Polar Coding

Status and Prospects

Erdal Arıkan

Electrical-Electronics Engineering Department
Bilkent University
Ankara, Turkey

1 August 2011
The IEEE International Symposium on Information Theory
ISIT'2011
Saint Petersburg, Russia

The channel

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

$$X \longrightarrow \boxed{W} \longrightarrow Y$$

- 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

$$X \longrightarrow \boxed{W} \longrightarrow Y$$

- 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

$$X \longrightarrow \boxed{\;W\;} \longrightarrow Y$$

- 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

$$X \longrightarrow \boxed{\quad W \quad} \longrightarrow Y$$

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

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

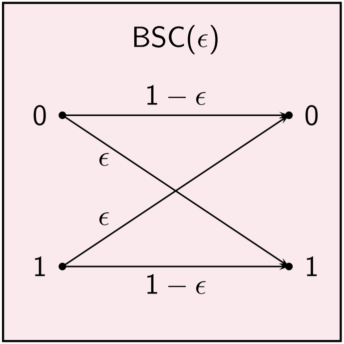
Symmetry assumption

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

Symmetry assumption

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

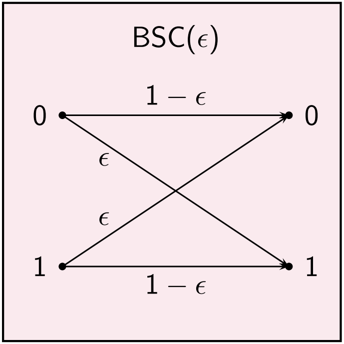
Examples:

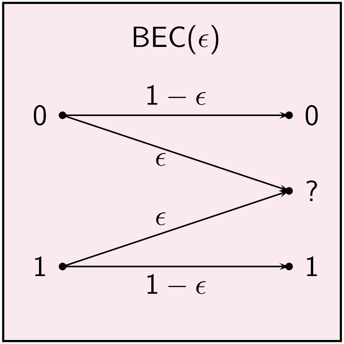


Symmetry assumption

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

Examples:





Capacity

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

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

Capacity

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

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

Use base-2 logarithms:

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

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

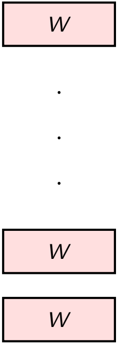
- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary W into such extreme channels

The main idea

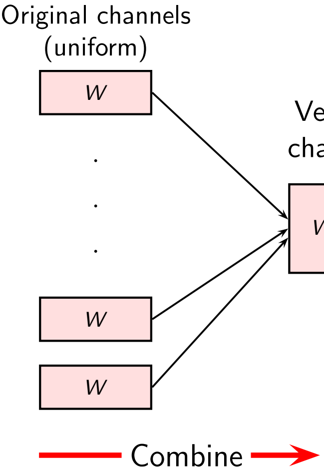
- Channel coding problem trivial for two types of channels
  - Perfect: $C(W) = 1$
  - Useless: $C(W) = 0$
- Transform ordinary W into such extreme channels

The method: aggregate and redistribute capacity

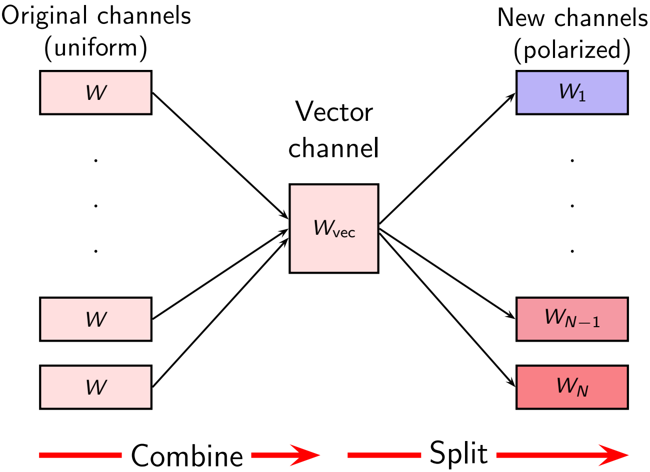
Original channels
(uniform)



The method: aggregate and redistribute capacity



The method: aggregate and redistribute capacity



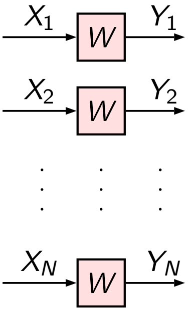
Combining

- 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

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



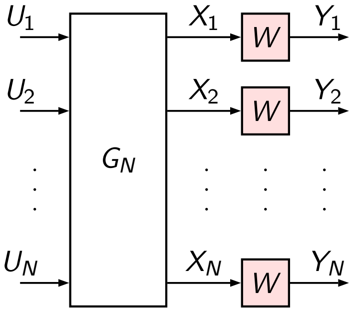
Combining

- 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

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



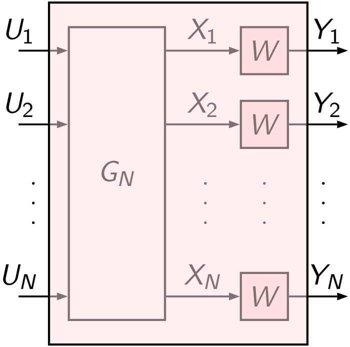
Combining

- 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

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



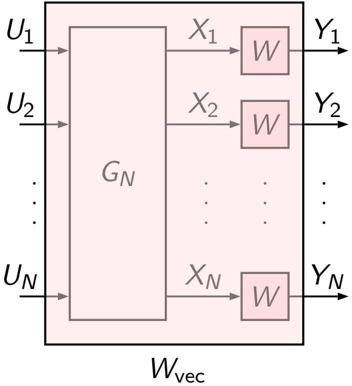
W_{vec}

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

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

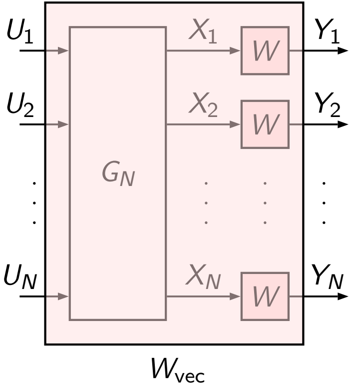


Conservation of capacity

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

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

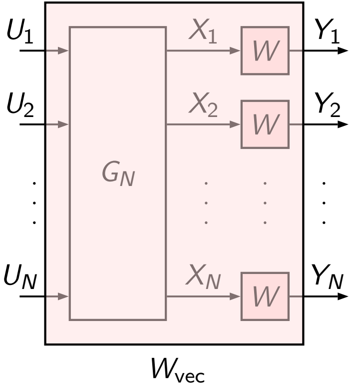


Conservation of capacity

Combining operation is lossless:

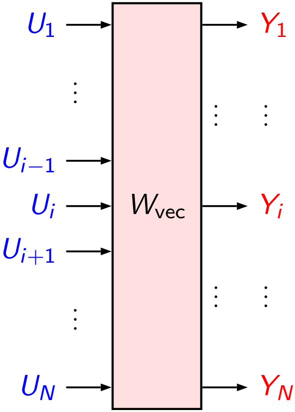
- Take $U_1, \ldots, U_N$ i.i.d. unif. $\{0,1\}$
- then, $X_1, \ldots, X_N$ i.i.d. unif. $\{0,1\}$
- and

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



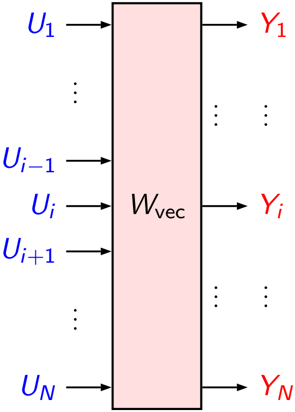
Splitting

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



Splitting

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

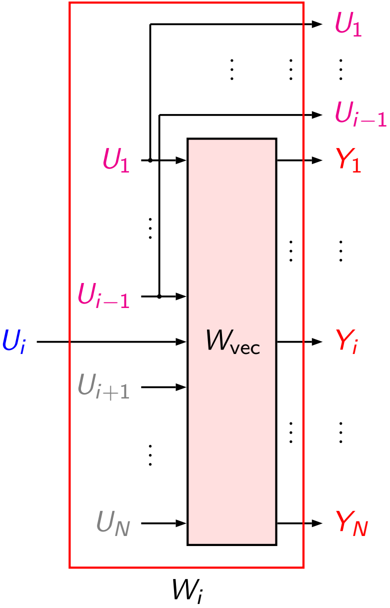


Splitting

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

Define bit-channels

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

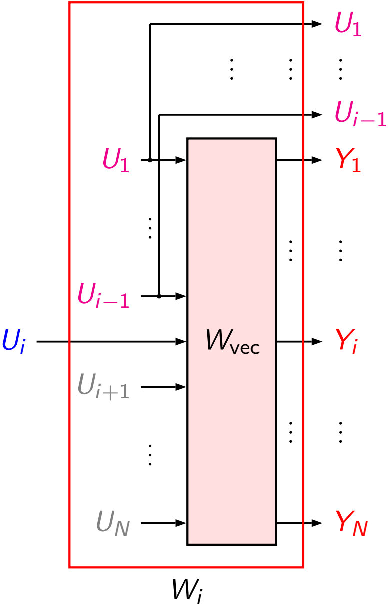


Splitting

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

Define bit-channels

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

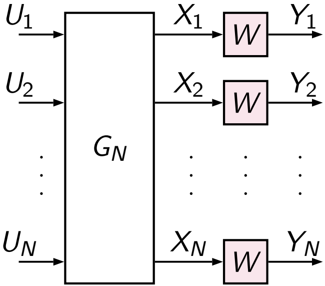


Polarization is commonplace

- 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



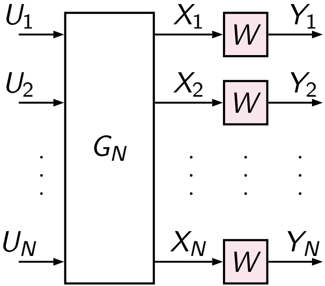
Polarization is commonplace

- 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

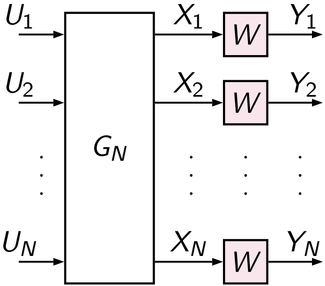


Polarization is commonplace

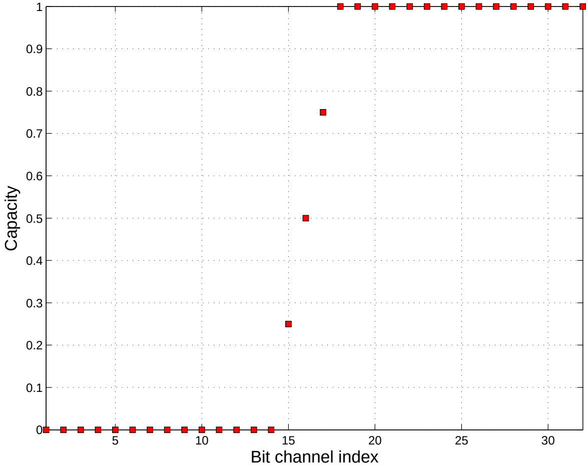
- Polarization is the rule not the exception
- A random permutation

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

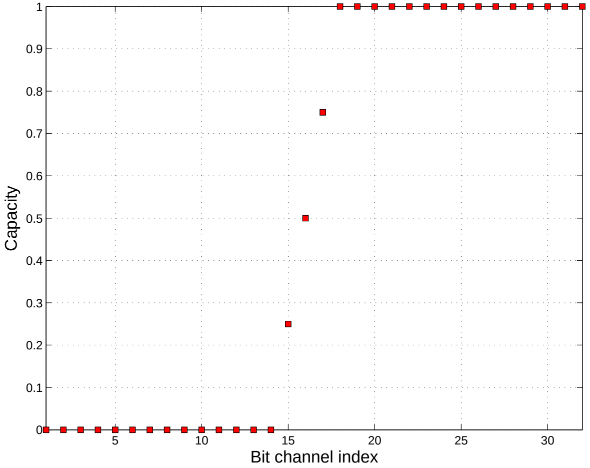
is a good polarizer with high probability
- Equivalent to Shannon's random coding approach



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

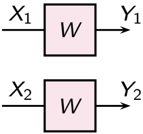
- Random polarizers lack structure, too complex to implement
- Need a low-complexity polarizer
- May sacrifice stepwise, isotropic properties of random polarizers in return for less complexity

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

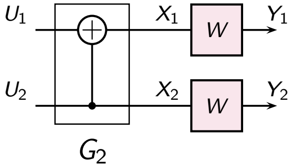
Basic module for a low-complexity scheme

Combine two copies of W



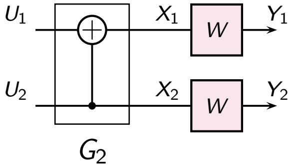
Basic module for a low-complexity scheme

Combine two copies of W



Basic module for a low-complexity scheme

Combine two copies of W

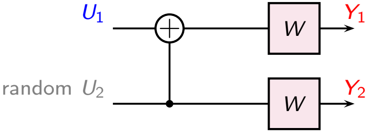


and split to create two bit-channels

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

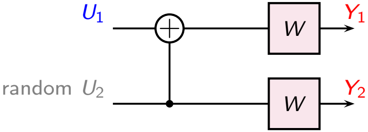
The first bit-channel W_1

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



The first bit-channel W_1

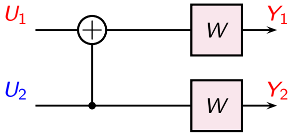
$$W_1 : U_1 \to (Y_1, Y_2)$$



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

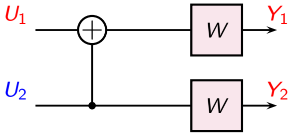
The second bit-channel W_2

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



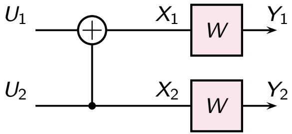
The second bit-channel W_2

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



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

Capacity conserved but redistributed unevenly



- Conservation:

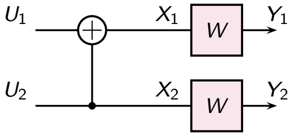
$$C(W_1) + C(W_2) = 2C(W)$$

- 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

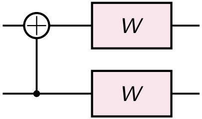
$$(W, W) \to (W_1, W_2)$$

will be denoted also as

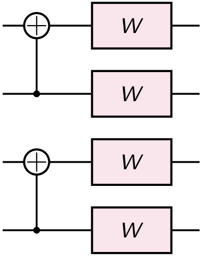
$$W^- = W_1 \quad \text{and} \quad W^+ = W_2$$

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

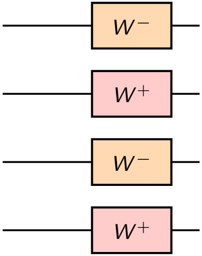
For the size-4 construction



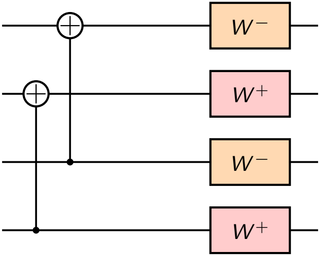
… duplicate the basic transform



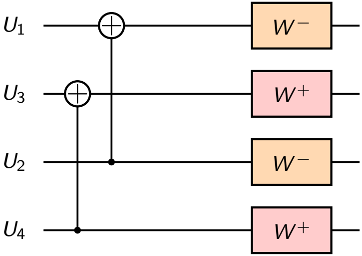
... obtain a pair of W^- and W^+ each



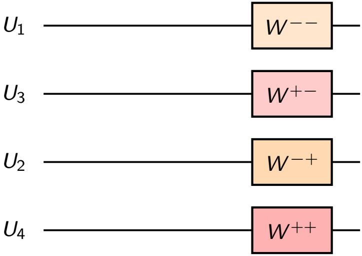
... apply basic transform on each pair



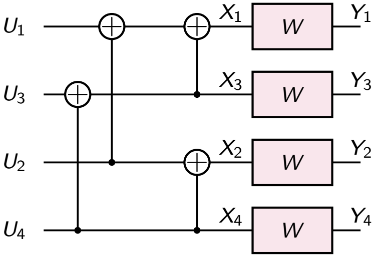
... decode in the indicated order



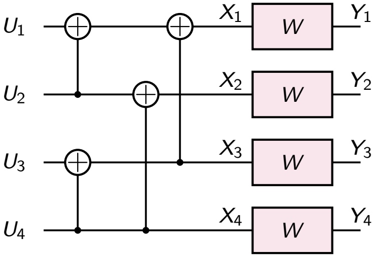
… obtain the four new bit-channels



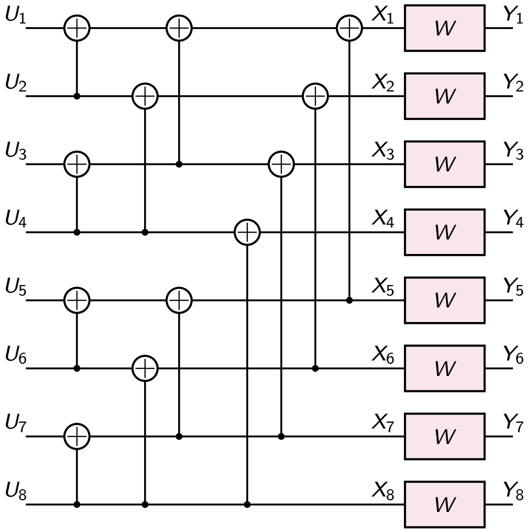
Overall size-4 construction



“Rewire” for standard-form size-4 construction



Size 8 construction



Demonstration of polarization

Polarization is easy to analyze when W is a BEC.

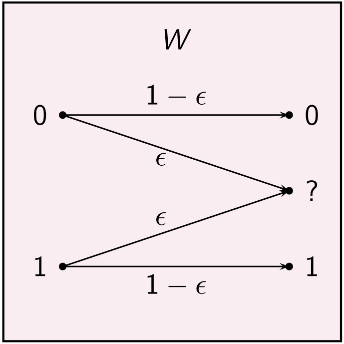
If W is a BEC(ϵ), then so are W^- and W^+, with erasure probabilities

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

and

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

respectively.



Demonstration of polarization

Polarization is easy to analyze when W is a BEC.

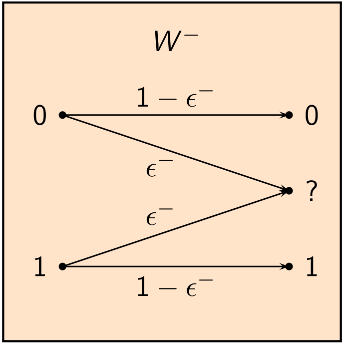
If W is a BEC(ϵ), then so are W^- and W^+, with erasure probabilities

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

and

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

respectively.



Demonstration of polarization

Polarization is easy to analyze when W is a BEC.

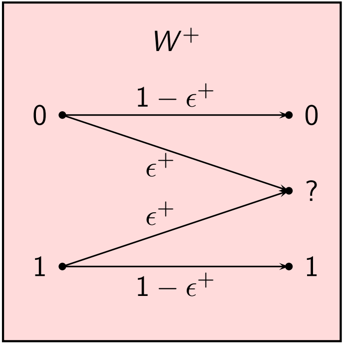
If W is a BEC(ϵ), then so are W^- and W^+, with erasure probabilities

$$\epsilon^- \stackrel{\Delta}{=} 2\epsilon - \epsilon^2$$

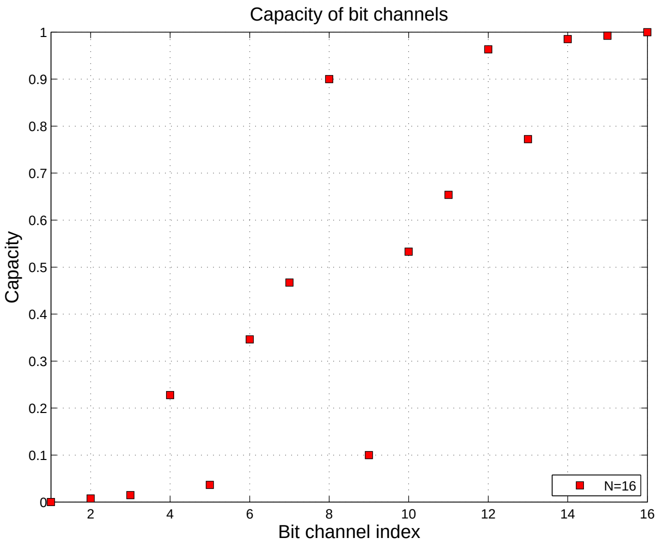
and

$$\epsilon^+ \stackrel{\Delta}{=} \epsilon^2$$

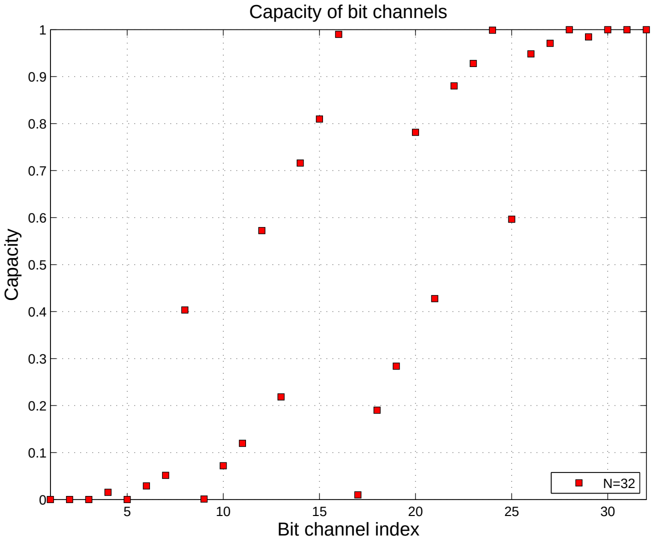
respectively.



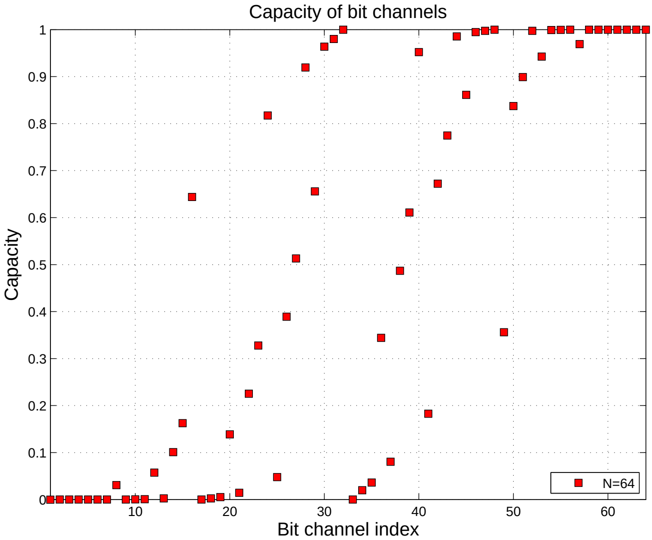
Polarization for BEC($\frac{1}{2}$): $N = 16$



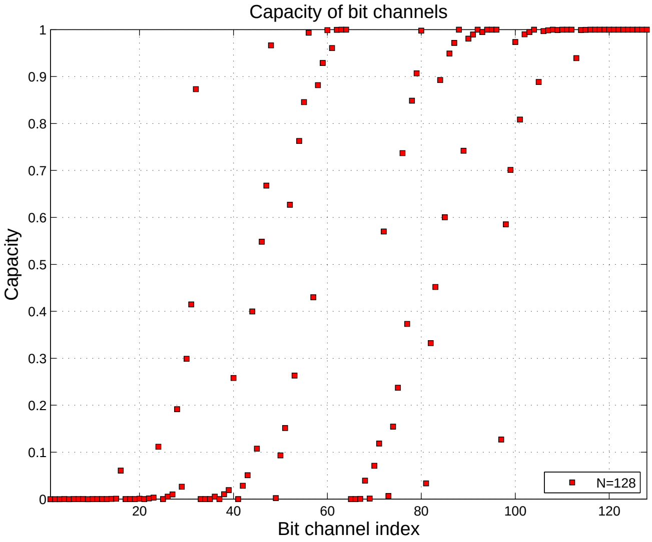
Polarization for BEC($\frac{1}{2}$): $N = 32$



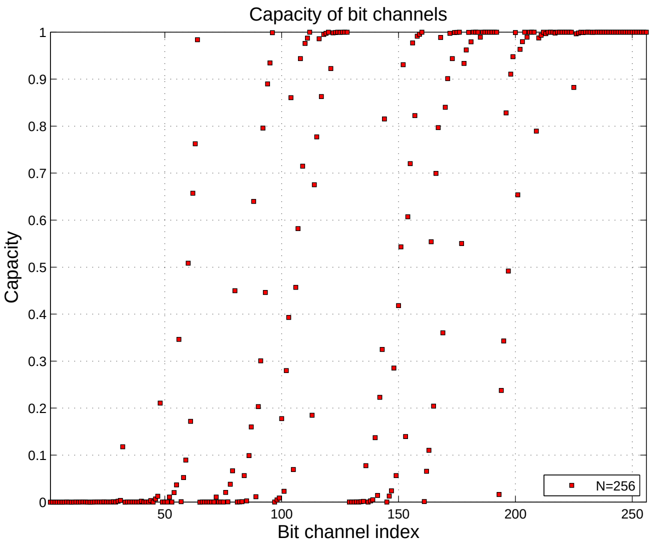
Polarization for BEC($\frac{1}{2}$): $N = 64$



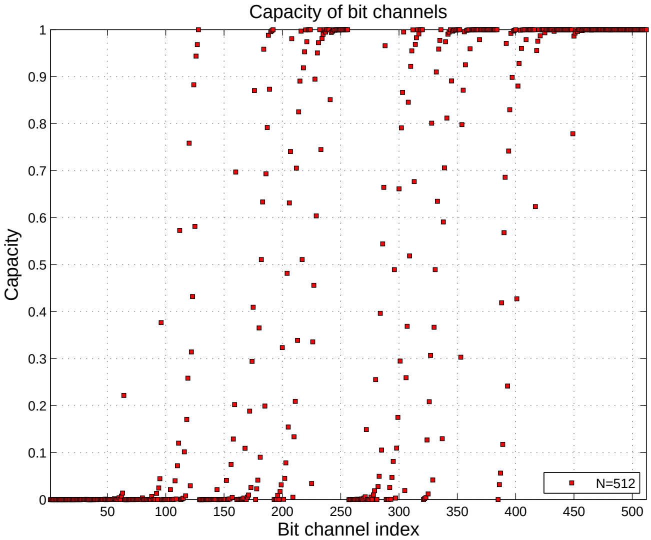
Polarization for BEC($\frac{1}{2}$): $N = 128$



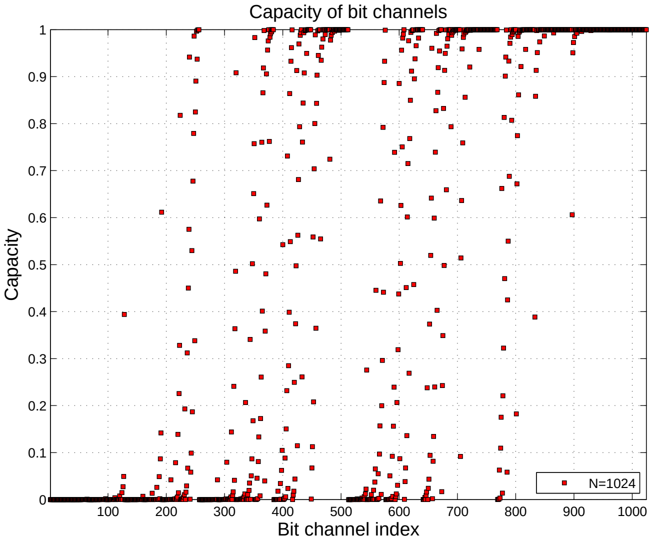
Polarization for BEC($\frac{1}{2}$): $N = 256$



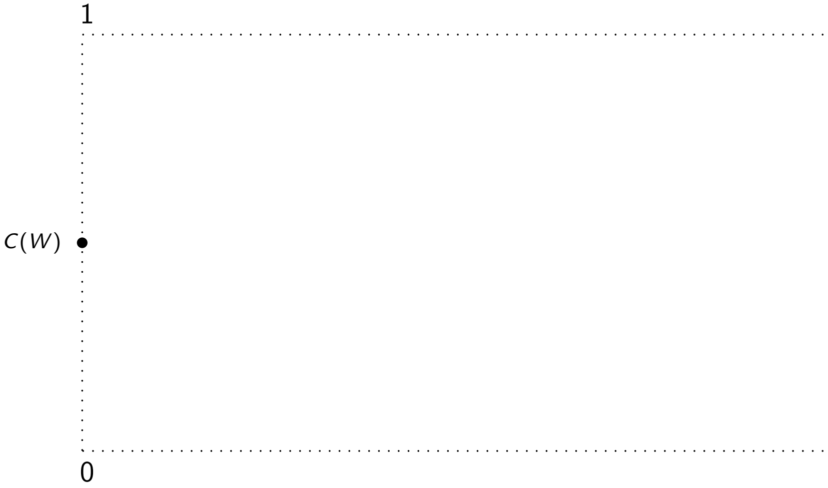
Polarization for BEC($\frac{1}{2}$): $N = 512$



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

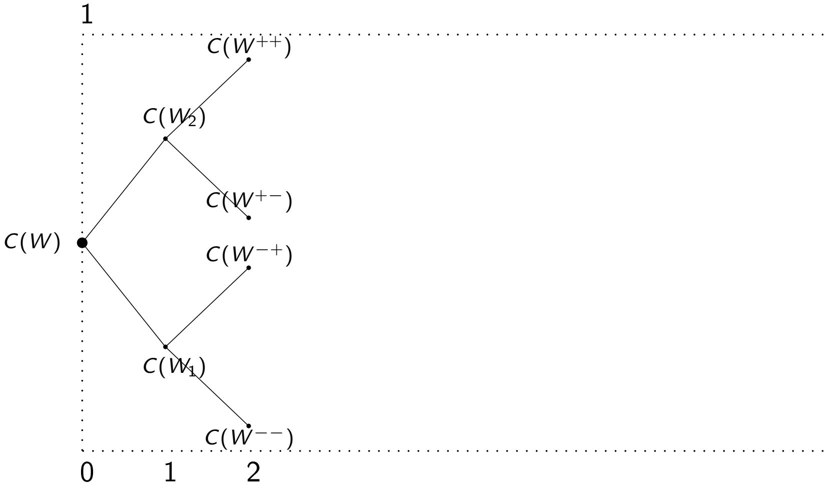


Polarization martingale

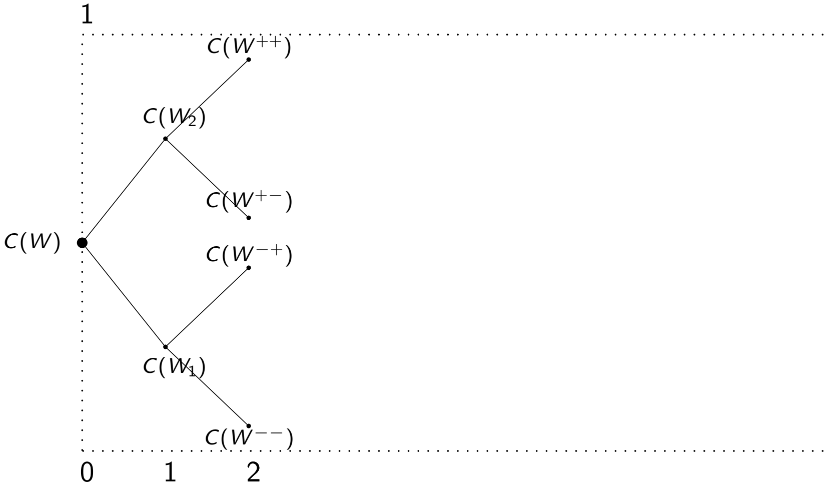


Polarization martingale

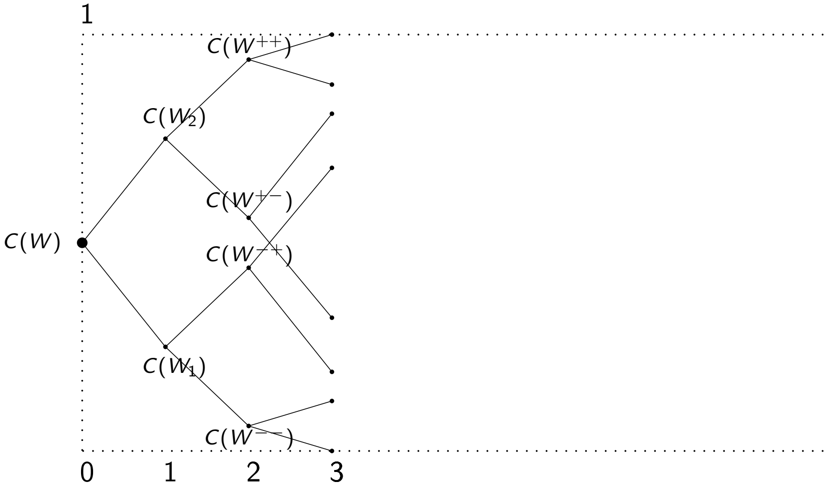
Polarization martingale



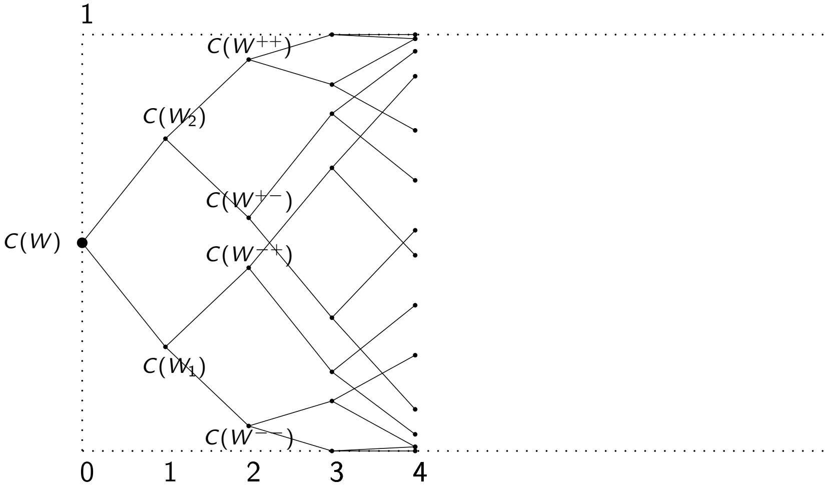
Polarization martingale



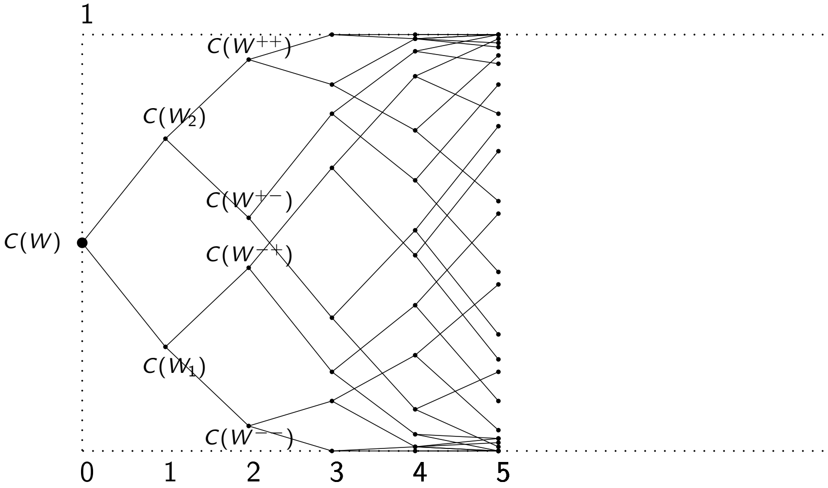
Polarization martingale



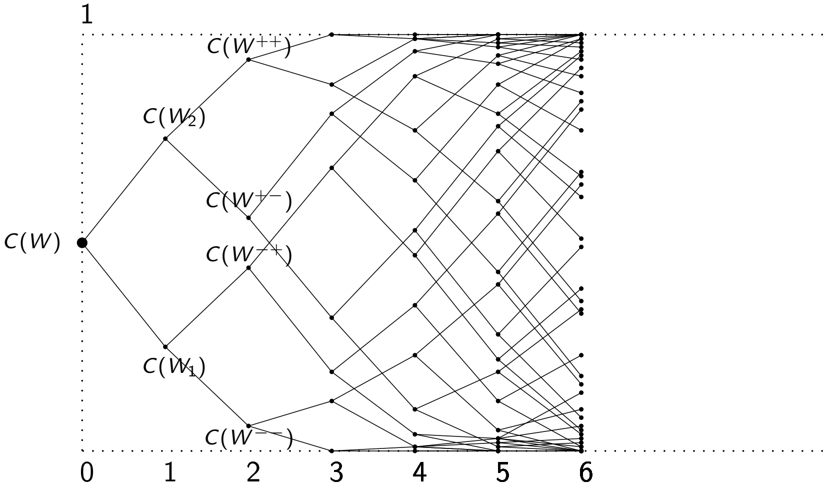
Polarization martingale



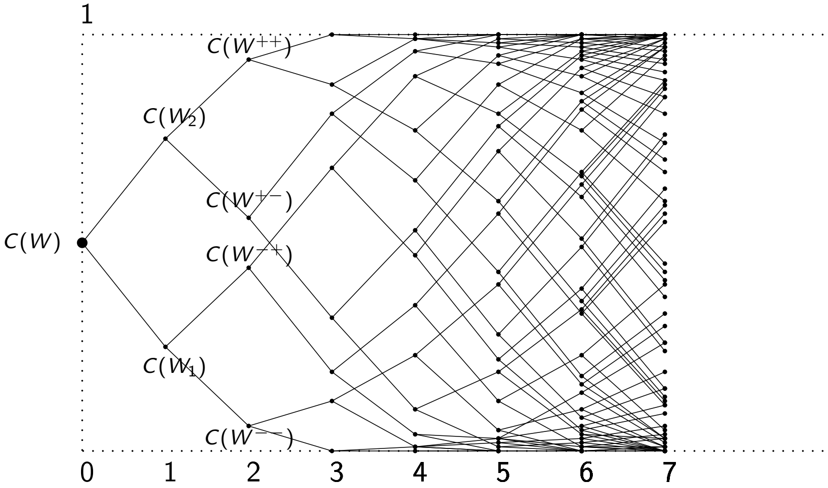
Polarization martingale



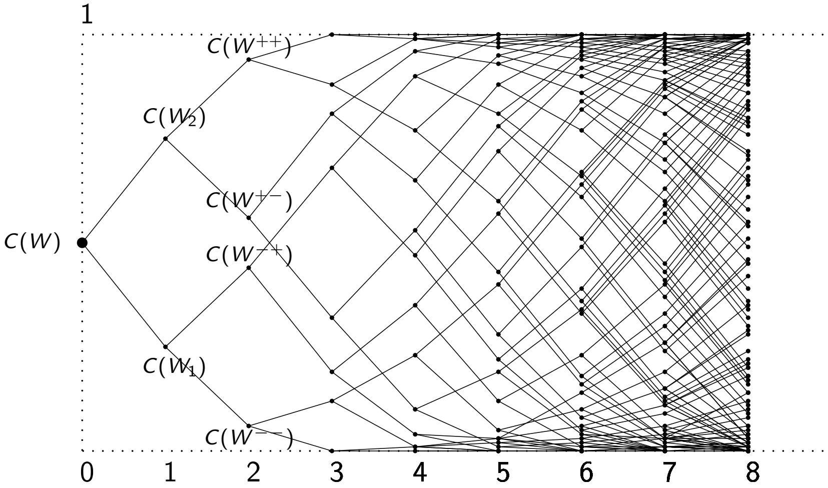
Polarization martingale



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*

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

and

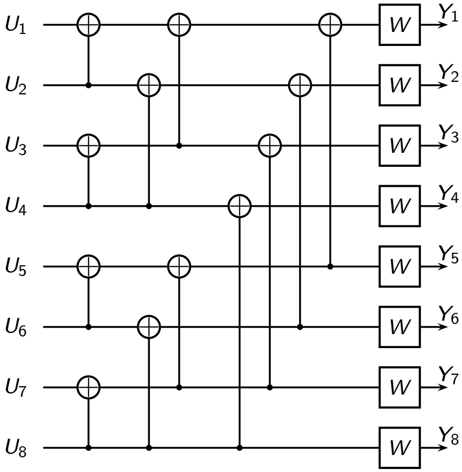
$$\left[\frac{\text{no. channels with } C(W_i) < \delta}{N}\right] \longrightarrow 1 - C(W)$$

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

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

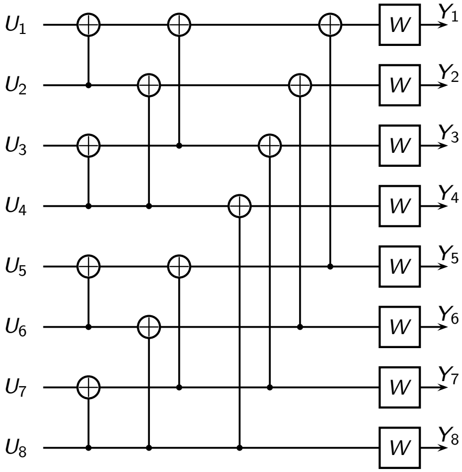
Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$C(W_i)$
0.0039
0.1211
0.1914
0.6836
0.3164
0.8086
0.8789
0.9961



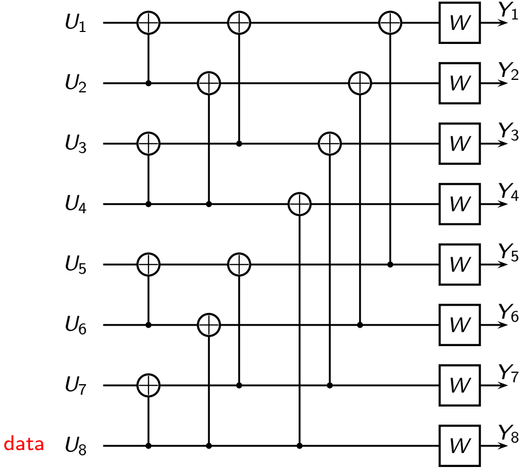
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$C(W_i)$	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



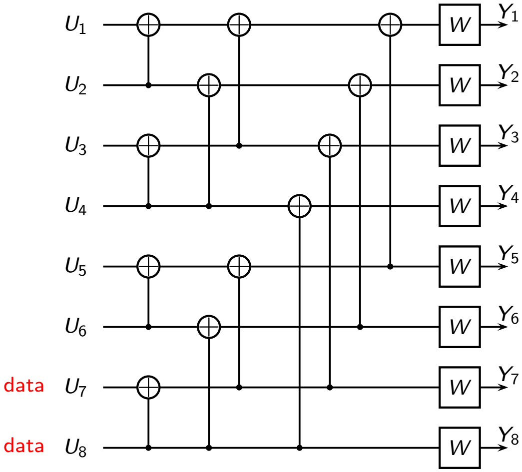
Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$C(W_i)$	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



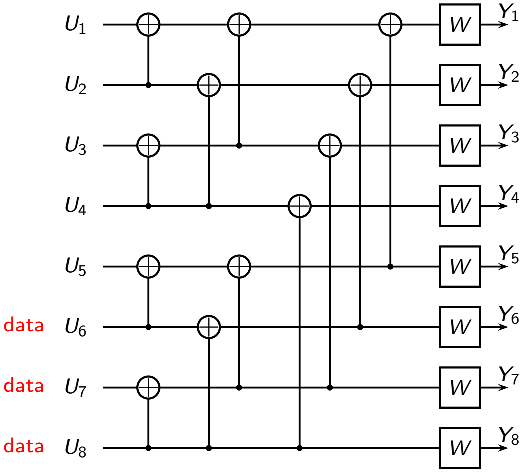
Polar code example: $W = \text{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$C(W_i)$	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



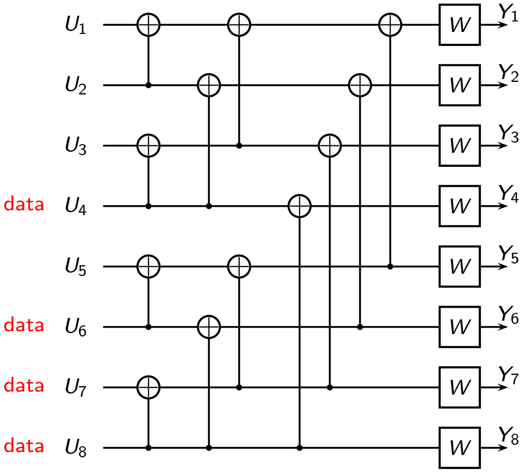
Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$C(W_i)$	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



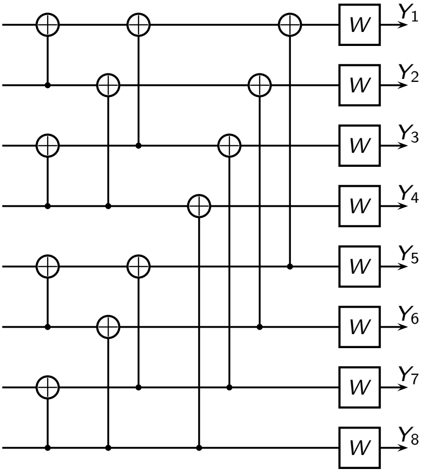
Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$C(W_i)$	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



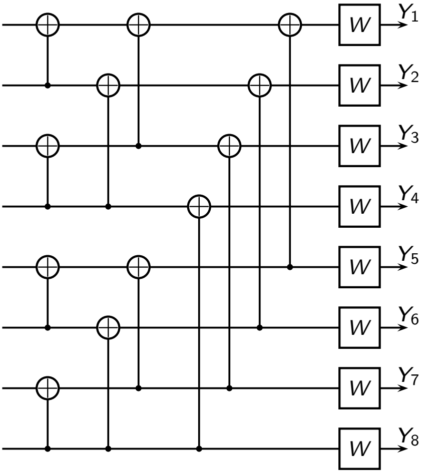
Polar code example: $W = \mathrm{BEC}(\frac{1}{2})$, $N = 8$, rate $1/2$

$C(W_i)$	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 $1/2$

$C(W_i)$	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



Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT'2011)

Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT'2011)

Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT'2011)

Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT'2011)

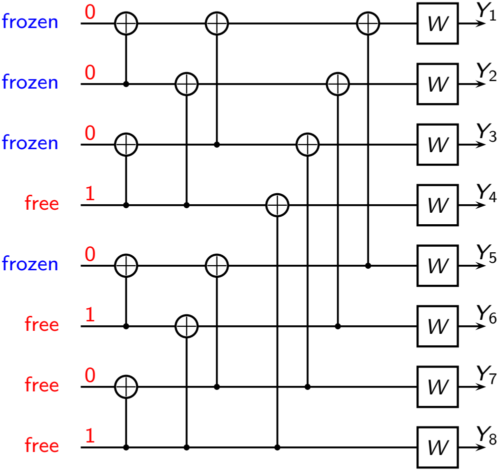
Construction complexity

- An $\mathcal{O}(N)$ construction algorithm exists that uses density-evolution
  - First proposed by Mori and Tanaka, without finite-precision implementation details
  - Tal and Vardy introduced smart quantization methods for a practical implementation
- The algorithm works well in practice but a precise proof of $\mathcal{O}(N)$ complexity still lacking
- Recent work: Pedarsani, Hassani, Tal, and Telatar (ISIT'2011)

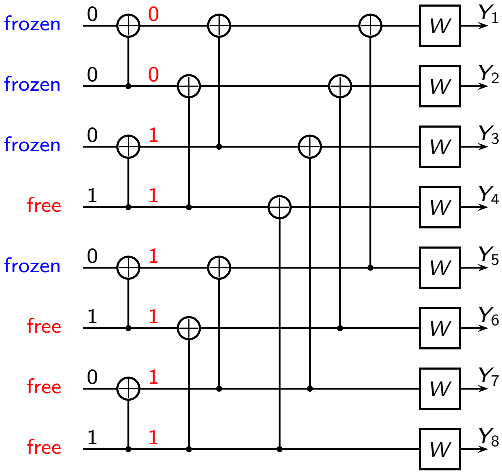
Encoding complexity

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

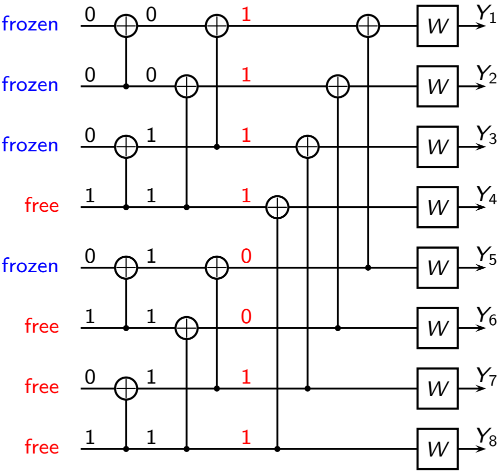
Encoding: an example



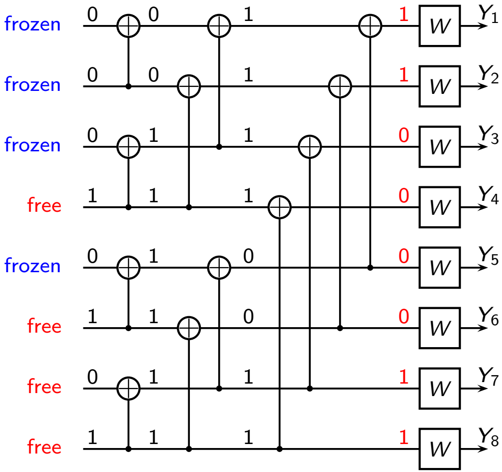
Encoding: an example



Encoding: an example



Encoding: an example



Successive cancellation decoding complexity

(A. 2007)

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

Successive cancellation decoding complexity

(A. 2007)

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

Earlier work on similar decoders:

- Kabatiansky (1990)
- Schnabl and Bossert (1996)
- Dumer and co-authors (from 1990s)
- Burnashev and Dumer (2006-2009)

Performance of SC decoder

(A. and Telatar, 2008)

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

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

- Prior result (A. 2007): $P_e(N, R) = o\left(N^{-1/4}\right)$.
- Latest result: A rate-dependent refinement of the "square-root" bound has been given by Tanaka & Mori (2010) and Hassani & Urbanke (2010)

Performance of SC decoder

(A. and Telatar, 2008)

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

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

- Prior result (A. 2007): $P_e(N, R) = o\left(N^{-1/4}\right)$.
- Latest result: A rate-dependent refinement of the "square-root" bound has been given by Tanaka & Mori (2010) and Hassani & Urbanke (2010)

Performance of SC decoder

(A. and Telatar, 2008)

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

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

- Prior result (A. 2007): $P_e(N, R) = o\left(N^{-1/4}\right)$.
- Latest result: A rate-dependent refinement of the “square-root” bound has been given by Tanaka & Mori (2010) and Hassani & Urbanke (2010)

Polar coding summary

Given W, $N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $\mathcal{O}(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$

Polar coding summary

Given W, $N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $\mathcal{O}(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$

Polar coding summary

Given W, $N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $\mathcal{O}(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$

Polar coding summary

Given W, $N = 2^n$, and $R < C(W)$, a polar code with these parameters has

- construction complexity $\mathcal{O}(N)$ (conjecture),
- encoding complexity $\approx N \log N$,
- decoding complexity $\approx N \log N$,
- and frame error probability $P_e(N, R) \approx o(2^{-\sqrt{N}})$

Polar coding in other contexts

- Source coding (lossless)
- Source coding in the presence of memory
- Lossy source coding
- Slepian-Wolf problem
- Wyner-Ziv problem
- Gelfand-Pinsker problem
- MAC
- Degraded-broadcast channel
- Wyner wiretap channel
- Randomness extraction
- ...

Channel-coding scenarios

Topic	Ref.
Single-user q-ary channels	Şaşoğlu, Telatar, A. (2009)
Multi-access channels	Şaşoğlu, Telatar, Yeh (2010)
m-user MAC	Abbe and Telatar (2010)
Wyner wiretap channel	Mahdavifar and Vardy (2009)
〃	Hof and Shamai (2010)
〃	Koyluoglu and El Gamal (2010)
〃	Andersson et al. (2010)
Relay channel	Andersson et al. (2010)
〃	Blasco-Serrano et al. (2010)
〃	Karzand (2011)
Compund channel coding	Hassani, Korada, Urbanke (2009)

Source-coding scenarios

Topic	Ref.
Lossless source coding	Hussami, Korada, Urbanke (2009)
Rate-distortion coding	Korada and Urbanke (2009)
q-ary lossless source coding	Karzand and Telatar (2010)
Direct source polarization	A. (2010)
Universal polar coding	Abbe (2010)
Sparse recovery	Abbe (2010)
Randomness extraction	Abbe (2011)
Ergodic source polarization	Şaşoğlu (2011)

Scenarios with side-information

Topic	Ref.
Wyner-Ziv coding	Korada and Urbanke (2009)
Gelfand-Pinsker coding	Korada and Urbanke (2009)
Slepian-Wolf coding	Hussami, Korada, Urbanke (2009)

Generalized polarization schemes

q: alphabet size

ℓ: dimension of basic transform (kernel)

E: rate of polarization exponent

q	ℓ	Exponent E	Kernel	Ref.
2	2 to 15	$\leq 1/2$	Any linear	KSU (2009)
2	16	0.51828	BCH	KSU (2009)
2	31	0.52643	BCH	KSU (2009)
4	4	0.573120	Reed-Solomon	MT (2010)
2	14	0.50193	Nonlinear	PSL (2011)
2	15	0.50773	Nonlinear	PSL (2011)
2	16	0.52742	Nonlinear	PSL (2011)

KSU: Korada, Şaşoğlu, Urbanke

MT: Mori and Tanaka

PSL: Presman, Shapira, Litsyn

Performance comparison: Polar vs. Turbo

Turbo code

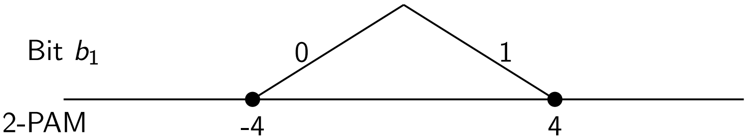
- WiMAX CTC
- Duobinary, memory 3
- QAM over AWGN channel
- Gray mapping
- BICM
- Simulator: “Coded Modulation Library”

Polar code

- Standard construction
- Successive cancellation decoding
- QAM over AWGN channel
- Natural mapping
- Multi-level PAM
- PAM over AWGN channel

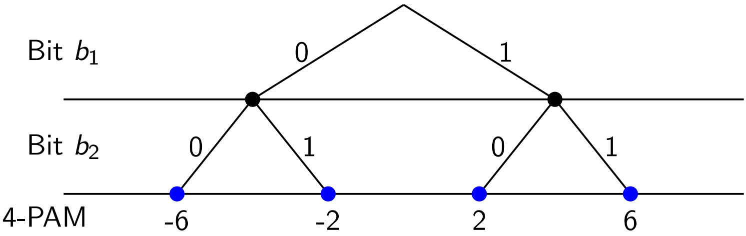
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits (b_1, b_2, b_3)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order b_3, b_2, b_1



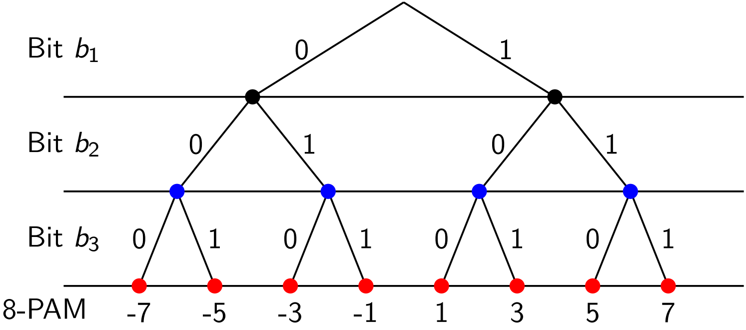
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits (b_1, b_2, b_3)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order b_3, b_2, b_1



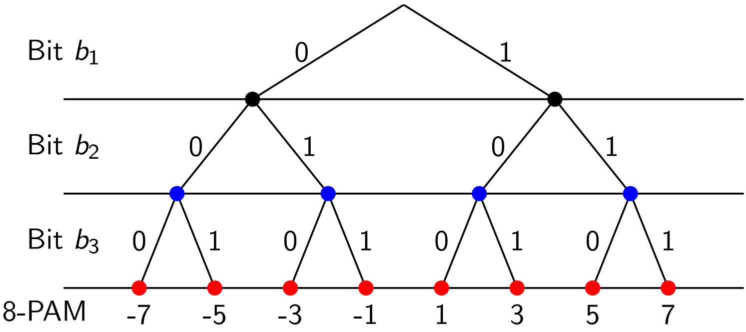
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits (b_1, b_2, b_3)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order b_3, b_2, b_1



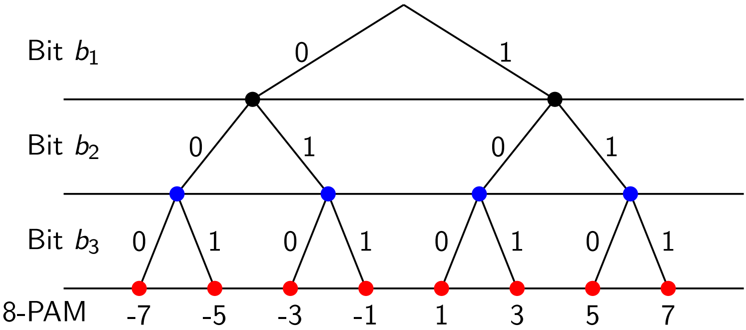
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits (b_1, b_2, b_3)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order b_3, b_2, b_1



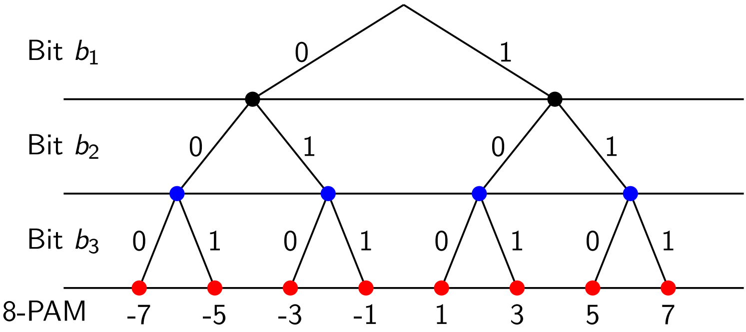
Example: 8-PAM as 3 bit channels

- PAM signals selected by three bits (b_1, b_2, b_3)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order b_3, b_2, b_1

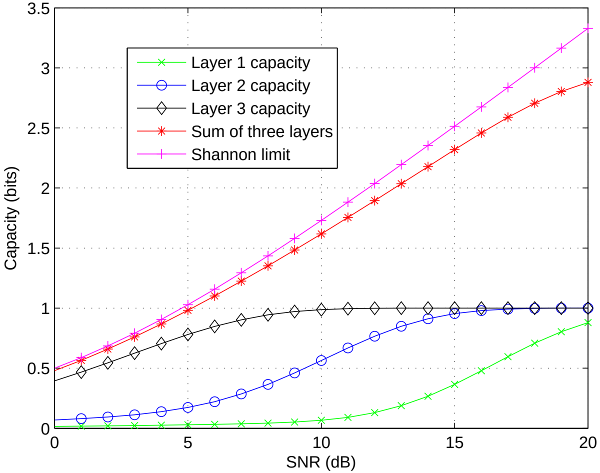


Example: 8-PAM as 3 bit channels

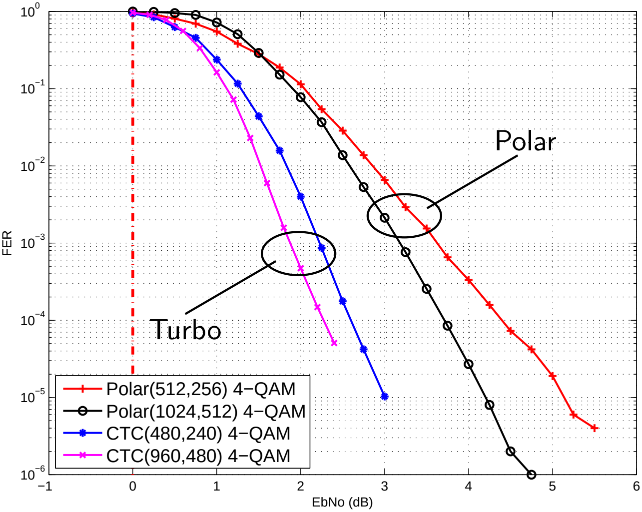
- PAM signals selected by three bits (b_1, b_2, b_3)
- Three layers of binary channels created
- Each layer encoded independently
- Layers decoded in the order b_3, b_2, b_1



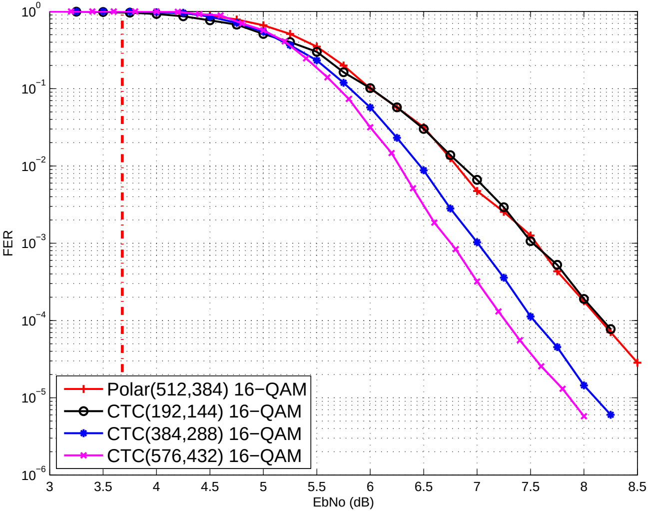
Multi-layering jump-starts polarization



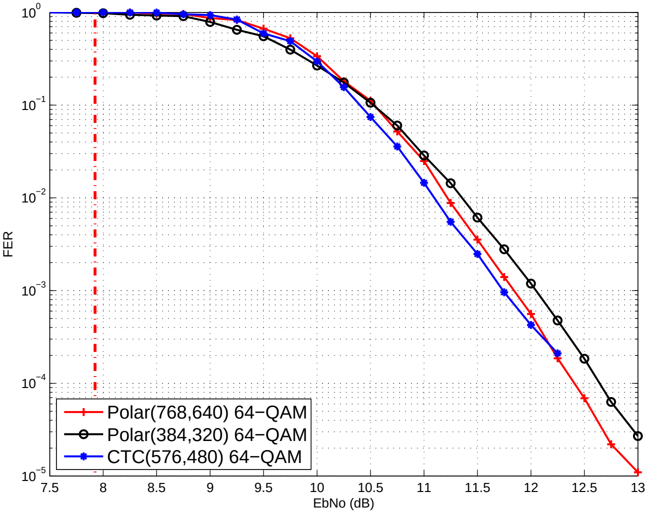
4-QAM, Rate 1/2


Polar
Turbo
Polar(512,256) 4−QAM
Polar(1024,512) 4−QAM
CTC(480,240) 4−QAM
CTC(960,480) 4−QAM
FER
EbNo (dB)

16-QAM, Rate 3/4



64-QAM, Rate 5/6



Complexity comparison: 64-QAM, Rate 5/6

Average decoding time in milliseconds per codeword (ms/cw)

E_b/N_0	CTC(576,432)	Polar(768,640)	Polar(384,320)
10 dB	6.23	0.92	0.48
11 dB	1.83	1.01	0.53

Both decoders implemented as MATLAB mex functions. Polar decoder is a successive cancellation decoder. CTC decoder is a public domain decoder (CML). Profiling done by MATLAB Profiler. Iteration limit for CTC decoder was 10; average no of iterations was 10 at 10 dB and 3.3 at 11 dB. CTC decoder used a linear approximation to log-MAP while polar decoder used exact log-MAP.

Complexity comparison: 64-QAM, Rate 5/6

Average decoding time in milliseconds per codeword (ms/cw)

E_b/N_0	CTC(576,432)	Polar(768,640)	Polar(384,320)
10 dB	6.23	0.92	0.48
11 dB	1.83	1.01	0.53

Polar codes show a complexity advantage against CTC codes.

Both decoders implemented as MATLAB mex functions. Polar decoder is a successive cancellation decoder. CTC decoder is a public domain decoder (CML). Profiling done by MATLAB Profiler. Iteration limit for CTC decoder was 10; average no of iterations was 10 at 10 dB and 3.3 at 11 dB. CTC decoder used a linear approximation to log-MAP while polar decoder used exact log-MAP.

Performance improvement for polar codes

- Concatenation to improve minimum distance
- List decoding to improve SC decoder performance

Performance improvement for polar codes

- Concatenation to improve minimum distance
- List decoding to improve SC decoder performance

Concatenation

Method	Ref
Block turbo coding with polar constituents	AKMOP (2009)
Generalized concatenated coding with polar inner	AM (2009)
Reed-Solomon outer, polar inner	BJE (2010)
Polar outer, block inner	SH (2010)
Polar outer, LDPC inner	EP (ISIT'2011)

AKMOP: A., Kim, Markarian, Özgür, Poyraz
GCC: A., Markarian
BJE: Bakshi, Jaggi, and Effros
SH: Seidl and Huber
EP: Eslami and Pishro-Nik

Tal-Vardy list decoder for polar codes

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

Tal-Vardy list decoder for polar codes

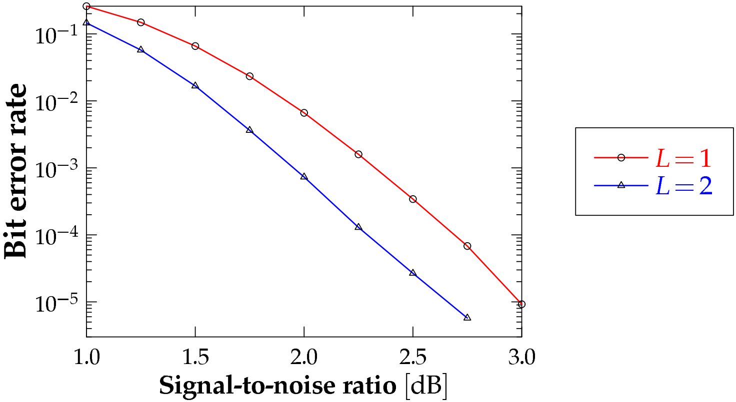
- First produce L candidate decisions
- Pick the most likely word from the list
- Complexity $\mathcal{O}(LN \log N)$

Tal-Vardy list decoder for polar codes

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

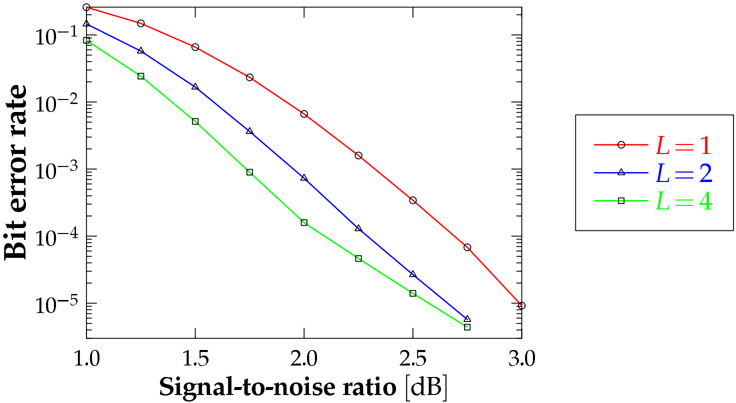
Tal-Vardy list decoder performance

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



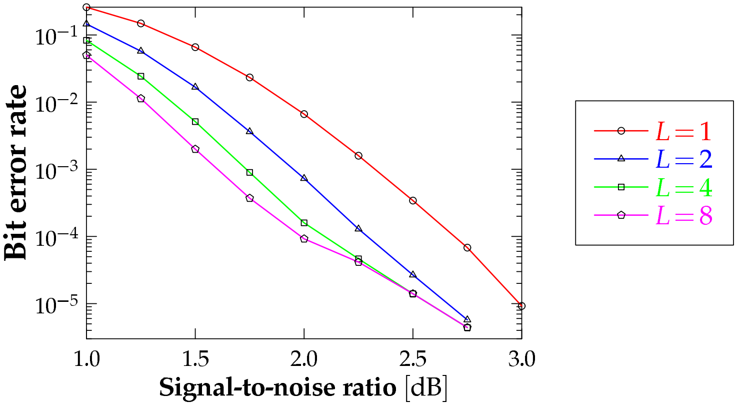
Tal-Vardy list decoder performance

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



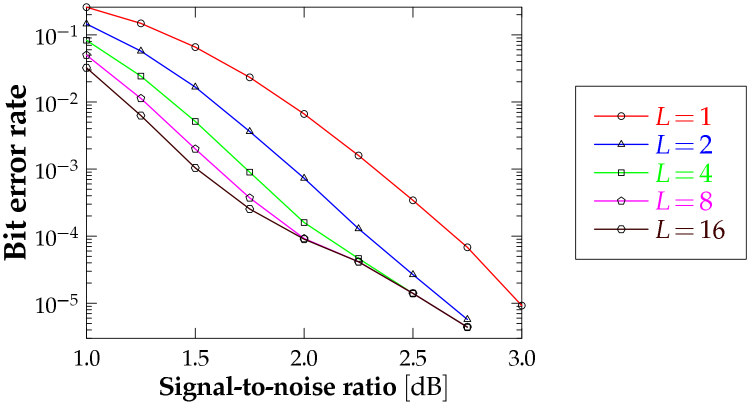
Tal-Vardy list decoder performance

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



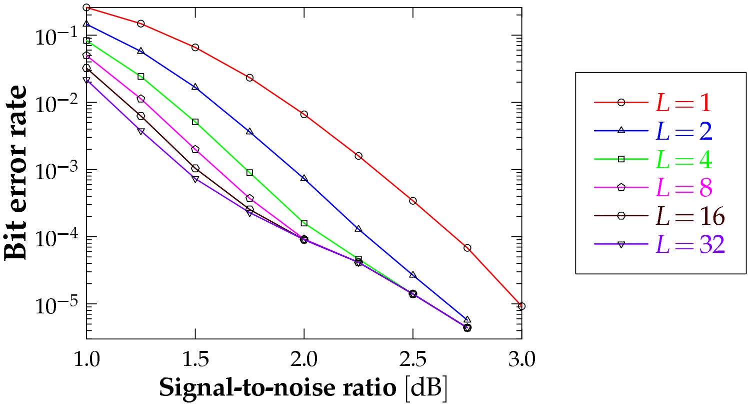
Tal-Vardy list decoder performance

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



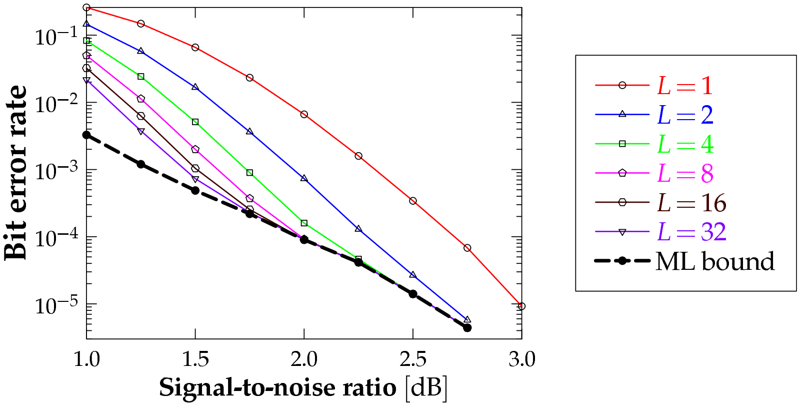
Tal-Vardy list decoder performance

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



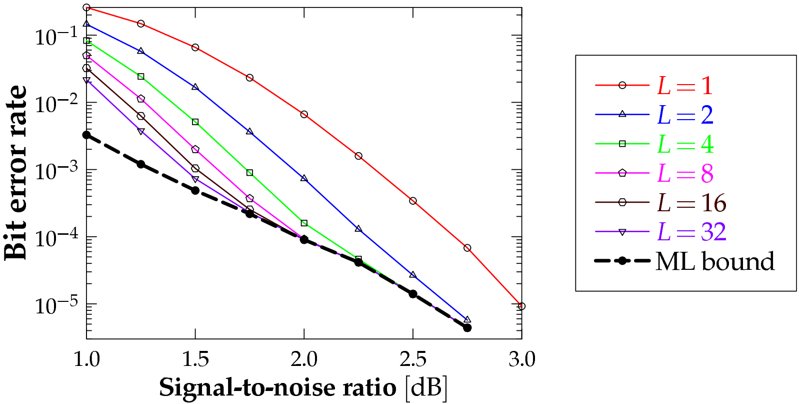
Tal-Vardy list decoder performance

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!

Tal-Vardy list decoder with CRC

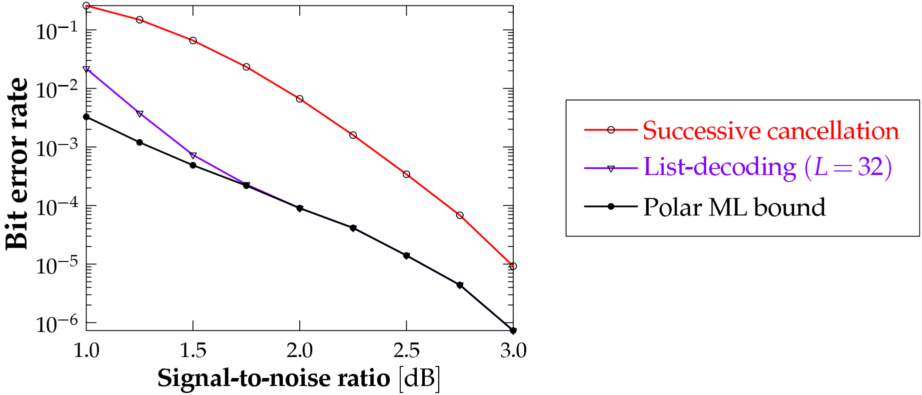
- Same decoder as before but data contains a built-in CRC
- Selection made by CRC and relative likelihood

Tal-Vardy list decoder with CRC

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

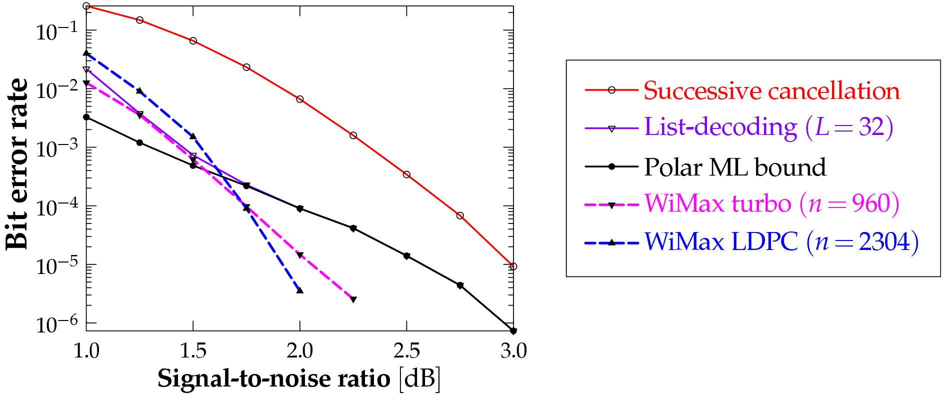
Tal-Vardy list decoder with CRC

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



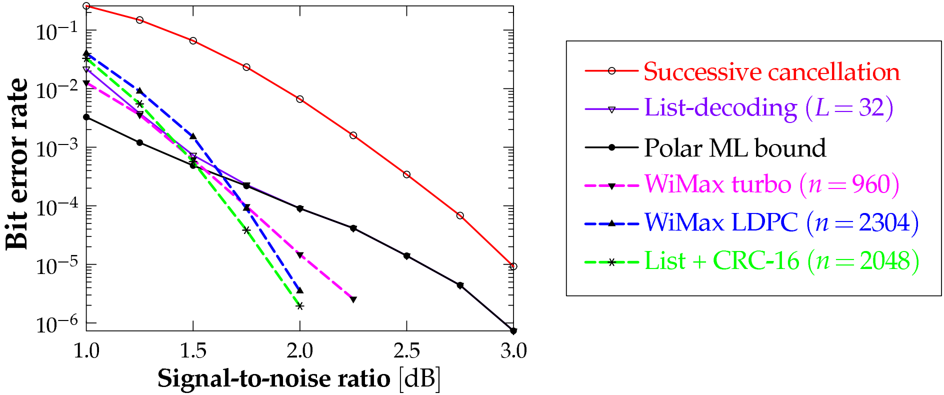
Tal-Vardy list decoder with CRC

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!

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts
- Disadvantages
  - High latency: $\mathcal{O}(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

References: A. (2007, 2010), Leroux, Tal, Vardy, Gross (2010), Leroux, Sarkis, and Gross (2011), Pamuk and A. (2011).

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts
- Disadvantages
  - High latency: $\mathcal{O}(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

References: A. (2007, 2010), Leroux, Tal, Vardy, Gross (2010), Leroux, Sarkis, and Gross (2011), Pamuk and A. (2011).

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts
- Disadvantages
  - High latency: $\mathcal{O}(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

References: A. (2007, 2010), Leroux, Tal, Vardy, Gross (2010), Leroux, Sarkis, and Gross (2011), Pamuk and A. (2011).

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts
- Disadvantages
  - High latency: $\mathcal{O}(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

References: A. (2007, 2010), Leroux, Tal, Vardy, Gross (2010), Leroux, Sarkis, and Gross (2011), Pamuk and A. (2011).

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts
- Disadvantages
  - High latency: $\mathcal{O}(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

References: A. (2007, 2010), Leroux, Tal, Vardy, Gross (2010), Leroux, Sarkis, and Gross (2011), Pamuk and A. (2011).

Hardware implementation of polar codes

- Advantages
  - Regular structure simplifies resource reuse
  - Lack of randomness helps avoid memory conflicts
- Disadvantages
  - High latency: $\mathcal{O}(N)$
  - Throughput bottleneck: 1/2 bits per clock-period

References: A. (2007, 2010), Leroux, Tal, Vardy, Gross (2010), Leroux, Sarkis, and Gross (2011), Pamuk and A. (2011).

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

Acknowledgements

Research sponsors

intel

TÜBİTAK

Special thanks to Emre Telatar, Alex Vardy, Ido Tal, and Warren Gross for sharing ideas, software and slides, and to Matthew Valenti for *Coded Modulation Library.*

Thank you!