EE 276: Information Theory

Polar Codes

Mert Pilanci

Stanford University

February 15, 2024

Outline

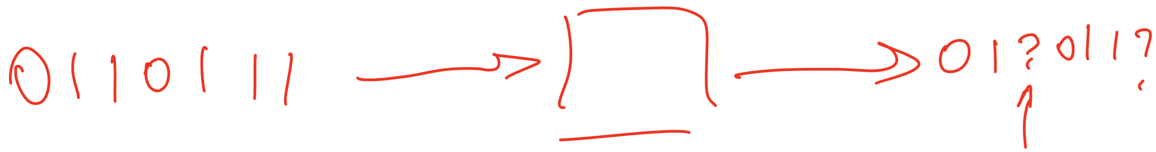
- Polar code construction
- Achieving channel capacity
- Decoding
- Applications and Extensions

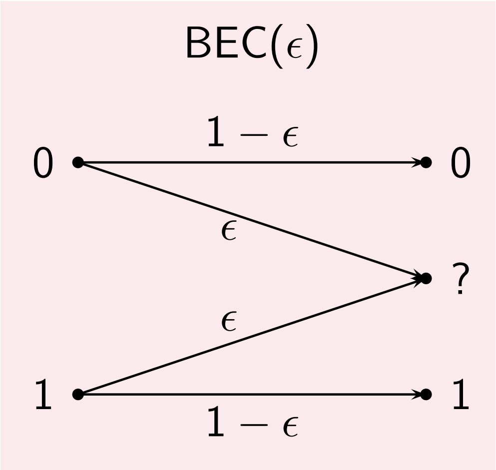
Channel Capacity

- Channel capacity C is the maximal rate of reliable communication
- Shannon's Second Fundamental Theorem (from Lecture 7) :

$$C = \max_{P_X} I(X;Y)$$

Capacity of the binary erasure channel (BEC)





Capacity of the BEC with erasure probability ϵ is $C = 1 - \epsilon$

Channel Coding

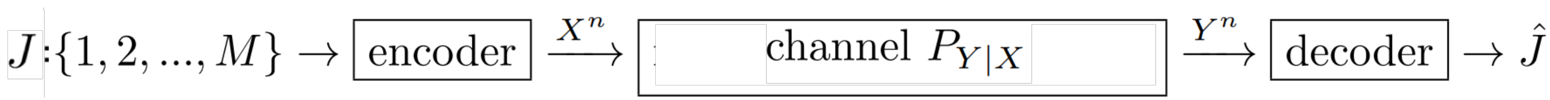
$J : \{1, 2, ..., M\} \rightarrow$ [encoder] $\xrightarrow{X^n}$ [channel $P_{Y|X}$] $\xrightarrow{Y^n}$ [decoder] $\rightarrow \hat{J}$

$K = \log_2 M$

rate: $R = \frac{\log M}{n}$ bits/channel use

probability of error $P_{\text{error}} =$ Probability$[\hat{J} \neq J]$

Channel Coding



rate: $R = \frac{\log M}{n}$ bits/channel use

probability of error $P_{\text{error}} = \text{Probability}[\hat{J} \neq J]$

- If $R < C$, then there exists a communication scheme with rate $\geq R$ and probability of error: $P_{\text{error}} \to 0$

Channel Coding

$$J:\{1,2,...,M\} \rightarrow \boxed{\text{encoder}} \xrightarrow{X^n} \boxed{\text{channel } P_{Y|X}} \xrightarrow{Y^n} \boxed{\text{decoder}} \rightarrow \hat{J}$$

rate: $R = \frac{\log M}{n}$ bits/channel use

probability of error $P_{\text{error}} =$ Probability$[\hat{J} \neq J]$

- If $R < C$, then there exists a communication scheme with rate $\geq R$ and probability of error: $P_{\text{error}} \rightarrow 0$
- If $R > C$, then rate R is not achievable (P_{error} is large)

Shannon's Second Theorem: Maximum rate of reliable communication is $C = \max_{P_X} I(X;Y)$

Shannon's Coding Method

11

0 1 1 0

0 1 ? ?

- random codebook (from Lecture 11) 0 1 1 0 1

codeword 1	0	1	1	0	1	...
codeword 2	1	1	0	1	0	...
codeword 3	1	1	0	1	0	...
codeword 4	0	0	0	0	1	...
codeword 5	1	0	0	0	0	...
codeword 6	0	1	0	0	1	...
codeword 7	0	0	1	1	0	...
codeword 8	1	0	0	0	1	...
⋮						

- not explicitly constructed
- shows the existence of good codes
- not computationally efficient

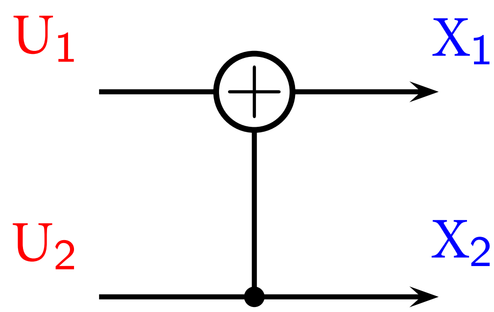
"Almost all codes are "good" codes except for the the ones that we can think of..."
Jack Wolfe

Today: Polar Codes

- Invented by Erdal Arıkan in 2009
- First code with an explicit construction to provably achieve the channel capacity
- Efficient encoding/decoding operations

(channel coding vs source coding)

Basic 2×2 transformation



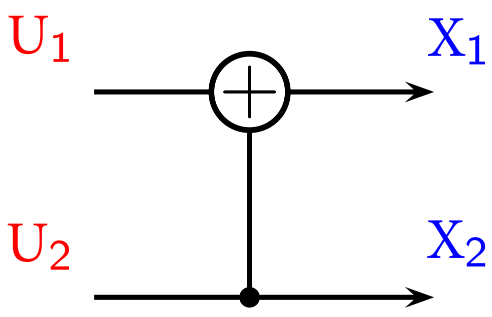
$X_1 = U_1 \oplus U_2 = U_1 + U_2 \mod 2$

XOR

$U_1 \oplus U_2$	$U_2 = 0$	$U_2 = 1$
$U_1 = 0$	0	1
$U_1 = 1$	1	0

$U_1, U_2 \in \{0, 1\}$ two input bits
$X_1, X_2 \in \{0, 1\}$ two output bits

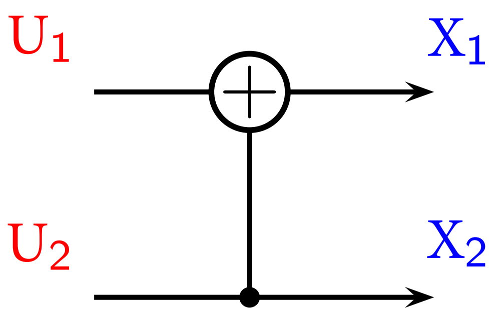
Basic 2×2 transformation



$U_1, U_2 \in \{0,1\}$ two input bits
$X_1, X_2 \in \{0,1\}$ two output bits

$$X_1 = U_1 \oplus U_2 = \; U_1 \text{ XOR } U_2$$
$$X_2 = U_2$$

Basic 2×2 transformation



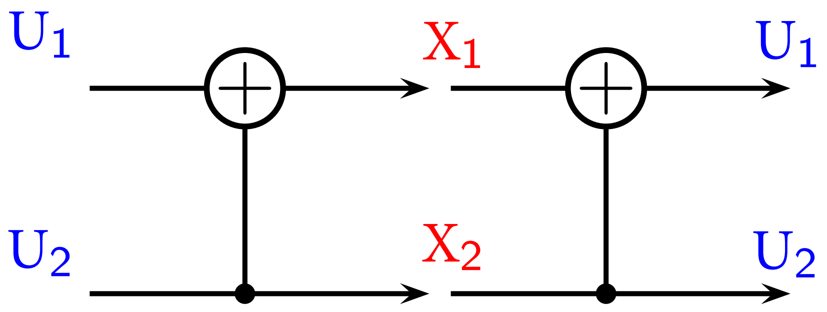
$U_1, U_2 \in \{0,1\}$ two input bits
$X_1, X_2 \in \{0,1\}$ two output bits

$$X_1 = U_1 \oplus U_2 = \; U_1 \text{ XOR } U_2$$
$$X_2 = U_2$$

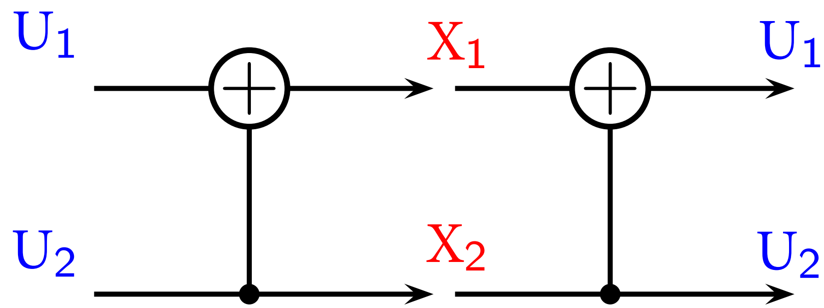
alternatively

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} \quad \text{modulo } 2$$

Inverting the transform



Inverting the transform

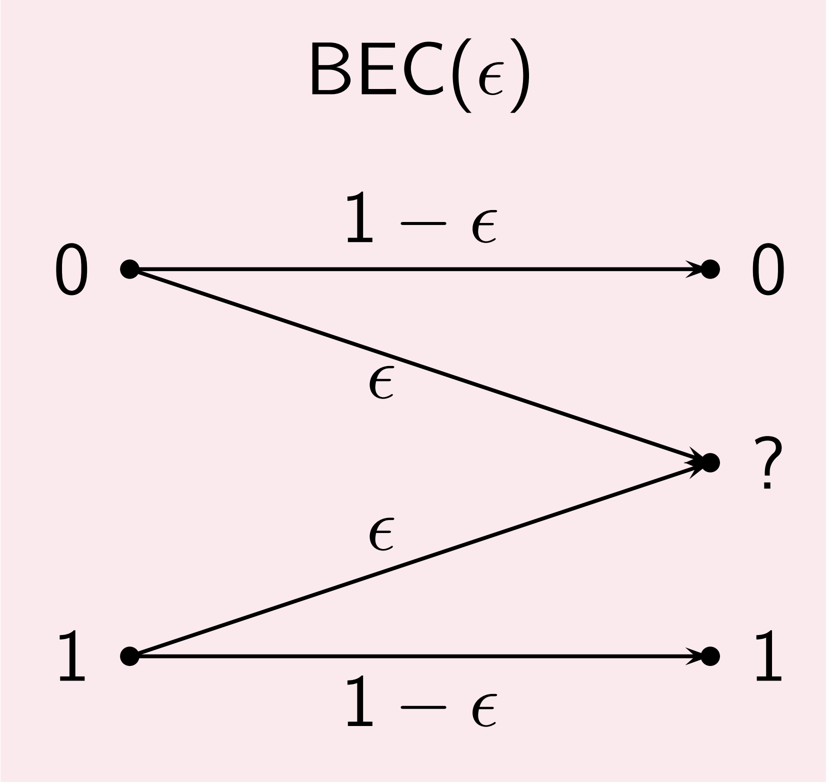


$$2 \times 2 \text{ transformation} \qquad G_2 := \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$G_2 G_2 U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$$

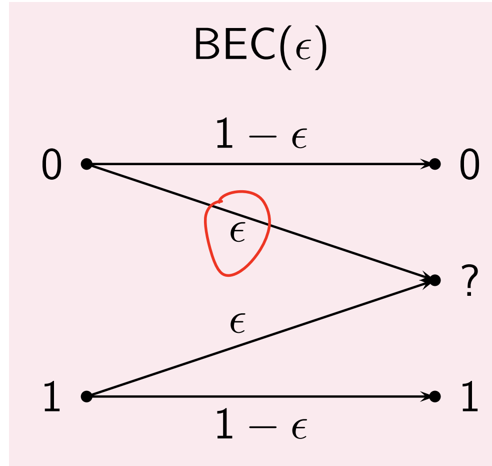
$$= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} U_1 \oplus U_2 \\ U_2 \end{bmatrix} = \begin{bmatrix} U_1 \oplus U_2 \oplus U_2 \\ U_2 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$$

Erasure channel

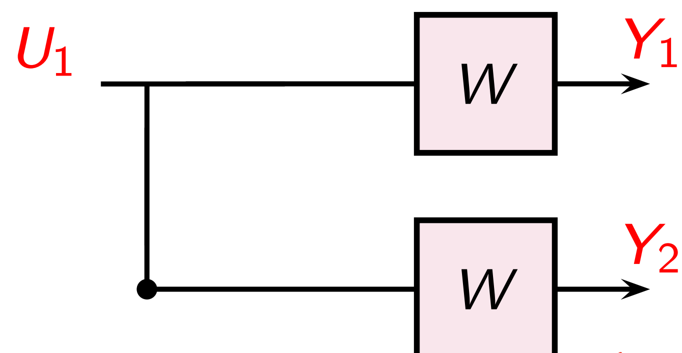


Naively combining erasure channels

W



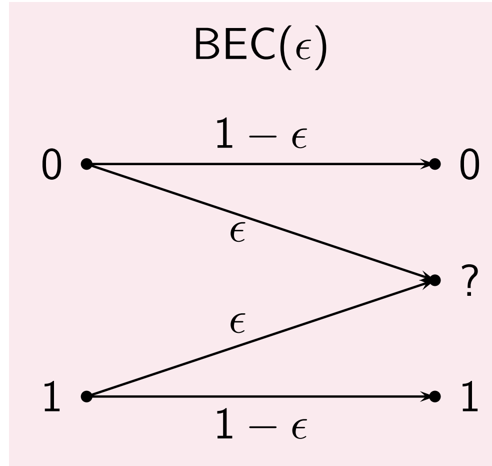
- Repetition coding



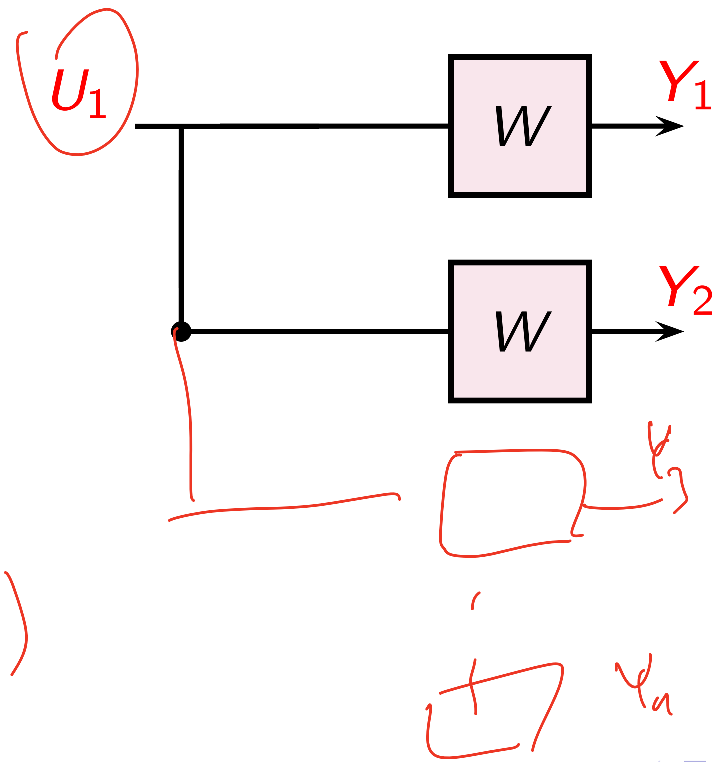
$P(U_1 \text{ is erased}) = P(Y_1 \text{ erased } \& \; Y_2 \text{ erased})$

$= \epsilon \cdot \epsilon = \epsilon^2$

Repetition code with rate $1/n$



- Repetition coding



0.1

$(0.1)^n$

$\text{Rate} = \frac{1}{n}$

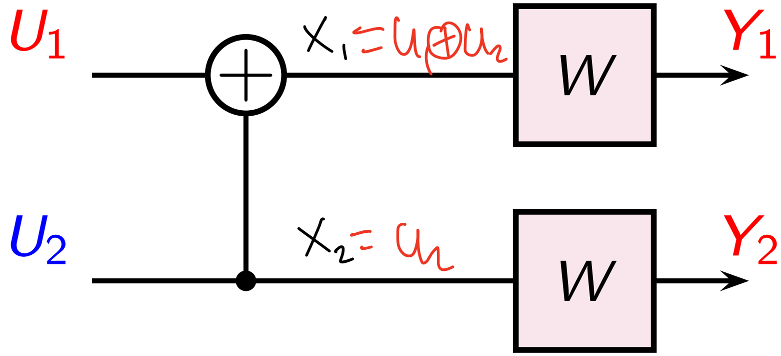
$\text{Rate} = 0.5$

$P(U_1 \text{ is erased}) =$

$P(\text{all } Y_i \text{ erased } i \in [n])$

$= \epsilon^n$

Combining two erasure channels



Invertible transformation does not alter capacity or mutual information: $I(U;Y) = I(X;Y)$

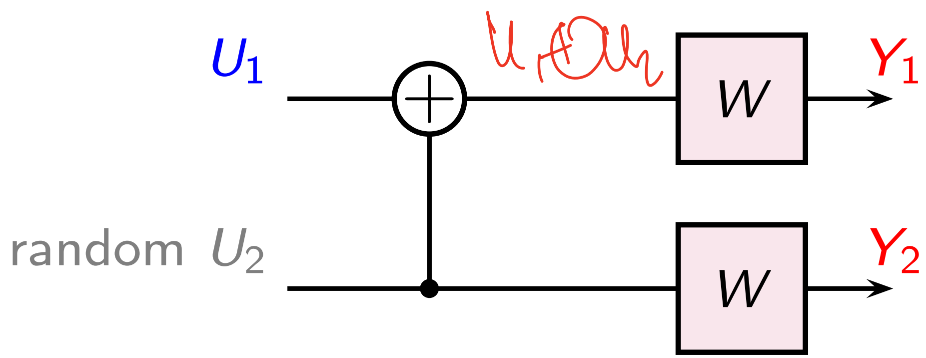
Sequential decoding: Decode U_1 and U_2 one by one

$U_1 \oplus U_2 \oplus U_2 = U_1$

$2 \cdot 0.1 - (0.1)^2$

First bit-channel $W_1 : U_1 \to (Y_1, Y_2)$

$\epsilon = 0.1$



$P(U_1 \text{ is erased}) = P(Y_1 \text{ erased or } Y_2 \text{ erased})$

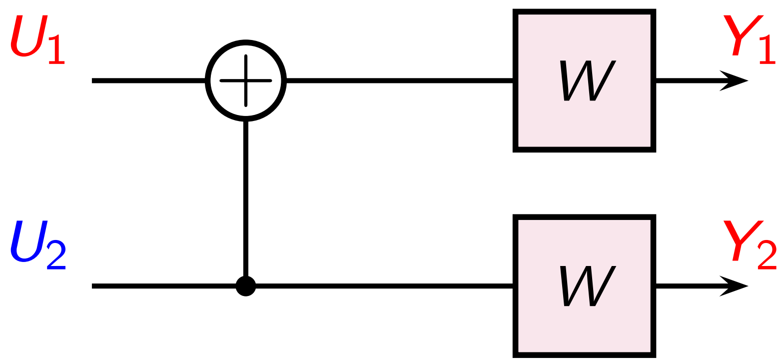
$= 1 - (1-\epsilon)^2$

$= 1 - (1 - 2\epsilon + \epsilon^2) = 2\epsilon - \epsilon^2$

probability	Y_1 erased	Y_1 not erased
Y_2 erased	$\epsilon \cdot \epsilon$	$(1-\epsilon)\epsilon$
Y_2 not erased	$(1-\epsilon)\epsilon$	$(1-\epsilon)^2$

suppose a '**genie**' provides this bit

Second bit-channel $W_2 : U_2 \to (Y_1, Y_2, U_1)$

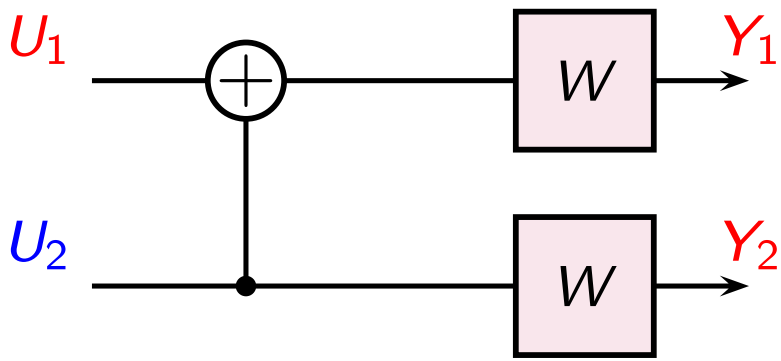


suppose a **'genie'** provides this bit

Second bit-channel $W_2 : U_2 \to (Y_1, Y_2, U_1)$

GPT4 prompt: generate an artistic depiction of a genie aided channel decoder

Second bit-channel $W_2 : U_2 \to (Y_1, Y_2, U_1)$

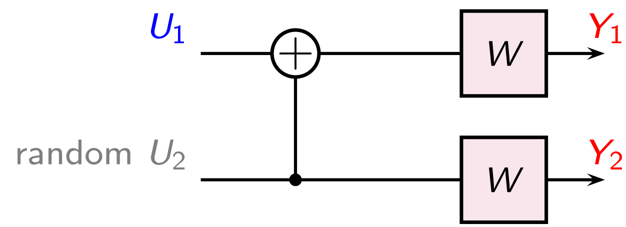


$P(U_2 \text{ is erased}) = \epsilon^2$

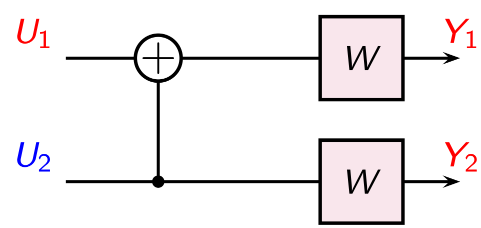
probability	Y_1 erased	Y_1 not erased
Y_2 erased	ϵ^2	$\epsilon(1-\epsilon)$
Y_2 not erased	$\epsilon(1-\epsilon)$	$(1-\epsilon)^2$

Two different cases: W_1 and W_2

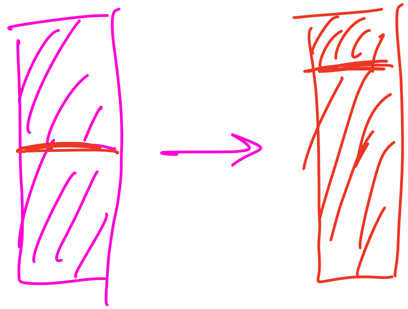
- W_1: Decoding U_1 when U_2 is **not** available
  U_1 is erased when Y_1 is erased **or** Y_2 is erased
  Failure probability $= 1 - (1-\epsilon)^2 = 2\epsilon - \epsilon^2$ **worse erasure channel**



- W_2: Decoding U_2 when U_1 is **available**
  U_2 is erased when Y_1 is erased **and** Y_2 is erased
  Failure probability $= \epsilon^2$ **better erasure channel**

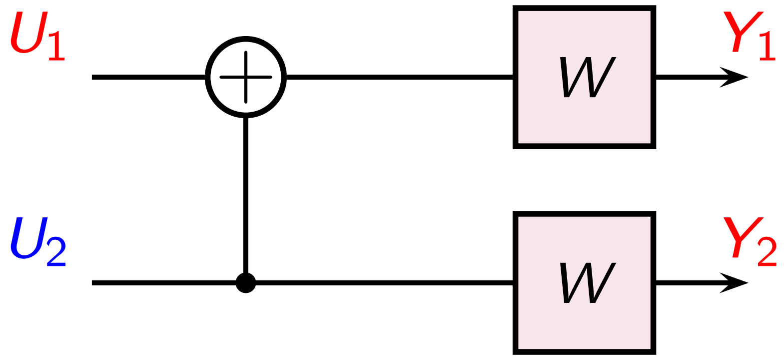


Capacity is conserved

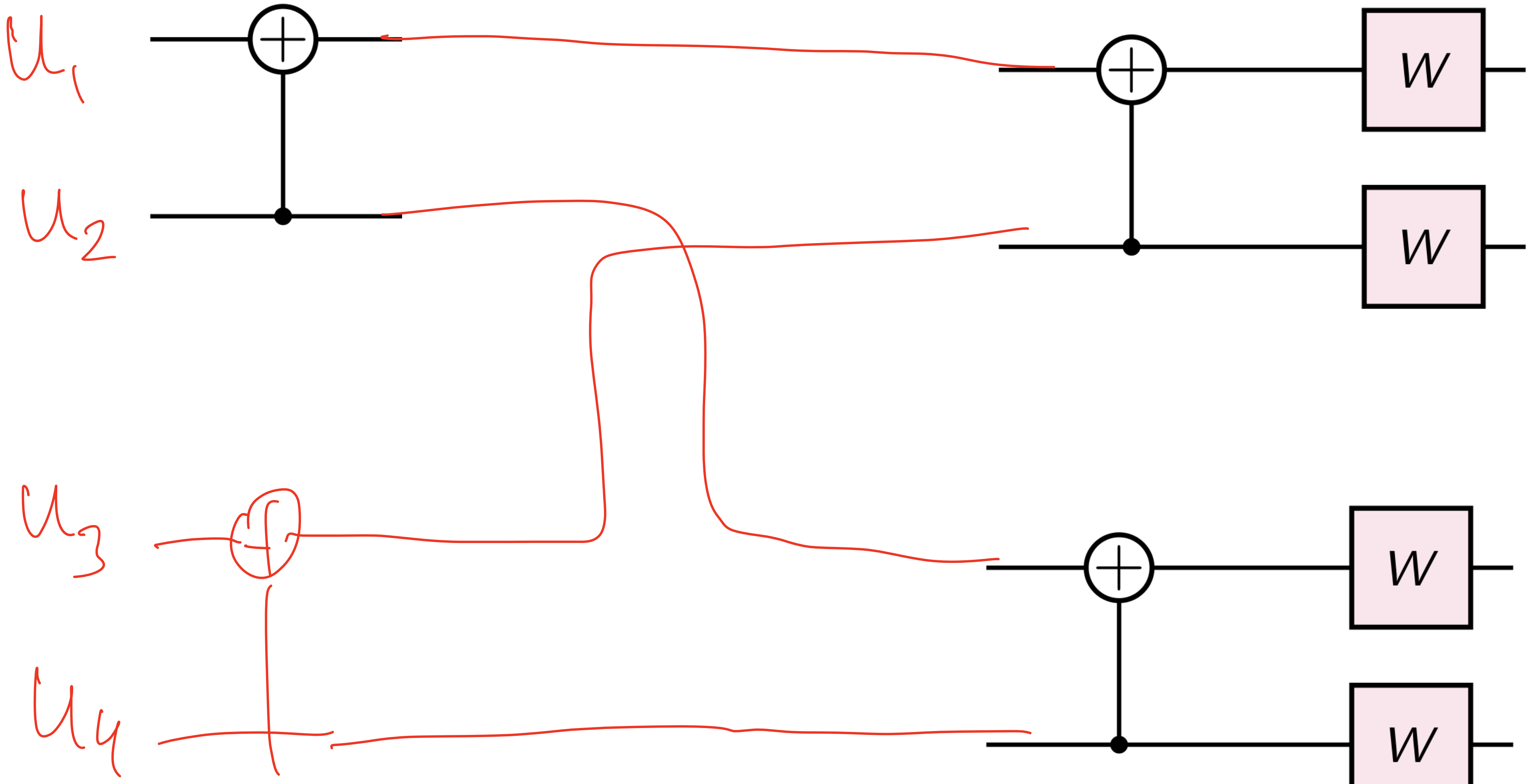


$$C(W_1) + C(W_2) = C(W) + C(W) = 2C(W)$$

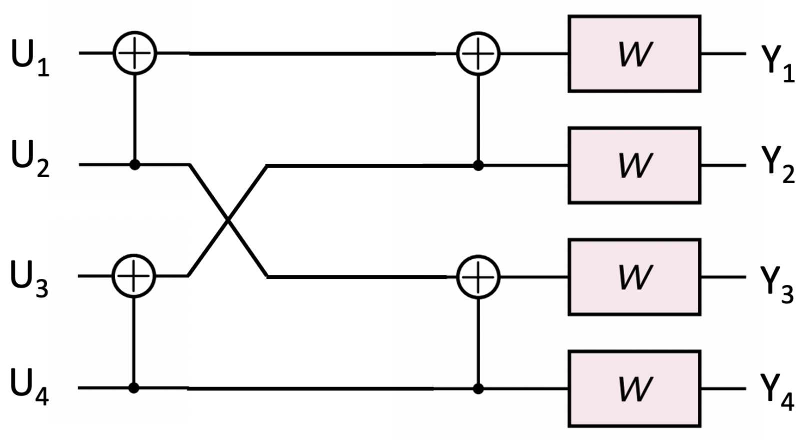
$$C(W_1) \leq C(W) \leq C(W_2)$$



Extending the size



Extending the size

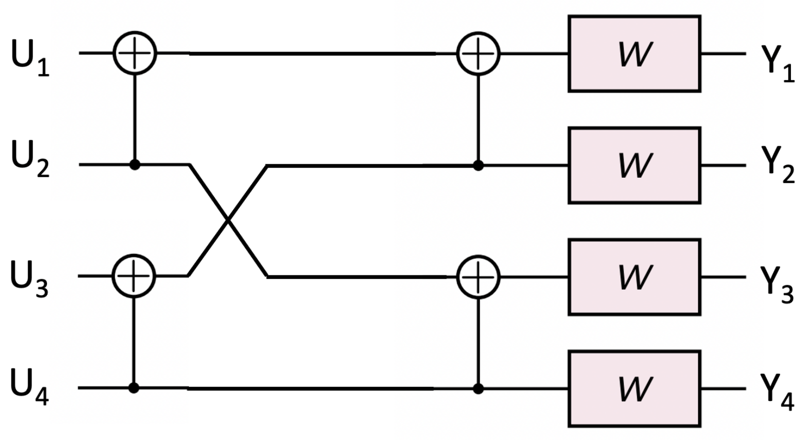


Sequential decoding:

- Decode U_1 from Y_1, Y_2, Y_3, Y_4

  erased if (Y_1 or Y_2 erased) **or** (Y_3 or Y_4 erased)

Extending the size



Sequential decoding:

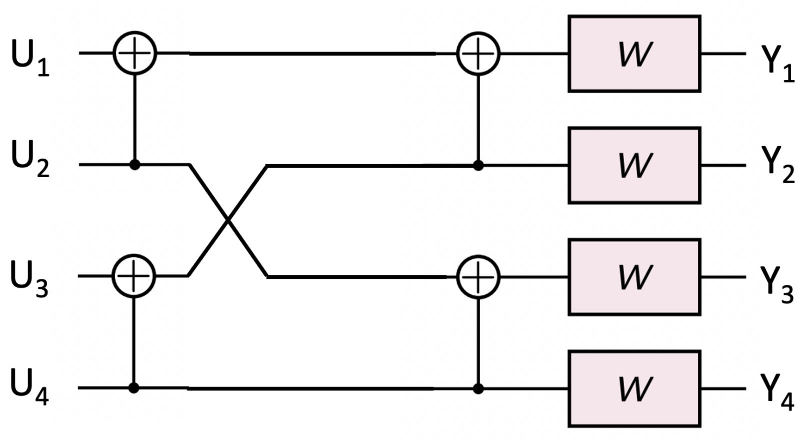
- Decode U_1 from Y_1, Y_2, Y_3, Y_4

  erased if (Y_1 or Y_2 erased) **or** (Y_3 or Y_4 erased)

- Decode U_2 from Y_1, Y_2, Y_3, Y_4, U_1

  erased if (Y_1 or Y_2 is erased) **and** (Y_3 or Y_4 is erased)

Extending the size



Sequential decoding:

- Decode U_1 from Y_1, Y_2, Y_3, Y_4

  erased if (Y_1 or Y_2 erased) **or** (Y_3 or Y_4 erased)
- Decode U_2 from Y_1, Y_2, Y_3, Y_4, U_1

  erased if (Y_1 or Y_2 is erased) **and** (Y_3 or Y_4 is erased)
- Decode U_3 from $Y_1, Y_2, Y_3, Y_4, U_1, U_2$

  erased if (Y_1 and Y_2 is erased) **or** (Y_3 and Y_4 is erased)
- Decode U_4 from $Y_1, Y_2, Y_3, Y_4, U_1, U_2, U_3$

  erased if (Y_1 and Y_2 erased) **and** (Y_3 and Y_4 erased)

Recursive Calculation of Failure Probability

Sequential decoding:

$\hat{\epsilon} = 2\epsilon - \epsilon^2$

- Decode U_1

  erased if (Y_1 or Y_2 erased) **or** (Y_3 or Y_4 erased)

failure probability$= 2\hat{\epsilon} - \hat{\epsilon}^2 = 2(2\epsilon - \epsilon^2) - (2\epsilon - \epsilon^2)^2$

Recursive Calculation of Failure Probability

Sequential decoding:

- Decode U_1

  erased if (Y_1 or Y_2 erased) **or** (Y_3 or Y_4 erased)

  failure probability$= 2\hat{\epsilon} - \hat{\epsilon}^2 = 2(2\epsilon - \epsilon^2) - (2\epsilon - \epsilon^2)^2$

- Decode U_2

  erased if (Y_1 or Y_2 is erased) **and** (Y_3 or Y_4 is erased)

  failure probability$= \hat{\epsilon}^2 = (2\epsilon - \epsilon^2)^2$

Recursive Calculation of Failure Probability

Sequential decoding:

- Decode U_1

  erased if (Y_1 or Y_2 erased) **or** (Y_3 or Y_4 erased)

  failure probability$= 2\hat{\epsilon} - \hat{\epsilon}^2 = 2(2\epsilon - \epsilon^2) - (2\epsilon - \epsilon^2)^2$
- Decode U_2

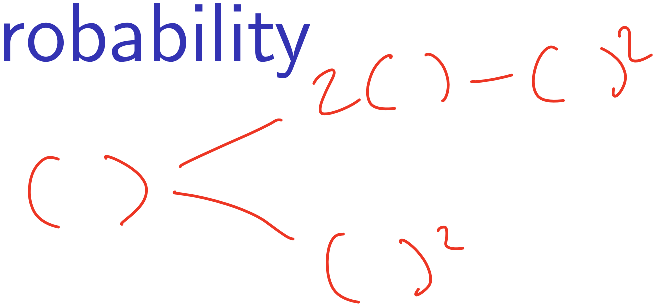
  erased if (Y_1 or Y_2 is erased) **and** (Y_3 or Y_4 is erased)

  failure probability$= \hat{\epsilon}^2 = (2\epsilon - \epsilon^2)^2$
- Decode U_3

  erased if (Y_1 and Y_2 is erased) **or** (Y_3 and Y_4 is erased)

  failure probability$= 2\tilde{\epsilon} - \tilde{\epsilon}^2 = 2(\epsilon^2) - (\epsilon^2)^2$

Recursive Calculation of Failure Probability



Sequential decoding:

- Decode U_1

  erased if (Y_1 or Y_2 erased) **or** (Y_3 or Y_4 erased)

  failure probability$= 2\hat{\epsilon} - \hat{\epsilon}^2 = 2(2\epsilon - \epsilon^2) - (2\epsilon - \epsilon^2)^2$

- Decode U_2

  erased if (Y_1 or Y_2 is erased) **and** (Y_3 or Y_4 is erased)

  failure probability$= \hat{\epsilon}^2 = (2\epsilon - \epsilon^2)^2$

- Decode U_3

  $\tilde{\epsilon} = \epsilon^2$

  erased if (Y_1 and Y_2 is erased) **or** (Y_3 and Y_4 is erased)

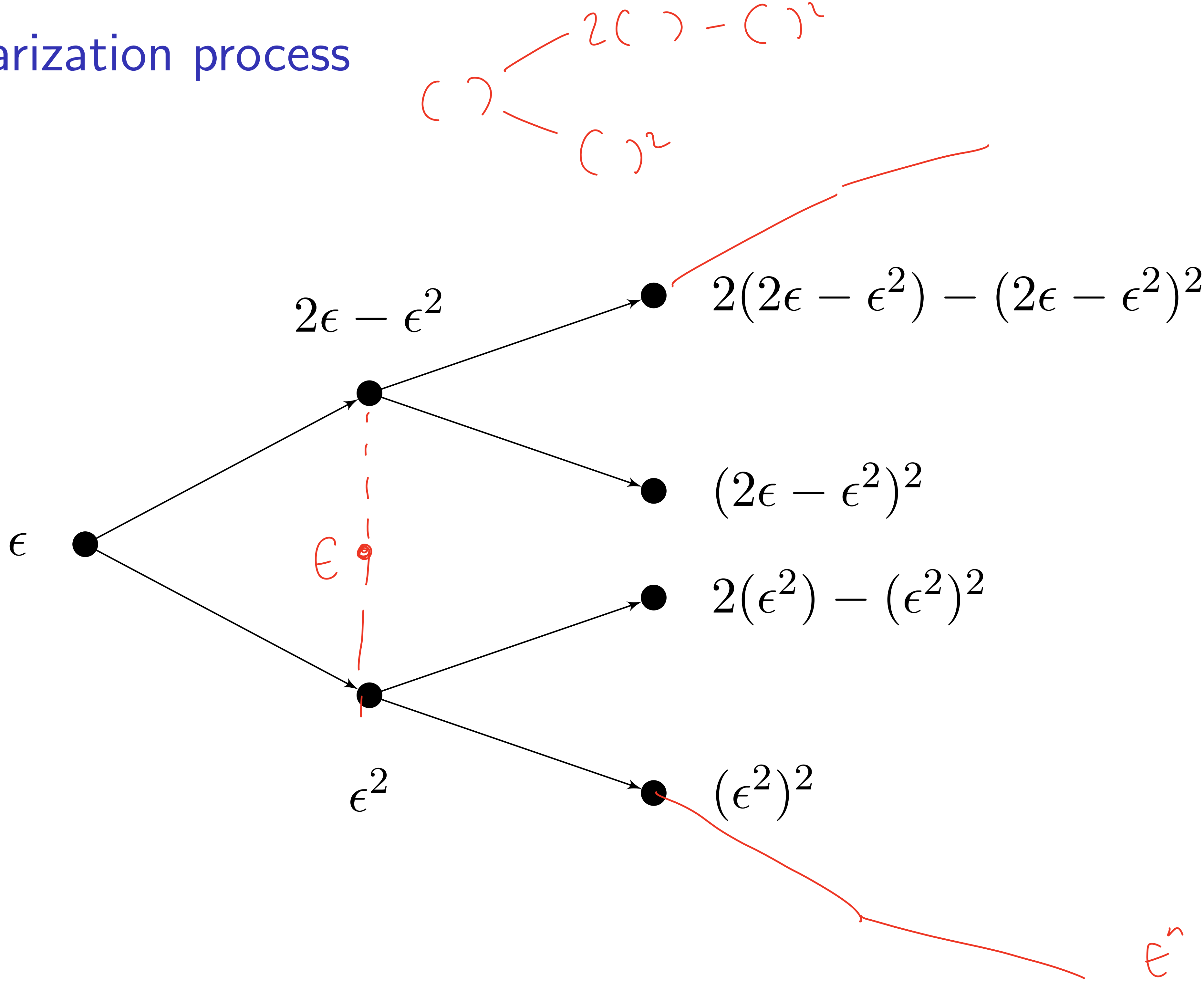
  failure probability$= 2\tilde{\epsilon} - \tilde{\epsilon}^2 = 2(\epsilon^2) - (\epsilon^2)^2$

- Decode U_4

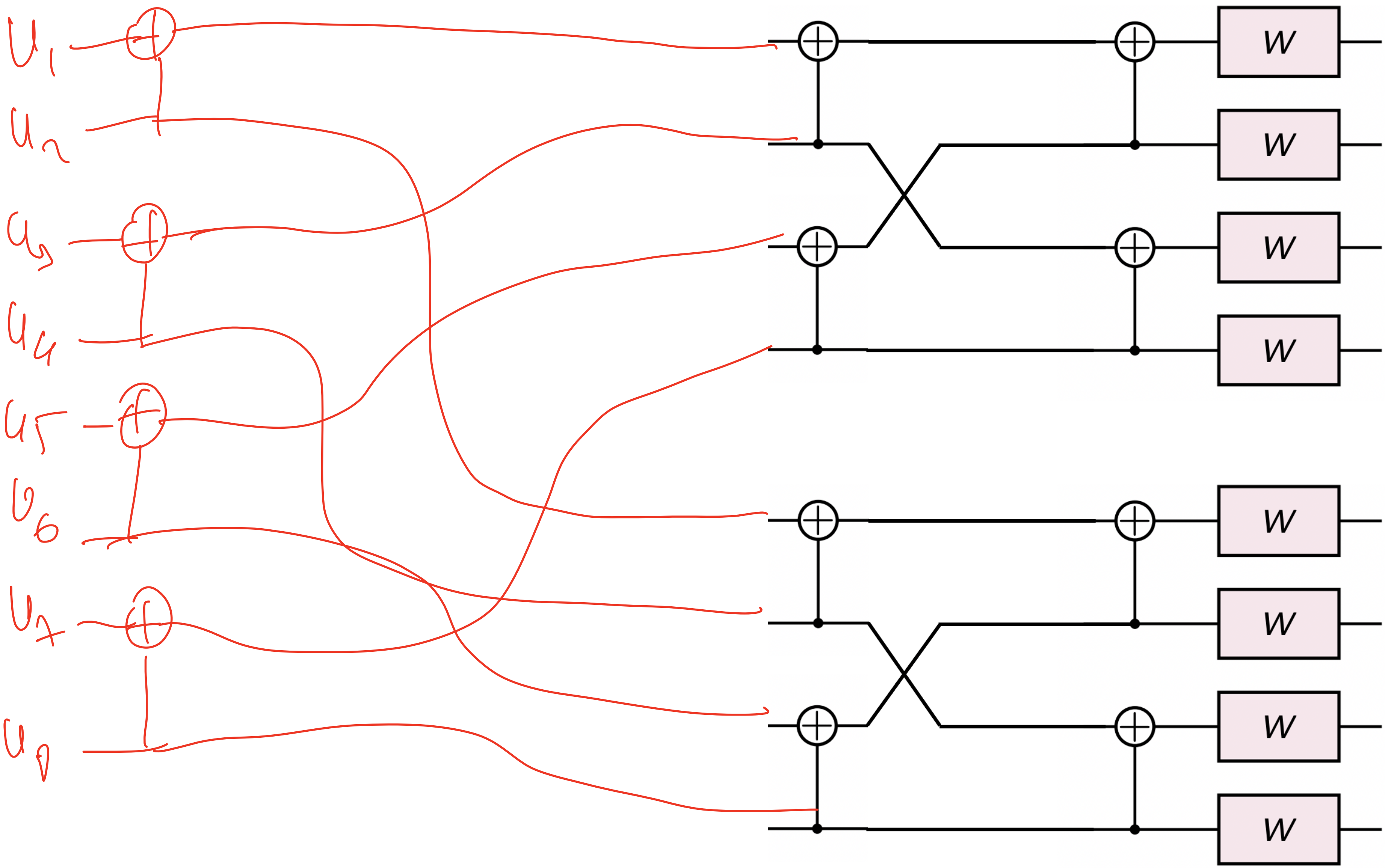
  erased if (Y_1 and Y_2 erased) **and** (Y_3 and Y_4 erased)

  failure probability$= \tilde{\epsilon}^2 = (\epsilon^2)^2$

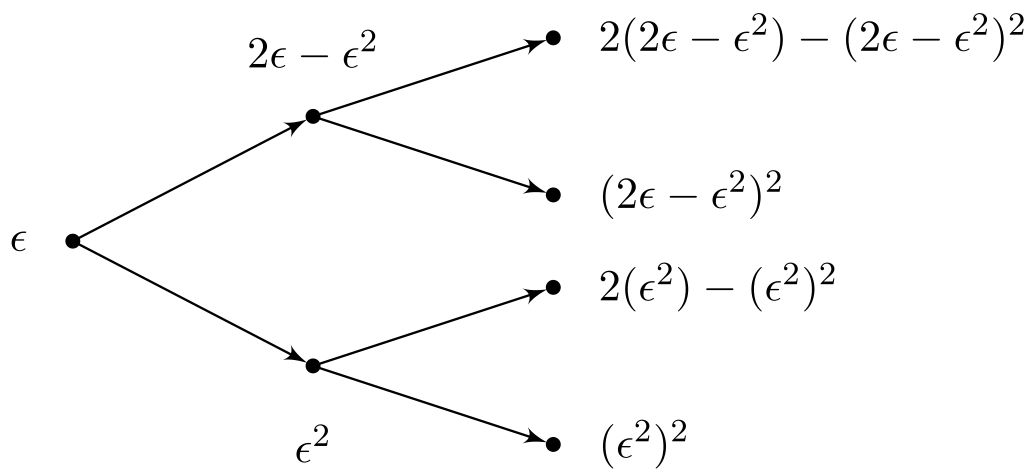
Polarization process



Larger construction

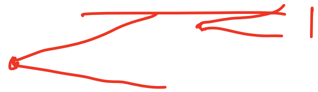


Larger construction



Question: **What happens if we keep extending the size?**

Larger construction



Gambling and Martingales

- you can bet **red** or **black**

  Probability[**red**] $= \frac{1}{2}$ Probability[**black**] $= \frac{1}{2}$
- a betting strategy that always wins (!)[1]: double the bet after every loss

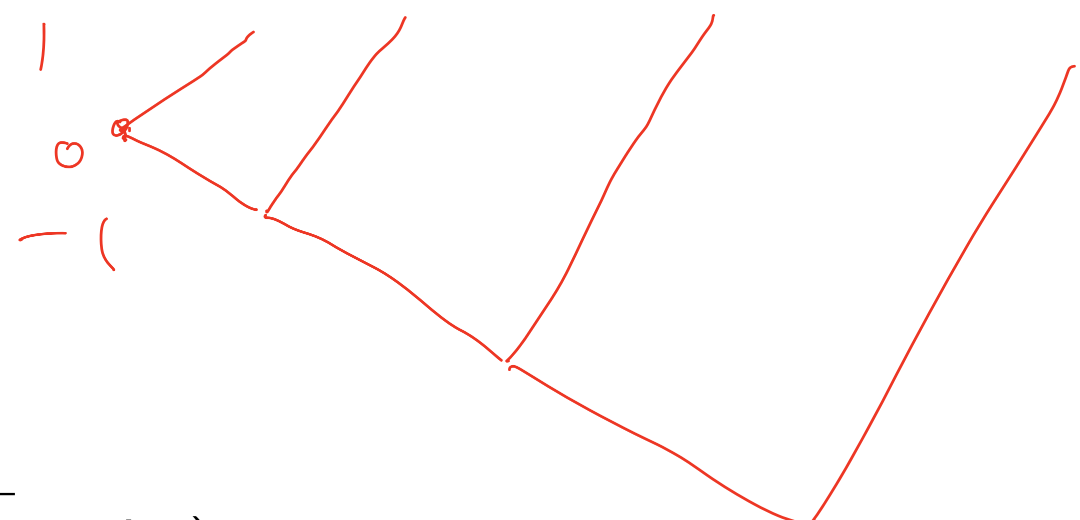
  until you win:

  bet \$1 on **black**

  bet \$2 on **black**

  bet \$4 on **black**

  ⋮



[1]do not try this at home (or at the casino)

Gambling and Martingales

until you win:

bet \$1 on **black**

bet \$2 on **black**

bet \$4 on **black**

bet \$8 on **black**

bet \$16 on **black**

bet \$32 on **black**

$\vdots$

- Martingale betting strategy is a winning strategy only if you have **unbounded wealth**
- It is **not sustainable**

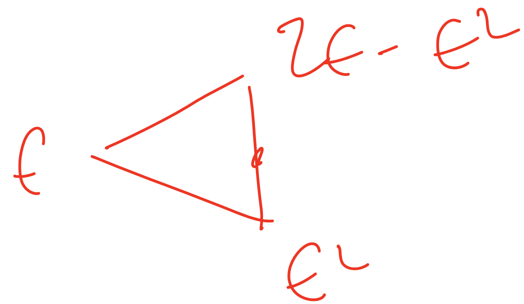
  Probability[Loosing 6 in a row]$=\frac{1}{2^6} \approx 0.016$. This will eventually happen if you repeat many times

Martingale Processes

A martingale process is a sequence of random variables for which the conditional expectation of the next value in the sequence is equal to the present value, regardless of all prior values.

- Martingales processes are important finance, e.g., in stock trading, Black Scholes option pricing model (which won the Nobel prize in economics)

Back to Polar Codes



Let e_t be random ± 1 for $t = 1, 2 \ldots$. The polarization process is

$$w_{t+1} = w_t + e_t w_t (1 - w_t)$$

$$= \begin{cases} 2w_t - w_t^2 & e_t = +1 \\ w_t^2 & e_t = -1 \end{cases}$$

Sample paths

Martingales

$$w_{t+1} = w_t + e_t w_t (1 - w_t)$$

- the expectation of the next value is equal to the previous value: $\mathbb{E}[w_{t+1}|w_t] = w_t$

Martingales

$$w_{t+1} = w_t + e_t w_t (1 - w_t)$$

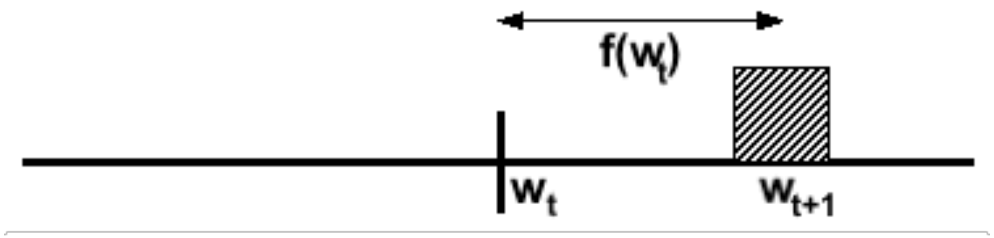
- the expectation of the next value is equal to the previous value: $\mathbb{E}[w_{t+1}|w_t] = w_t$
- Martingale processes converge to a limiting distribution if they are bounded
- polarization process $w_{t+1} = w_t + e_t w_t (1 - w_t)$ converges to $w(1-w) = 0$ **why?**
  $w = 0$ (erasure probability one) or
  $w = 1$ (erasure probability zero)

w = 0 or

w = 1

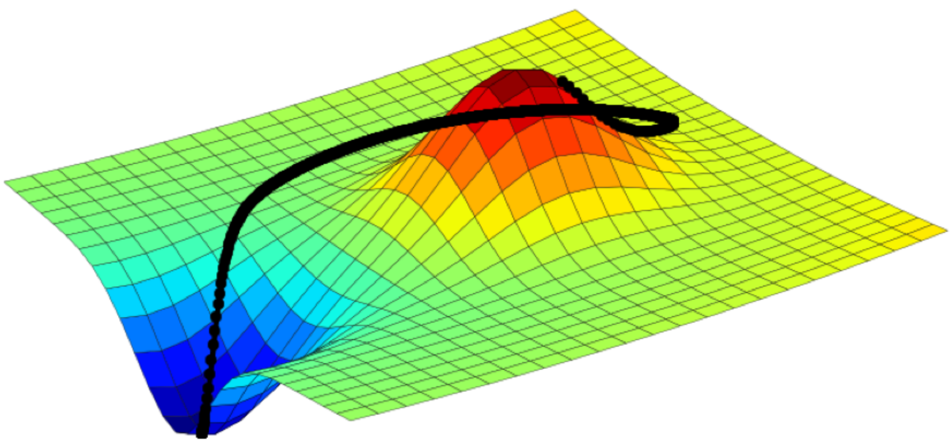
Evolution of physical systems

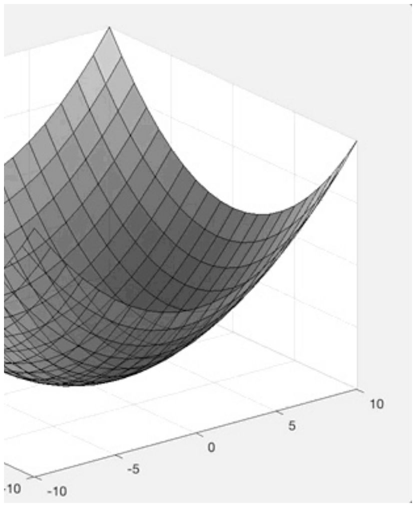
$$\underbrace{w_{t+1}}_{\text{next position}} = \underbrace{w_t}_{\text{current position}} + \underbrace{f(w_t)}_{\text{displacement}}$$



Gradient descent

$$\underbrace{w_{t+1}}_{\text{next parameters}} = \underbrace{w_t}_{\text{current parameters}} + \underbrace{f(w_t)}_{\text{-gradient}}$$



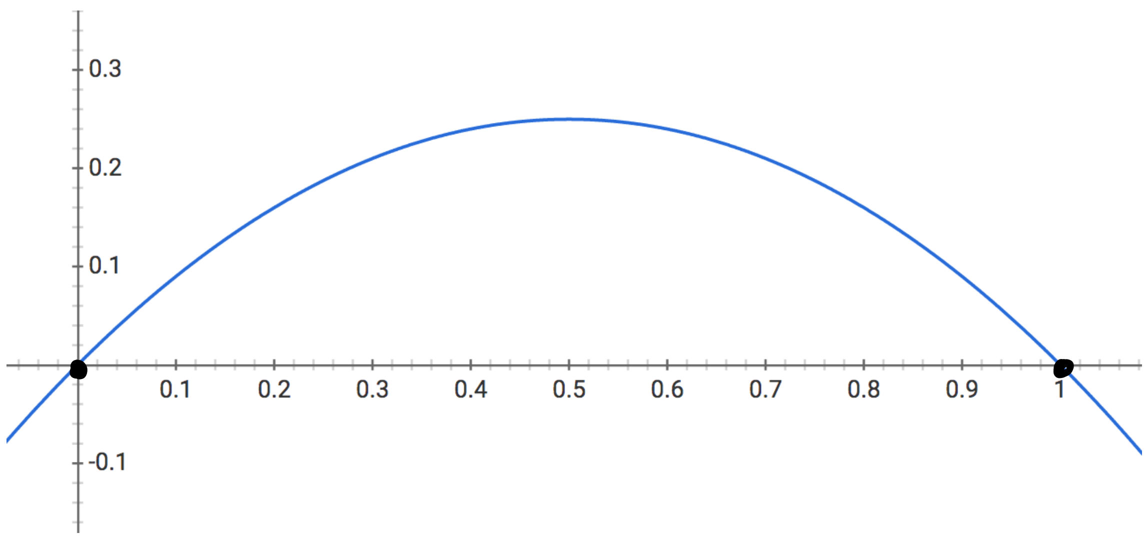


$$w_{t+1} = w_t + e_t w_t(1 - w_t)$$

converges, i.e., $w_{t+1} = w_t$ when $f(w_t) = w_t(1 - w_t) = 0$

$$w_{t+1} = w_t + e_t w_t(1 - w_t)$$

plot of $f(w) = w(1 - w)$



Polarization theorem

- the process

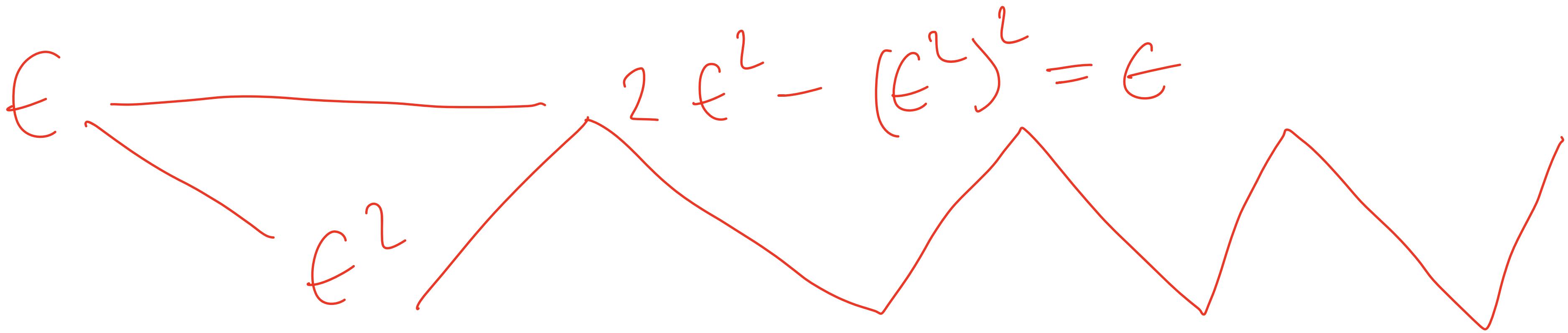
$$w_{t+1} = w_t + e_t w_t (1 - w_t)$$

  converges to either zero or one with probability one!
- implies that almost all channels are either perfect or completely noisy

Non-convergent paths

- Down - Up - Down - Up ....

  $\epsilon \searrow \epsilon^2 \nearrow 2\epsilon^2 - \epsilon^4 =? \epsilon$



Non-convergent paths

- Down - Up - Down - Up ....

  $\epsilon \searrow \epsilon^2 \nearrow 2\epsilon^2 - \epsilon^4 = \epsilon$ if $\epsilon = \frac{\sqrt{5}}{2} - \frac{1}{2} = \frac{1}{\phi} \approx 0.61803398875$

Non-convergent paths

- Down - Up - Down - Up ....

  $\epsilon \searrow \epsilon^2 \nearrow 2\epsilon^2 - \epsilon^4 = \epsilon$ if $\epsilon = \frac{\sqrt{5}}{2} - \frac{1}{2} = \frac{1}{\phi} \approx 0.61803398875$

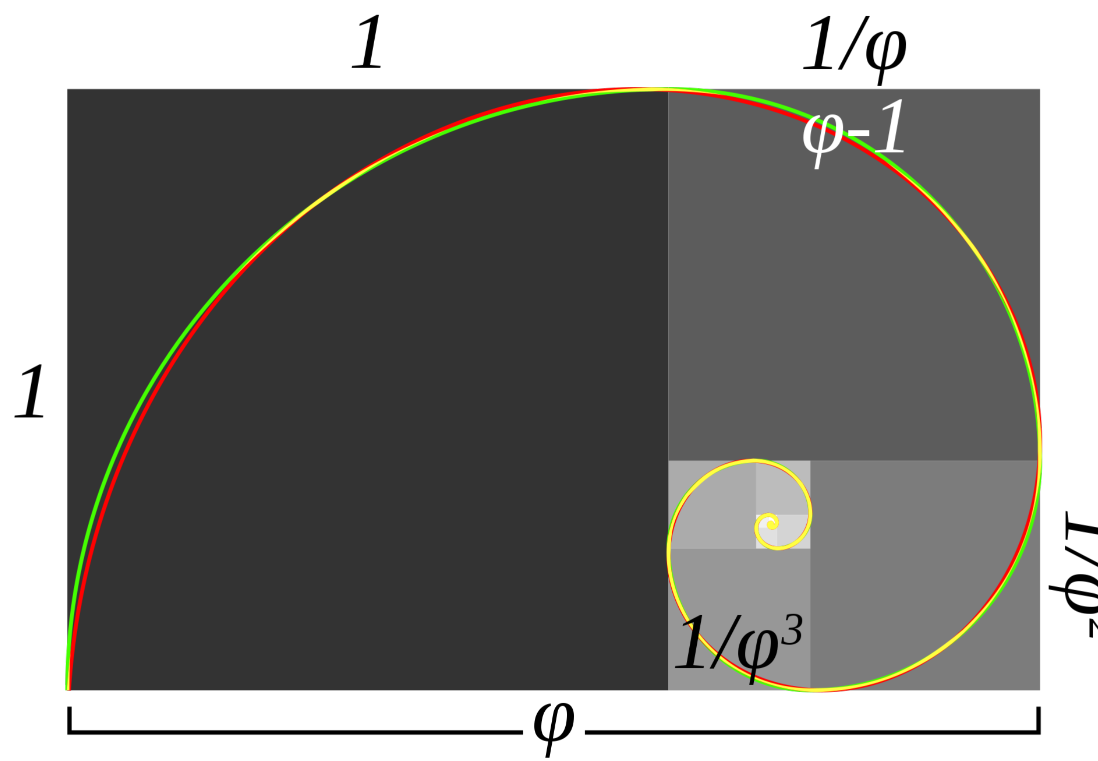
$$\text{Golden ratio :} \quad \phi := \frac{1+\sqrt{5}}{2} \approx 1.61803398875$$

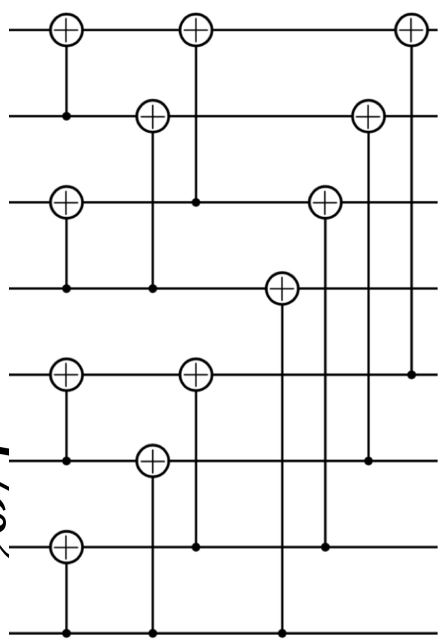
Non-convergent paths

- Down - Up - Down - Up ....

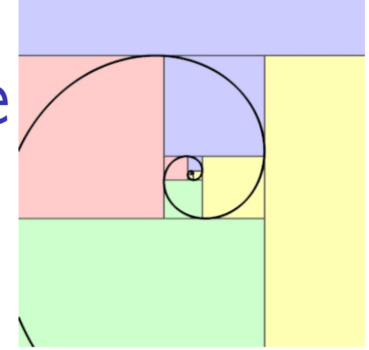
$$\epsilon \searrow \epsilon^2 \nearrow 2\epsilon^2 - \epsilon^4 = \epsilon \text{ if } \epsilon = \frac{\sqrt{5}}{2} - \frac{1}{2} = \frac{1}{\phi} \approx 0.61803398875$$

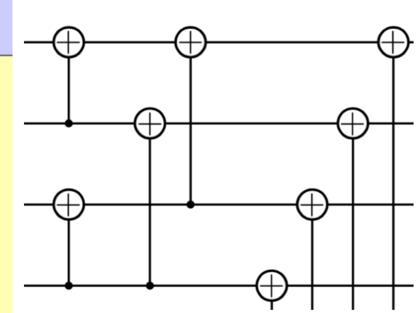
$$\text{Golden ratio :} \quad \phi := \frac{1+\sqrt{5}}{2} \approx 1.61803398875$$

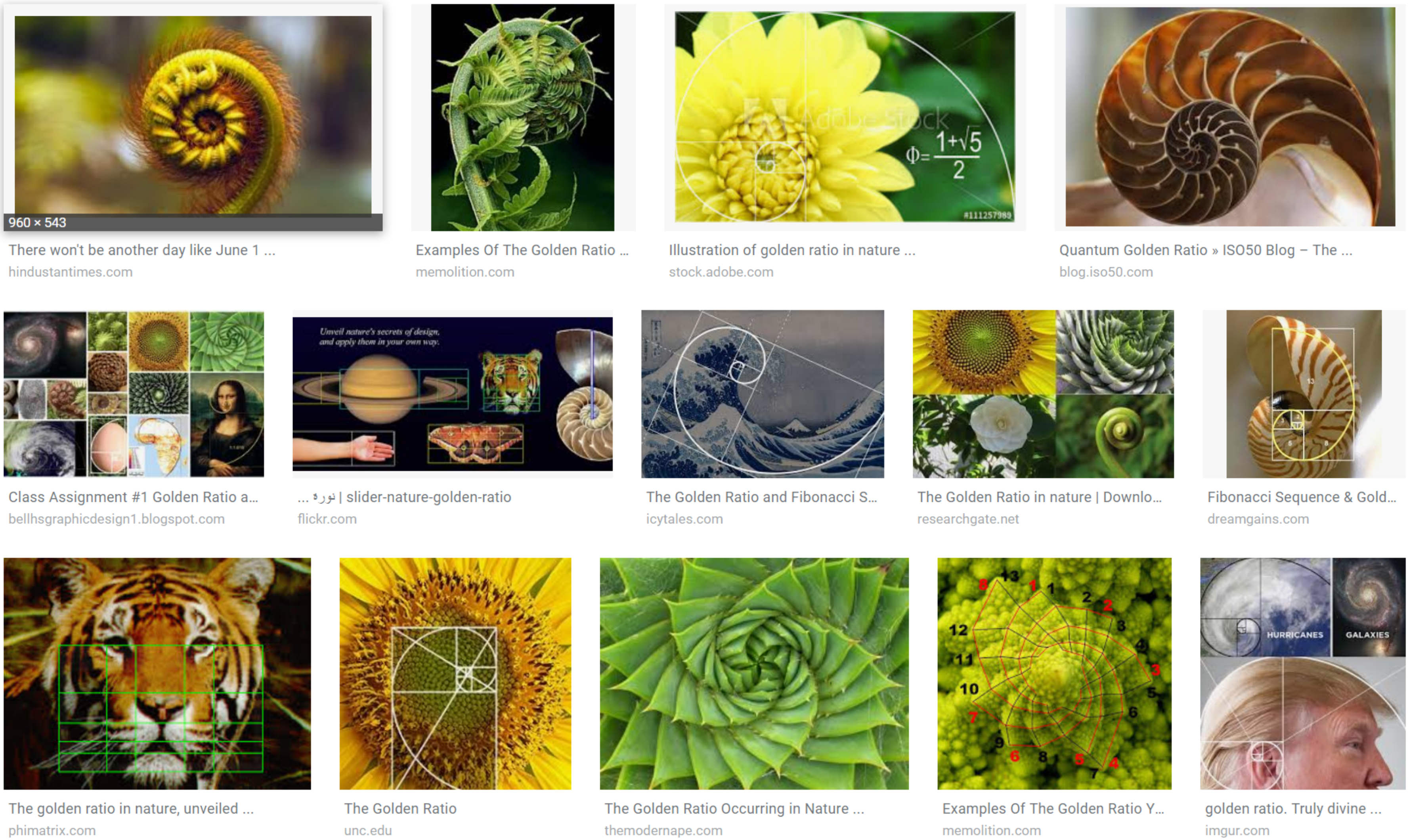




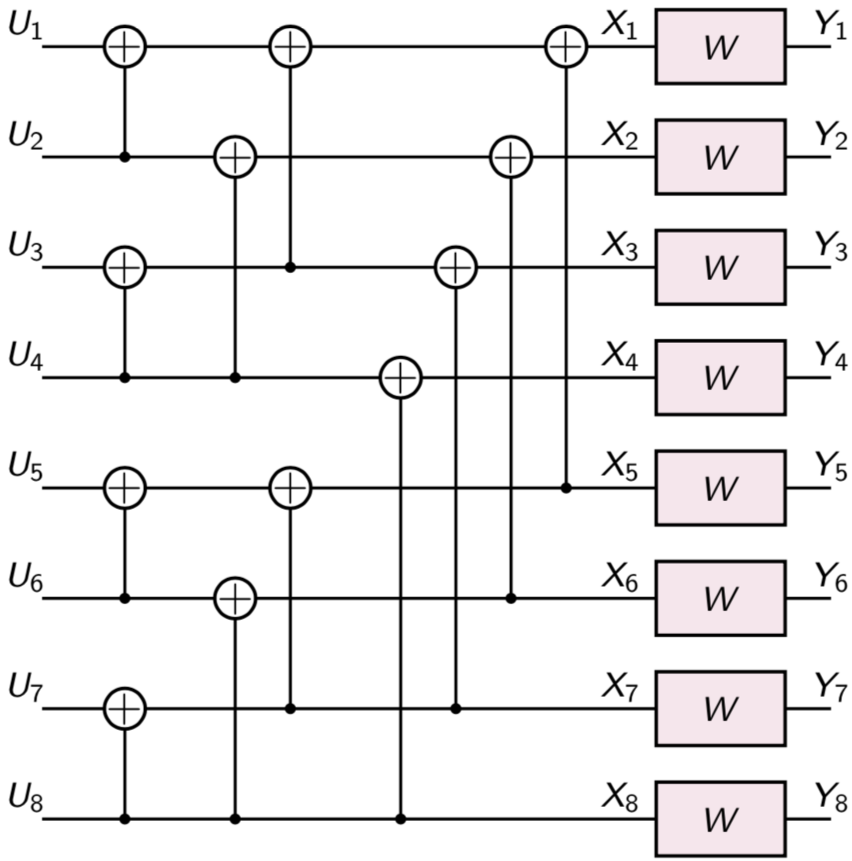
Google images: golden ratio in nature







Encoding circuit



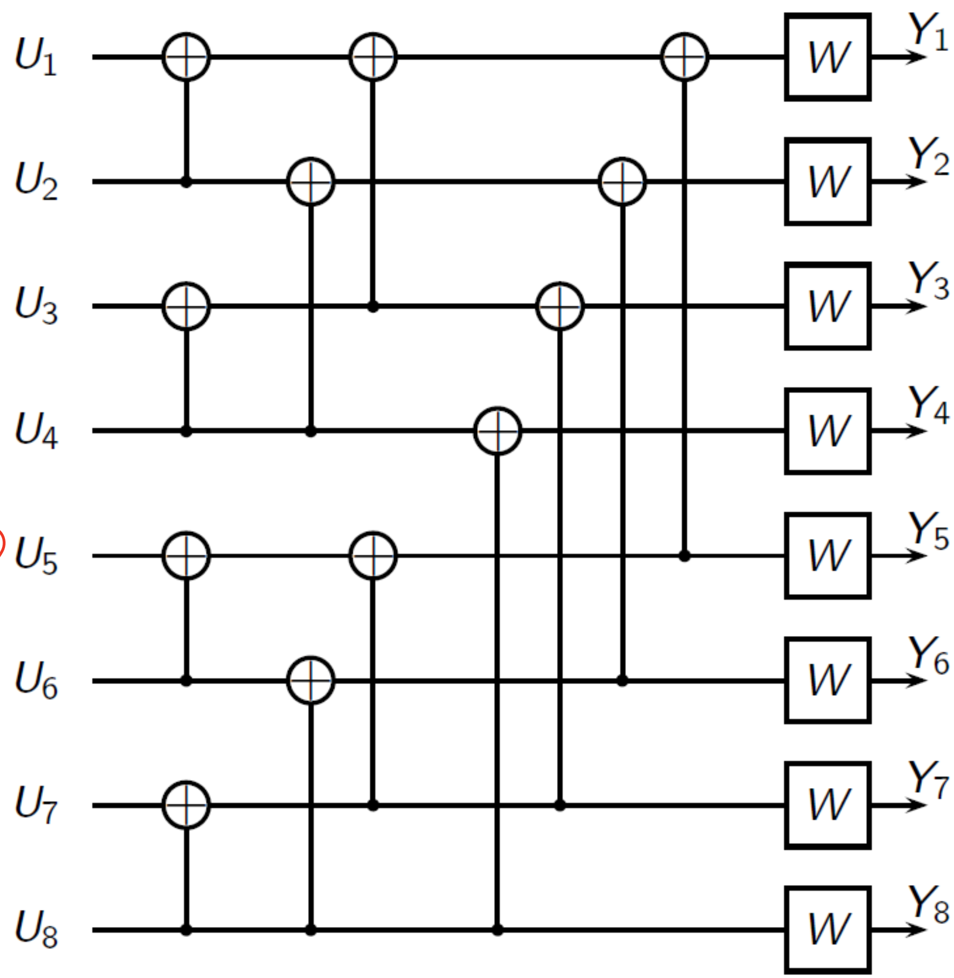
$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

What do we do with the noisy channels?

Rate = $\frac{1}{2}$

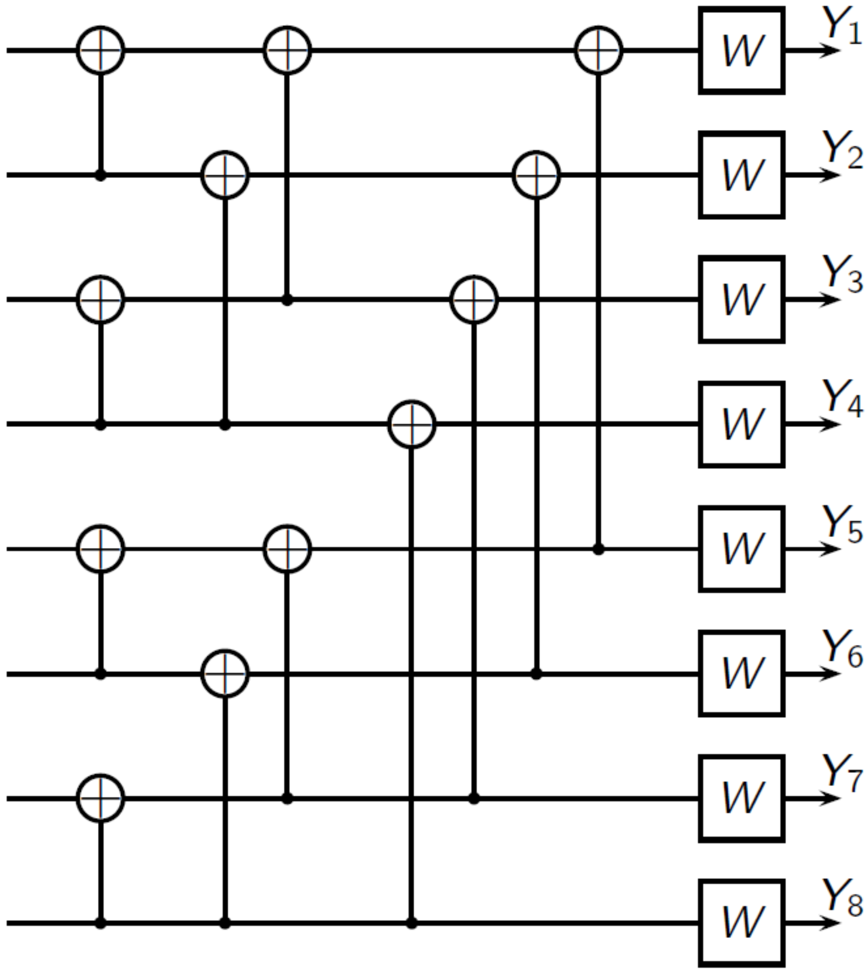
C=1-P[erasure]	rank
0.0039	8
0.1211	7
0.1914	6
✓ 0.6836	4
0.3164	5
✓ 0.8086	3
✓ 0.8789	2
✓ 0.9961	1

$u_1 = 0$
$u_2 = 0$
$u_3 = 0$
$u_5 = 0$



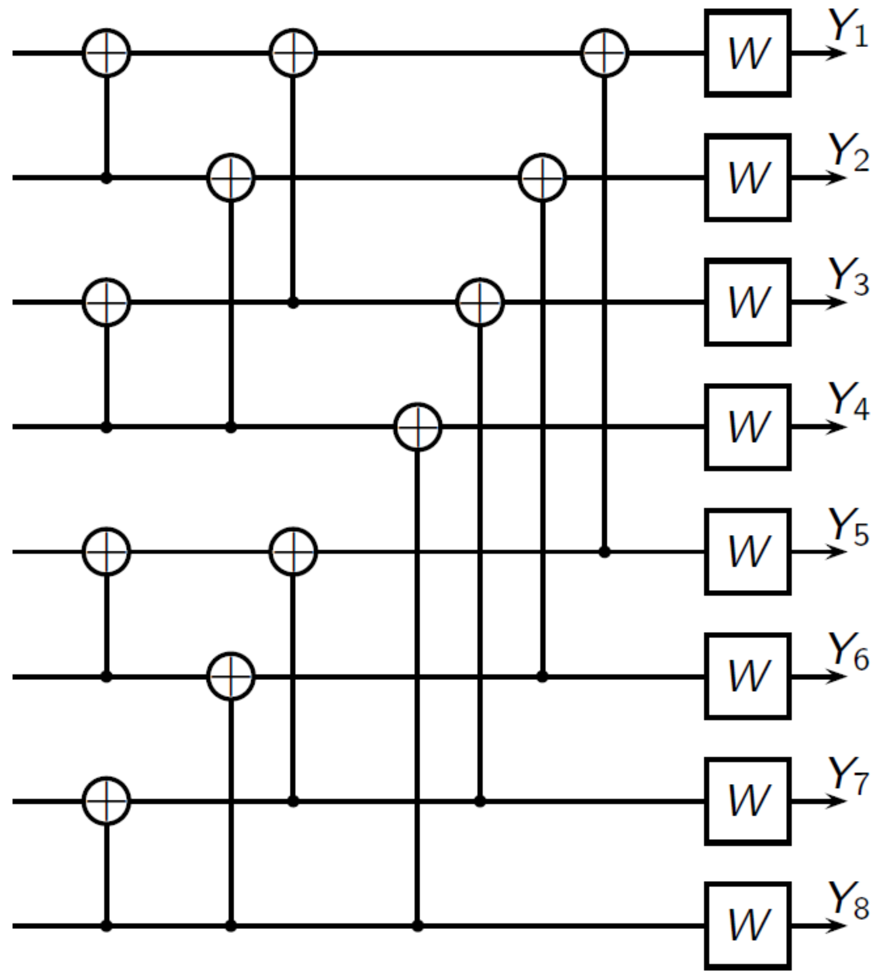
We can freeze noisy channels!

C=1-P[erasure]	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



Freezing noisy channels

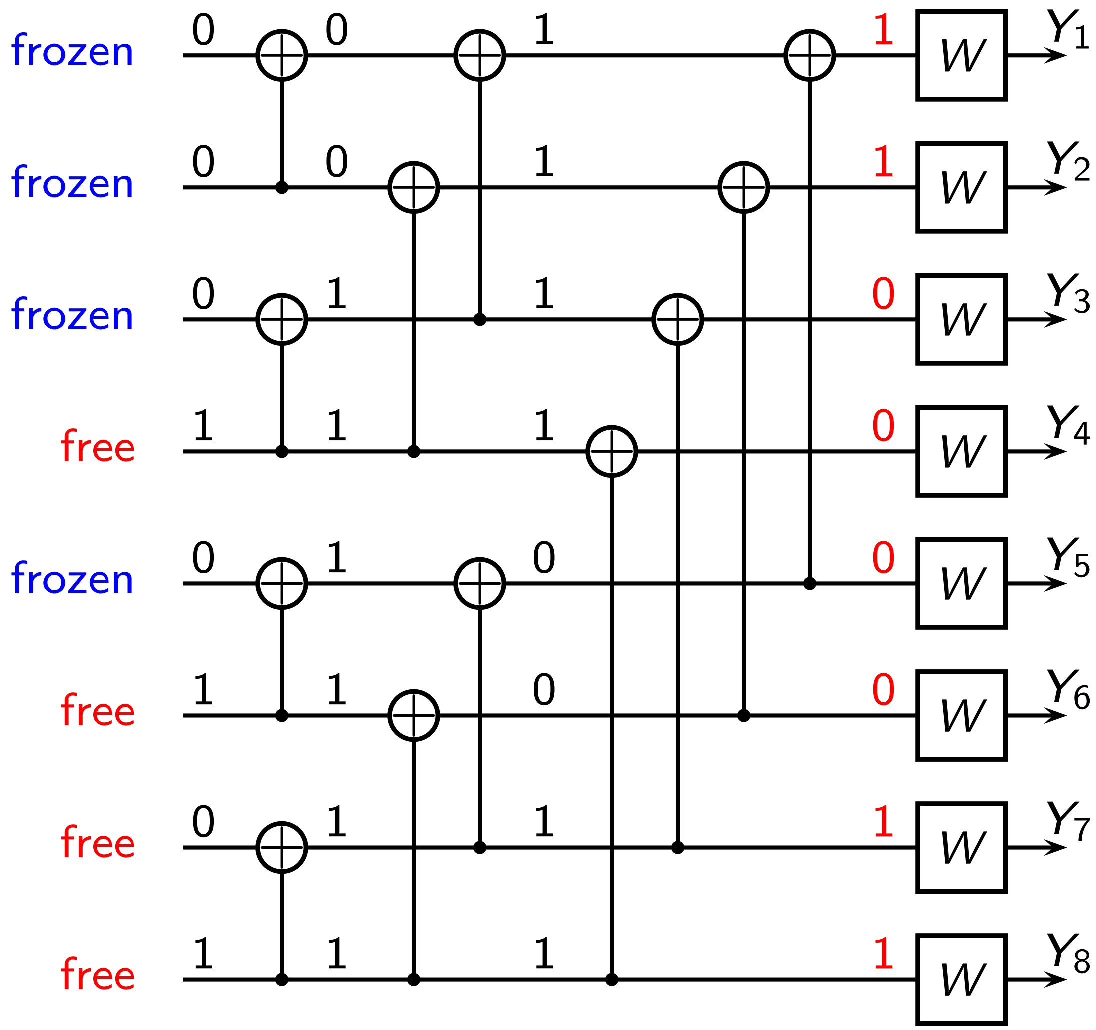
C=1-P[erasure]	rank		
0.0039	8	frozen	0
0.1211	7	frozen	0
0.1914	6	frozen	0
0.6836	4	data	U_4
0.3164	5	frozen	0
0.8086	3	data	U_6
0.8789	2	data	U_7
0.9961	1	data	U_8



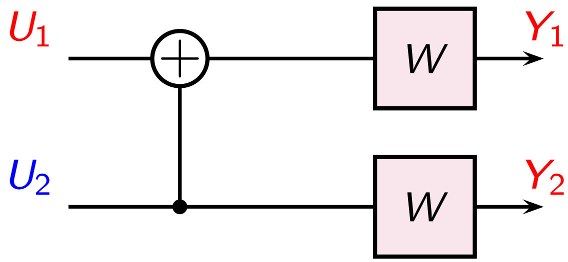
Encoding and decoding

all the bad channels are frozen

- successive cancellation decoder will correctly recover the message with high probability!



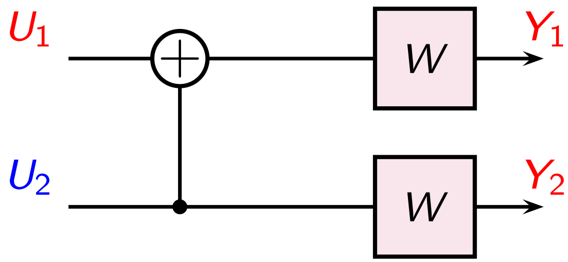
Polarization of general channels



$$W^{-}(Y_1, Y_2 | U_1) = \frac{1}{2} \sum_{u_2} W_1(y_1 | u_1 \oplus u_2) W_2(y_2 | u_2)$$

$$W^{+}(Y_1, Y_2, U_1 | U_2) = \frac{1}{2} W_1(y_1 | u_1 + u_2) W_2(y_2 | u_2)$$

Polarization of general channels

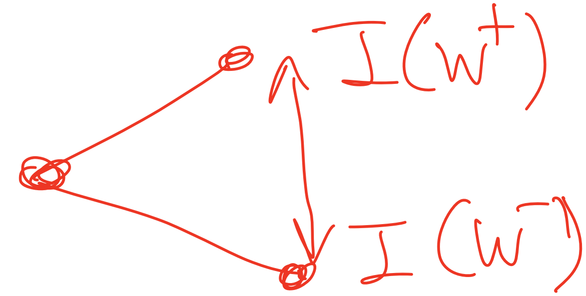


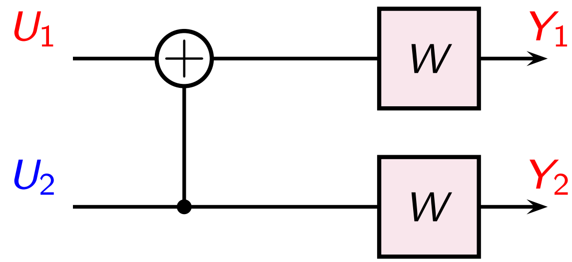
$$W^-(Y_1, Y_2|U_1) = \frac{1}{2}\sum_{u_2} W_1(y_1|u_1 \oplus u_2)W_2(y_2|u_2)$$

$$W^+(Y_1, Y_2, U_1|U_2) = \frac{1}{2} W_1(y_1|u_1 + u_2)W_2(y_2|u_2)$$

$$I(W^-) + I(W^+) = I(W) + I(W) = 2I(W)$$

Polarization of general channels





$$W^-(Y_1, Y_2|U_1) = \frac{1}{2}\sum_{u_2} W_1(y_1|u_1 \oplus u_2)W_2(y_2|u_2)$$

$$W^+(Y_1, Y_2, U_1|U_2) = \frac{1}{2}W_1(y_1|u_1 + u_2)W_2(y_2|u_2)$$

$$I(W^-) + I(W^+) = I(W) + I(W) = 2I(W)$$

2p(1-p)=p
p=0 or 1/2

Mrs Gerber's Lemma: If $I(W) = 1 - \mathcal{H}(p)$, then

$$\frac{1}{2}(I(W^+) - I(W^-)) \geq \mathcal{H}(2p(1-p)) - \mathcal{H}(p) > 0$$

Polarization theorem

$C(W) = 1 - \epsilon$

$C(W)$ = capacity of the original channel

- $C(W)$ fraction of channels converge to noiseless channels with mutual information ≈ 1
- $1 - C(W)$ fraction of channels converge noisy channels with mutual information ≈ 0

$C(W)$ fraction

$C(W)$—

$1 - C(W)$ fraction

Polarization theorem

$C(W) =$ capacity of the original channel

- $C(W)$ fraction of channels converge to noiseless channels with mutual information ≈ 1
- $1 - C(W)$ fraction of channels converge noisy channels with mutual information $\approx$ 0

n total channel uses:

$nC(W)$ noiseless and $n(1 - C(W))$ noisy

- By freezing the noisy channels to zero we get $Rate \rightarrow \frac{nC(W)}{n} = C(W)$
- Achieves capacity as n gets large! This is true for any symmetric channel!

Polarization Theorem (formal)

Theorem

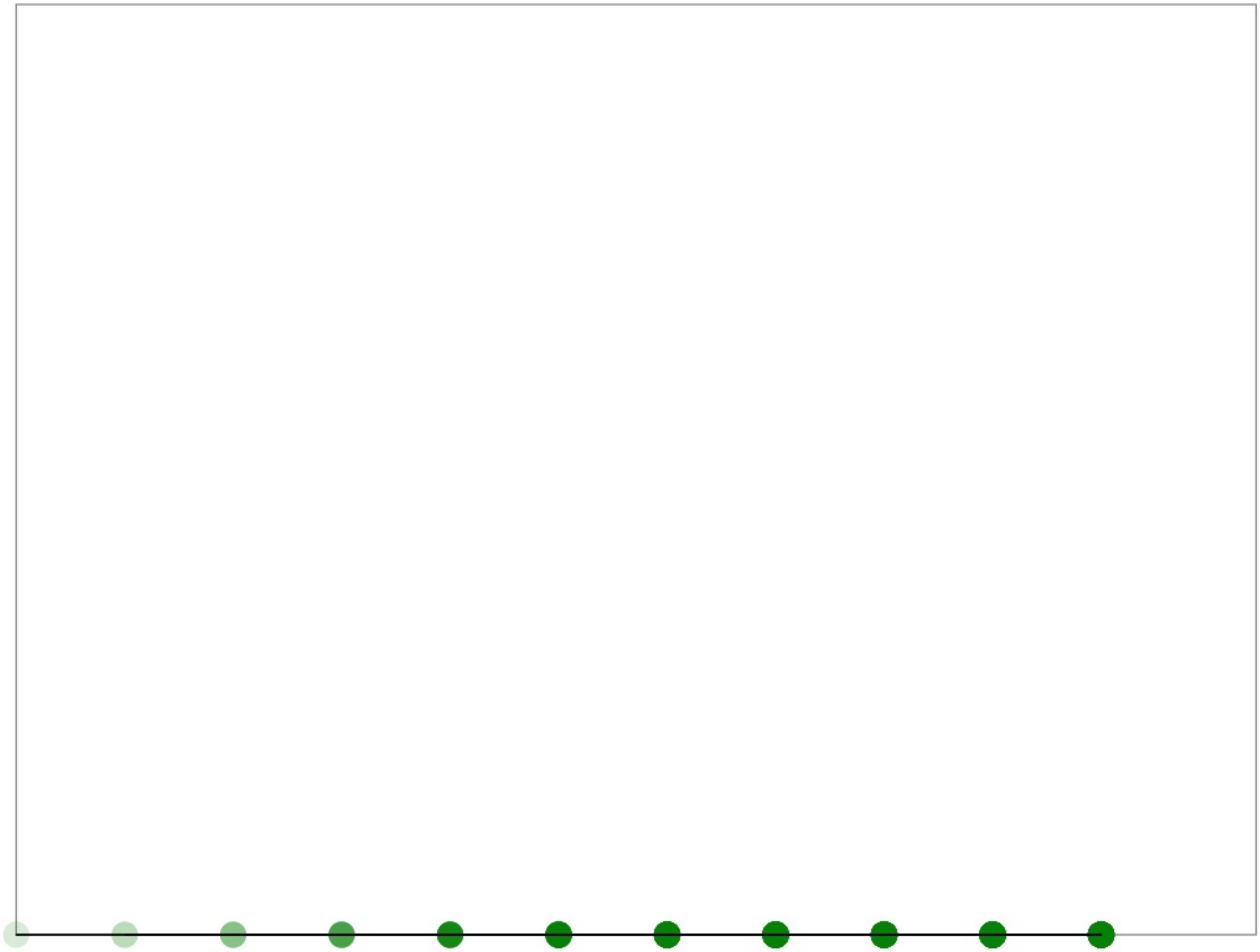
The bit-channel capacities $\{C(W_i)\}$ *polarize: for any* $\delta \in (0,1)$, *as the construction size N grows*

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

and

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

Polarization as capacity changes



Consequence of the Polarization Theorem

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

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

random codes

$$2^{-\frac{N}{2}}$$

5G Communications

- The jump from 4G to 5G is far larger than any previous jumps–from 2G to 3G; 3G to 4G
- The global 5G market is expected reach a value of **251 Bn** by 2025

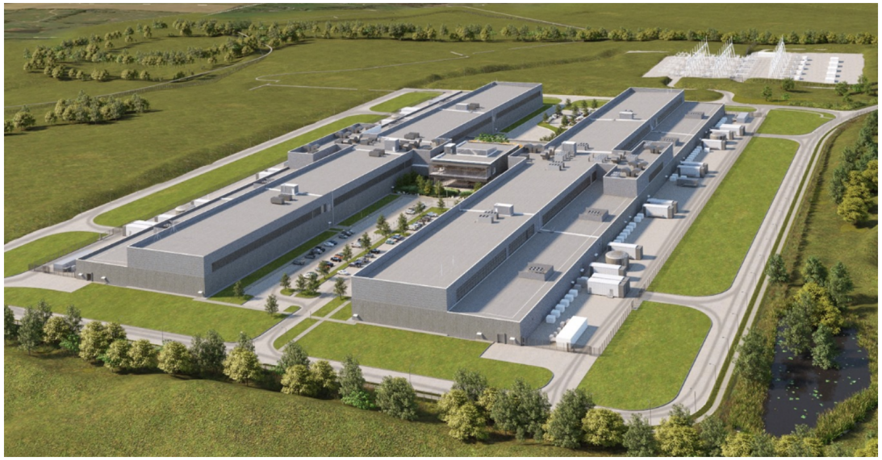
5G Communications

- The jump from 4G to 5G is far larger than any previous jumps–from 2G to 3G; 3G to 4G
- The global 5G market is expected reach a value of **251 Bn** by 2025
- In 2016, researchers reached **27 Gbps** downlink using Polar Codes
- Current LTE download speed is **5-12 Mbps**

5G Communications

- The jump from 4G to 5G is far larger than any previous jumps–from 2G to 3G; 3G to 4G
- The global 5G market is expected reach a value of **251 Bn** by 2025
- In 2016, researchers reached **27 Gbps** downlink using Polar Codes
- Current LTE download speed is **5-12 Mbps**
- In November 2016, 3GPP agreed to adopt Polar codes for control channels in 5G. LDPC codes will be used in data channels.

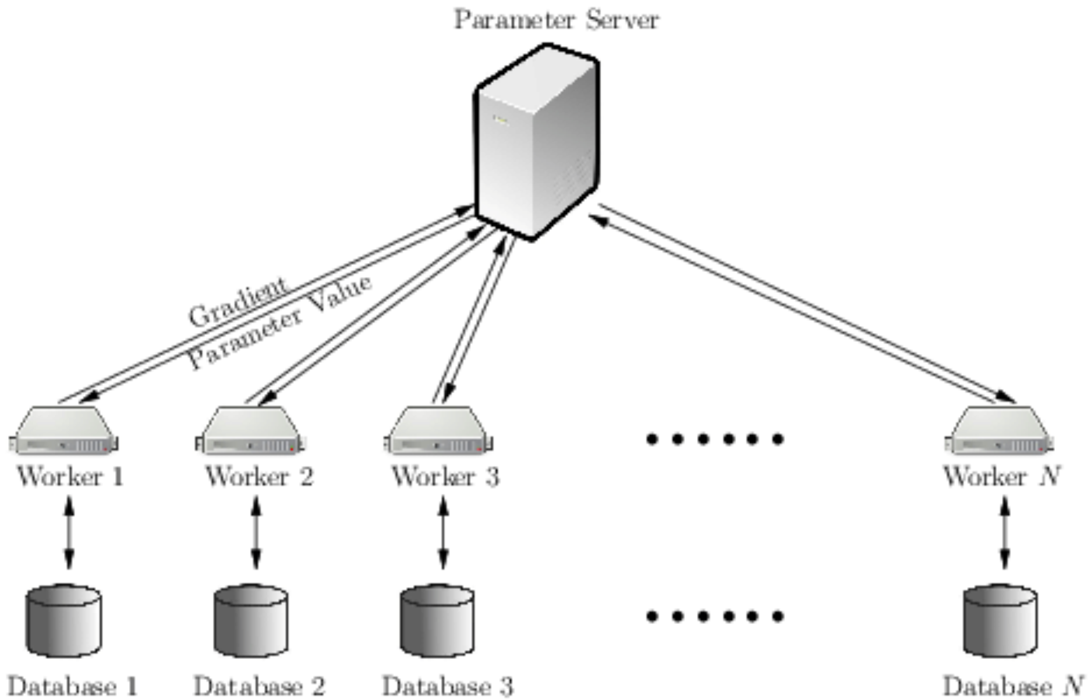
Other Applications: Distributed Computing in Data Centers



Facebook Data Center, New Albany OH.

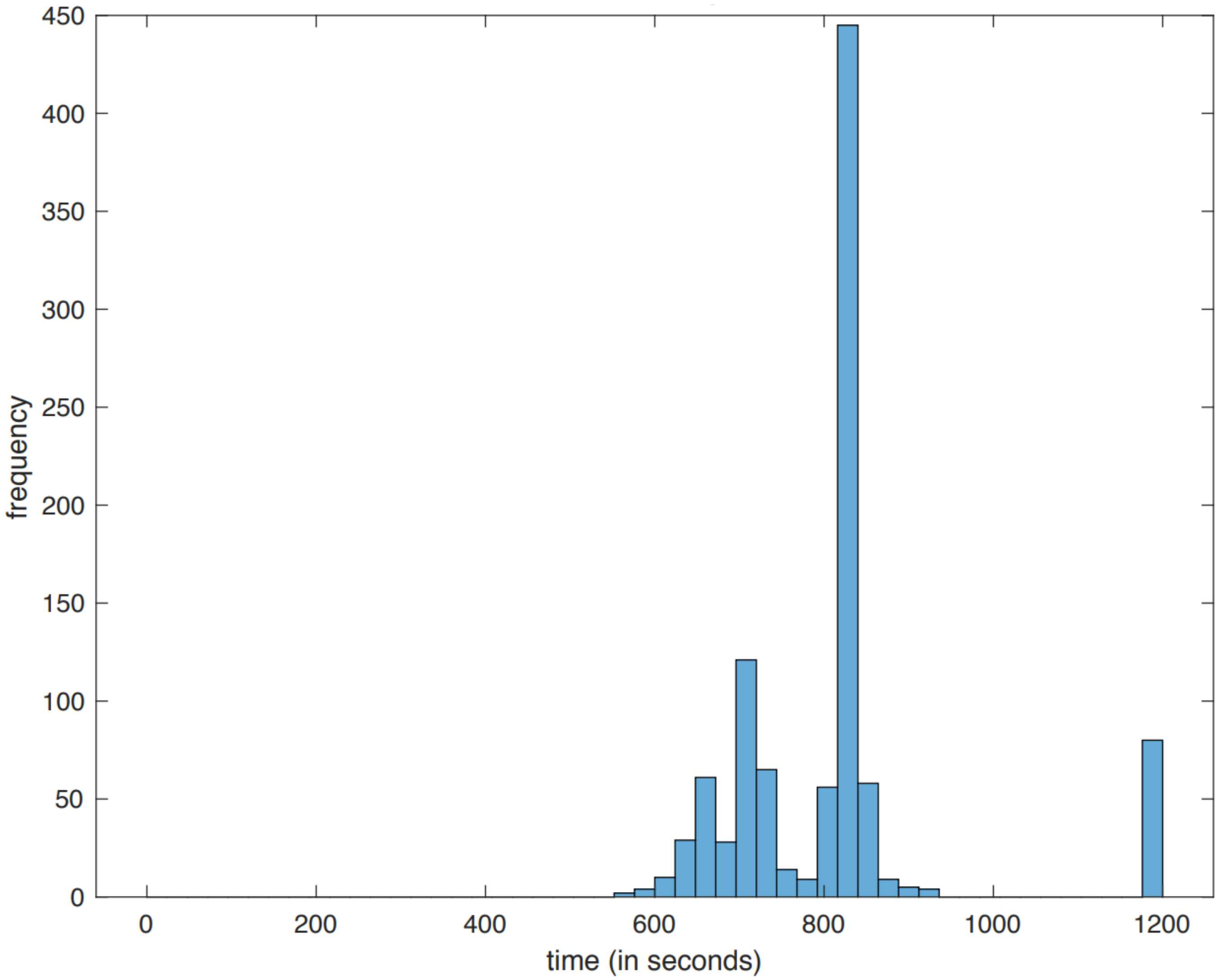
Data Centers

Distributed Computing

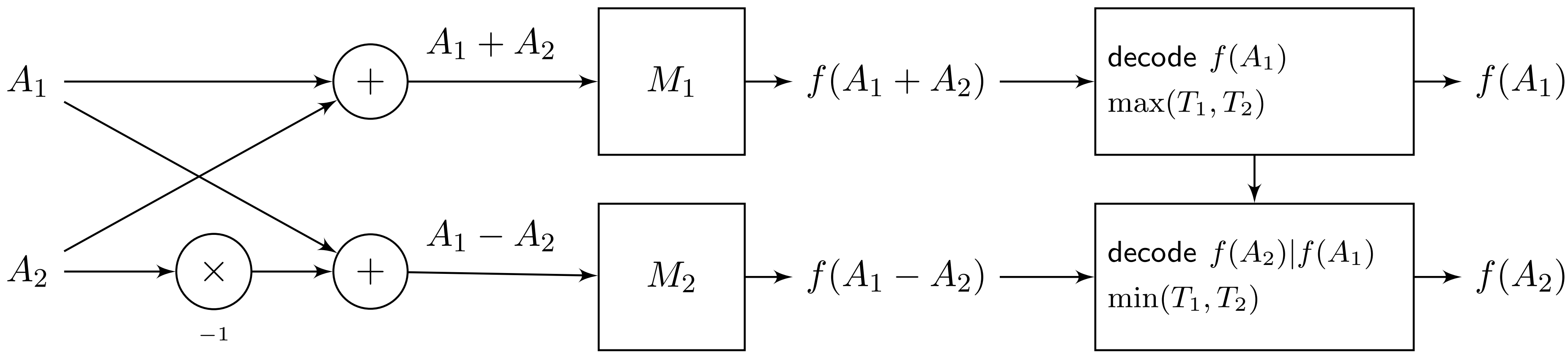


need to wait workers to finish local computations

Distributed Computing



Computational Polarization

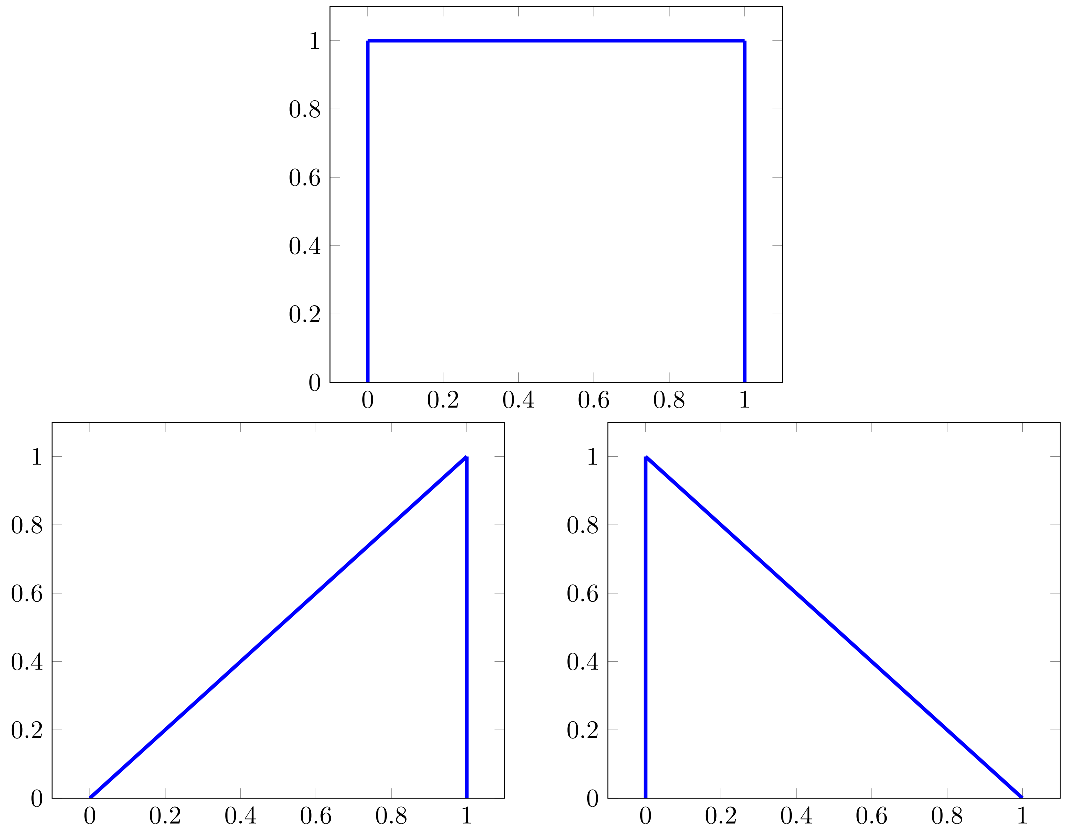


$f(A_1) = A_1 D$

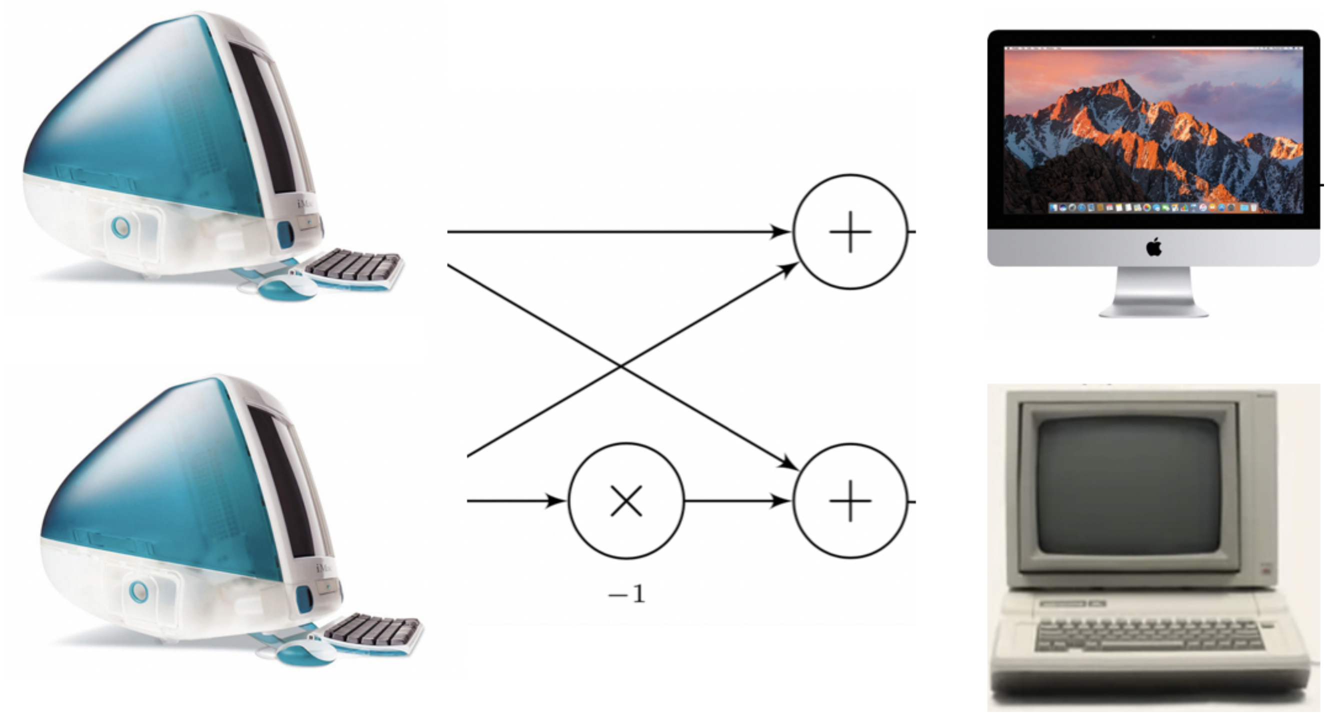
M. Pilanci, **Computational Polarization: An Information-Theoretic Method for Resilient Computing,** IEEE Transactions on Information Theory, 2022

B. Bartan and M. Pilanci, **Straggler Resilient Serverless Computing Based on Polar Codes**, Annual Allerton Conference on Communication, Control, and Computing 2019. arxiv.org/pdf/1901.06811

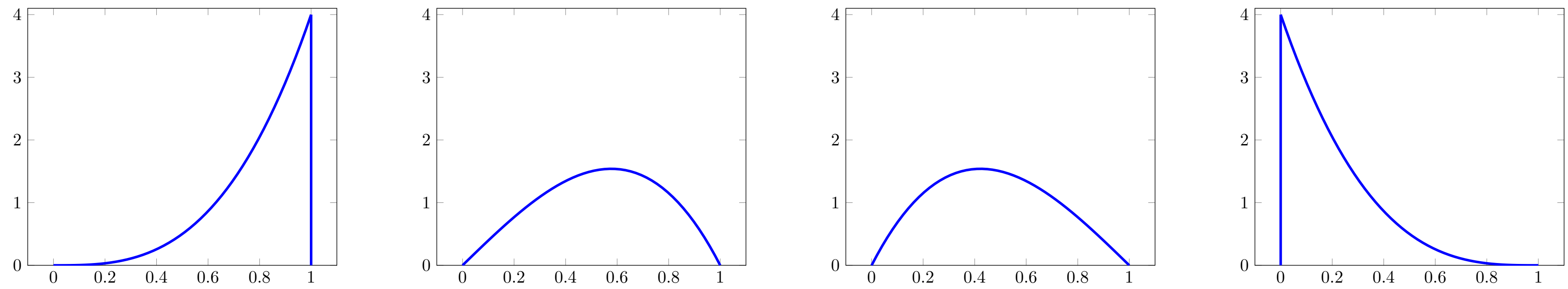
Polarization of computation times



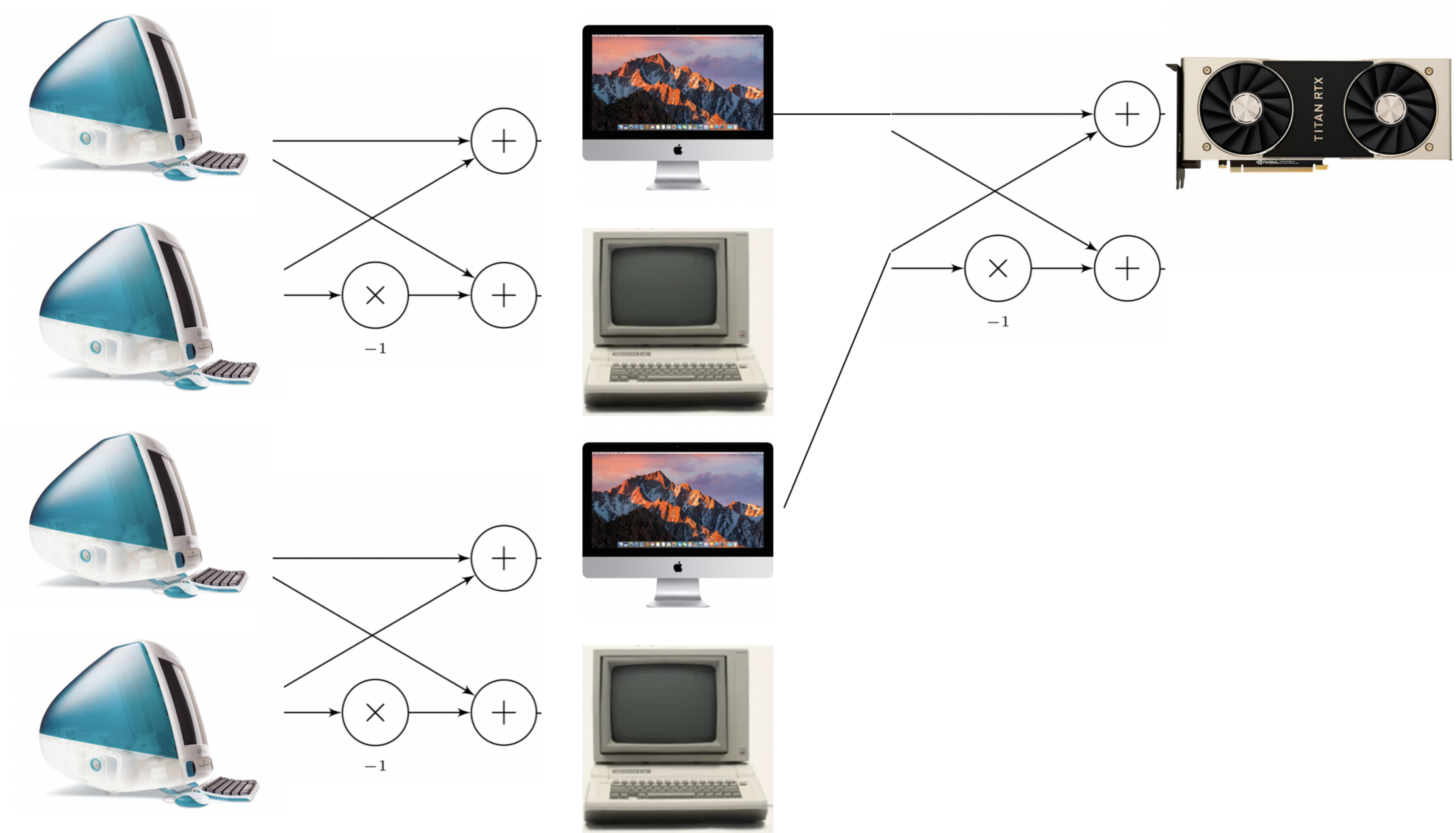
Polarization for computation



Polarization of computation times



Polar codes for computation



Polar coded machine learning

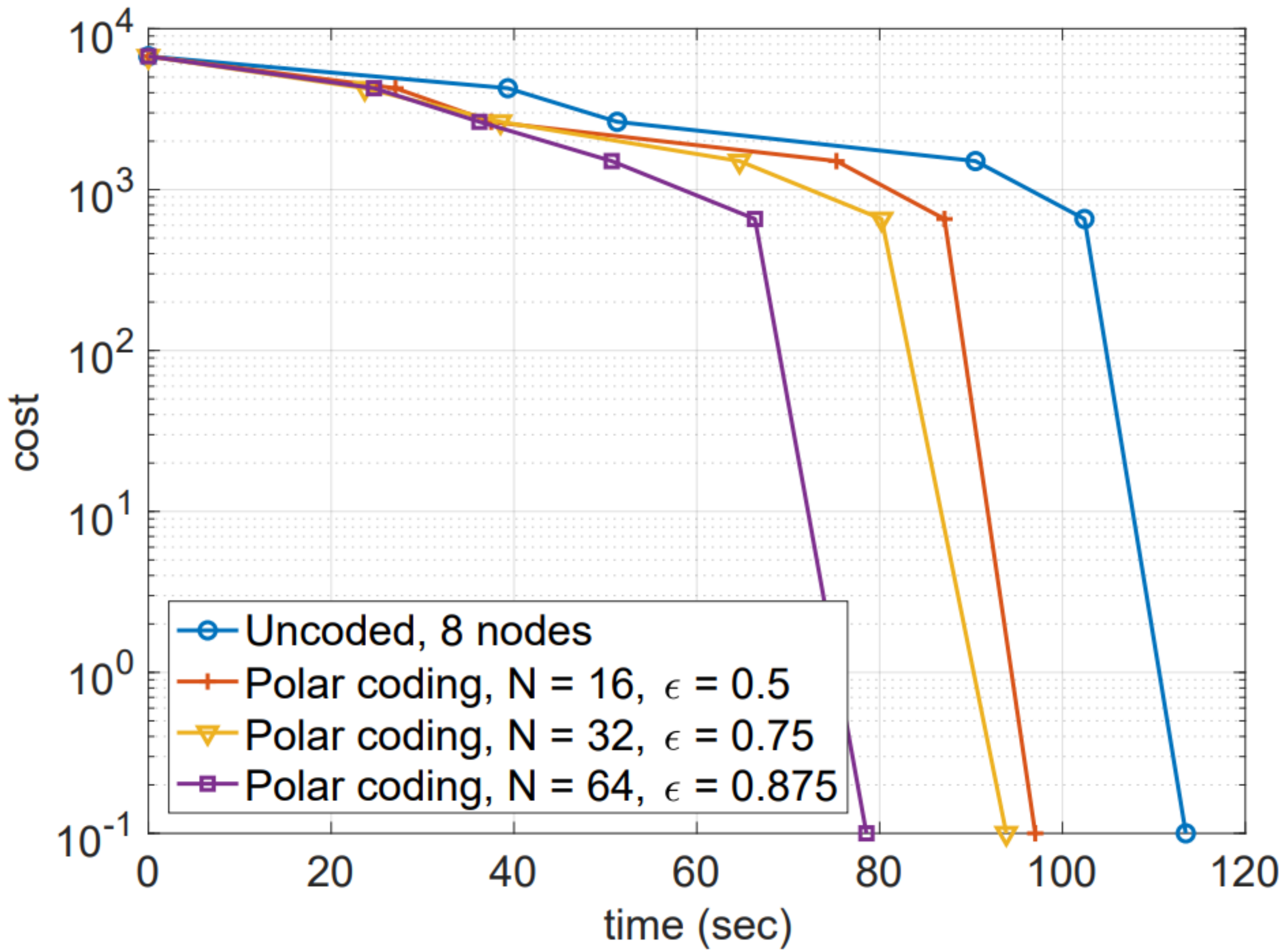


Fig. 8. Cost vs time for the gradient descent example.

B. Bartan and M. Pilanci, **Straggler Resilient Serverless Computing Based on Polar Codes**, Annual Allerton Conference on Communication, Control, and Computing 2019. arxiv.org/pdf/1901.06811

Questions?

Cloud computing on Amazon Lambda

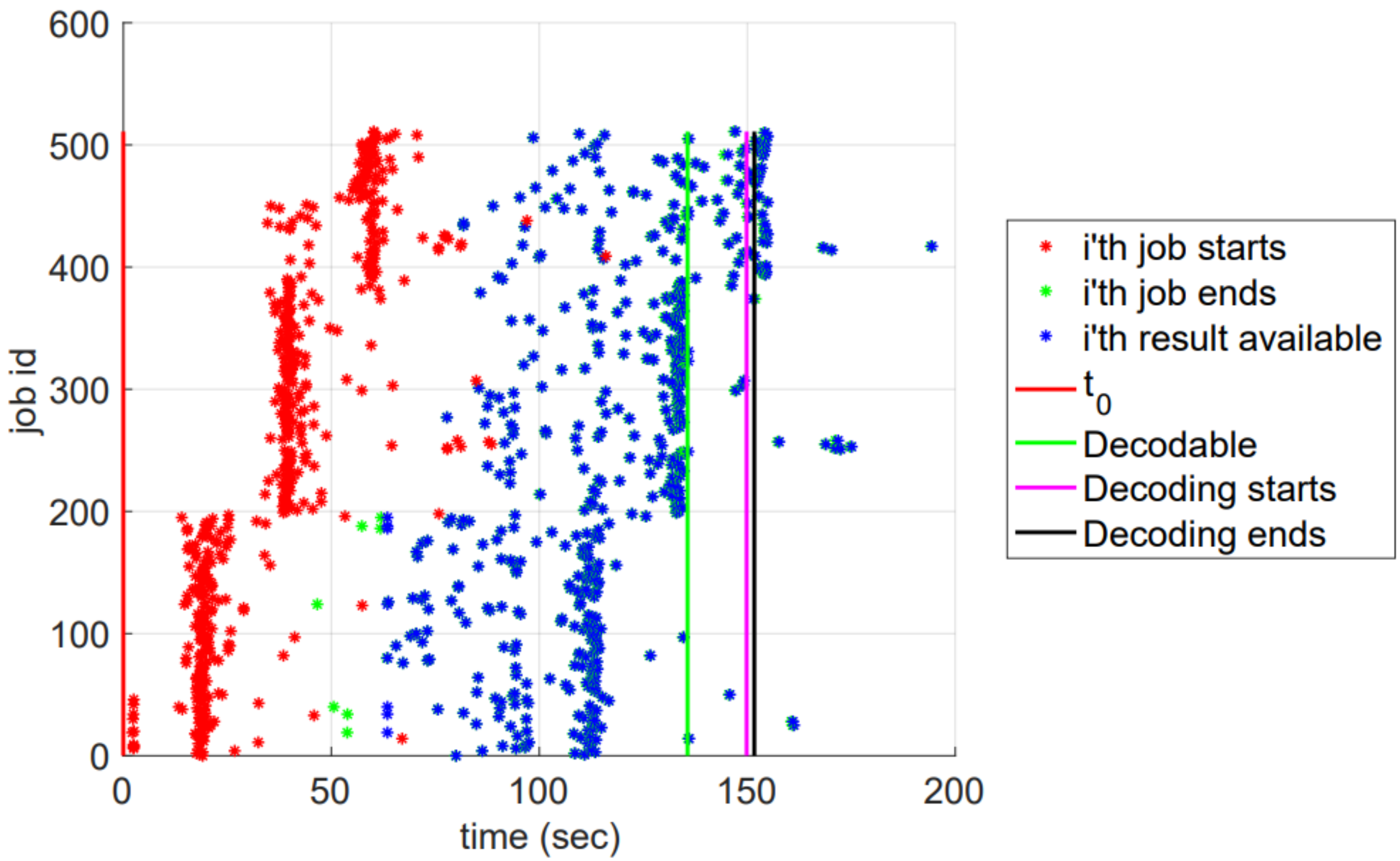
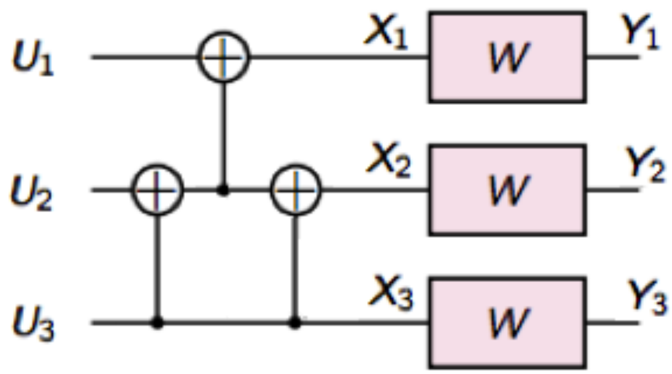


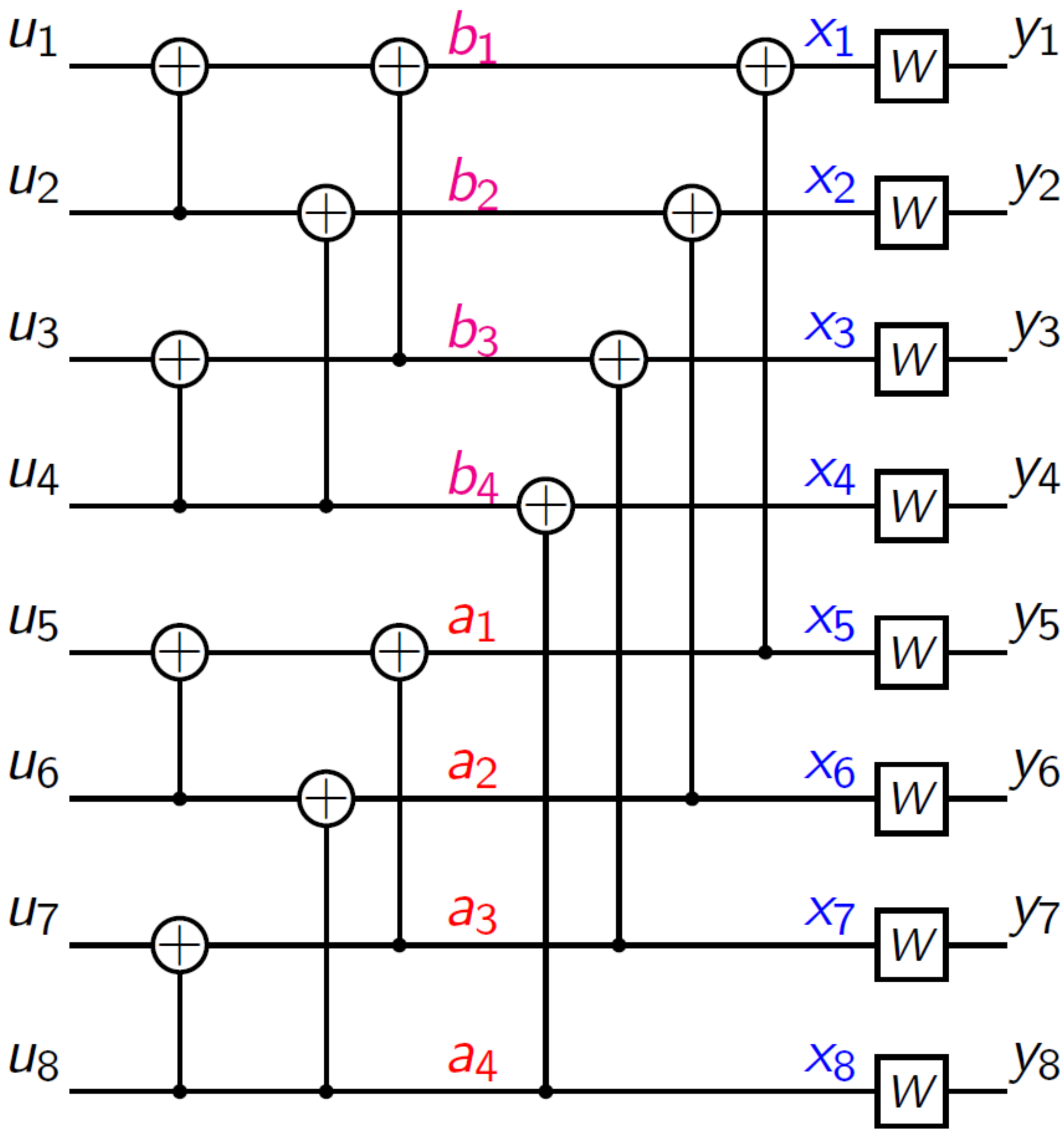
Fig. 7. Job output times and decoding times for $N = 512$.

Extensions

- Ternary Erasure Channel

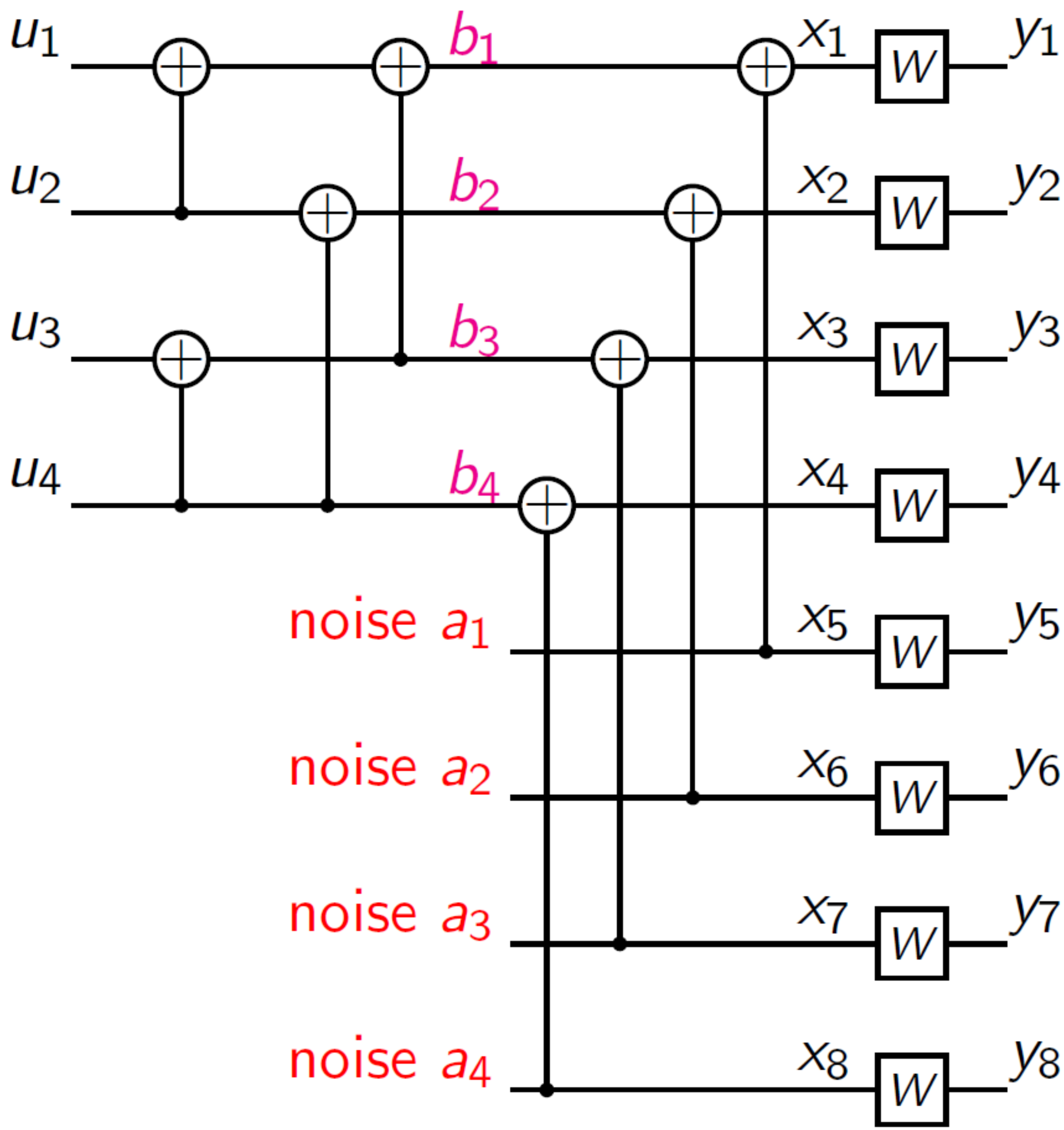


Details on Decoding: Divide and Conquer



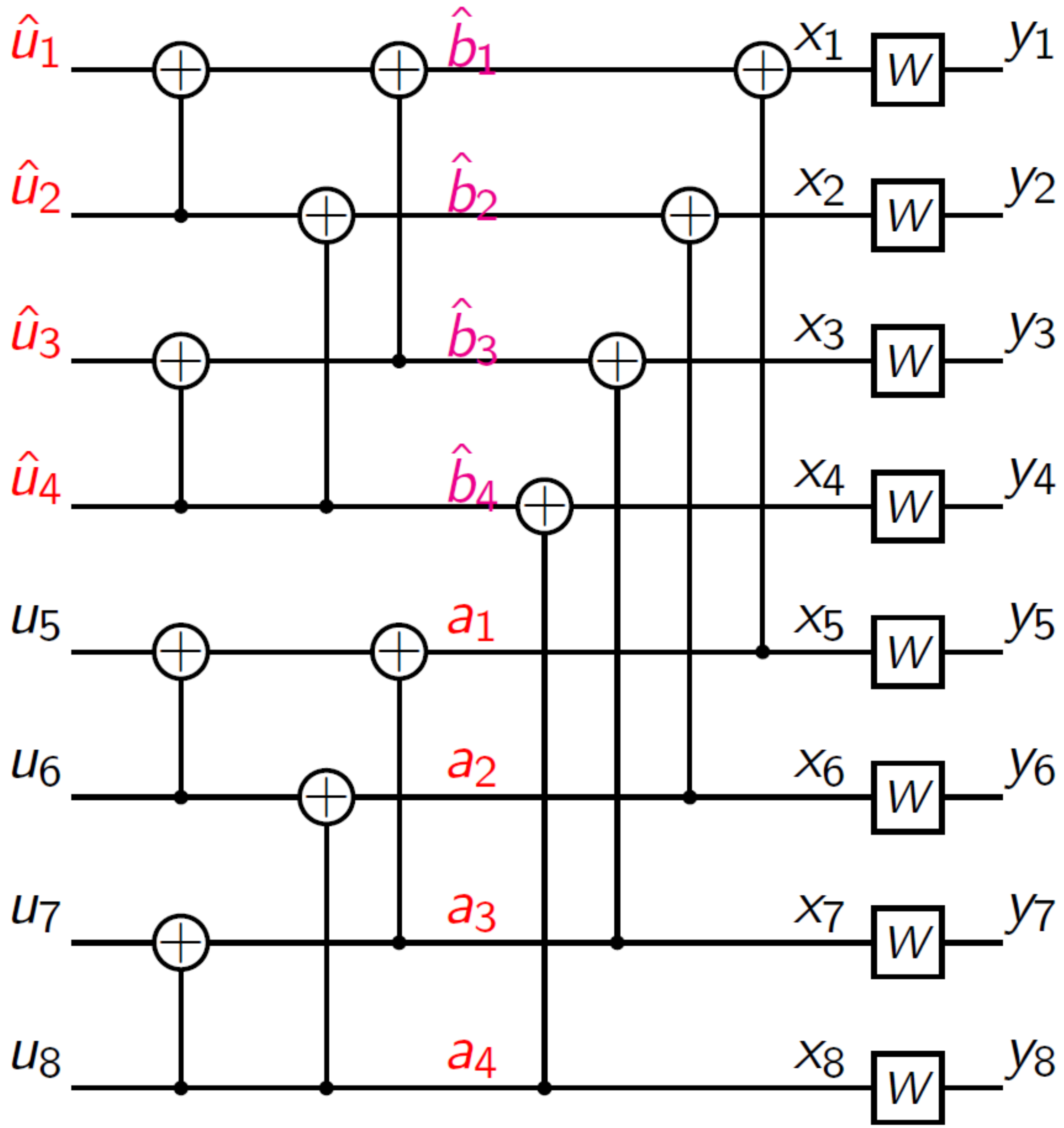
Successive Cancellation Decoder

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



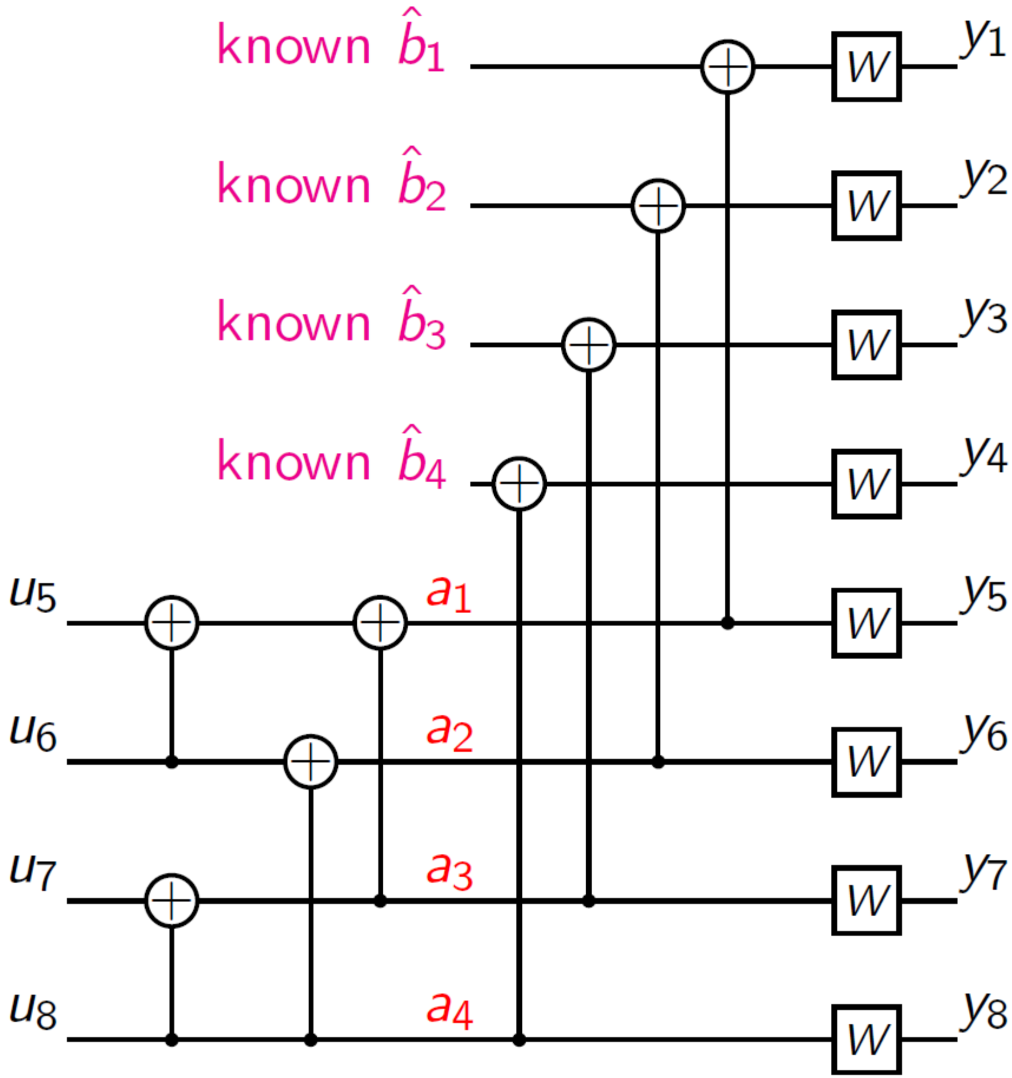
Successive Cancellation Decoder

End of first phase



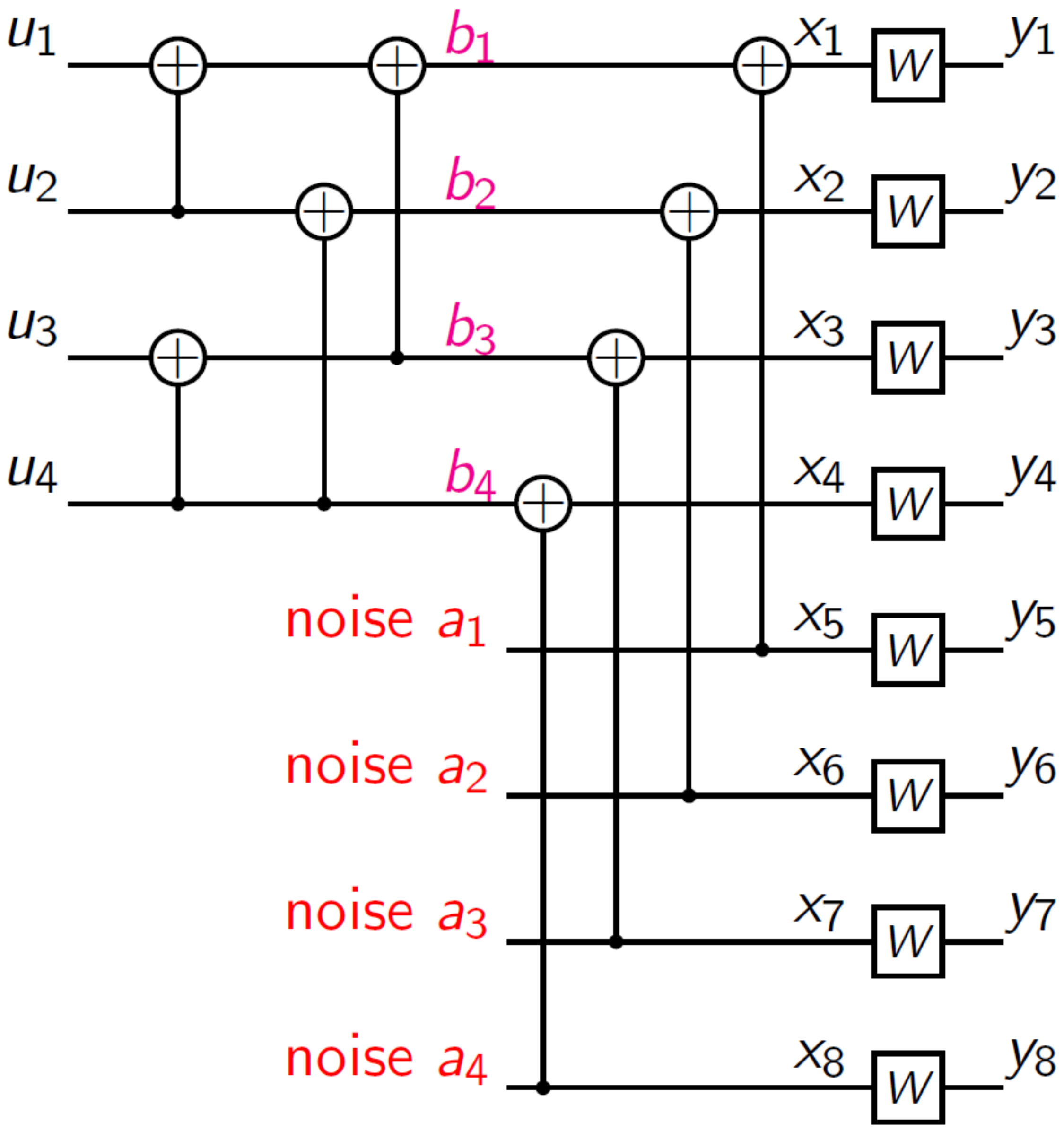
Successive Cancellation Decoder

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



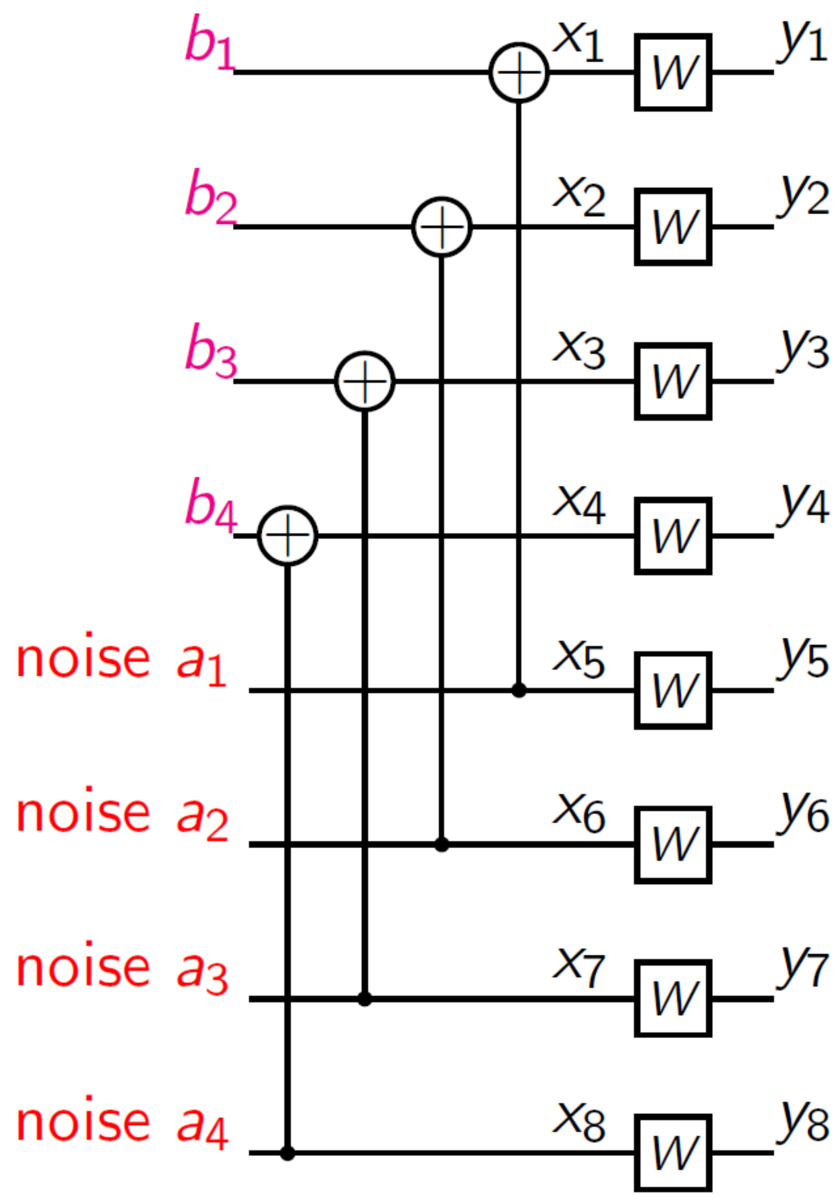
Successive Cancellation Decoder

First phase in detail



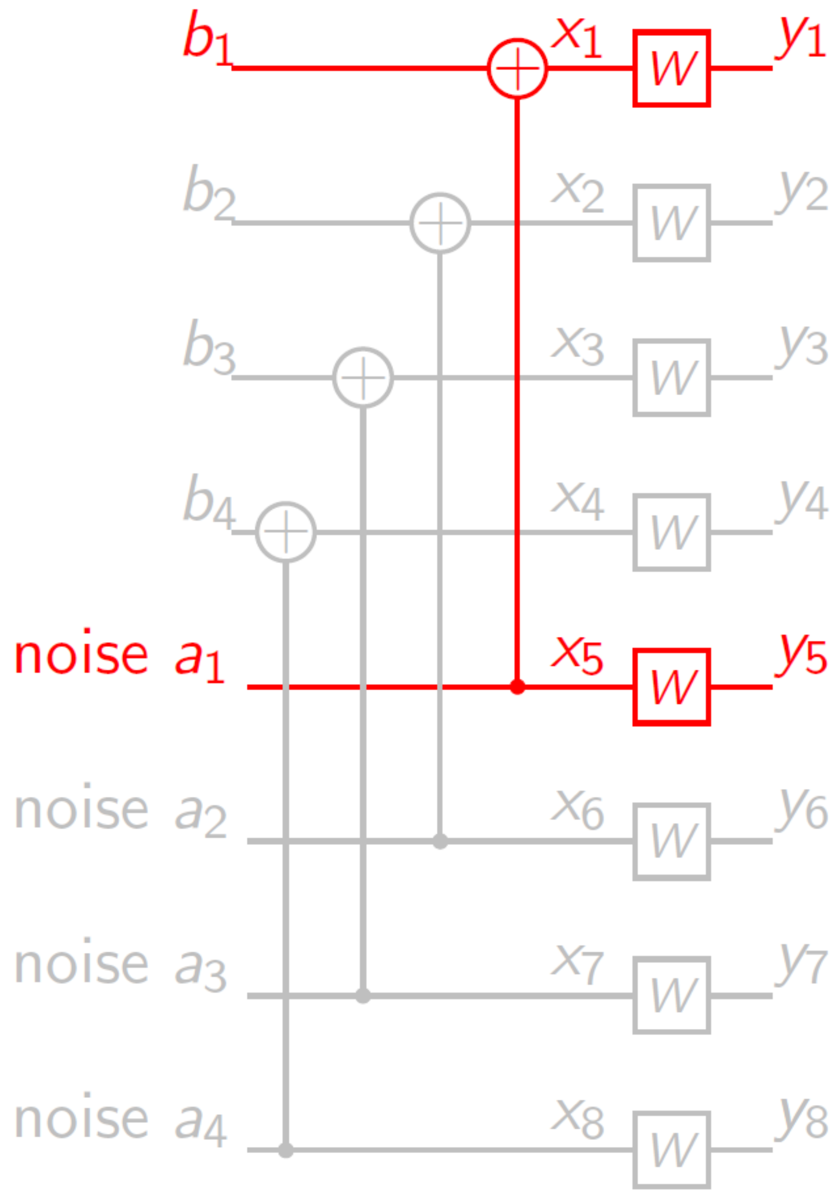
Successive Cancellation Decoder

Equivalent channel model



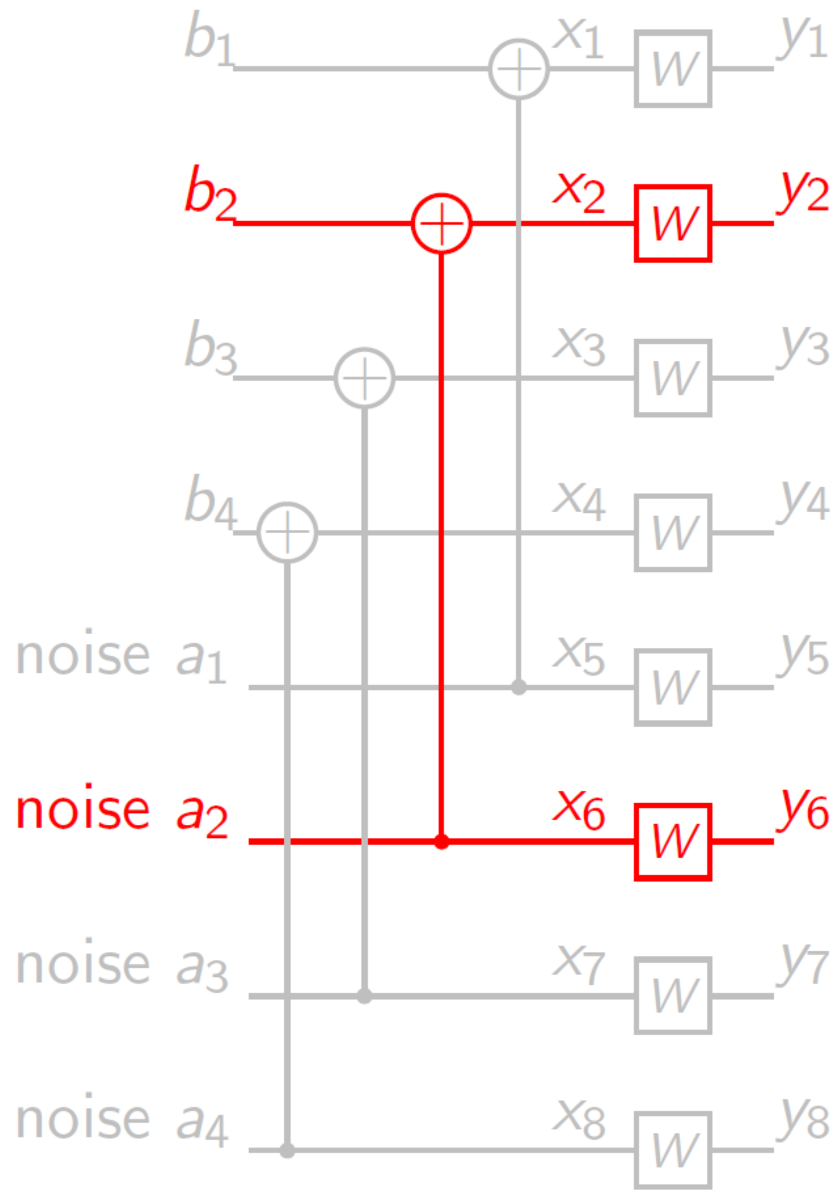
Successive Cancellation Decoder

First copy of W^-



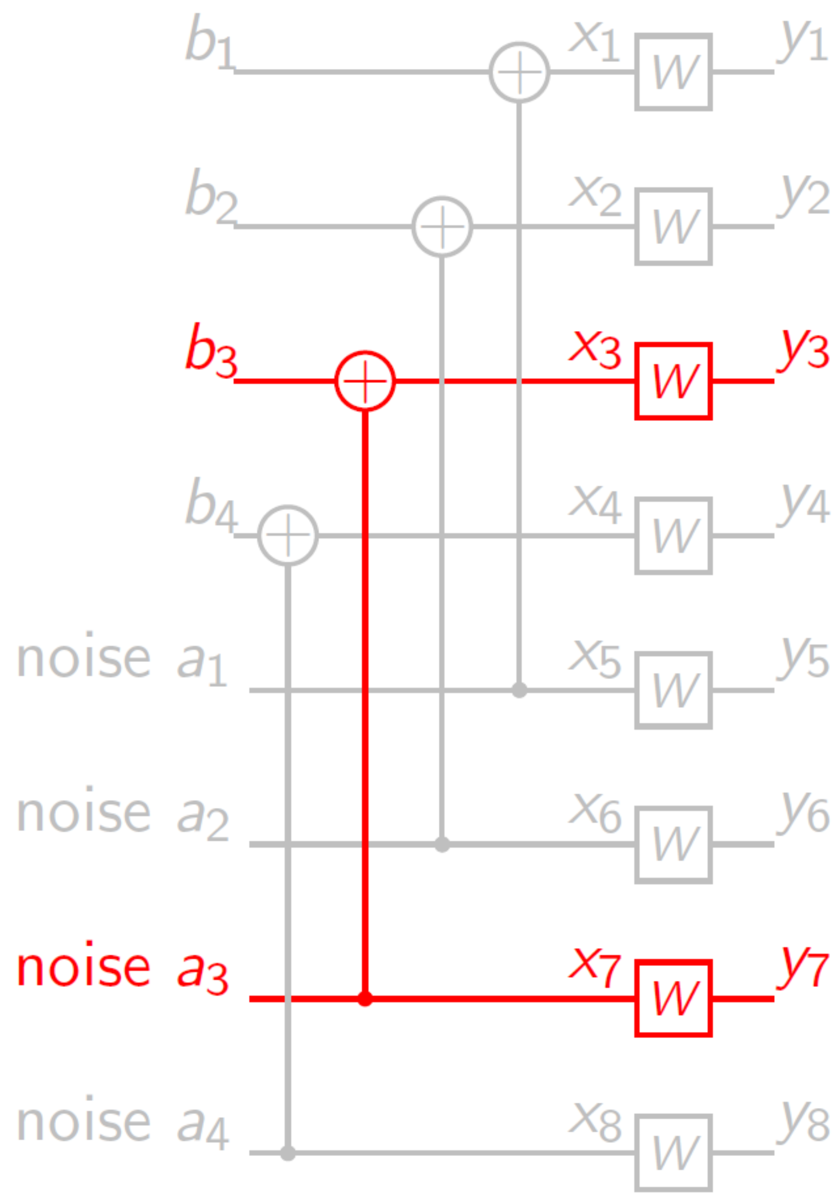
Successive Cancellation Decoder

Second copy of W^-



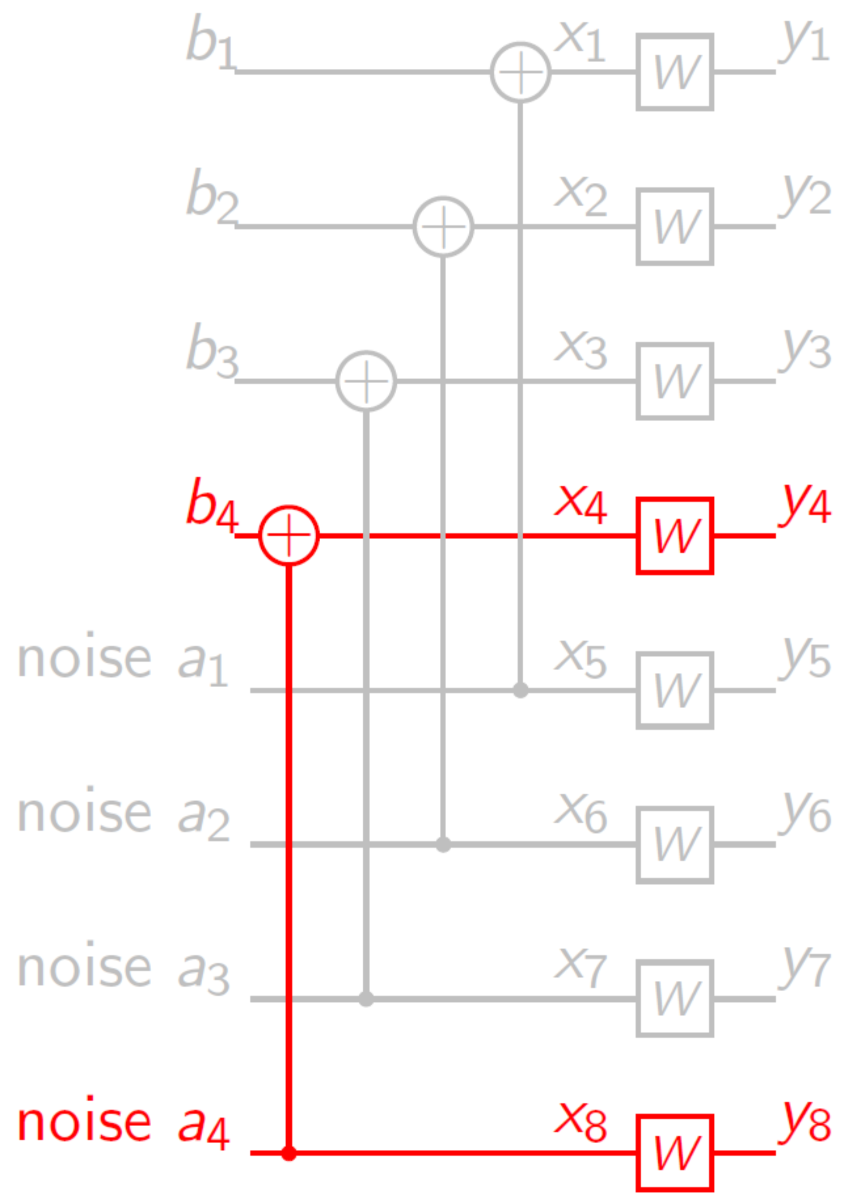
Successive Cancellation Decoder

Third copy of W^-

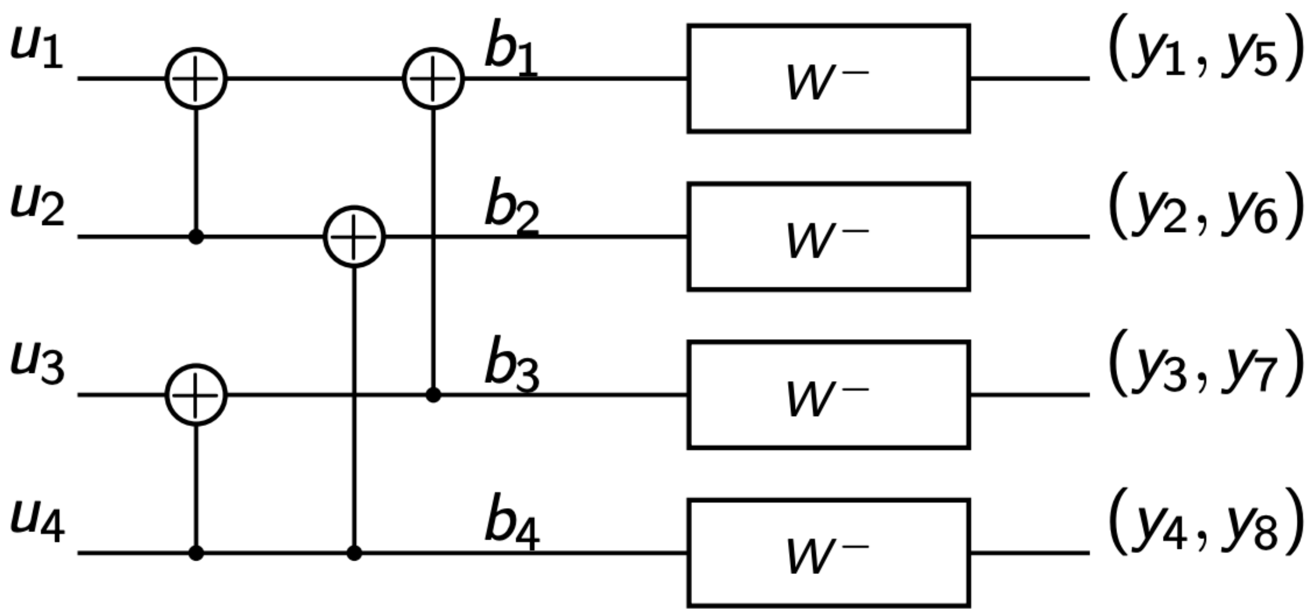


Successive Cancellation Decoder

Fourth copy of W^-

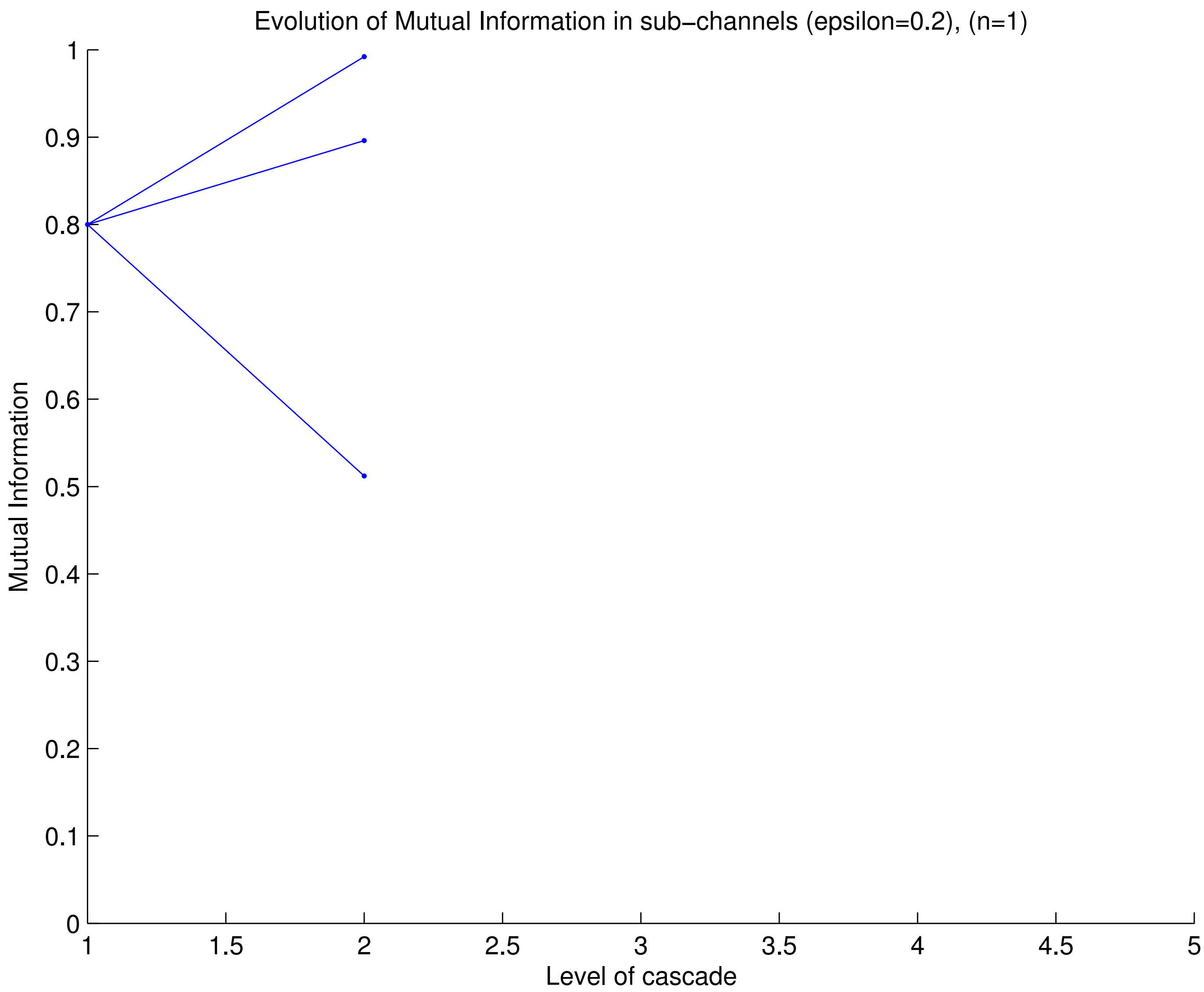


Reduction to 4×4

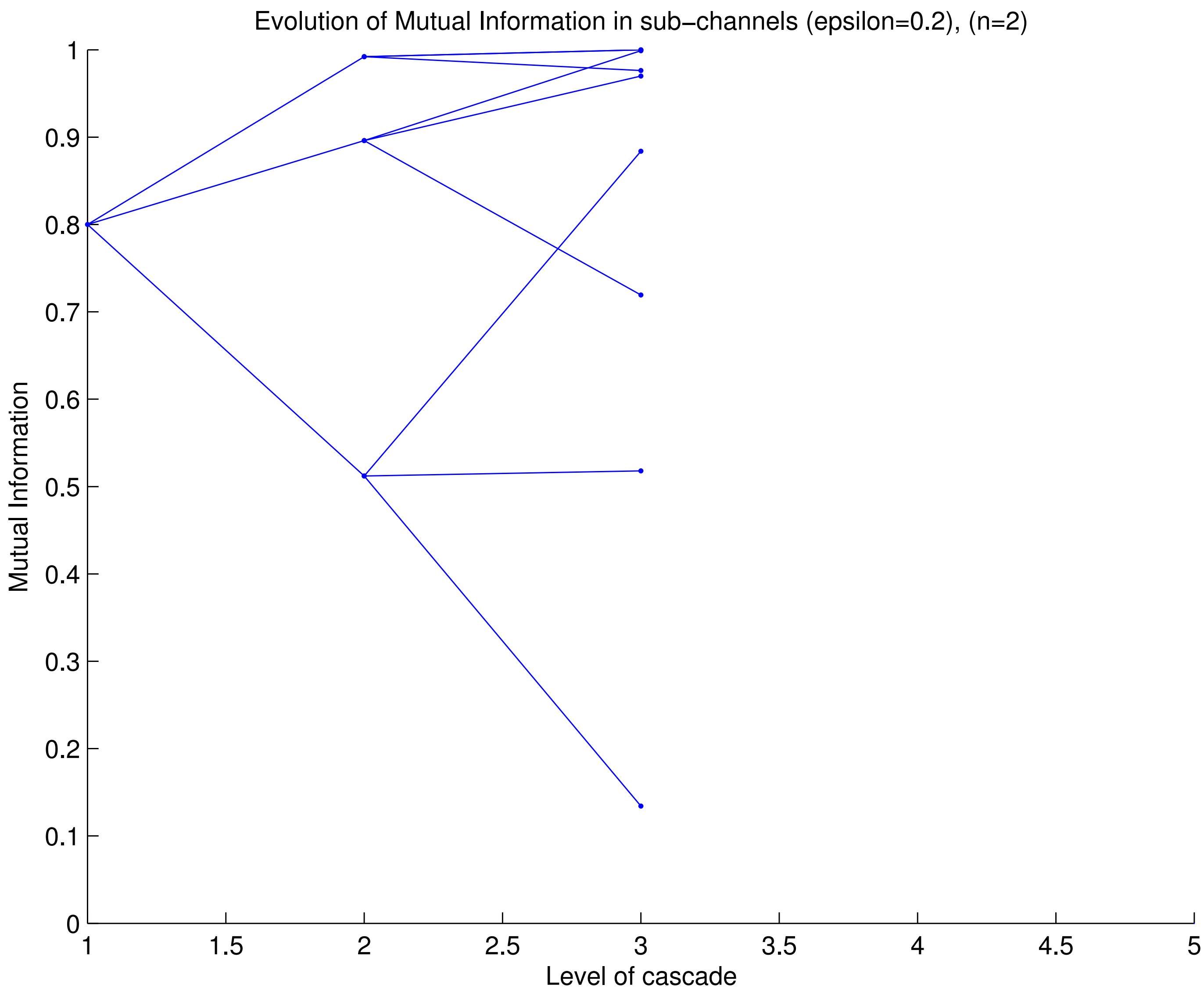


- Decoding the lower block u_5, u_6, u_7, u_8 is done similarly with a 4×4 block of W^+ channels

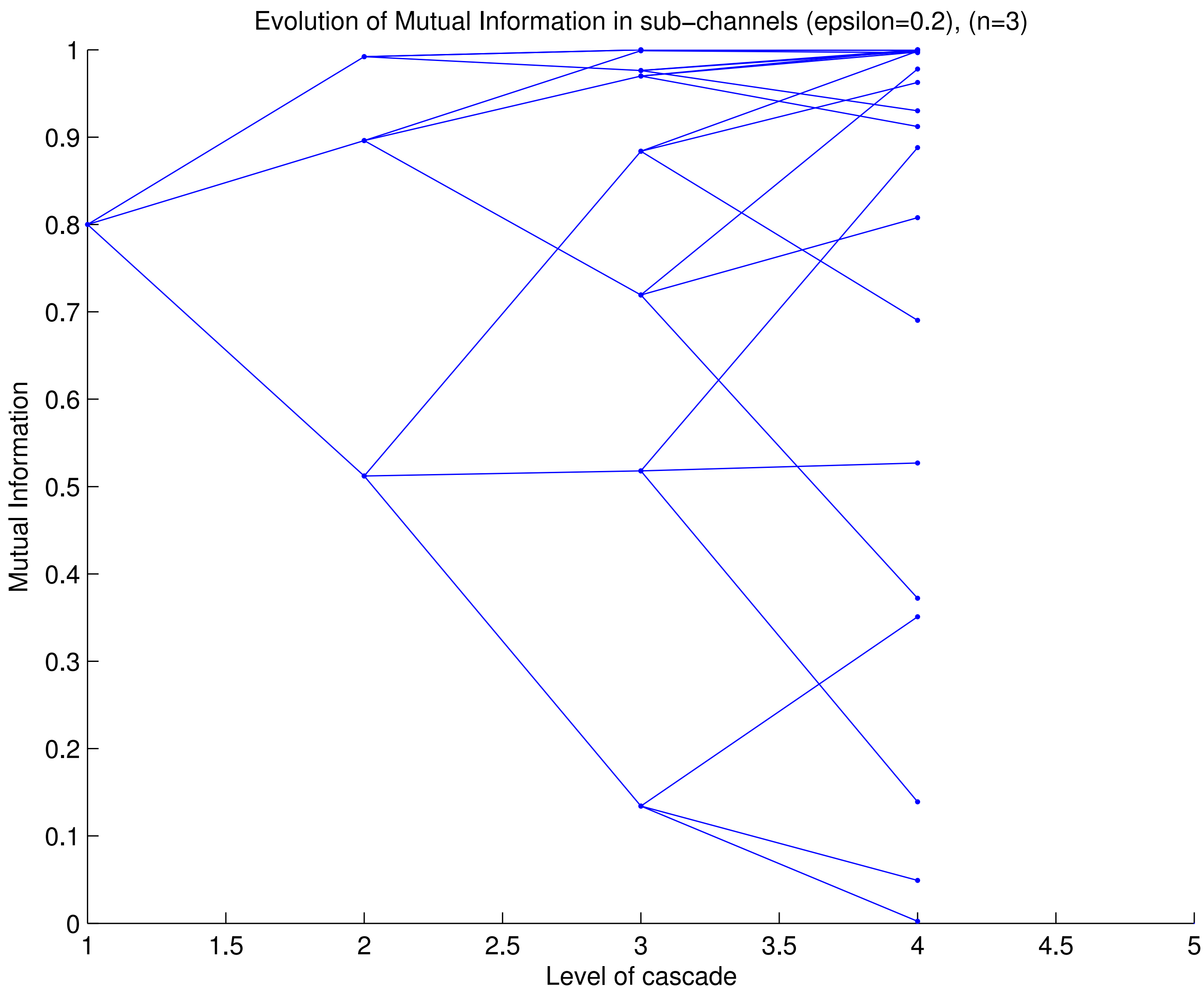
Mutual Information in TEC



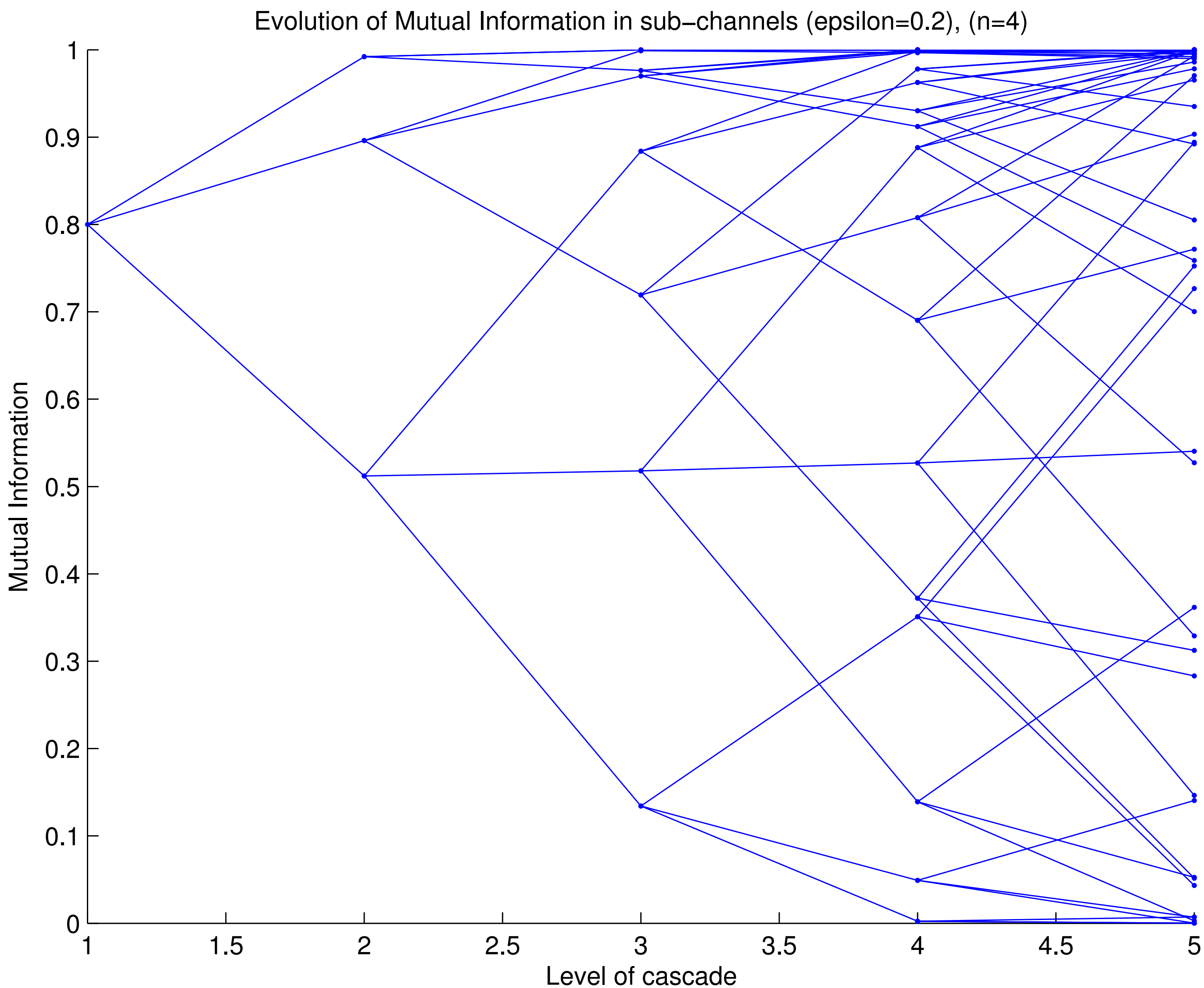
Mutual Information in TEC



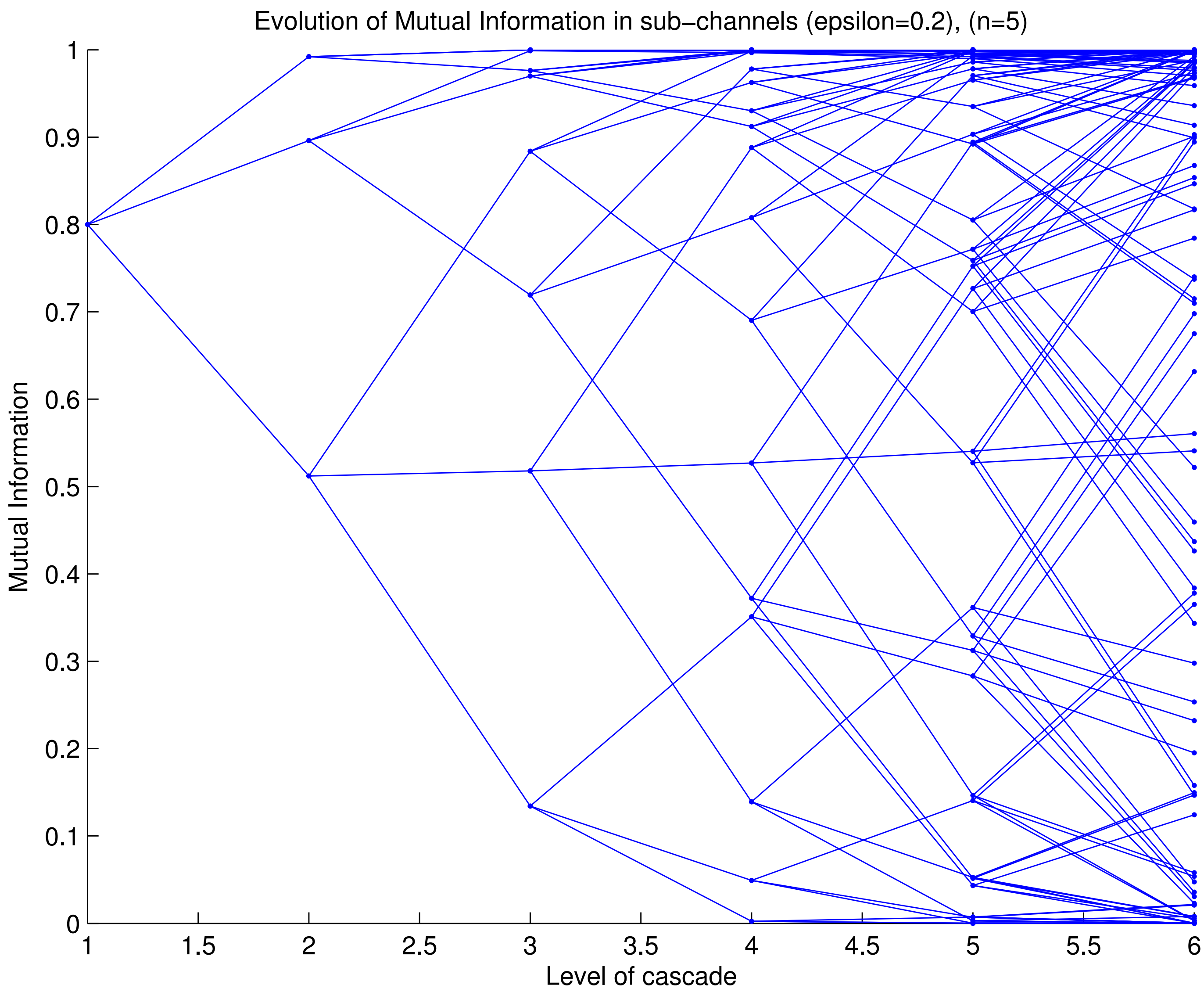
Mutual Information in TEC



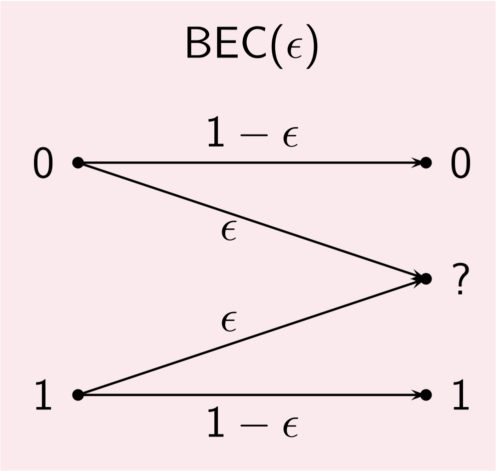
Mutual Information in TEC



Mutual Information in TEC



Capacity of the binary erasure channel (BEC)



$$
\begin{aligned}
I(X;Y) &= H(X) - H(X|Y) \\
&= H(X) - H(X)\epsilon - 0\,P(Y=0) - 0\,P(Y=1) \\
&= (1-\epsilon)H(X)
\end{aligned}
$$

Picking $X \sim Ber(\frac{1}{2})$, we have $H(X) = 1$. Thus, the capacity of BEC is $C = 1 - \epsilon$ Capacity of the BEC with erasure probability ϵ is $C = 1 - \epsilon$

References

- Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels, E. Arikan - IEEE Transactions on information Theory, 2009
- A Short Course on Polar Coding, E. Arikan, 2016