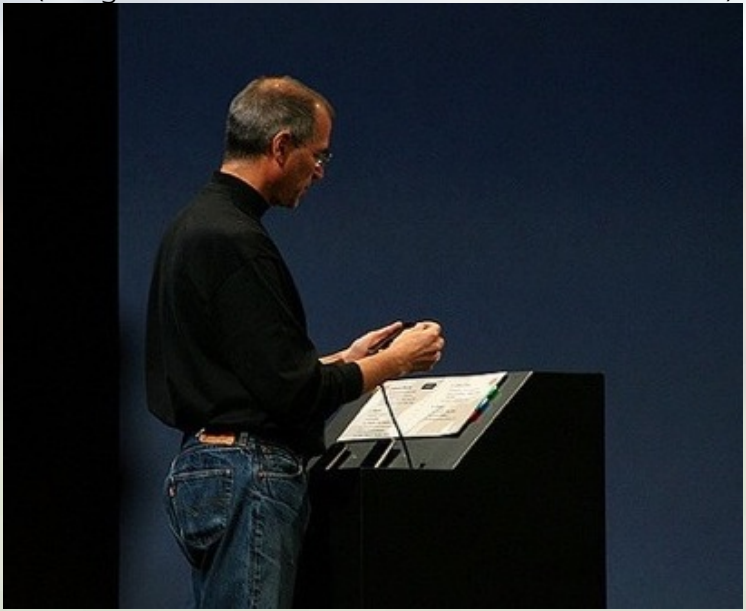
Channel Manipulation as a Coding Technique

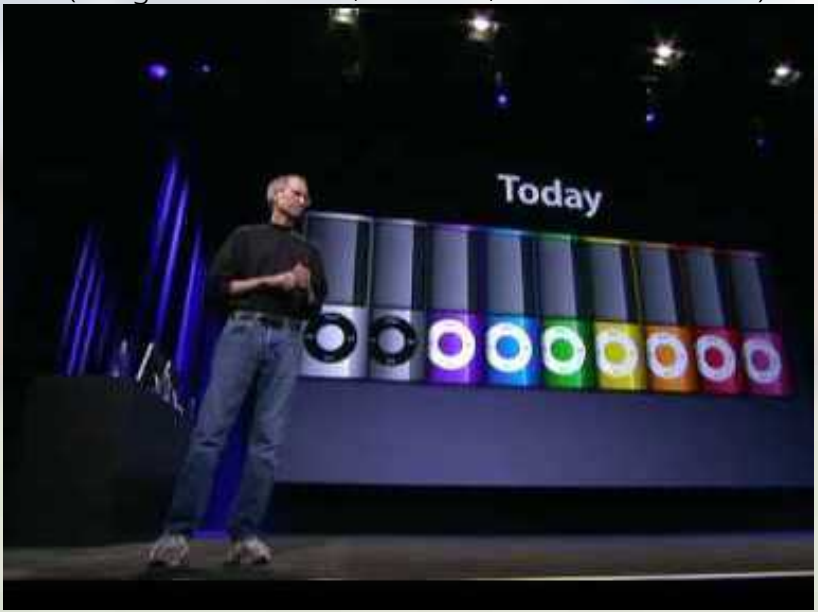
Hsin-Po Wang (EECS, UC Berkeley)

What do the following pictures have in common?

(Images from Wired, Youtube, iPhone in Canada)

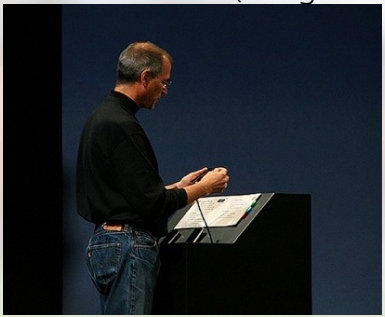


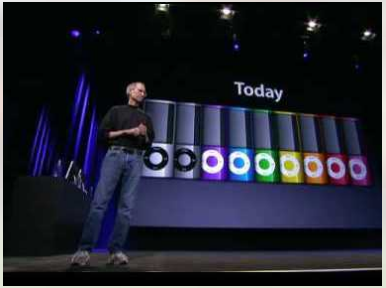
(Images from Wired, Youtube, iPhone in Canada)



(Images from Wired, Youtube, iPhone in Canada)

(Images from Wired, Youtube, iPhone in Canada)



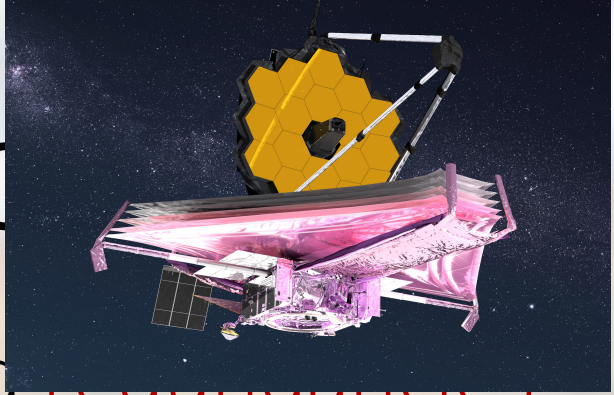


https://job  lk.symbol.codes

Coding

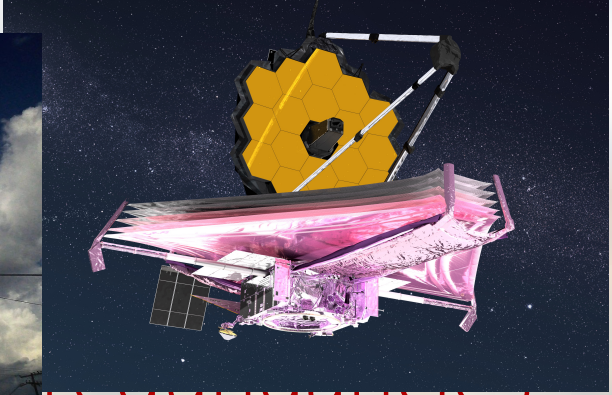
= adding redundancies
in a smart way

Co



= adding redundancies
in a smart way

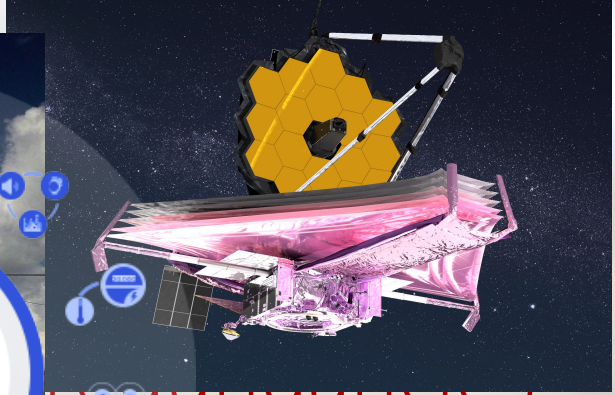
(Images from Wikipedia, NASA, IBM, Qualcomm)



redundancies

in a smart way

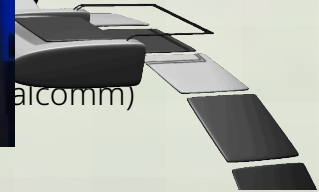
(Images from Wikipedia, NASA, IBM, Qualcomm)



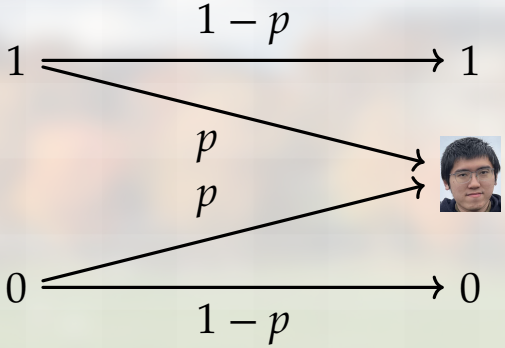
smart way

(Images from Wikipedia, NASA, IBM, Qualcomm)

(Images from Wikipe… NAS… SM, Qualcomm)



Math Framework of Coding



Binary erasure channel with erasure probability p: BEC(p)

V = 86 T = 84 M = 77 a = 97 t = 116 h = 104

The “VTMath” polynomial:

$$f(x) = 86 + 84x + 77x^2 + 97x^3 + 116x^4 + 104x^5$$

V = 86 T = 84 M = 77 a = 97 t = 116 h = 104

The “VTMath” polynomial:

$f(x) =$  5

$f(-3) = 968 \quad f(-2) = 22 \quad f(-1) = -6 \quad f(0) = 86$

$f(1) = 564 \quad f(2) = 522 \quad f(3) = 318 \quad f(4) = 54$

V = 86 T = 84 M = 77 a = 97 t = 116 h = 104

The “VTMath” polynomial:

$f(x) =$  5

$f(-3) = 968 \quad f(-2) = 22 \quad f(-1) = -6 \quad f(0) = 8$ 

$f(1) = 564 \quad f(2) = 522 \quad f(3) = 318 \quad f(4) = 54$

V = 86 T = 84 M = 77 a = 97 t =
116 h = 104

The “VTMath” polynomial:

$f(x) =$ 

$f(-3) = 968$ $f(-2) = 22$ $f(-1) = -6$ $f(0) = 8$ 

$f(1) =$  $f(2) = 522$ $f(3) = 318$ $f(4) = 54$

V = 86 T = 84 M = 77 a = 97 t = 116 h = 104

The “VTMath” polynomial:

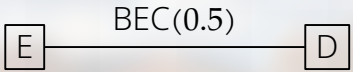
$f(x) =$ 

$f(-3) = 968 \quad f(-2) = 22 \quad f(-1) = -6 \quad f(0) = 8$ 

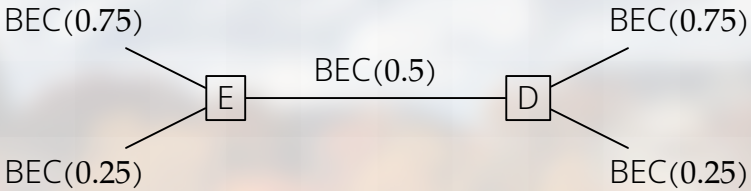
$f(1) =$  $\quad f(2) = 522 \quad f(3) =$  $\quad f(4) = 54$

New Idea

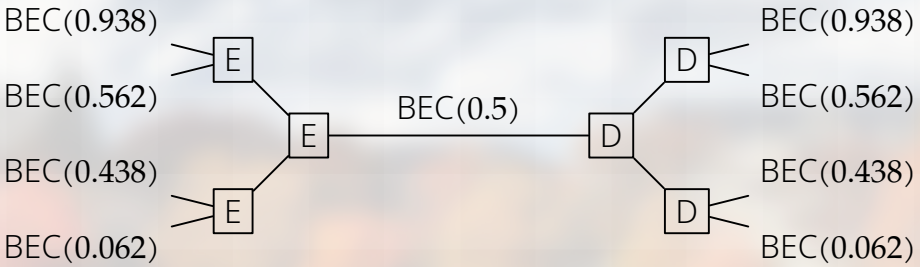
Polar Codes



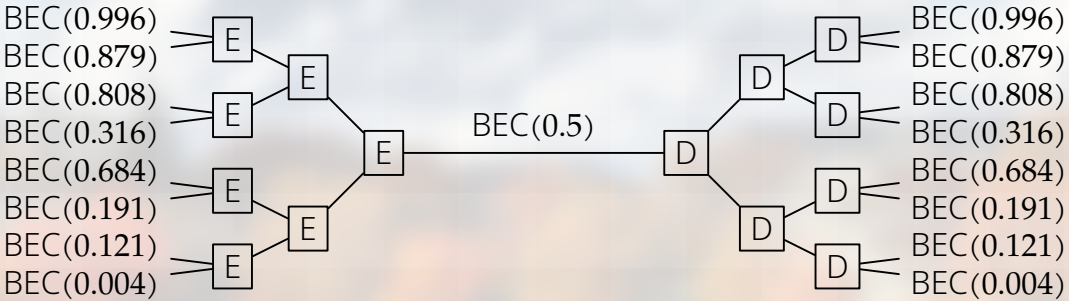
Suppose there are magic devices E and D that turns BEC(x) into BEC(x^2) and BEC($2x - x^2$).



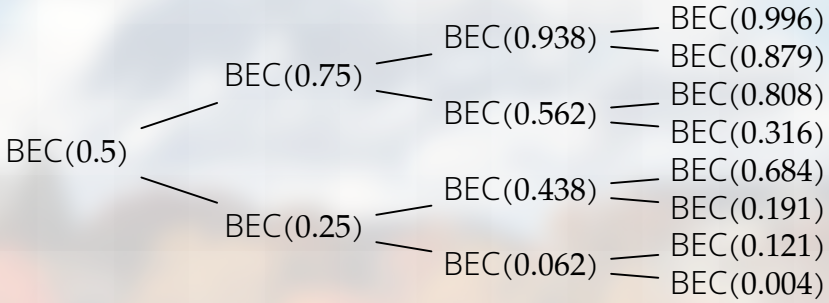
Suppose there are magic devices E and D that turns BEC(x) into BEC(x^2) and BEC($2x - x^2$).



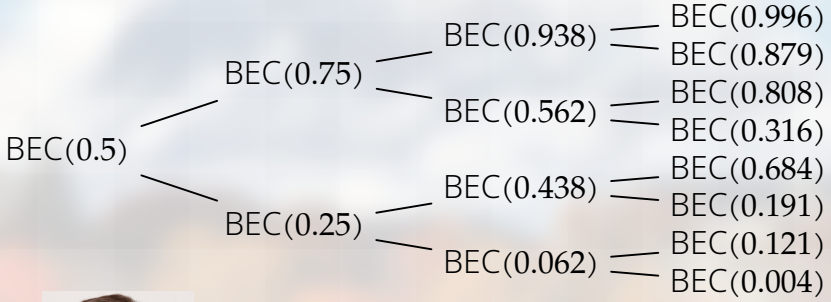
What if we apply apply more magic devices?



What if we apply apply more magic devices?
And more and more and more???

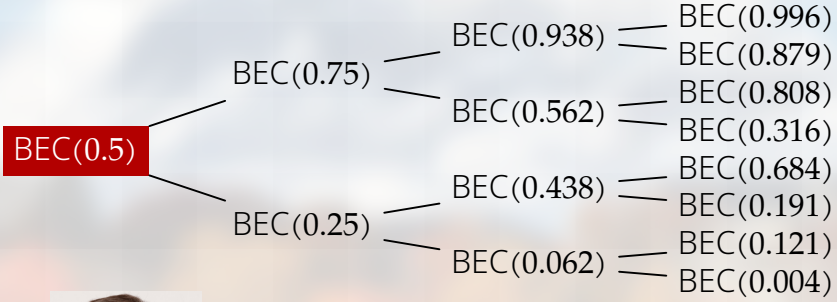


This is a tree



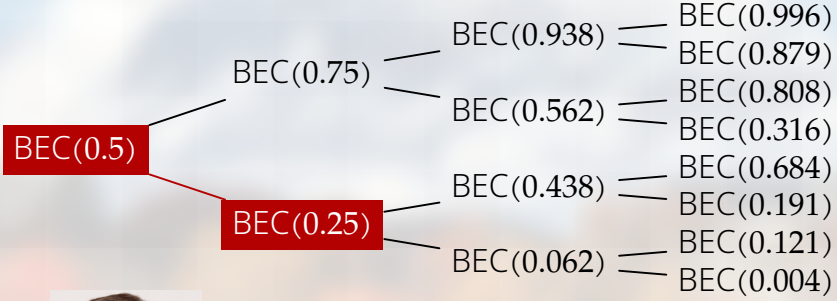
: This is a martingale

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$



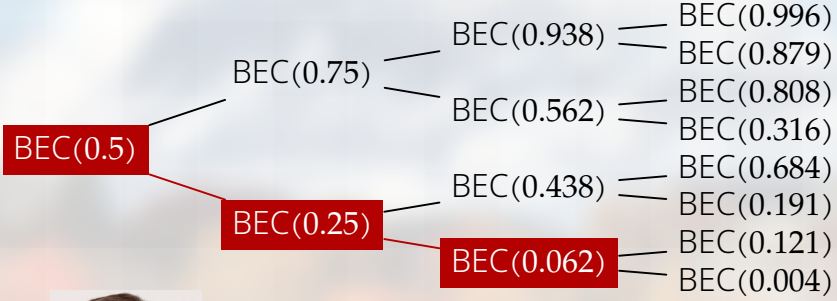
: This is a martingale

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$



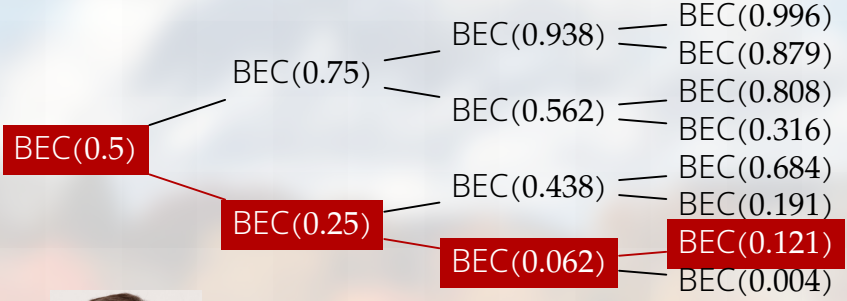
: This is a martingale

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$



: This is a martingale

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$



: This is a martingale

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$

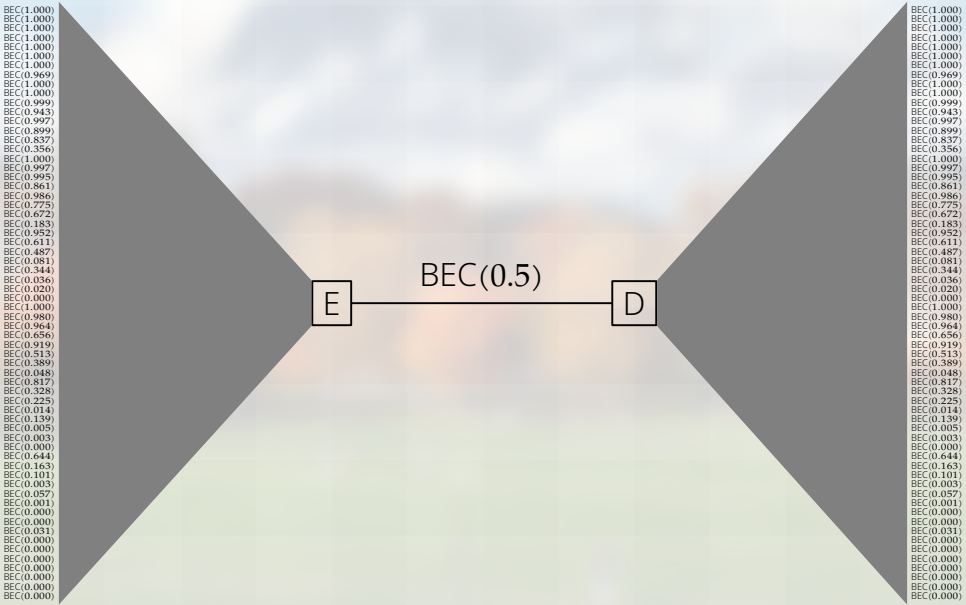
: Bounded martingale converges.

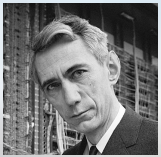
$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$

: Bounded martingale converges.

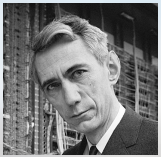
: BECs converges to 0 or 1.

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$


BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(0.969)
BEC(1.000)
BEC(1.000)
BEC(0.999)
BEC(0.943)
BEC(0.997)
BEC(0.899)
BEC(0.837)
BEC(0.356)
BEC(1.000)
BEC(0.997)
BEC(0.995)
BEC(0.861)
BEC(0.986)
BEC(0.775)
BEC(0.672)
BEC(0.183)
BEC(0.952)
BEC(0.611)
BEC(0.487)
BEC(0.081)
BEC(0.344)
BEC(0.036)
BEC(0.020)
BEC(0.000)
BEC(1.000)
BEC(0.980)
BEC(0.964)
BEC(0.656)
BEC(0.919)
BEC(0.513)
BEC(0.389)
BEC(0.048)
BEC(0.817)
BEC(0.328)
BEC(0.225)
BEC(0.014)
BEC(0.139)
BEC(0.005)
BEC(0.003)
BEC(0.000)
BEC(0.644)
BEC(0.163)
BEC(0.101)
BEC(0.003)
BEC(0.057)
BEC(0.001)
BEC(0.000)
BEC(0.000)
BEC(0.031)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
E
BEC(0.5)
D
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(0.969)
BEC(1.000)
BEC(1.000)
BEC(0.999)
BEC(0.943)
BEC(0.997)
BEC(0.899)
BEC(0.837)
BEC(0.356)
BEC(1.000)
BEC(0.997)
BEC(0.995)
BEC(0.861)
BEC(0.986)
BEC(0.775)
BEC(0.672)
BEC(0.183)
BEC(0.952)
BEC(0.611)
BEC(0.487)
BEC(0.081)
BEC(0.344)
BEC(0.036)
BEC(0.020)
BEC(0.000)
BEC(1.000)
BEC(0.980)
BEC(0.964)
BEC(0.656)
BEC(0.919)
BEC(0.513)
BEC(0.389)
BEC(0.048)
BEC(0.817)
BEC(0.328)
BEC(0.225)
BEC(0.014)
BEC(0.139)
BEC(0.005)
BEC(0.003)
BEC(0.000)
BEC(0.644)
BEC(0.163)
BEC(0.101)
BEC(0.003)
BEC(0.057)
BEC(0.001)
BEC(0.000)
BEC(0.000)
BEC(0.031)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)

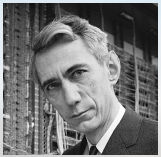


: 5000 bits / 10000 channel uses



: 5000 bits / 10000 channel uses

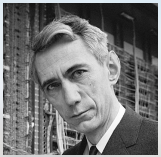
: Polar 4900 bits / 10000 uses



: 5000 bits / 10000 channel uses

: Polar 4900 bits / 10000 uses

In general, $(1/2 - \varepsilon)N$ bits / N uses



: 5000 bits / 10000 channel uses

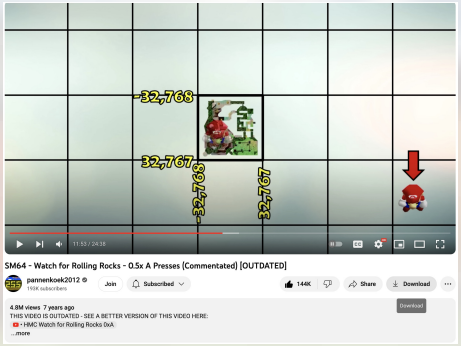
: Polar 4900 bits / 10000 uses

In general, $(1/2 - \varepsilon)N$ bits / N uses

THE END?

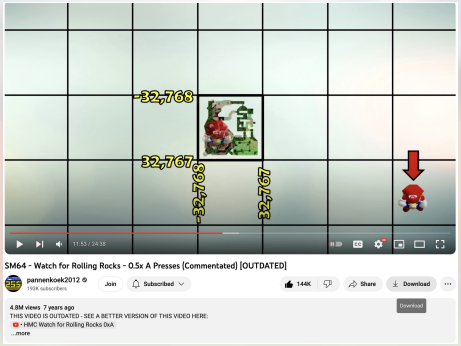
5G is for

5G is for



video streaming

5G is for

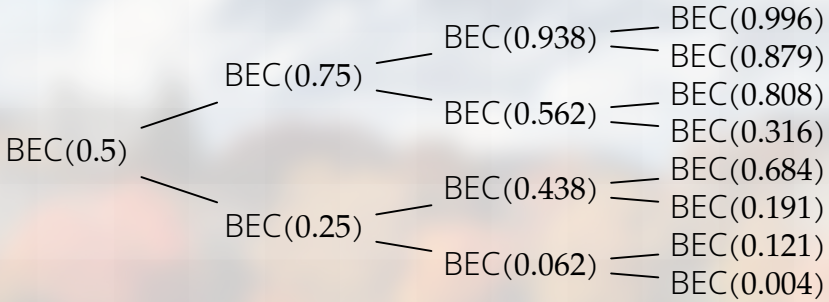


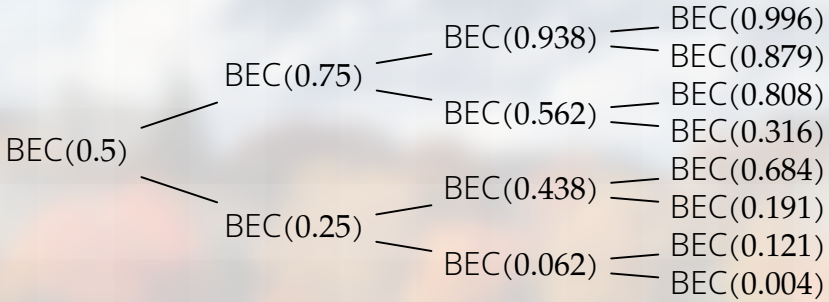
video streaming

flying drone

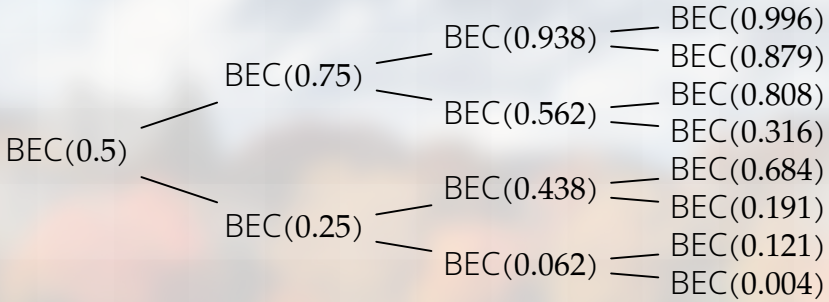
Channel Manipulation is Coding

H.-P. Wang


BEC(0.5)
BEC(0.75)
BEC(0.25)
BEC(0.938)
BEC(0.562)
BEC(0.438)
BEC(0.062)
BEC(0.996)
BEC(0.879)
BEC(0.808)
BEC(0.316)
BEC(0.684)
BEC(0.191)
BEC(0.121)
BEC(0.004)



Recall tree: early channels not useful.



Recall tree: early channels not useful. How to accelerate polarization?

Martingale–Code Rate Thm [folklore]

Martingale–Code Rate Thm [folklore]

$$\text{Prob}\{4^{-n} < M_n < 1 - 4^{-n}\} < 2^{-\rho n}$$

implies

$(1/2 - N^{-\rho})N$ bits / N channel uses.

$(1/2 - N^{-\rho})N$ bits / N channel uses

History of ρ over BMSC:0 $1/2$

$(1/2 - N^{-\rho})N$ bits / N channel uses

History of ρ over BMSC: 0 $1/2$

2015 Guruswami–Xia

2012 Goli–Hassani–Urbanke

2014 Hassani–Alishahi–Urbanke

2014 Goldin–Burshtein

2016 Mondelli–Hassani–Urbanke

$(1/2 - N^{-\rho})N$ bits / N channel uses

History of ρ over BMSC: 0 $1/2$

2015 Guruswami–Xia ⟷

2012 Goli–Hassani–Urbanke ⟷

2014 Hassani–Alishahi–Urbanke ⟷

2014 Goldin–Burshtein ⟷

2016 Mondelli–Hassani–Urbanke ↔

2022  –Lin–Vardy–Gabrys ↔

Improve ρ over BEC: 0 $1/2$


Improve ρ over BEC: 0
1/2
2010 Hassani–Alishahi–Urbanke 2×2
2010 Korada–Montanari–Telatar–Urbanke 2×2
2014 Fazeli–Vardy 8×8
2021 Trofimiuk–Trifonov 16×16
2021 Trofimiuk 24×24
2021 Yao–Fazeli–Vardy 32×32
2021 Yao–Fazeli–Vardy 64×64


Improve ρ over BEC: 0
1/2
2010 Hassani–Alishahi–Urbanke 2×2
2010 Korada–Montanari–Telatar–Urbanke 2×2
2014 Fazeli–Vardy 8×8
2021 Trofimiuk–Trifonov 16×16
2022 Duursma–Gabrys–Guruswami–Lin– $2 \times 2/GF4$
2021 Trofimiuk 24×24
2021 Yao–Fazeli–Vardy 32×32
2021 Yao–Fazeli–Vardy 64×64
Channel Manipulation is Coding
H.-P. Wang

The optimal ρ: 0 1/2

The optimal ρ: 0 $1/2$

2019 Pfister–Urbanke
q-ary erasure channel, $q \to \infty$

2021 Fazeli–Hassani–Mondelli–Vardy
binary erasure channel

2022 Guruswami–Riazanov–Ye
binary symmetric memoryless channel

The optimal ρ: 0 1/2

2019 Pfister–Urbanke
q-ary erasure channel, $q \to \infty$

2021 Fazeli–Hassani–Mondelli–Vardy
binary erasure channel

2022 Guruswami–Riazanov–Ye
binary symmetric memoryless channel

2021  –Duursma
discrete memoryless channel

2011 Alamdar-Yazdi–Kschischang:
Prune the tree to reduce complexity.

2017 El-Khamy–Mahdavifar–Feygin–Lee–Kang:
Pruning reduces complexity by a scalar; still $O(N \log N)$.

2021 Mondelli–Hashemi–Cioffi–Goldsmith,
2021 Hashemi–Mondelli–Fazeli–Vardy–Cioffi–Goldsmith:
Study parallelism vs latency.

2011 Alamdar-Yazdi–Kschischang:
Prune the tree to reduce complexity.

2017 El-Khamy–Mahdavifar–Feygin–Lee–Kang:
Pruning reduces complexity by a scalar; still $O(N \log N)$.

2021  –Duursma: $O(N \log \log N)$
if relax the performance requirement.

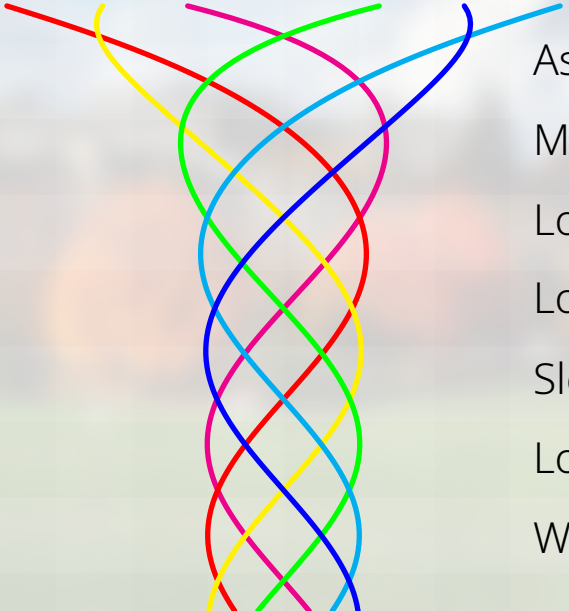
Trade-off: complexity $\approx O(N \log(-\log(\text{decode error})))$.

2021 Mondelli–Hashemi–Cioffi–Goldsmith,
2021 Hashemi–Mondelli–Fazeli–Vardy–Cioffi–Goldsmith:
Study parallelism vs latency.

Polar code is a mathy code

Polar code is a mathy code

Polar achieves the capacity of



Asymmetric channel

Multiple access channel

Lossless compression

Lossy compression

Slepian–Wolf

Lossless compression w/ helper

Wiretap channel (degradation)

Deletion channel ... (good error prob)

Broadcast channel ... (good error prob)

Channel with memory ... (good error prob)

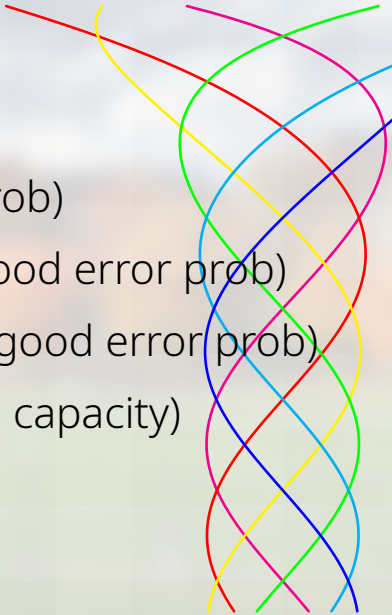
Wiretap channel (no degradation) ... (good error prob)

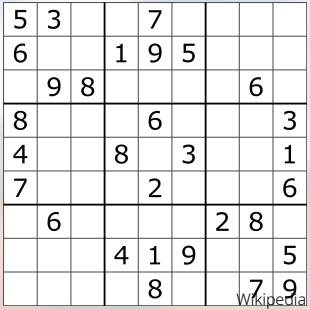
Hidden Markov chain channel state ... (good error prob)

Non-stationary channel ... (good gap to capacity)

Classical–Quantum channel ... (yes)

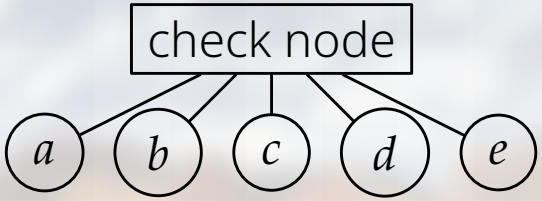
Quantum–Quantum channel ... (?)

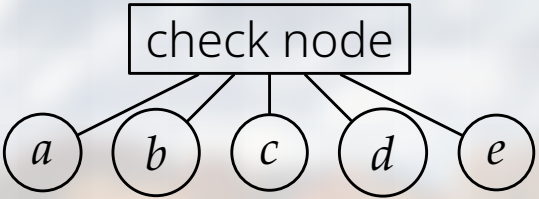




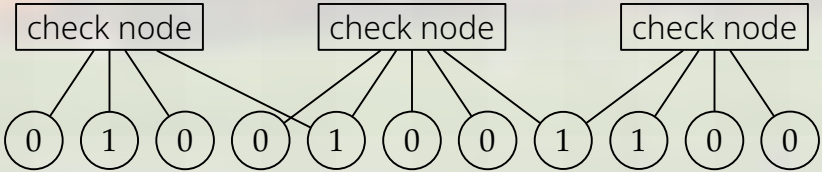
Low-Density Parity-Check (LDPC) Codes

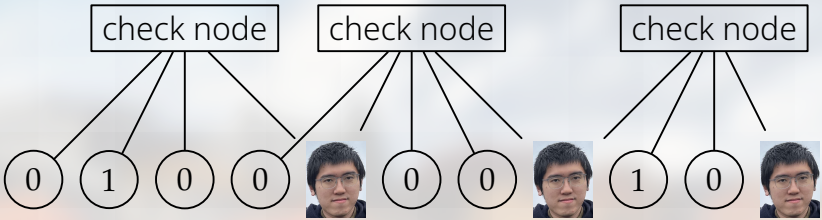
Rule: Every Sum to an even number

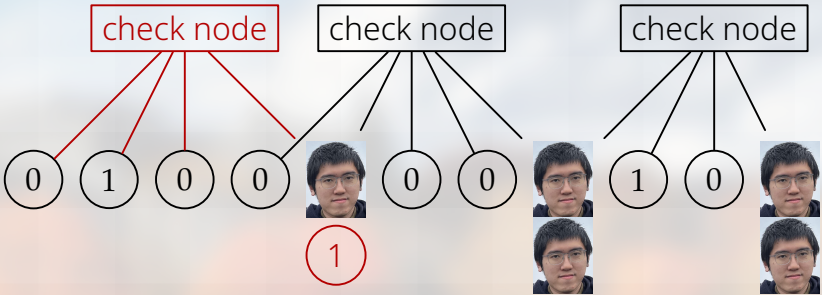


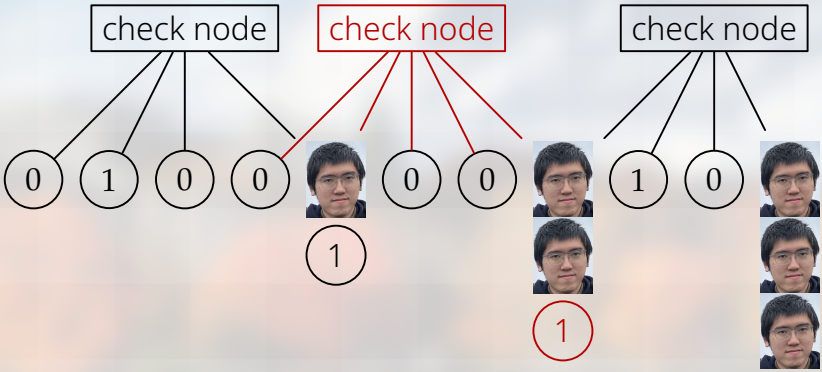


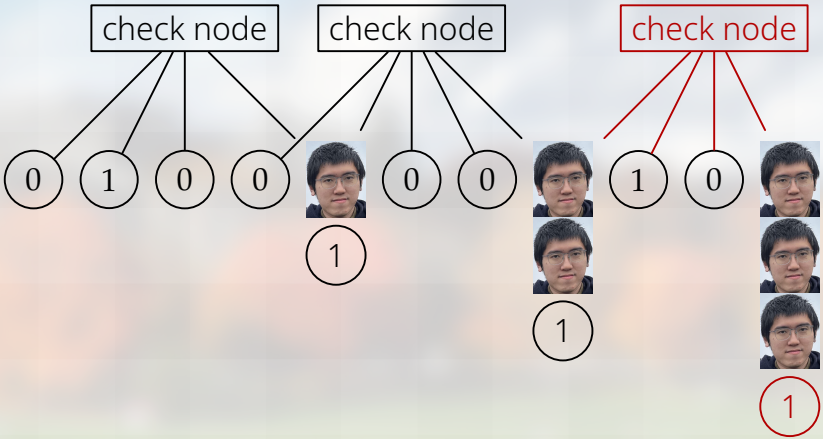
Rule: Every Sum to an even number

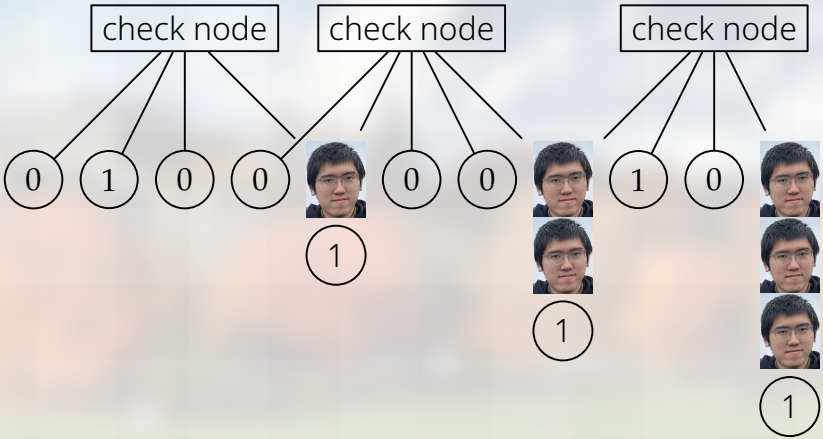



check node
check node
check node
0
1
0
0
0
0
1
0


check node
check node
check node
0
1
0
0
1
0
0
1
0
1


check node
check node
check node
0
1
0
0
1
0
0
1
1
0
1


check node
check node
check node
0
1
0
0
1
0
0
1
1
0
1
1



How to analyze, mathematically?

check node
BEC(a)
BEC(b)
BEC(c)
BEC(x)

check node
BEC(a)
BEC(b)
BEC(c)
BEC(x)
BEC(x − x(1 − a)(1 − b)(1 − c))

check node
check node
check node
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)

check node
check node
check node
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.44)
BEC(0.44)
BEC(0.44)
BEC(0.47)
BEC(0.41)
BEC(0.47)
BEC(0.47)
BEC(0.41)
BEC(0.44)
BEC(0.44)
BEC(0.44)

check node
check node
check node
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.44)
BEC(0.44)
BEC(0.44)
BEC(0.41)
BEC(0.47)
BEC(0.47)
BEC(0.47)
BEC(0.41)
BEC(0.44)
BEC(0.44)
BEC(0.44)
BEC(0.47)
BEC(0.38)
BEC(0.38)
BEC(0.38)
BEC(0.32)
BEC(0.43)
BEC(0.43)
BEC(0.43)
BEC(0.32)
BEC(0.38)
BEC(0.38)
BEC(0.38)
BEC(0.43)


0:05 / 0:39
How To Make An Ice Bubble
Rosemary Danielis
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers

Subscribe

732 Share Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers

Subscribe

732 Share Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

Problem: Theory–Reality Duality

Problem: Theory–Reality Duality

Theory:

Keep track of channel capacity

Problem: Theory–Reality Duality

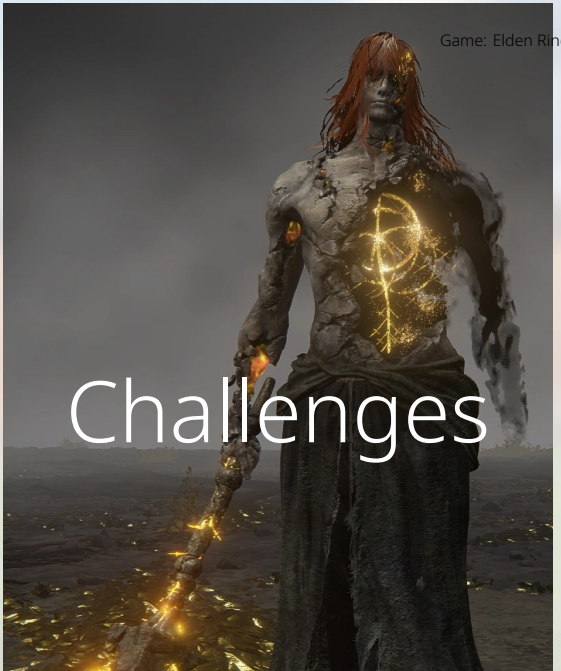
Theory:

Keep track of channel capacity

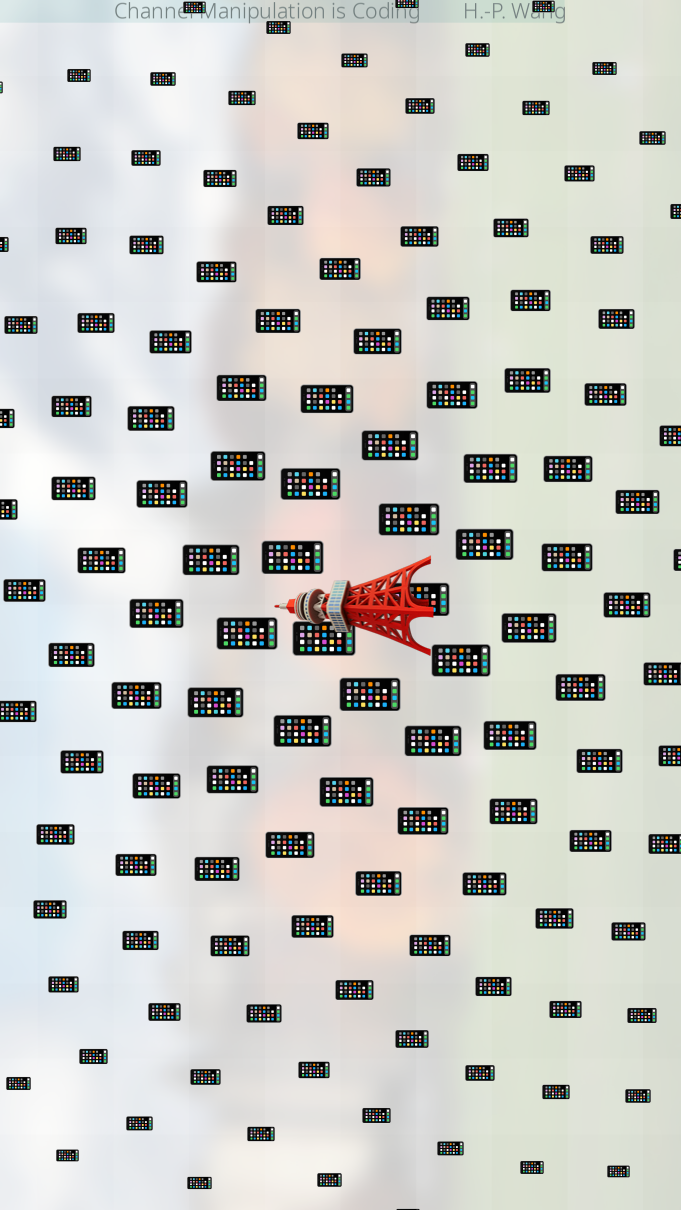
Reality:

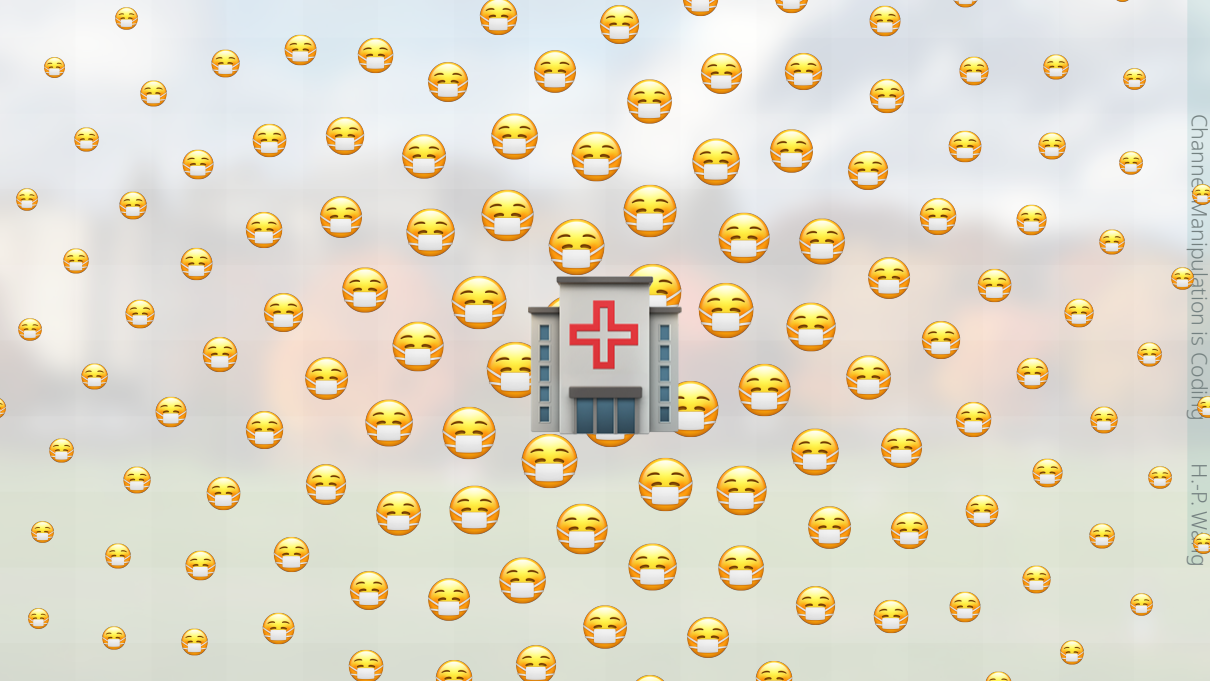
“Shape” of channels matters (BEC/BSC/AWGN…)


Wikipedia
Life
LDPC
Polar
Turbo
Reed–Muller
Channel Evolution



Challenges




Channel Manipulation is Coding
H.-P. Weng

Problem	There are n	k of them	To solve, use m
Radio protocol	cellphones	want to talk	frequencies
Disease control	people	are sick	virus tests
Search engine	webpages	contain keywords	bits per keyword

Problem	There are n	k of them	To solve, use m
Radio protocol	cellphones	want to talk	frequencies
Disease control	people	are sick	virus tests
Search engine	webpages	contain keywords	bits per keyword
Radio protocol	messages	will be sent	frequencies
Genotyping	genes	cause cancer	gene tests
Computer forensics	files	will be modified	bits of storage
Property-preserving hash	properties	appears in a file	bits per file
Image compression	wavelet coefficients	are nonzero	bits per digit
Traitor tracing	users	resell keys	keys
Heavy hitter / DoS	users	are spamming	virtual servers

Proposal: Prove the following.

Proposal: Prove the following.

Lemma [Your Name Here]

If we can solve (n, k)-RAC
(RAC = random access channel),

then we can solve $(3n, 2k)$-RAC.

Game: Elden Ring

DNA Coding

Substitution

Insertion

Deletion

Transposition

Amplification

Permutation

DNA Coding

ATTCCG

ATTACG

Substitution

Insertion

Deletion

Transposition

Amplification

Permutation

DNA Coding

ATT CCG

G

Substitution

Insertion

Deletion

Transposition

Amplification

Permutation

DNA Coding

Substitution

Insertion

Deletion

Transposition

Amplification

Permutation

DNA Coding

ATTCCG

ATCTCG

- Substitution
- Insertion
- Deletion
- Transposition
- Amplification
- Permutation

DNA Coding

Substitution

Insertion

Deletion

Transposition

Amplification

Permutation

DNA Coding

Substitution

Insertion

Deletion

Transposition

Amplification

Permutation

List of Publications

[1] polar codes' ρ: symbolic calculus, interval arithmetics

[2] polar codes' improved ρ: symbolic experiments, compactness argument

[3] polar's optimal ρ: large deviation toolbox, analytic continuation

[4] polar codes' complexity: optional stopping time theorem

[5] ordering bit-channels: Picard–Lindelöf, IVP version of Green's

[6] ordering bit-channels: real-root counting, Galois theory

[7] ordering bit-channels: fundamental theorem of algebra

[8] group testing: generating function, Möbius transformation

[9] group testing: tropical arithmetic, count–min sketch

[10] cloud storage: chain complex, tensor and alternating algebras

[11] cloud storage: symmetric algebra, polynomial identity test

[12] cloud matrix multiplication (MM): fast MM + distributed MM

Hsin-Po Wang, Ting-Chun Lin, Alexander Vardy, and Ryan Gabrys.
Sub-4.7 scaling exponent of polar codes.
IEEE Transactions on Information Theory, 69(7):4235–4254, July 2023.

Iwan Duursma, Ryan Gabrys, Venkatesan Guruswami, Ting-Chun Lin, and Hsin-Po Wang.
Accelerating Polarization via Alphabet Extension.
In Amit Chakrabarti and Chaitanya Swamy, editors, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022), volume 245 of Leibniz International Proceedings in Informatics (LIPIcs), pages 17:1–17:15, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Hsin-Po Wang and Iwan M. Duursma.
Polar codes' simplicity, random codes' durability.
IEEE Transactions on Information Theory, 67(3):1478–1508, 2021.

Hsin-Po Wang and Iwan M. Duursma.

Log-logarithmic time pruned polar coding.
IEEE Transactions on Information Theory, 67(3):1509–1521, March 2021.

Ting-Chun Lin and Hsin-Po Wang.
Optimal self-dual inequalities to order polarized becs.
In 2023 IEEE International Symposium on Information Theory (ISIT), pages 1550–1555, June 2023.

Hsin-Po Wang and Chi-Wei Chin.
Density devolution for ordering synthetic channels.
In 2023 IEEE International Symposium on Information Theory (ISIT), pages 1544–1549, June 2023.

Hsin-Po Wang and Vlad-Florin Drăgoi.
Fast methods for ranking synthetic becs.
In 2023 IEEE International Symposium on Information Theory (ISIT), pages 1562–1567, June 2023.

Hsin-Po Wang, Ryan Gabrys, and Venkatesan Guruswami.

Quickly-decodable group testing with fewer tests: Price–scarlett's nonadaptive splitting with explicit scalars.
In 2023 IEEE International Symposium on Information Theory (ISIT), pages 1609–1614, June 2023.

Hsin-Po Wang, Ryan Gabrys, and Alexander Vardy.
Tropical group testing.
IEEE Transactions on Information Theory, 69(9):6098–6120, Sep. 2023.

Iwan Duursma, Xiao Li, and Hsin-Po Wang.
Multilinear algebra for distributed storage.
SIAM Journal on Applied Algebra and Geometry, 5(3):552–587, 2021.

Iwan Duursma and Hsin-Po Wang.
Multilinear algebra for minimum storage regenerating codes: a generalization of the product-matrix construction.
Applicable Algebra in Engineering, Communication and Computing, 34(4):717–743, 2023.

Hsin-Po Wang and Iwan Duursma.
Parity-checked strassen algorithm, 2022.