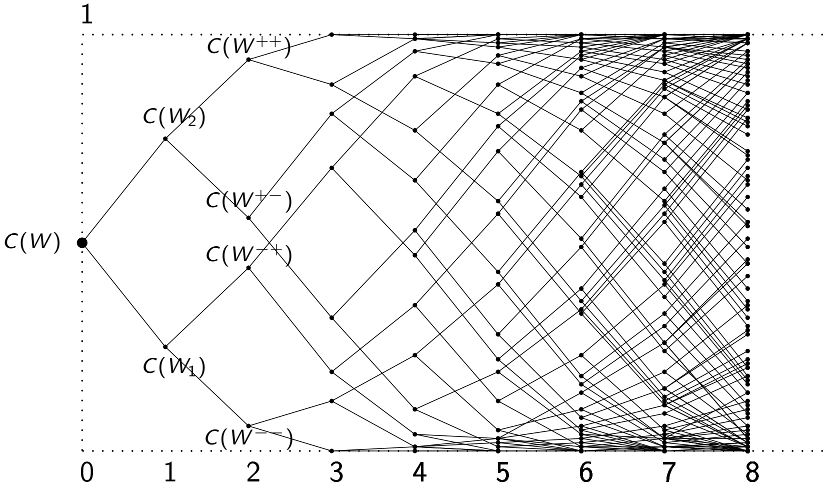
Polarization martingale



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*

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

and

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

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*

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

and

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

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

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

Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$I(W_i)$

0.0039

0.1211

0.1914

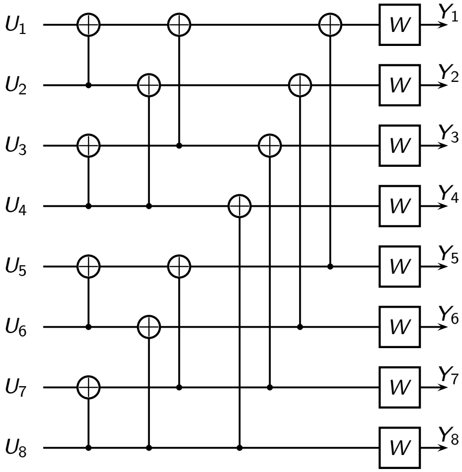
0.6836

0.3164

0.8086

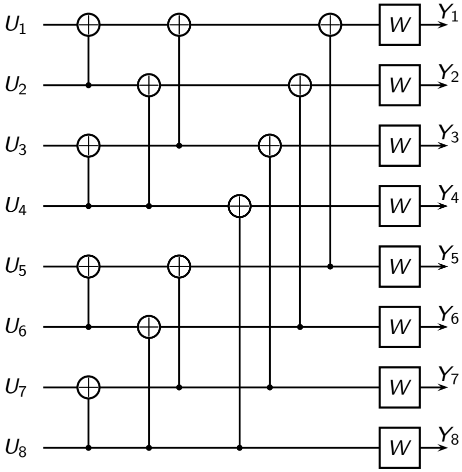
0.8789

0.9961



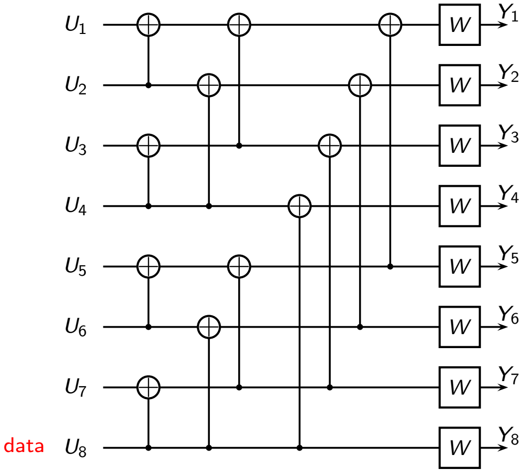
Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate 1/2

$I(W_i)$	Rank
0.0039	8
0.1211	7
0.1914	6
0.6836	4
0.3164	5
0.8086	3
0.8789	2
0.9961	1



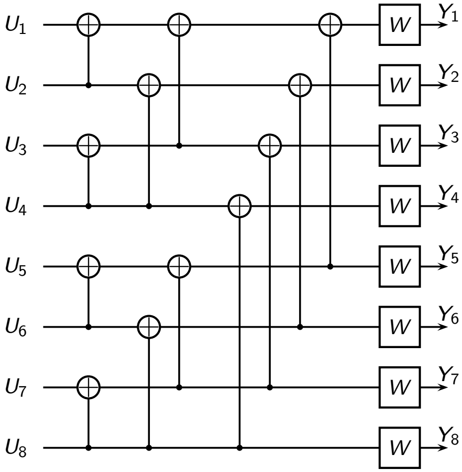
Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$I(W_i)$	Rank
0.0039	8
0.1211	7
0.1914	6
0.6836	4
0.3164	5
0.8086	3
0.8789	2
0.9961	1



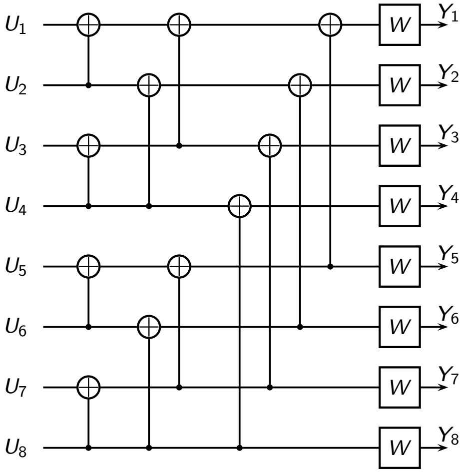
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$I(W_i)$	Rank	
0.0039	8	
0.1211	7	
0.1914	6	
0.6836	4	
0.3164	5	
0.8086	3	
0.8789	2	data
0.9961	1	data



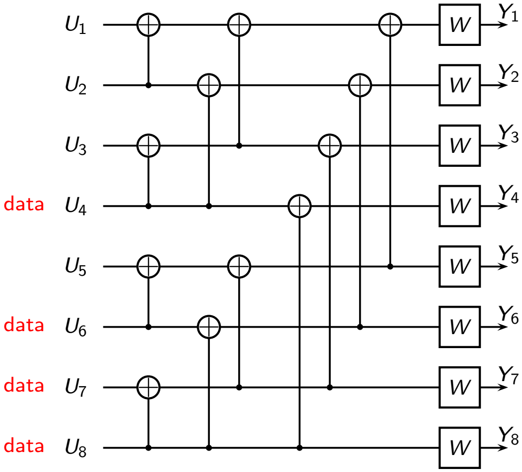
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$I(W_i)$	Rank	
0.0039	8	
0.1211	7	
0.1914	6	
0.6836	4	
0.3164	5	
0.8086	3	data
0.8789	2	data
0.9961	1	data



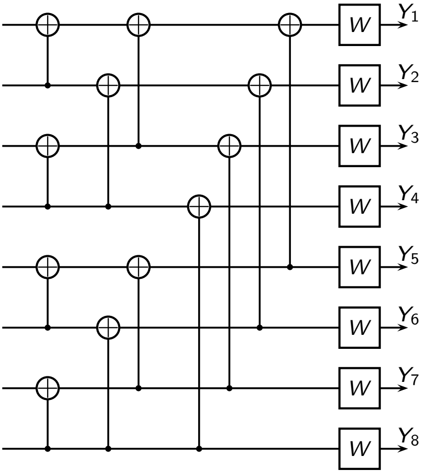
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$I(W_i)$	Rank
0.0039	8
0.1211	7
0.1914	6
0.6836	4
0.3164	5
0.8086	3
0.8789	2
0.9961	1



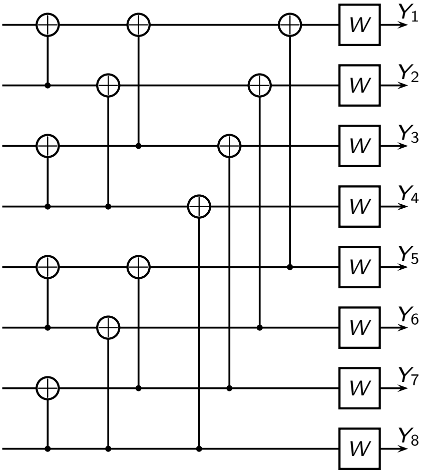
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$I(W_i)$	Rank		
0.0039	8	frozen	U_1
0.1211	7	frozen	U_2
0.1914	6	frozen	U_3
0.6836	4	data	U_4
0.3164	5	frozen	U_5
0.8086	3	data	U_6
0.8789	2	data	U_7
0.9961	1	data	U_8



Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$I(W_i)$	Rank		
0.0039	8	frozen	0
0.1211	7	frozen	0
0.1914	6	frozen	0
0.6836	4	data	U_4
0.3164	5	frozen	0
0.8086	3	data	U_6
0.8789	2	data	U_7
0.9961	1	data	U_8



Encoding complexity

Theorem

Encoding complexity for polar coding is $\mathcal{O}(N \log N)$.

Proof:

- Polar coding transform can be represented as a graph with $N[1 + \log(N)]$ variables.
- The graph has $(1 + \log(N))$ levels with N variables at each level.
- Computation begins at the source level and can be carried out level by level.
- Space complexity $O(N)$, time complexity $O(N \log N)$.

Encoding complexity

Theorem

Encoding complexity for polar coding is $\mathcal{O}(N \log N)$.

Proof:

- Polar coding transform can be represented as a graph with $N[1 + \log(N)]$ variables.
- The graph has $(1 + \log(N))$ levels with N variables at each level.
- Computation begins at the source level and can be carried out level by level.
- Space complexity $O(N)$, time complexity $O(N \log N)$.

Encoding complexity

Theorem

Encoding complexity for polar coding is $\mathcal{O}(N \log N)$.

Proof:

- Polar coding transform can be represented as a graph with $N[1 + \log(N)]$ variables.
- The graph has $(1 + \log(N))$ levels with N variables at each level.
- Computation begins at the source level and can be carried out level by level.
- Space complexity $O(N)$, time complexity $O(N \log N)$.

Encoding complexity

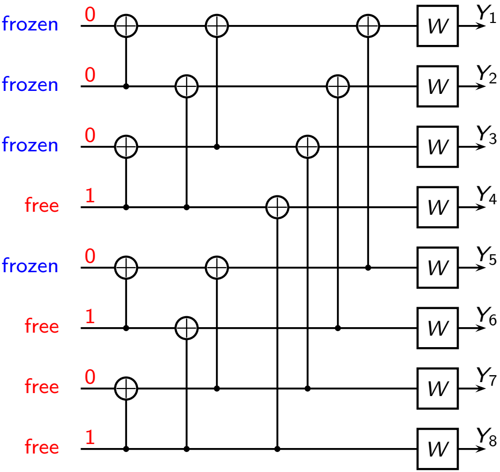
Theorem

Encoding complexity for polar coding is $\mathcal{O}(N \log N)$.

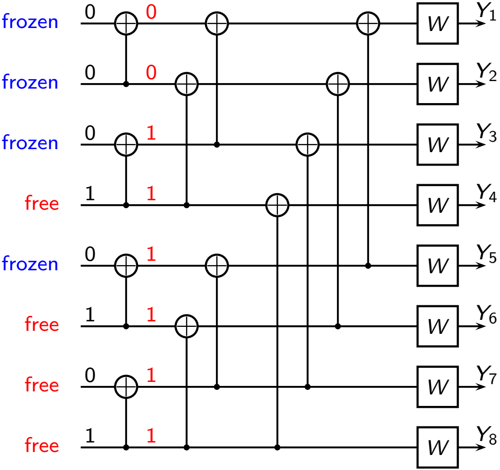
Proof:

- Polar coding transform can be represented as a graph with $N[1 + \log(N)]$ variables.
- The graph has $(1 + \log(N))$ levels with N variables at each level.
- Computation begins at the source level and can be carried out level by level.
- Space complexity $O(N)$, time complexity $O(N \log N)$.

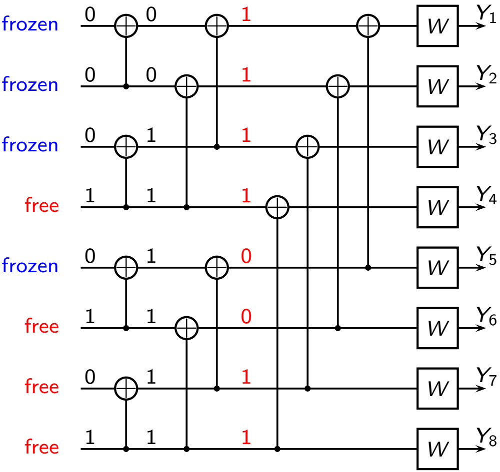
Encoding: an example



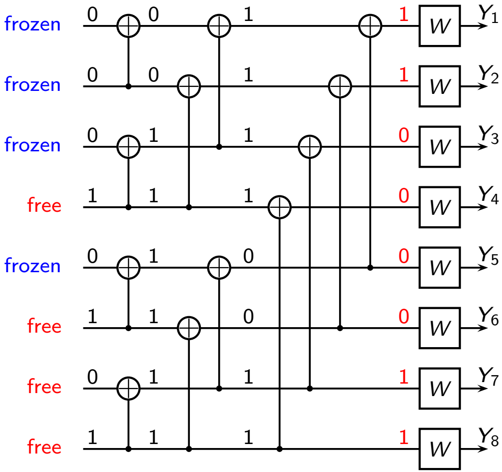
Encoding: an example



Encoding: an example



Encoding: an example



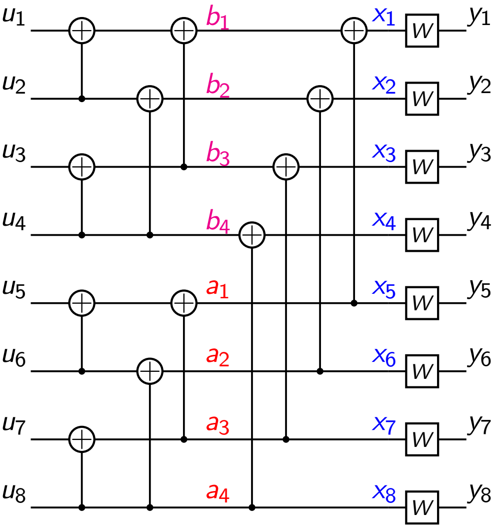
Successive Cancellation Decoding (SCD)

Theorem

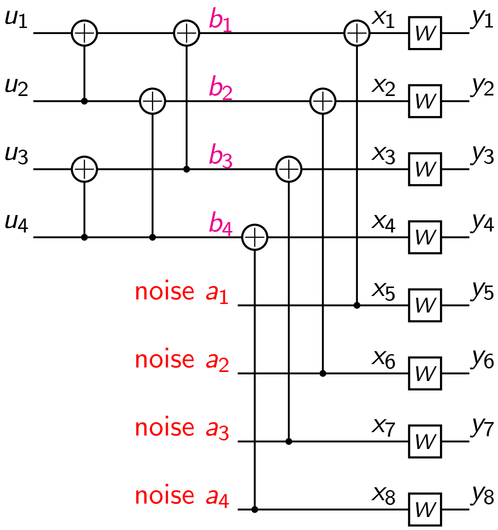
The complexity of successive cancellation decoding for polar codes is $\mathcal{O}(N \log N)$.

Proof: Given below.

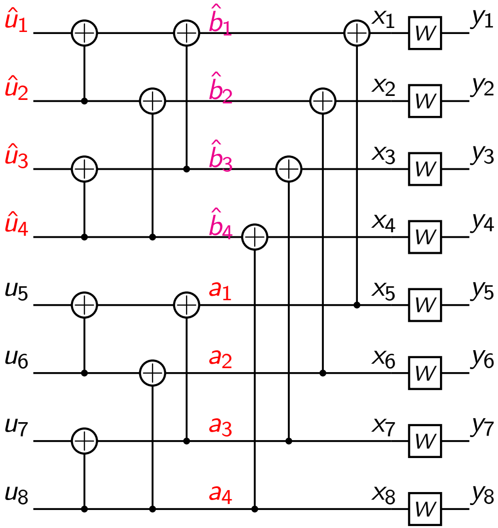
SCD: Exploit the $\mathbf{x} = |\mathbf{a}|\mathbf{a} + \mathbf{b}|$ structure



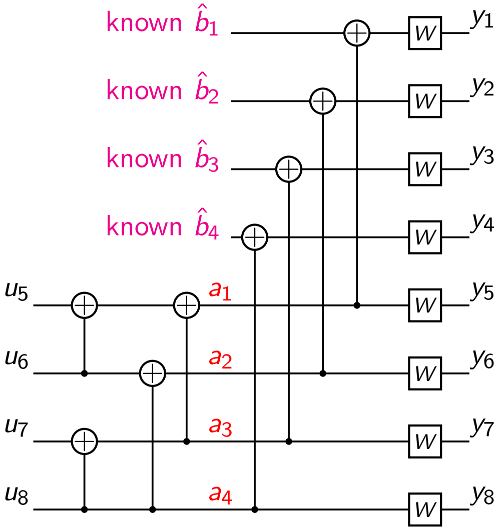
First phase: treat **a** as noise, decode (u_1, u_2, u_3, u_4)



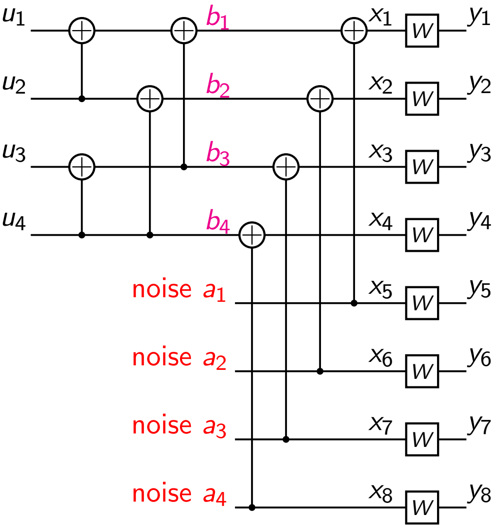
End of first phase



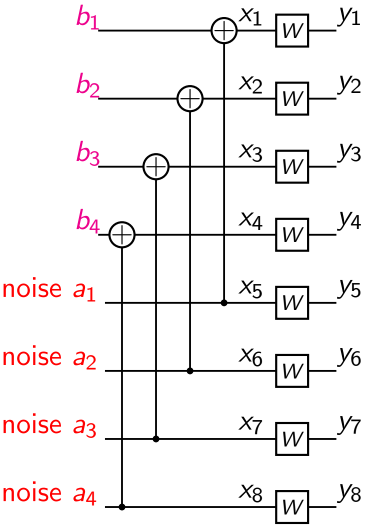
Second phase: Treat $\hat{\mathbf{b}}$ as known, decode (u_5, u_6, u_7, u_8)



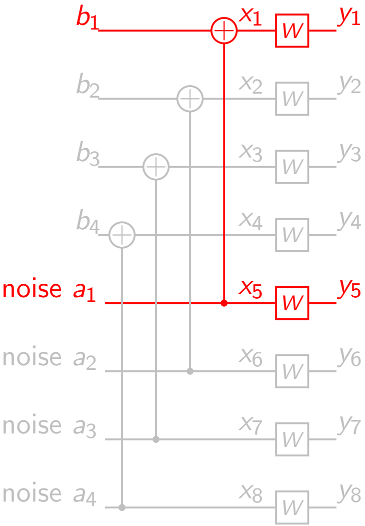
First phase in detail



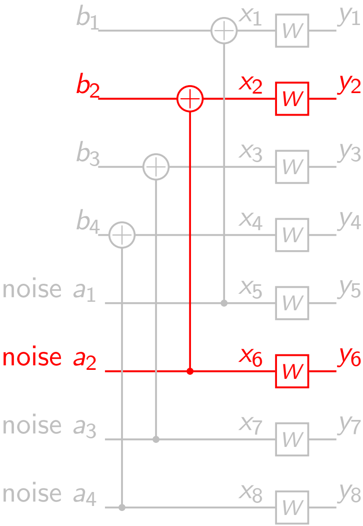
Equivalent channel model



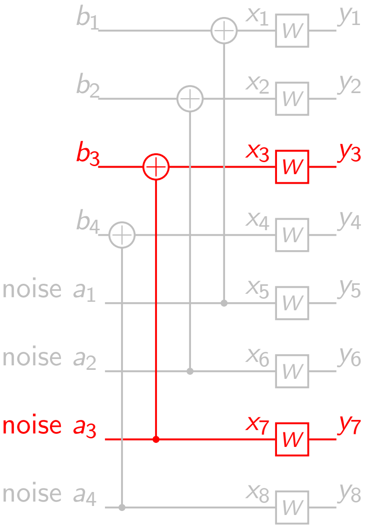
First copy of W^-



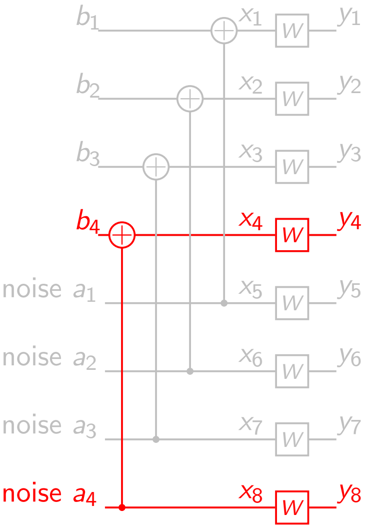
Second copy of W^-



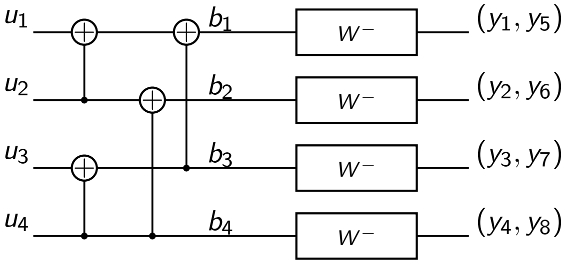
Third copy of W^-



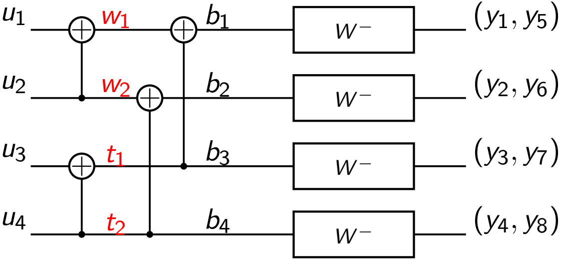
Fourth copy of W^-



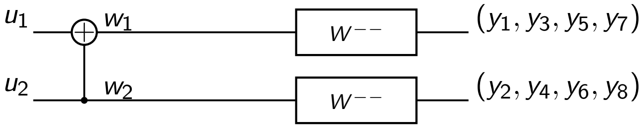
Decoding on W^-



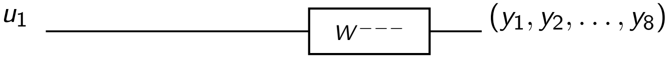
$\mathbf{b} = |\mathbf{t}|\mathbf{t} + \mathbf{w}|$



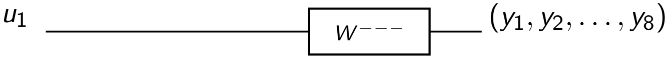
Decoding on W^{--}



Decoding on W^{---}



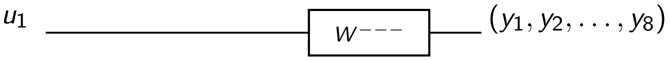
Decoding on W^{---}



Compute

$$L^{---} \triangleq \frac{W^{---}(y_1, \ldots, y_8 \mid u_1 = 0)}{W^{---}(y_1, \ldots, y_8 \mid u_1 = 1)}.$$

Decoding on W^{---}



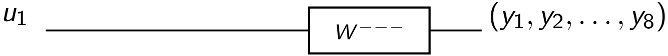
Compute

$$L^{---} \triangleq \frac{W^{---}(y_1, \ldots, y_8 \mid u_1 = 0)}{W^{---}(y_1, \ldots, y_8 \mid u_1 = 1)}.$$

Set

$$\hat{u}_1 = \begin{cases} u_1 & \text{if } u_1 \text{ is frozen} \\ 0 & \text{else if } L^{---} > 0 \\ 1 & \text{else} \end{cases}$$

Decoding on W^{---}



Compute

$$L^{---} \triangleq \frac{W^{---}(y_1, \ldots, y_8 \mid u_1 = 0)}{W^{---}(y_1, \ldots, y_8 \mid u_1 = 1)}.$$

Set

$$\hat{u}_1 = \begin{cases} u_1 & \text{if } u_1 \text{ is frozen} \\ 0 & \text{else if } L^{---} > 0 \\ 1 & \text{else} \end{cases}$$

Decoding on W^{---}

$$u_1 \;\text{———}\; \boxed{W^{---}} \;\text{——}\; (y_1, y_2, \ldots, y_8)$$

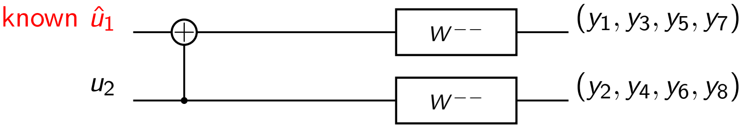
Compute

$$L^{---} \triangleq \frac{W^{---}(y_1, \ldots, y_8 \mid u_1 = 0)}{W^{---}(y_1, \ldots, y_8 \mid u_1 = 1)}.$$

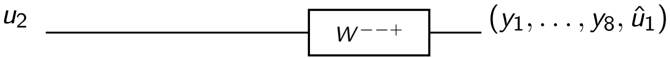
Set

$$\hat{u}_1 = \begin{cases} u_1 & \text{if } u_1 \text{ is frozen} \\ 0 & \text{else if } L^{---} > 0 \\ 1 & \text{else} \end{cases}$$

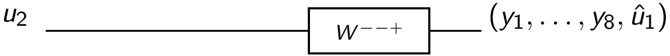
Decoding on W^{--+}



Decoding on W^{--+}



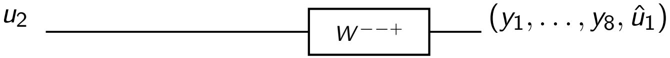
Decoding on W^{--+}



Compute

$$L^{--+} \triangleq \frac{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 0)}{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 1)}.$$

Decoding on W^{--+}



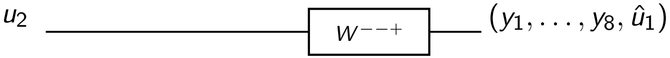
Compute

$$L^{--+} \triangleq \frac{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 0)}{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 1)}.$$

Set

$$\hat{u}_2 = \begin{cases} u_2 & \text{if } u_2 \text{ is frozen} \\ 0 & \text{else if } L^{--+} > 0 \\ 1 & \text{else} \end{cases}$$

Decoding on W^{--+}



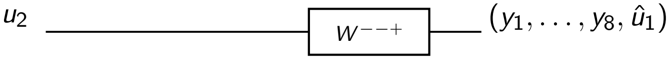
Compute

$$L^{--+} \triangleq \frac{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 0)}{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 1)}.$$

Set

$$\hat{u}_2 = \begin{cases} u_2 & \text{if } u_2 \text{ is frozen} \\ 0 & \text{else if } L^{--+} > 0 \\ 1 & \text{else} \end{cases}$$

Decoding on W^{--+}



Compute

$$L^{--+} \triangleq \frac{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 0)}{W^{--+}(y_1, \ldots, y_8, \hat{u}_1 \mid u_2 = 1)}.$$

Set

$$\hat{u}_2 = \begin{cases} u_2 & \text{if } u_2 \text{ is frozen} \\ 0 & \text{else if } L^{--+} > 0 \\ 1 & \text{else} \end{cases}$$

Complexity for successive cancelation decoding

- Let C_N be the complexity of decoding a code of length N
- Decoding problem of size N for W reduced to two decoding problems of size $N/2$ for W^- and W^+
- So

$$C_N = 2C_{N/2} + kN$$

  for some constant k
- This gives $C_N = \mathcal{O}(N \log N)$

Complexity for successive cancelation decoding

- Let C_N be the complexity of decoding a code of length N
- Decoding problem of size N for W reduced to two decoding problems of size $N/2$ for W^- and W^+
- So
$$C_N = 2C_{N/2} + kN$$
for some constant k
- This gives $C_N = \mathcal{O}(N \log N)$

Complexity for successive cancelation decoding

- Let C_N be the complexity of decoding a code of length N
- Decoding problem of size N for W reduced to two decoding problems of size $N/2$ for W^- and W^+
- So

$$C_N = 2C_{N/2} + kN$$

  for some constant k
- This gives $C_N = \mathcal{O}(N \log N)$

Complexity for successive cancelation decoding

- Let C_N be the complexity of decoding a code of length N
- Decoding problem of size N for W reduced to two decoding problems of size $N/2$ for W^- and W^+
- So

$$C_N = 2C_{N/2} + kN$$

  for some constant k
- This gives $C_N = \mathcal{O}(N \log N)$

Performance of polar codes

Theorem

For any rate $R < I(W)$ and block-length N, the probability of frame error for polar codes under successive cancelation decoding is bounded as

$$P_e(N, R) = o\left(2^{-\sqrt{N}+o(\sqrt{N})}\right)$$

Proof: Given in the next presentation.

Construction complexity

Theorem

Given W and a rate $R < I(W)$, a polar code can be constructed in $\mathcal{O}(N\text{poly}(log(N)))$ time that achieves under SCD the performance

$$P_e = o\left(2^{-\sqrt{N}+o(\sqrt{N})}\right)$$

Proof: Given in the next presentation.

Polar coding summary

Summary

Given W, $N = 2^n$, and $R < I(W)$, a polar code can be constructed such that it has

- construction complexity $\mathcal{O}(N\text{poly}(log(N)))$,
- encoding complexity $\approx N \log N$,
- successive-cancellation decoding complexity $\approx N \log N$,
- frame error probability $P_e(N, R) = o\left(2^{-\sqrt{N}+o(\sqrt{N})}\right)$.

Polar coding summary

Summary

Given W, $N = 2^n$, and $R < I(W)$, a polar code can be constructed such that it has

- construction complexity $\mathcal{O}(N\text{poly}(log(N)))$,
- encoding complexity $\approx N \log N$,
- successive-cancellation decoding complexity $\approx N \log N$,
- frame error probability $P_e(N, R) = o\left(2^{-\sqrt{N}+o(\sqrt{N})}\right)$.

Polar coding summary

Summary

Given W, $N = 2^n$, and $R < I(W)$, a polar code can be constructed such that it has

- construction complexity $\mathcal{O}(N\text{poly}(log(N)))$,
- encoding complexity $\approx N \log N$,
- successive-cancellation decoding complexity $\approx N \log N$,
- frame error probability $P_e(N, R) = o\left(2^{-\sqrt{N}+o(\sqrt{N})}\right)$.

Polar coding summary

Summary

Given W, $N = 2^n$, and $R < I(W)$, a polar code can be constructed such that it has

- construction complexity $\mathcal{O}(N\text{poly}(log(N)))$,
- encoding complexity $\approx N \log N$,
- successive-cancellation decoding complexity $\approx N \log N$,
- frame error probability $P_e(N, R) = o\left(2^{-\sqrt{N}+o(\sqrt{N})}\right)$.

List decoder for polar codes

Developed by Tal and Vardy (2011); similar to Dumer's list decoder for Reed-Muller codes.

- First produce L candidate decisions
- Pick the most likely word from the list
- Complexity $\mathcal{O}(LN \log N)$

List decoder for polar codes

Developed by Tal and Vardy (2011); similar to Dumer's list decoder for Reed-Muller codes.

- First produce L candidate decisions
- Pick the most likely word from the list
- Complexity $\mathcal{O}(LN \log N)$

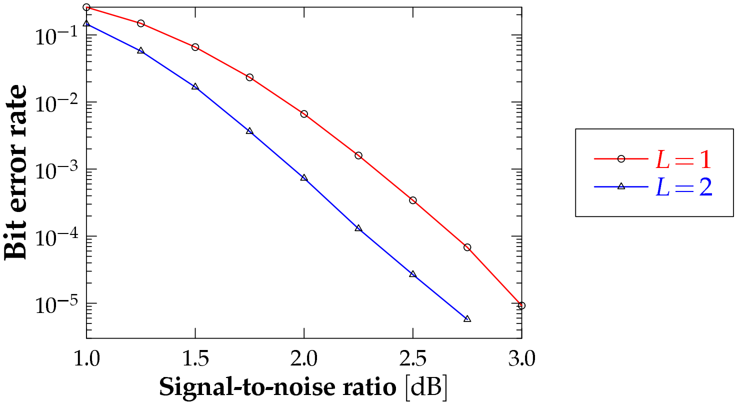
List decoder for polar codes

Developed by Tal and Vardy (2011); similar to Dumer's list decoder for Reed-Muller codes.

- First produce L candidate decisions
- Pick the most likely word from the list
- Complexity $\mathcal{O}(LN \log N)$

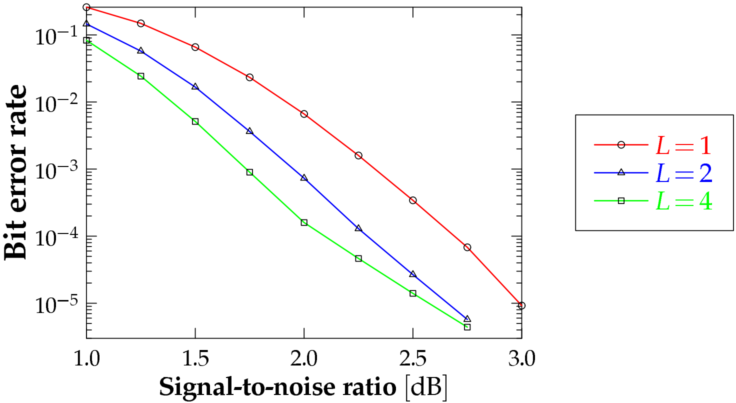
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



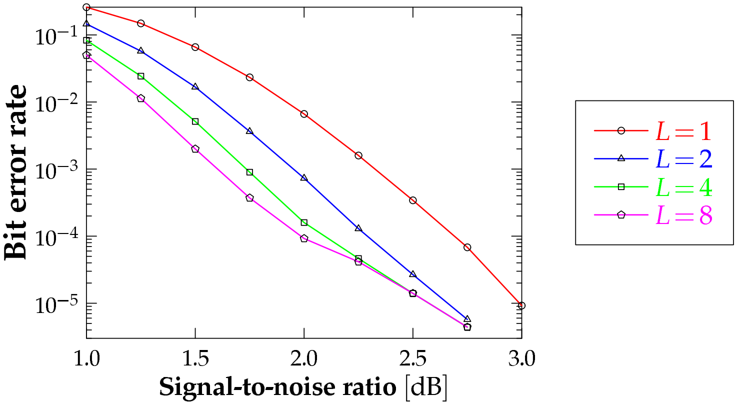
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



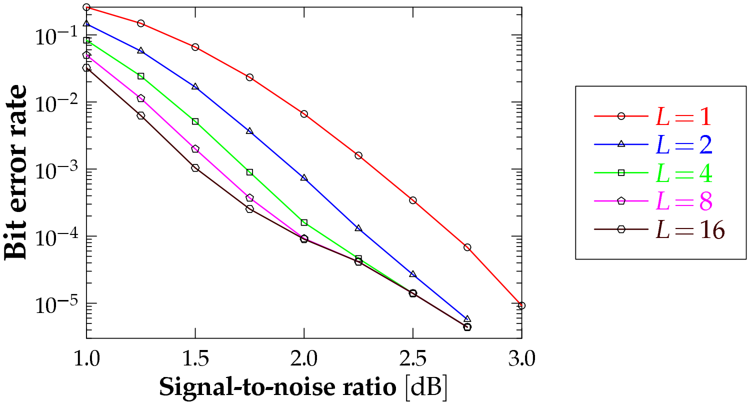
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



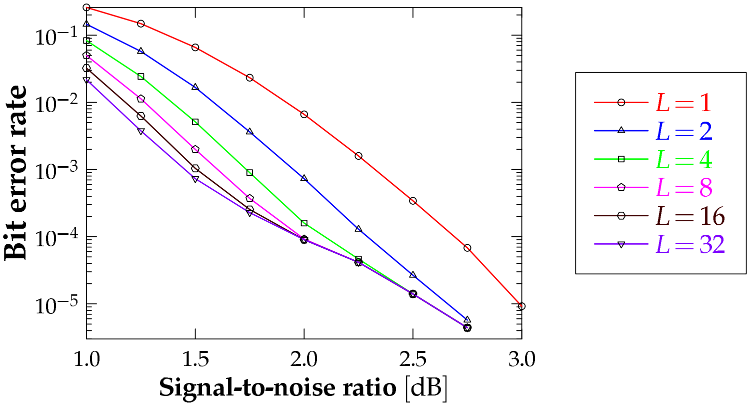
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



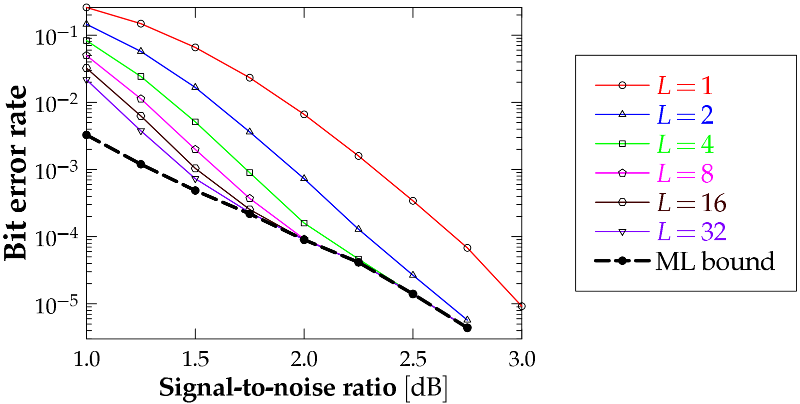
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



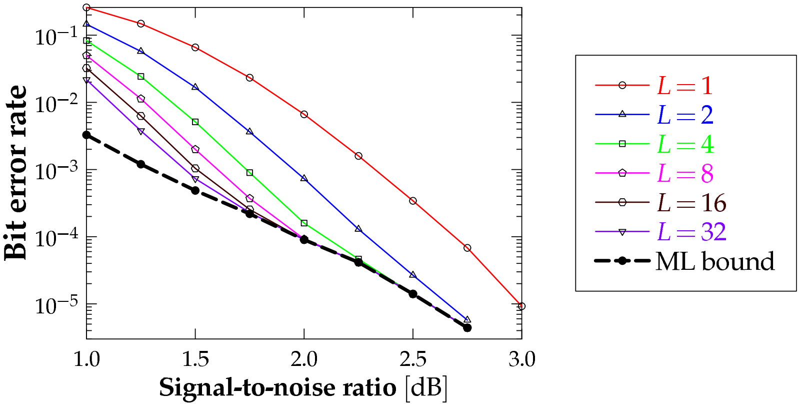
Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



Tal-Vardy list decoder performance

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



List-of-L performance quickly approaches ML performance!

List decoder with CRC

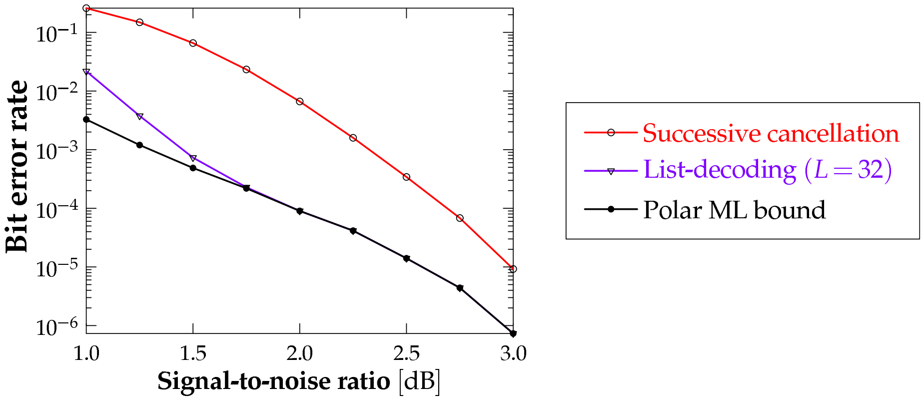
- Same decoder as before but data contains a built-in CRC
- Selection made by CRC and relative likelihood

List decoder with CRC

- Same decoder as before but data contains a built-in CRC
- Selection made by CRC and relative likelihood

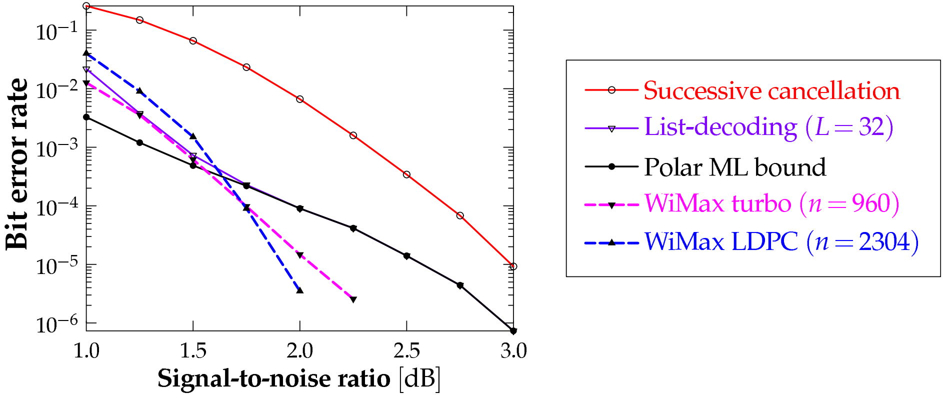
Tal-Vardy list decoder with CRC

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



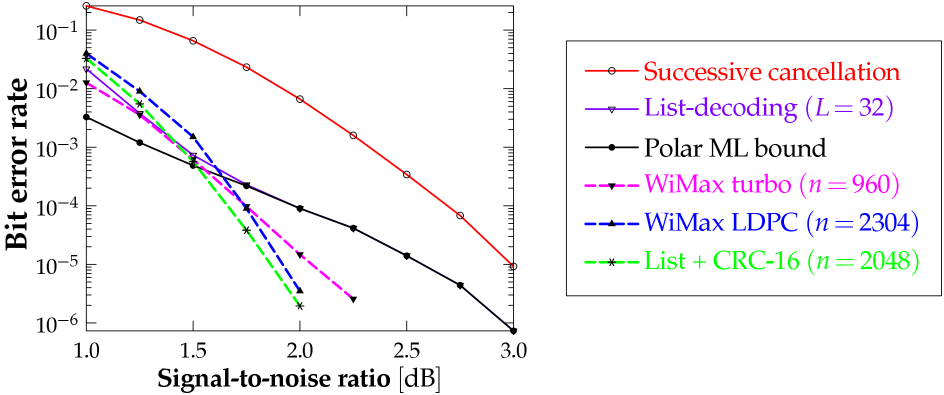
Tal-Vardy list decoder with CRC

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



Tal-Vardy list decoder with CRC

Length $n = 2048$, rate $R = 0.5$, BPSK-AWGN channel, list-size L.



Polar codes (+CRC) achieve state-of-the-art performance!

Summary

- Polarization is a commonplace phenomenon – almost unavoidable
- Polar codes are low-complexity methods designed to exploit polarization for achieving Shannon limits
- Polar codes with some help from other methods perform competitively with the state-of-the-art codes in terms of complexity and performance

Summary

- Polarization is a commonplace phenomenon – almost unavoidable
- Polar codes are low-complexity methods designed to exploit polarization for achieving Shannon limits
- Polar codes with some help from other methods perform competitively with the state-of-the-art codes in terms of complexity and performance

Summary

- Polarization is a commonplace phenomenon – almost unavoidable
- Polar codes are low-complexity methods designed to exploit polarization for achieving Shannon limits
- Polar codes with some help from other methods perform competitively with the state-of-the-art codes in terms of complexity and performance