On Asymptotic Behavior of Sums of Random Variables, Codes, and Structured Codes

Hsin-Po Wang (EECS, UC Berkeley)

02022-12-09

Part I

A Tale

One day, some alien visits the Earth.

Alien: (very politely) We want to invite you carbon-based life to join Unite Galaxies of Andromeda. But before that happens, we have some questions so we understand each other more.

Alien: So you call this $A = \lambda mnfx.mf(nfx)$ addition, right? And you call this $M = \lambda mnfx.m(nf)x$ multiplication. How fast can you multiply two numbers?

Human: We use a positional numeral system with base ● ● ● ● ● ● ● ● ● ● and we can multiply two integers, each with n digits, in $O(n \log n)$ operations.

One day, some alien visits the Earth.

Alien: (very politely) We want to invite you carbon-based life to join Unite Galaxies of Andromeda. But before that happens, we have some questions so we understand each other more.

Alien: So you call this $A = \lambda mnfx.mf(nfx)$ addition, right? And you call this $M = \lambda mnfx.m(nf)x$ multiplication. How fast can you multiply two numbers?

Human: We use a positional numeral system with base ● ● ● ● ● ● ● ● ● ● and we can multiply two integers, each with n digits, in $O(n \log n)$ operations.

Alien: Not bad. Not bad. The optimal complexity is in fact $n \log n \div \log \log n \times \log \log \log n \div \cdots$, you see what I mean? But that one is impractical. Let's call it a day.

One day, some alien visits the Earth.

Alien: (very politely) We want to invite you carbon-based life to join Unite Galaxies of Andromeda. But before that happens, we have some questions so we understand each other more.

Alien: So you call this $A = \lambda mnfx.mf(nfx)$ addition, right? And you call this $M = \lambda mnfx.m(nf)x$ multiplication. How fast can you multiply two numbers?

Human: We use a positional numeral system with base ● ● ● ● ● ● ● ● ● ● and we can multiply two integers, each with n digits, in $O(n \log n)$ operations.

Alien: Not bad. Not bad. The optimal complexity is in fact $n \log n \div \log \log n \times \log \log \log n \div \cdots$, you see what I mean? But that one is impractical. Let's call it a day.

One day, some alien visits the Earth.

Alien: (very politely) We want to invite you carbon-based life to join Unite Galaxies of Andromeda. But before that happens, we have some questions so we understand each other more.

Alien: So you call this $A = \lambda mnfx.mf(nfx)$ addition, right? And you call this $M = \lambda mnfx.m(nf)x$ multiplication. How fast can you multiply two numbers?

Human: We use a positional numeral system with base ●●●●●●●●●● and we can multiply two integers, each with n digits, in $O(n \log n)$ operations.

Alien: Not bad. Not bad. The optimal complexity is in fact $n \log n \div \log \log n \times \log \log \log n \div \cdots$, you see what I mean? But that one is impractical. Let's call it a day.

One day, some alien visits the Earth.

Alien: (very politely) We want to invite you carbon-based life to join Unite Galaxies of Andromeda. But before that happens, we have some questions so we understand each other more.

Alien: So you call this $A = \lambda mnfx.mf(nfx)$ addition, right? And you call this $M = \lambda mnfx.m(nf)x$ multiplication. How fast can you multiply two numbers?

Human: We use a positional numeral system with base ●●●●●●●●●● and we can multiply two integers, each with n digits, in $O(n \log n)$ operations.

Alien: Not bad. Not bad. The optimal complexity is in fact $n \log n \div \log \log n \times \log \log \log n \div \cdots$, you see what I mean? But that one is impractical. Let's call it a day.

Alien: Good morning our beloved *Homo sapiens*. Let's continue where we left off yesterday.

Alien: So you know you can arrange numbers in a square array (using drones to show a square array) and you can define an associative but not commutative operation on those arrays, right?

Human: It must be matrix multiplication you are talking about. Glad you ask because we have done so much about that. In fact, we even use matrix multiplication to build a machine that teaches us how to do matrix multiplication faster. Anyway, we are currently under $\omega < 2.4$ but struggling with breaking the 2.3-barrier.

Alien: Very impressive. We are not allowed to say too much. But group theory is the way to go. Goodnight.

Alien: Good morning our beloved *Homo sapiens*. Let's continue where we left off yesterday.

Alien: So you know you can arrange numbers in a square array (using drones to show a square array) and you can define an associative but not commutative operation on those arrays, right?

Human: It must be matrix multiplication you are talking about. Glad you ask because we have done so much about that. In fact, we even use matrix multiplication to build a machine that teaches us how to do matrix multiplication faster. Anyway, we are currently under $\omega < 2.4$ but struggling with breaking the 2.3-barrier.

Alien: Very impressive. We are not allowed to say too much. But group theory is the way to go. Goodnight.

Alien: Good morning our beloved *Homo sapiens*. Let's continue where we left off yesterday.

Alien: So you know you can arrange numbers in a square array (using drones to show a square array) and you can define an associative but not commutative operation on those arrays, right?

Human: It must be matrix multiplication you are talking about. Glad you ask because we have done so much about that. In fact, we even use matrix multiplication to build a machine that teaches us how to do matrix multiplication faster. Anyway, we are currently under $\omega < 2.4$ but struggling with breaking the 2.3-barrier.

Alien: Very impressive. We are not allowed to say too much. But group theory is the way to go. Goodnight.

Alien: Good morning our beloved *Homo sapiens*. Let's continue where we left off yesterday.

Alien: So you know you can arrange numbers in a square array (using drones to show a square array) and you can define an associative but not commutative operation on those arrays, right?

Human: It must be matrix multiplication you are talking about. Glad you ask because we have done so much about that. In fact, we even use matrix multiplication to build a machine that teaches us how to do matrix multiplication faster. Anyway, we are currently under $\omega < 2.4$ but struggling with breaking the 2.3-barrier.

Alien: Very impressive. We are not allowed to say too much. But group theory is the way to go. Goodnight.

Alien: What a beautiful day, huh? We are almost done with the standard interview checklist.

Alien: This is the last question. You know that joining the United Galaxies means we need to communicate and collaborate a lot, right? How are your communication skills?

Human: Well, well, well, we know there is this thing called shannon limit and we can be quite close, only 0.0045dB ... (signal interrupted for 30 minutes)

Human: In fact, we know a way to reach shannon limit in $O(n \log n)$ operations. We call it polar code.

Alien: What a beautiful day, huh? We are almost done with the standard interview checklist.

Alien: This is the last question. You know that joining the United Galaxies means we need to communicate and collaborate a lot, right? How are your communication skills?

Human: Well, well, well, we know there is this thing called shannon limit and we can be quite close, only 0.0045dB ... (signal interrupted for 30 minutes)

Human: In fact, we know a way to reach shannon limit in $O(n \log n)$ operations. We call it polar code.

Alien: What a beautiful day, huh? We are almost done with the standard interview checklist.

Alien: This is the last question. You know that joining the United Galaxies means we need to communicate and collaborate a lot, right? How are your communication skills?

Human: Well, well, well, we know there is this thing called shannon limit and we can be quite close, only 0.0045dB ... (signal interrupted for 30 minutes)

Human: In fact, we know a way to reach shannon limit in $O(n \log n)$ operations. We call it polar code.

Alien: What a beautiful day, huh? We are almost done with the standard interview checklist.

Alien: This is the last question. You know that joining the United Galaxies means we need to communicate and collaborate a lot, right? How are your communication skills?

Human: Well, well, well, we know there is this thing called shannon limit and we can be quite close, only 0.0045dB ... (signal interrupted for 30 minutes)

Human: In fact, we know a way to reach shannon limit in $O(n \log n)$ operations. We call it polar code.

Alien: Outstanding. But let me remind you, one in a billion and one in a trillion make a huge difference when it comes to fault tolerance. Do you have any idea how fast polar code achieves shannon limit?

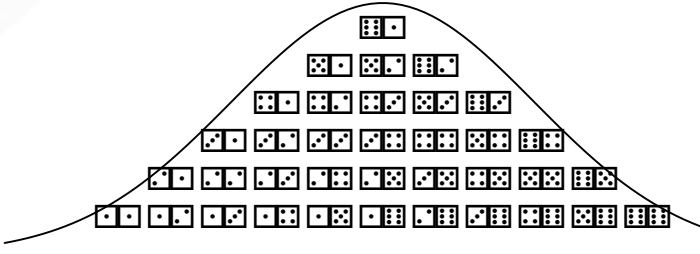
The following are the slides that were presented in the Talk to the Alien Committee (TACo) *before human answered the last question.*

Alien: Outstanding. But let me remind you, one in a billion and one in a trillion make a huge difference when it comes to fault tolerance. Do you have any idea how fast polar code achieves shannon limit?

The following are the slides that were presented in the Talk to the Alien Committee (TACo) *before human answered the last question.*

Part II

A 3x3 Array of Theories

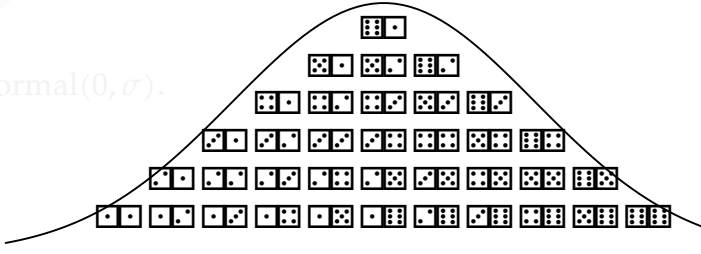


Probability theory review.
Let $S_n := X_1 + X_2 + \cdots + X_n$. (Mean zero.)

Law of large numbers: $\frac{S_n}{n} \approx 0$.

Large deviation theory: $\frac{S_n}{n} > \delta$ exponentially rare: $\approx \exp(-I(\delta)n)$.

Central limit theorem: $\frac{S_n}{\sqrt{n}} \sim \mathrm{Normal}(0, \sigma)$.

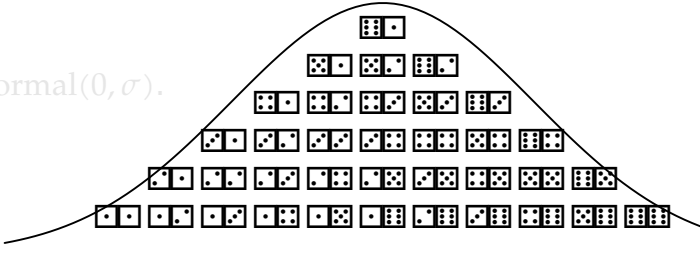


Probability theory review.
Let $S_n := X_1 + X_2 + \cdots + X_n$. (Mean zero.)

Law of large numbers: $\frac{S_n}{n} \approx 0$.

Large deviation theory: $\frac{S_n}{n} > \delta$ exponentially rare: $\approx \exp(-I(\delta)n)$.

Central limit theorem: $\frac{S_n}{\sqrt{n}} \sim \text{Normal}(0, \sigma)$.

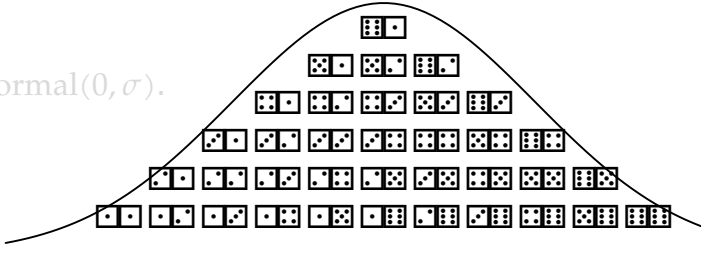


Probability theory review.
Let $S_n := X_1 + X_2 + \cdots + X_n$. (Mean zero.)

Law of large numbers: $\frac{S_n}{n} \approx 0$.

Large deviation theory: $\frac{S_n}{n} > \delta$ exponentially rare: $\approx \exp(-I(\delta)n)$.

Central limit theorem: $\frac{S_n}{\sqrt{n}} \sim \text{Normal}(0, \sigma)$.

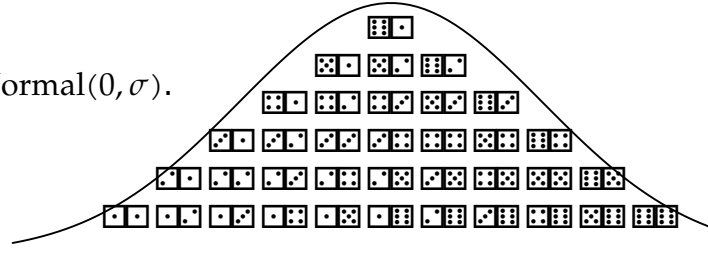


Probability theory review.
Let $S_n := X_1 + X_2 + \cdots + X_n$. (Mean zero.)

Law of large numbers: $\frac{S_n}{n} \approx 0$.

Large deviation theory: $\frac{S_n}{n} > \delta$ exponentially rare: $\approx \exp(-I(\delta)n)$.

Central limit theorem: $\frac{S_n}{\sqrt{n}} \sim \text{Normal}(0, \sigma)$.



Coding theory.

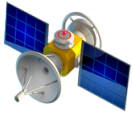
Information density: $i(x,y) := \log \dfrac{P_{XY}(x,y)}{P_X(y)P_Y(y)}$.

As a random variable: $i(X,Y)$.

Law of large numbers $\Leftrightarrow$ Shannon's noisy channel coding theory.

Large deviations $\Leftrightarrow$ Gallager's error exponent: decode error $\approx \exp(-\mathrm{E_r}\, n)$.

Central limit theorem $\Leftrightarrow$ Polyanskiy–Poor–Verdu's finite blocklength characterization: gap to capacity $\approx 1/\sqrt{n}$.



Coding theory.

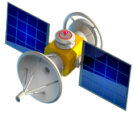
Information density: $i(x,y) := \log \dfrac{P_{XY}(x,y)}{P_X(y)P_Y(y)}$.

As a random variable: $i(X,Y)$.

Law of large numbers $\Leftrightarrow$ Shannon's noisy channel coding theory.

Large deviations $\Leftrightarrow$ Gallager's error exponent: decode error $\approx \exp(-\mathrm{E_r}\, n)$.

Central limit theorem $\Leftrightarrow$ Polyanskiy–Poor–Verdu's finite blocklength characterization: gap to capacity $\approx 1/\sqrt{n}$.



Coding theory.

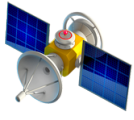
Information density: $i(x,y) := \log \frac{P_{XY}(x,y)}{P_X(y)P_Y(y)}$.

As a random variable: $i(X,Y)$.

Law of large numbers $\Leftrightarrow$ Shannon's noisy channel coding theory.

Large deviations $\Leftrightarrow$ Gallager's error exponent: decode error $\approx \exp(-\mathrm{E_r}\, n)$.

Central limit theorem $\Leftrightarrow$ Polyanskiy–Poor–Verdu's finite blocklength characterization: gap to capacity $\approx 1/\sqrt{n}$.



Coding theory.

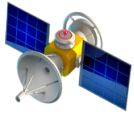
Information density: $i(x,y) := \log \dfrac{P_{XY}(x,y)}{P_X(y)P_Y(y)}$.

As a random variable: $i(X,Y)$.

Law of large numbers $\Leftrightarrow$ Shannon's noisy channel coding theory.

Large deviations $\Leftrightarrow$ Gallager's error exponent: decode error $\approx \exp(-\mathrm{E_r}\, n)$.

Central limit theorem $\Leftrightarrow$ Polyanskiy–Poor–Verdu's finite blocklength characterization: gap to capacity $\approx 1/\sqrt{n}$.



probability theory	theory of random codes
law of large numbers	Shannon’s noisy channel coding thm
large deviations	Gallager’s error exponent
central limit theorem	PPV’s finite blocklength characteriz’n

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

Polar code.

Shannon’s channel coding thm ⇔ 2009 Arıkan’s channel polarization.

Gallager’s error exponent ⇔ 2009 Arıkan–Telatar’s rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV’s finite blocklength ⇔ 2010’s Urbanke’s scaling exponent
(gap to capacity $n^{-1/4}$).

Polar code.

Shannon’s channel coding thm ⇔ 2009 Arıkan’s channel polarization.

Gallager’s error exponent ⇔ 2009 Arıkan–Telatar’s rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV’s finite blocklength ⇔ 2010’s Urbanke’s scaling exponent
(gap to capacity $n^{-1/4}$).

Polar code.

Shannon’s channel coding thm ⇔ 2009 Arıkan’s channel polarization.

Gallager’s error exponent ⇔ 2009 Arıkan–Telatar’s rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV’s finite blocklength ⇔ 2010’s Urbanke’s scaling exponent
(gap to capacity $n^{-1/4}$).

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

History of scaling over BMSC: 0 $1/2$

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

History of scaling over BMSC: 0 $1/2$
2015 Guruswami–Xia $\longleftrightarrow$

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

History of scaling over BMSC: 0 $1/2$

2015 Guruswami–Xia ⟷

2012 Goli–Hassani–Urbanke ⟷

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

History of scaling over BMSC: 0 $1/2$

2015 Guruswami–Xia

2012 Goli–Hassani–Urbanke

2014 Hassani–Alishahi–Urbanke

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

History of scaling over BMSC: 0 $1/2$

2015 Guruswami–Xia

2012 Goli–Hassani–Urbanke

2014 Hassani–Alishahi–Urbanke

2014 Goldin–Burshtein

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

History of scaling over BMSC: 0 $1/2$

2015 Guruswami–Xia ⟻——————⟼

2012 Goli–Hassani–Urbanke ⟻————⟼

2014 Hassani–Alishahi–Urbanke ⟻——⟼

2014 Goldin–Burshtein ⟻——⟼

2016 Mondelli–Hassani–Urbanke ⟻——⟼

Polar code.

Shannon's channel coding thm ⇔ 2009 Arıkan's channel polarization.

Gallager's error exponent ⇔ 2009 Arıkan–Telatar's rate of polarization
(decode error $\approx \exp(-\sqrt{n})$).

PPV's finite blocklength ⇔ 2010's Urbanke's scaling exponent
(gap to capacity $n^{-1/4}$).

History of scaling over BMSC: 0 $1/2$

2015 Guruswami–Xia ⟷

2012 Goli–Hassani–Urbanke ⟷

2014 Hassani–Alishahi–Urbanke ⟷

2014 Goldin–Burshtein ⟷

2016 Mondelli–Hassani–Urbanke ⟷

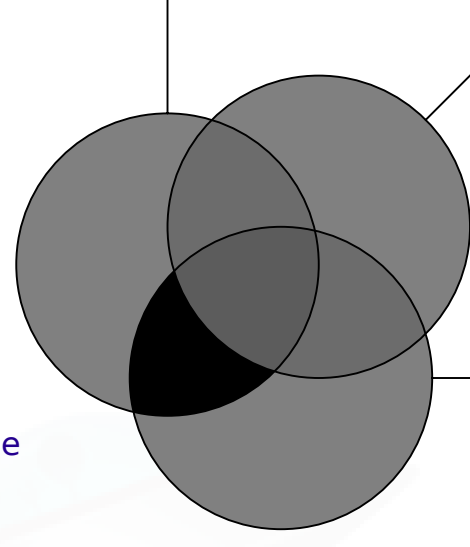
2022 W.–Lin–Vardy–Gabrys ⟷

probability theory	random code	polar code
law of large numbers	channel capacity	channel polarization
large deviations	error exponent	rate of convergence
central limit theorem	finite blocklength	scaling exponent

probability theory	random code	polar code
law of large numbers	channel capacity	channel polarization
large deviations	$\exp(-I(\delta)n)$	$\exp(-\sqrt{n})$
central limit theorem	$1/\sqrt{n}$	$n^{-1/4}$

Part III

Improve Polar Code



Improve polar for large deviations /
error exponent / rate of polarization.

2010 Korada–Sasoglu–Urbanke:
decode error $\approx \exp(-n^{\beta})$ with any matrix.

2014 Mori–Tanaka: same error $\exp(-n^{\beta})$
but for any finite field.

⇒ Reed–Solomon kernel (Vandermonde matrix)
⇒ best distance profile.

Improve polar for large deviations /
error exponent / rate of polarization.

2010 Korada–Sasoglu–Urbanke:
decode error $\approx \exp(-n^{\beta})$ with any matrix.

2014 Mori–Tanaka: same error $\exp(-n^{\beta})$
but for any finite field.

⇒ Reed–Solomon kernel (Vandermonde matrix)
⇒ best distance profile.

Improve polar for large deviations /
error exponent / rate of polarization.

2010 Korada–Sasoglu–Urbanke:
decode error $\approx \exp(-n^{\beta})$ with any matrix.

2014 Mori–Tanaka: same error $\exp(-n^{\beta})$
but for any finite field.

⇒ Reed–Solomon kernel (Vandermonde matrix)
⇒ best distance profile.

Improve polar for large deviations /
error exponent / rate of polarization.

2010 Korada–Sasoglu–Urbanke:
decode error $\approx \exp(-n^{\beta})$ with any matrix.

2014 Mori–Tanaka: same error $\exp(-n^{\beta})$
but for any finite field.

⇒ Reed–Solomon kernel (Vandermonde matrix)
⇒ best distance profile.

probability theory	random code	polar code
law of large numbers	channel capacity	channel polarization
large deviations	$\exp(-I(\delta)n)$	$\exp(-n^{1-\varepsilon})$
central limit theorem	$1/\sqrt{n}$	$n^{-1/4}$

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

2010 Hassani–Alishahi–Urbanke 2×2 χ

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

2010 Hassani–Alishahi–Urbanke 2×2

2010 Korada–Montanari–Telatar–Urbanke 2×2

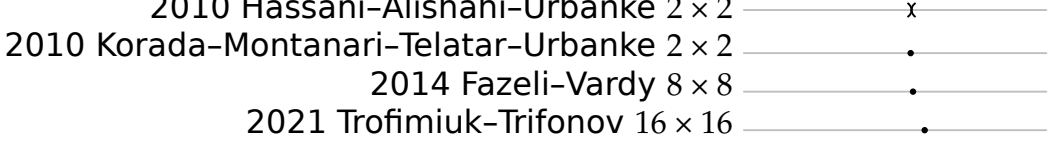
Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

2010 Hassani–Alishahi–Urbanke 2×2
2010 Korada–Montanari–Telatar–Urbanke 2×2
2014 Fazeli–Vardy 8×8

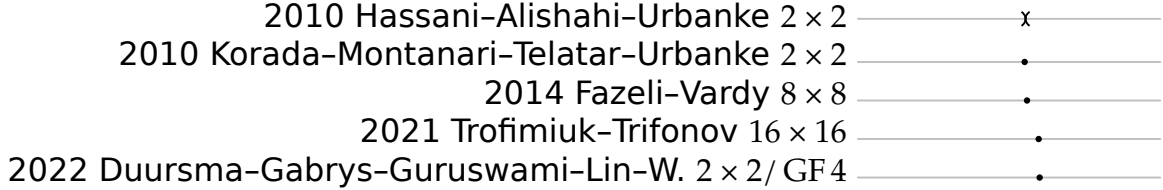
Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.



Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

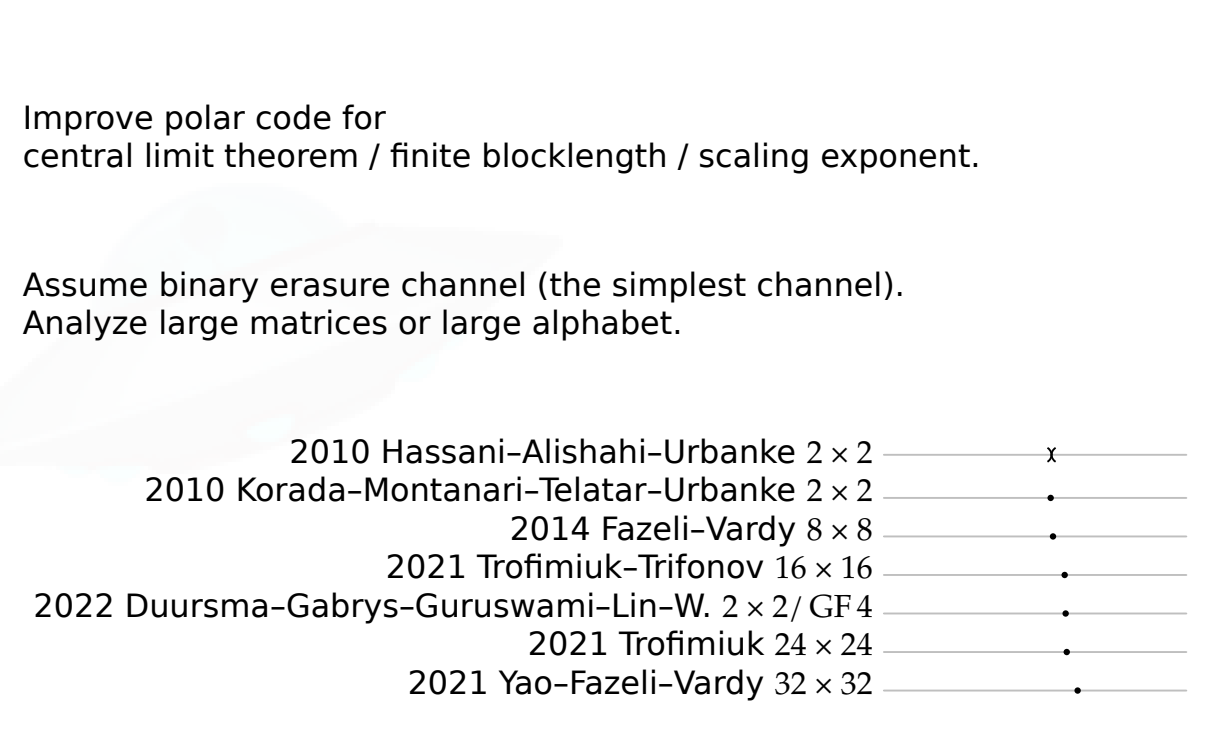


Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.



Improve polar code for
central limit theorem / finite blocklength / scaling exponent.

Assume binary erasure channel (the simplest channel).
Analyze large matrices or large alphabet.

probability theory	random code	polar code
law of large numbers	channel capacity	channel polarization
large deviations	$\exp(-I(\delta)n)$	$\exp(-n^{1-\varepsilon})$
central limit theorem	$1/\sqrt{n}$	$n^{-1/3}$

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.
But this time, aim for the optimal gap to capacity $n^{-1/2+\varepsilon}$.

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 Wu–Duursma: GF.

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.
But this time, aim for the optimal gap to capacity $n^{-1/2+\varepsilon}$.

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 W.–Duursma: discrete memoryless channel.

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.
But this time, aim for the optimal gap to capacity $n^{-1/2+\varepsilon}$.

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 W.–Duursma: discrete memoryless channel.

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.
But this time, aim for the optimal gap to capacity $n^{-1/2+\varepsilon}$.

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 W.–Duursma: discrete memoryless channel.

Improve polar code for
central limit theorem / finite blocklength / scaling exponent.
But this time, aim for the optimal gap to capacity $n^{-1/2+\varepsilon}$.

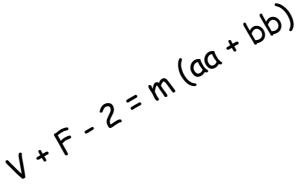
2019 Pfister–Urbanke: q-ary erasure channel, $q \to \infty$.

2021 Fazeli–Hassani–Mondelli–Vardy: binary erasure channel.

2022 Guruswami–Riazanov–Ye: binary symmetric memoryless channel.

2021 W.–Duursma: discrete memoryless channel.

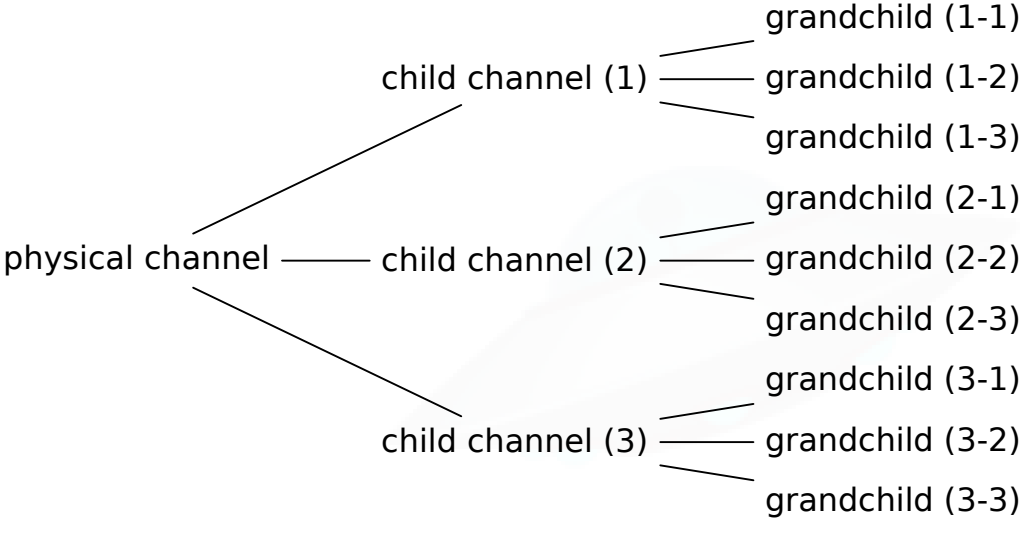
probability theory	random code	polar code
law of large numbers	channel capacity	channel polarization
large deviations	$\exp(-I(\delta)n)$	$\exp(-n^{1-\varepsilon})$
central limit theorem	$1/\sqrt{n}$	$n^{-1/2+\varepsilon}$



Part IV

Proof Techniques

Polar code is a technique
to generate “child channels” out of “parent channel”.



To get a good polar code, the “descendant channels” must polarize.
That is, some descendants must be very good channels,
others very bad.

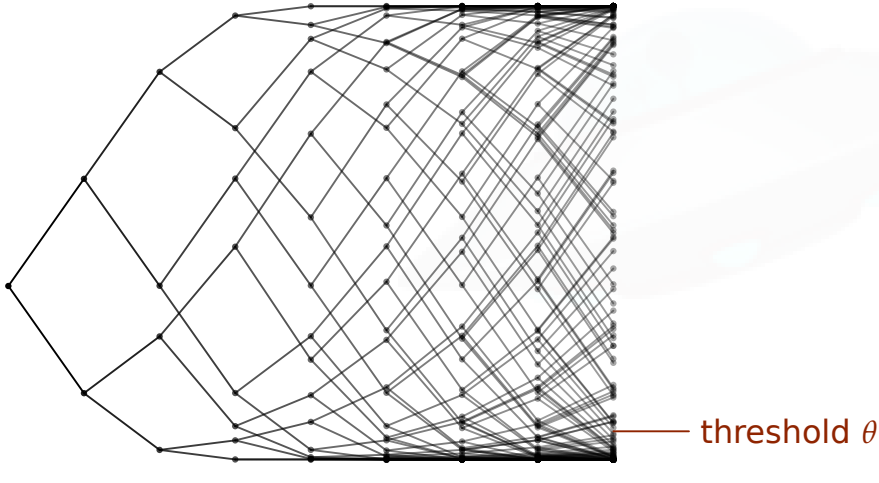
It suffices to investigate each “nuclear family” .

Ask: Are the good children good enough compared to parent?

bad bad

To get a good polar code, the “descendant channels” must polarize.
That is, some descendants must be very good channels,
others very bad.

It suffices to investigate each “nuclear family”  .
Ask: Are the good children good enough compared to parent?
bad bad


threshold θ

Now, optimal scaling law $n^{1/2+\varepsilon}$, how to?
Warning: over-simplified but still highly technical.

Let G be a random invertible $\ell \times \ell$ matrix.
With high probability, the lower half of G is a good linear code.
Use Gallager's error exponent argument: decode error $\approx \exp(-\mathrm{E_r}\,\ell)$.
$\mathrm{E_r}$ is about $(\text{gap to capacity})^2$ — Cramér function is locally parabola.
The square is the same square as in finite blocklength's $1/\sqrt{n}$.

The other half of the story:
With high probability, the upper half of G is good against wiretapping.
Use Hayashi's secrecy exponent argument: info leaked $\approx \exp(-\mathrm{E_r}\,\ell)$.

Now, optimal scaling law $n^{1/2+\varepsilon}$, how to?
Warning: over-simplified but still highly technical.

Let G be a random invertible $\ell \times \ell$ matrix.
With high probability, the lower half of G is a good linear code.
Use Gallager's error exponent argument: decode error $\approx \exp(-\mathrm{E_r}\,\ell)$.
$\mathrm{E_r}$ is about $(\text{gap to capacity})^2$ — Cramér function is locally parabola.
The square is the same square as in finite blocklength's $1/\sqrt{n}$.

The other half of the story:
With high probability, the upper half of G is good against wiretapping.
Use Hayashi's secrecy exponent argument: info leaked $\approx \exp(-\mathrm{E_r}\,\ell)$.

Now, optimal scaling law $n^{1/2+\varepsilon}$, how to?
Warning: over-simplified but still highly technical.

Let G be a random invertible $\ell \times \ell$ matrix.
With high probability, the lower half of G is a good linear code.
Use Gallager's error exponent argument: decode error $\approx \exp(-\mathrm{E_r}\,\ell)$.
$\mathrm{E_r}$ is about $(\text{gap to capacity})^2$ — Cramér function is locally parabola.
The square is the same square as in finite blocklength's $1/\sqrt{n}$.

The other half of the story:
With high probability, the upper half of G is good against wiretapping.
Use Hayashi's secrecy exponent argument: info leaked $\approx \exp(-\mathrm{E_r}\,\ell)$.

Now, optimal scaling law $n^{1/2+\varepsilon}$, how to?
Warning: over-simplified but still highly technical.

Let G be a random invertible $\ell \times \ell$ matrix.
With high probability, the lower half of G is a good linear code.
Use Gallager's error exponent argument: decode error $\approx \exp(-\mathrm{E_r}\,\ell)$.
$\mathrm{E_r}$ is about $(\text{gap to capacity})^2$ — Cramér function is locally parabola.
The square is the same square as in finite blocklength's $1/\sqrt{n}$.

The other half of the story:
With high probability, the upper half of G is good against wiretapping.
Use Hayashi's secrecy exponent argument: info leaked $\approx \exp(-\mathrm{E_r}\,\ell)$.

Funny 1: we use error exponent bounds to attack finite blocklength.

Now, optimal scaling law $n^{1/2+\varepsilon}$, how to?
Warning: over-simplified but still highly technical.

Let G be a random invertible $\ell \times \ell$ matrix.
With high probability, the lower half of G is a good linear code.
Use Gallager's error exponent argument: decode error $\approx \exp(-\mathrm{E_r}\,\ell)$.
$\mathrm{E_r}$ is about (gap to capacity)^2 — Cramér function is locally parabola.
The square is the same square as in finite blocklength's $1/\sqrt{n}$.

The other half of the story:
With high probability, the upper half of G is good against wiretapping.
Use Hayashi's secrecy exponent argument: info leaked $\approx \exp(-\mathrm{E_r}\,\ell)$.

Funny 1: we use error exponent bounds to attack finite blocklength.

Funny 2: we use wiretapping results to attack noisy channel coding.

Part V

The Trade-off Row

probability theory	random code	polar code
law of large numbers	channel capacity	channel polarization
large deviations	$\exp(-I(\delta)n)$	$\exp(-n^{1-\varepsilon})$
central limit theorem	$1/\sqrt{n}$	$n^{-1/2+\varepsilon}$
moderate deviations	???	???

In probability theory: $\frac{S_n}{n} \approx n^{-(1-\beta)/2}$ with probability $\approx \exp(-n^\beta)$.

For random code: gap to capacity $\approx n^{-(1-\beta)/2}$; decode error $\approx \exp(-n^\beta)$.
2014 Altug–Wagner. 2010 Polyanskiy–Verdu.

For polar code: gap to capacity $\approx n^{-(1-\beta)/2+\varepsilon}$; decode error $\approx \exp(-n^{\beta-\varepsilon})$.
2021 W.–Duursma.

In probability theory: $\frac{S_n}{n} \approx n^{-(1-\beta)/2}$ with probability $\approx \exp(-n^\beta)$.

For random code: gap to capacity $\approx n^{-(1-\beta)/2}$; decode error $\approx \exp(-n^\beta)$.
2014 Altug–Wagner. 2010 Polyanskiy–Verdu.

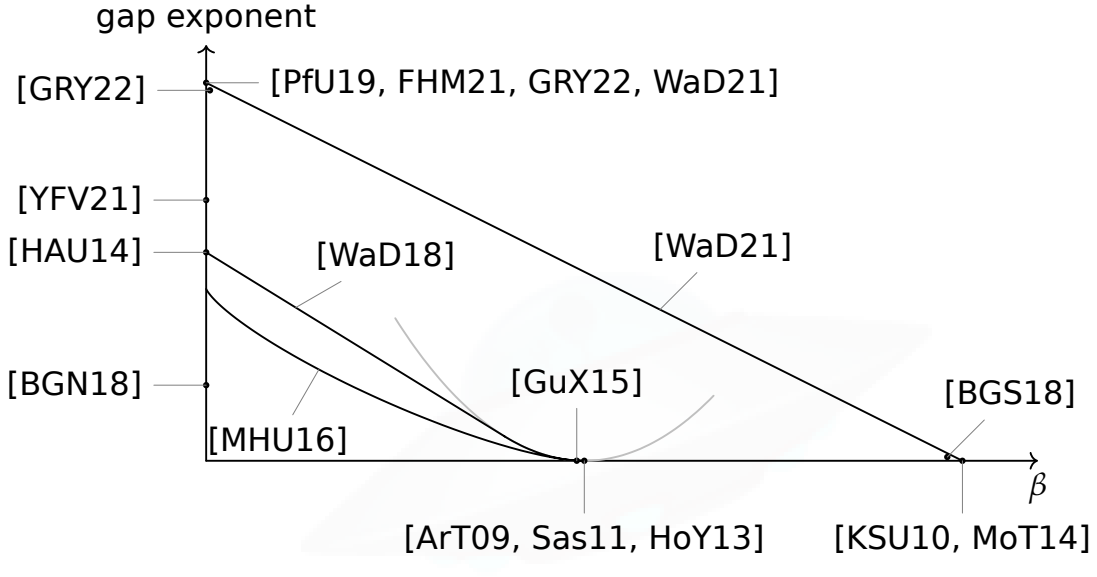
For polar code: gap to capacity $\approx n^{-(1-\beta)/2+\varepsilon}$; decode error $\approx \exp(-n^{\beta-\varepsilon})$.
2021 W.–Duursma.

In probability theory: $\frac{S_n}{n} \approx n^{-(1-\beta)/2}$ with probability $\approx \exp(-n^\beta)$.

For random code: gap to capacity $\approx n^{-(1-\beta)/2}$; decode error $\approx \exp(-n^\beta)$.
2014 Altug–Wagner. 2010 Polyanskiy–Verdu.

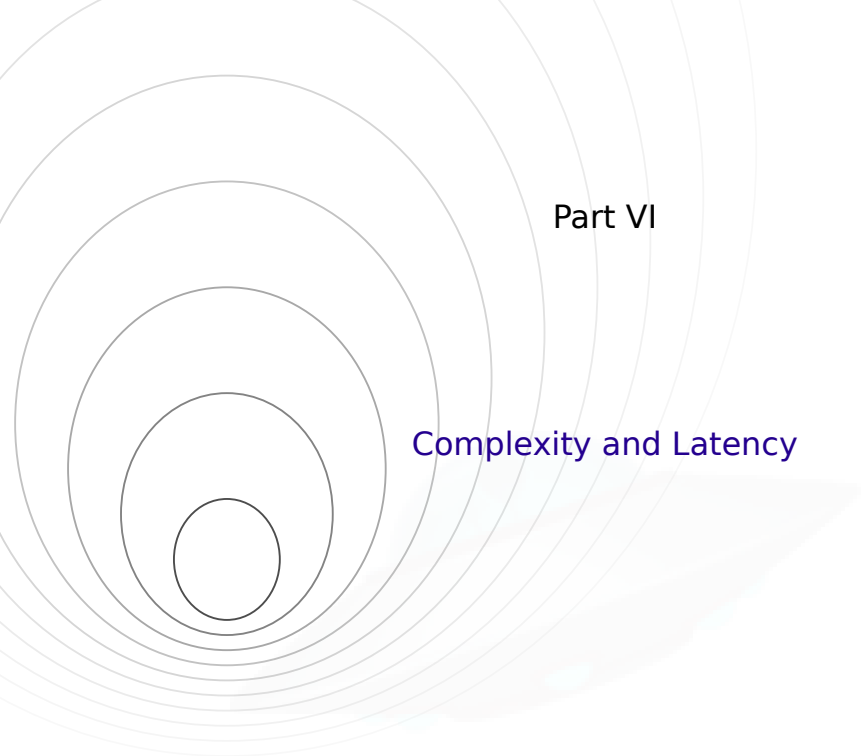
For polar code: gap to capacity $\approx n^{-(1-\beta)/2+\varepsilon}$; decode error $\approx \exp(-n^{\beta-\varepsilon})$.
2021 W.–Duursma.

probability theory	random code	polar code
law of large numbers	channel capacity	channel polarization
large deviations	$\exp(-I(\delta)n)$	$\exp(-n^{1-\varepsilon})$
central limit theorem	$1/\sqrt{n}$	$n^{-1/2+\varepsilon}$
moderate deviations	trade-off	trade-off up to ε


gap exponent
[GRY22]
[PfU19, FHM21, GRY22, WaD21]
[YFV21]
[HAU14]
[WaD18]
[WaD21]
[BGN18]
[GuX15]
[BGS18]
[MHU16]
β
[ArT09, Sas11, HoY13]
[KSU10, MoT14]

	Symmetric					Asymmetric	
	BEC	BMSC	p-ary	q-ary	finite	binary	finite
LLN	Ari09	Ari09	STA09	STA09	STA09	SRD12	WaD21
LD	ArT09	ArT09	STA09	MoT14	Sas11	HoY13	WaD21
CLT	KMT10	HAU14	BGN18	WaD21	WaD21	WaD21	WaD21
MD	GrX15	GrX15	BGS18	WaD21	WaD21	WaD21	WaD21
LD★	KSU10	KSU10	WaD21	WaD21	WaD21	WaD21	WaD21
CLT★	FHM18	GRY20	WaD21	WaD21	WaD21	WaD21	WaD21
MD★	WaD21	WaD21	WaD21	WaD21	WaD21	WaD21	WaD21

Un-starred: existence of exponent; starred: optimal exponent.



Part VI

Complexity and Latency

Complexity: lower is better. But still achieve capacity.

Random code: naively, exponential in n.

Standard polar code: $O(n \log n)$ encoding and decoding per block.

Binary erasure channel: generalizations of repeat–accumulate code can do linear encoder and decoder, i.e., $O(n)$ per block.
2005 Pfister–Sason–Urbanke. 2007 Pfister–Sason.

Complexity: lower is better. But still achieve capacity.

Random code: naively, exponential in n.

Standard polar code: $O(n \log n)$ encoding and decoding per block.

Binary erasure channel: generalizations of repeat–accumulate code can do linear encoder and decoder, i.e., $O(n)$ per block.
2005 Pfister–Sason–Urbanke. 2007 Pfister–Sason.

General channel?

Complexity: lower is better. But still achieve capacity.

Random code: naively, exponential in n.

Standard polar code: $O(n \log n)$ encoding and decoding per block.

Binary erasure channel: generalizations of repeat–accumulate code can do linear encoder and decoder, i.e., $O(n)$ per block.
2005 Pfister–Sason–Urbanke. 2007 Pfister–Sason.

General channel?

Complexity: lower is better. But still achieve capacity.

Random code: naively, exponential in n.

Standard polar code: $O(n \log n)$ encoding and decoding per block.

Binary erasure channel: generalizations of repeat–accumulate code can do linear encoder and decoder, i.e., $O(n)$ per block.
2005 Pfister–Sason–Urbanke. 2007 Pfister–Sason.

General channel?

Complexity: lower is better. But still achieve capacity.

Random code: naively, exponential in n.

Standard polar code: $O(n \log n)$ encoding and decoding per block.

Binary erasure channel: generalizations of repeat–accumulate code can do linear encoder and decoder, i.e., $O(n)$ per block.
2005 Pfister–Sason–Urbanke. 2007 Pfister–Sason.

General channel?

2011 Alamdar-Yazdi–Kschischang: You can prune the family tree.

2017 El-Khamy–Mahdavifar–Feygin–Lee–Kang: If you want the same performance, pruning reduces complexity by a scalar; still $O(n \log n)$.

2021 W.–Duursma: $O(n \log \log n)$ if relax the performance requirement. Same gap to capacity;

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

2011 Alamdar-Yazdi–Kschischang: You can prune the family tree.

2017 El-Khamy–Mahdavifar–Feygin–Lee–Kang: If you want the same performance, pruning reduces complexity by a scalar; still $O(n \log n)$.

2021 W.–Duursma: $O(n \log \log n)$ if relax the performance requirement. Same gap to capacity;

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

2011 Alamdar-Yazdi–Kschischang: You can prune the family tree.

2017 El-Khamy–Mahdavifar–Feygin–Lee–Kang: If you want the same performance, pruning reduces complexity by a scalar; still $O(n \log n)$.

2021 W.–Duursma: $O(n \log \log n)$ if relax the performance requirement. Same gap to capacity;

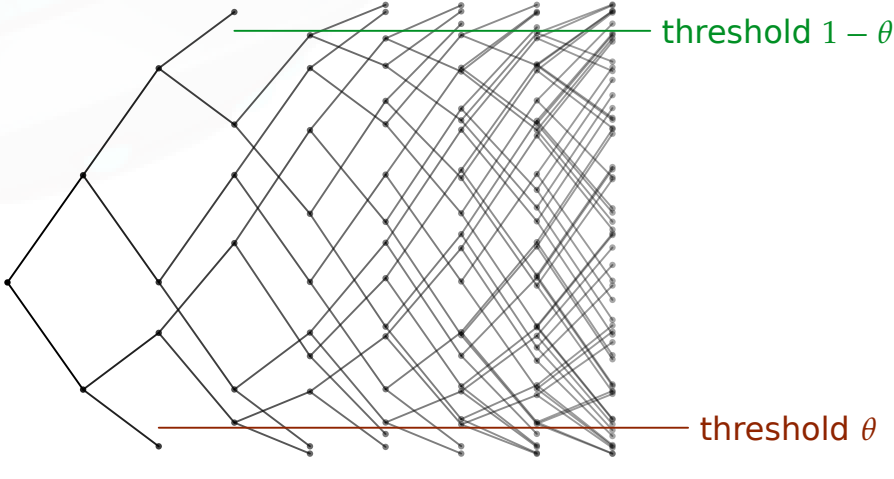
Trade-off: complexity $\approx O(n \log(-\log(\text{decode error})))$.

2011 Alamdar-Yazdi–Kschischang: You can prune the family tree.

2017 El-Khamy–Mahdavifar–Feygin–Lee–Kang: If you want the same performance, pruning reduces complexity by a scalar; still $O(n \log n)$.

2021 W.–Duursma: $O(n \log \log n)$ if relax the performance requirement. Same gap to capacity;

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


threshold $1-\theta$
threshold θ

Code	Error	Gap	Complexity	Channel
random	$\exp(-n^{\beta})$	$n^{-\alpha}$	$\exp(n)$	DMC
concatenation	$\exp(-n^{\beta})$	$\to 0$	$\mathrm{poly}(n)$	DMC
RM	$\to 0$	$\to 0$	$O(n^2)$	BEC
LDPC	$\to 0$	$\to 0$	unclear	BMSC
RA family	$\to 0$	$\to 0$	$O(n)$	BEC
MD-polar	$\exp(-n^{\beta})$	$n^{-\alpha}$	$O(n \log n)$	DMC
log-log-polar	$1/\mathrm{poly}(n)$	$n^{-\alpha}$	$O(n \log \log n)$	DMC

2021 Mondelli–Hashemi–Cioffi–Goldsmith.
2021 Hashemi–Mondelli–Fazeli–Vardy–Cioffi–Goldsmith.
Latency: number of operations that have to wait for each other.

No parallelism: $O(n \log\log n)$ latency.

Allow arbitrary parallelism: $O(n^{1-\alpha})$ latency.

Proof technique:
Full-parallel Latency: total number of nodes in the tree.
Complexity: sum of $2^{\text{total depth}-d} \times$ number of nodes at depth d.

2021 Mondelli–Hashemi–Cioffi–Goldsmith.
2021 Hashemi–Mondelli–Fazeli–Vardy–Cioffi–Goldsmith.
Latency: number of operations that have to wait for each other.

No parallelism: $O(n \log\log n)$ latency.

Allow arbitrary parallelism: $O(n^{1-\alpha})$ latency.

Proof technique:
Full-parallel Latency: total number of nodes in the tree.
Complexity: sum of $2^{\text{total depth}-d} \times$ number of nodes at depth d.

2021 Mondelli–Hashemi–Cioffi–Goldsmith.
2021 Hashemi–Mondelli–Fazeli–Vardy–Cioffi–Goldsmith.
Latency: number of operations that have to wait for each other.

No parallelism: $O(n \log\log n)$ latency.

Allow arbitrary parallelism: $O(n^{1-\alpha})$ latency.

Proof technique:
Full-parallel Latency: total number of nodes in the tree.
Complexity: sum of $2^{\text{total depth}-d} \times$ number of nodes at depth d.

2021 Mondelli–Hashemi–Cioffi–Goldsmith.
2021 Hashemi–Mondelli–Fazeli–Vardy–Cioffi–Goldsmith.
Latency: number of operations that have to wait for each other.

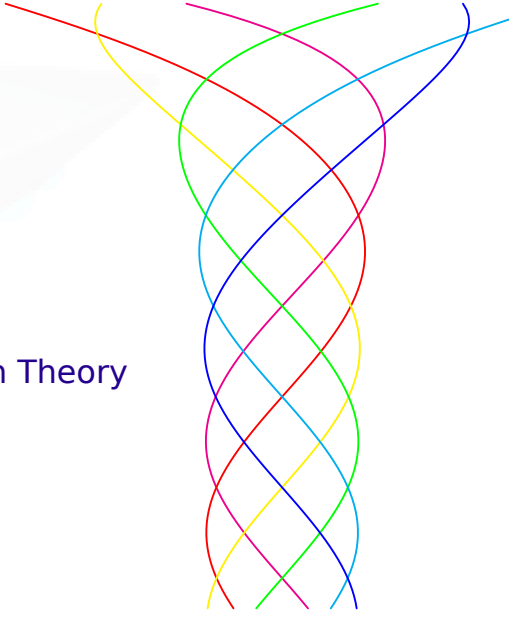
No parallelism: $O(n \log\log n)$ latency.

Allow arbitrary parallelism: $O(n^{1-\alpha})$ latency.

Proof technique:
Full-parallel Latency: total number of nodes in the tree.
Complexity: sum of $2^{\text{total depth}-d} \times$ number of nodes at depth d.

Part VII

Network Information Theory



Asymmetric channel ... check

Noisy channel coding ... check

Source coding (lossless compression) ... check

Lossy compression (rate–distortion theory) ... check

Multiple access channel ... check

Slepian–Wolf (losslessly compressing two sources) ... check

Lossless compression problem with a helper ... check

Wiretap channel (degradation) ... check

Wiretap channel (no degradation) ... LD

Broadcast channel ... LD

Hidden Markov chain input ... LD

Hidden Markov chain channel state ... LD

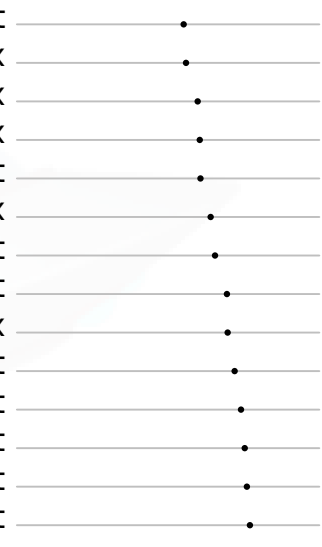
Deletion channel ... LD

Non-stationary channel ... CLT

Part VIII

Todos

(In preparation.)
Binary erasure channel.
Large matrix over binary alphabet
versus
2×2 matrix over large alphabet?



More directions — fine measurement.

True scaling exponent.

Beta expansion.

Coded computation.

More directions — bald improvement.

De-randomization.

De-dynamicity.

Epilogue

The alien is eventually satisfied with human’s answer and leaves.

Before the alien completely disappears, the last message is: The next presidential election will be held after half of Carbon-14 decays. Make your choice on a golden record and send it to the nearest black hole. The candidates are ...


Entrance exam
MIB


100