Sub-4.7 Scaling Exponent of Polar Codes

Hsin-Po Wang and Ting-Chun Lin and Alexander Vardy and Ryan Gabrys

***Abstract*—Polar code visibly approaches channel capacity in practice and is thereby a constituent code of the 5G standard. Compared to low-density parity-check code, however, the performance of short-length polar code has rooms for improvement that could hinder its adoption by a wider class of applications. As part of the program that addresses the performance issue at short length, it is crucial to understand how fast binary memoryless symmetric channels polarize. A number, called scaling exponent, was defined to measure the speed of polarization and several estimates of the scaling exponent were given in literature. As of 2022, the tightest overestimate is 4.714 made by Mondelli, Hassani, and Urbanke in 2015. We lower the overestimate to 4.63.**

I. Introduction

POLAR CODE was proved to be capacity achieving over any binary memoryless symmetric (BMS) channel [Ari09]. Polar code also shows great potential in practice and it was selected as part of the 5G standard for wireless communication. That being the case, polar coding for short block length has room for improvement when compared to low-density parity-check code, the other code in the 5G standard. Improving short-length polar code further can pave the way for applications such as Internet of Things, as some devices can only afford easily-decodable code and others must reply very promptly.

Now that improving the performance of polar code at finite block length is on the agenda, we first need to know how much we can say about the unmodified code. There are two regimes that were considered in literature. In the error exponent regime, the code rate is fixed and the asymptote of the error probability is evaluated. For polar code, it was shown that the block error probability scales as $\exp(-\sqrt{N})$, where N is the block length. For variations of polar code that use different matrices as the polarizing kernel, the asymptote of error can also be computed and is about $\exp(-N^\beta)$. Here, $\beta > 0$ is a number completely determined by the Hamming distances among the vector subspaces spanned by the rows of the kernel matrix. Long story short, predicting the behavior of error probability at a fixed code rate is straightforward. See [AT09; KŞU10; HMTU13; MT14] and those that cite them for more on this topic.

In the scaling exponent regime, the second approach that characterizes the performance of polar code, the error probability is fixed and the asymptote of the code rate is evaluated. It is observed that the *gap to capacity*, which is the difference between the channel capacity and code rate, scales as $N^{-1/\mu}$. Called the *scaling exponent*, this number μ is difficult to pinpoint exactly. Here is a list of progresses made before. It was shown in [HAU10] that $0.2786 \geqslant 1/\mu \geqslant 0.2669$ over binary erasure channels (BECs). It was shown in [KMTU10] that $\mu \approx 3.626$ over BECs. It was shown in [GHU12] that $3.553 \leqslant \mu$ over BMS channels. It was shown in [HAU14] that $3.579 \leqslant \mu \leqslant 6$ over BMS channels. It was shown in [GB14] that $\mu \leqslant 5.702$ over BMS channels. Mondelli, Hassani, and Urbanke showed in [MHU16] that $\mu \leqslant 4.714$ over BMS channels. The last record stood for seven years[1] and is the one we intend to improve upon.

TABLE I
Two key ways to synthesize channels.
SC stands for sequential cancellation decoder.

$W \boxminus W$	$W \circledast W$
serial combination	parallel combination
convolution at check node	convolution at variable node
guess $X_1 - X_2$ given Y_1, Y_2	guess X_2 given $Y_1, Y_2, X_1 - X_2$
decoded earlier in SC	decoded later in SC
named W' or W^- or $W^{(1)}$	named W'' or W^+ or $W^{(2)}$
more noisy than W	more reliable than W
still BSC if W is	not BSC if $Z(W) \neq 0, 1$
$1 - Z(W \boxminus W) \geqslant (1 - Z(W))^2$	$Z(W \circledast W) = Z(W)^2$

Scaling exponent's definition generalizes to other scenarios. To name a few: Over additive white Gaussian noise channels, $\mu \leqslant 4.714$ [FT17]. Over non-stationary BECs, $\mu \leqslant 7.34$; over non-stationary BMS channels, $\mu \leqslant 8.54$ [Mah20]. Over (hereafter stationary) BECs, permuting the rows of the Kronecker powers of Arıkan's kernel $\left[\begin{smallmatrix}1&0\\1&1\end{smallmatrix}\right]$ improves the scaling from $\mu \approx 3.627$ to $\mu \approx 3.479$ with little complexity overhead [BFS+17]. Using larger kernel matrices improves scaling exponents even further: over BECs, $\mu \approx 3.627$ for 2×2 kernel, $\mu \approx 3.577$ for 8×8 kernel [FV14], $\mu \approx 3.346$ for 16×16 kernel [TT21], $\mu \approx 3.122$ for 32×32 kernel, and $\mu \approx 2.87$ for 64×64 kernel[2] [YFV19]. (See [Tro21] for sizes between 9×9 and 31×31.) In general, any nontrivial matrix kernel over any alphabet has a finite scaling exponent over any discrete memoryless channel [Wan21, Chapter 5]. Meanwhile, dynamic kerneling is also shown, conceptually, to be improving the scaling exponent; for instance, $\mu \approx 4.938$ decreases to $\mu \approx 4.183$ for 3×3 kernels over BECs[3] [YB15]. Most challengingly, a series of works attempted to reach $\mu \approx 2$, the optimal scaling exponent, and succeeded. Pfister and Urbanke [PU19] showed that $\mu \approx 2$ can be reached using Reed–Solomon kernels over q-ary erasure channels as $q \to \infty$.

The authors are with University of California San Diego, CA, USA. Lin is also with Hon Hai (Foxconn) Research Institute, Taipei, Taiwan. This work was supported by NSF grants CCF-1764104 and CCF-2107346. Emails: {hsw001, til022, avardy, rgabrys} @ucsd.edu

[1] The preprint was first released in January 2015 at https://arxiv.org/abs/1501.02444

[2] The scaling exponent for the 64×64 kernel involves Monte Carlo method.

[3] We recalculate the exponents for dynamic kerneling using power iteration to even the baseline for comparison.

Fazeli, Hassani, Mondelli, and Vardy [FHMV21] showed that $\mu \approx 2$ can be reached using random linear kernel over BECs. Guruswami, Riazanov, and Ye [GRY22] showed that $\mu \approx 2$ can be reached using dynamic random linear kernels over BMS channels; plus the code construction is of polynomial complexity. Wang and Duursma [WD21b] showed that $\mu \approx 2$ can be reached using dynamic random linear kernels over discrete memoryless channels.

A good scaling exponent over BMS channels has several boarder impacts. One: One can now describe the trade-off between gap to capacity anderror probability; this is called the moderate deviation regime [Wan21, Section 2.6]. Two: For simplified decoders, the scaling exponent dictates how much soft-decision can be pruned away and controls the complexity [WD21a]. Three: For parallelized decoders, the scaling exponent dictates how much work still needs to be processed in serial and controls the latency [HMF+21]. Four: Polar code achieves the asymmetric capacity of any binary-input channel using the technique introduced in [HY13]; the corresponding scaling exponent assumes the same estimate as BMS does [Wan21, Chapter 3]. In fact, polar code achieves the same scaling exponent over discrete memoryless channels with constructions given in [RWLP22]. Five: For lossless [Ari10; CK10] and lossy [KU10] compression via polar coding, scaling exponent can be defined similarly and assumes the same bound [Wan21, Chapter 3]. Six: For multiple access channel, rate-splitting helps avoid time-sharing and achieve the same scaling exponent [CY18]; for distributed lossless compression, a similar technique applies [Wan21, Chapter 8]. Seven: Over wiretap channels, polar code achieves the secrecy capacity but consumes secrete keys shared between Alice and Bob; the scaling exponent gives prediction on the length of the secrete key [WU16; GB17]. Eight: For coded computation, scaling behavior is related to not only the code rate but also the waiting time [FM22].

The goal of this paper is to improve $\mu \leqslant 4.714$ to $\mu \leqslant 4.63$. The key idea is that a parallel combining followed by a serial combining makes a channel "less BSC" and hence some inequalities can be strengthened. On the execution side, we remix a handful of techniques that are versatile and flexible: We compute numerical convex envelopes to force functions become convex to apply Jensen's inequality; we use interval arithmetic library to obtain mathematically rigorous bounds to compensate coarse sampling; we use power iteration with a finite state automata to "remember" recent history.

This paper is organized as follows. Section II reviews notations and preliminary results. Section III reiterates the old proof of $\mu \leqslant 4.714$; we did not add anything new; the intention is to provide a baseline for comparison. Section IV introduces tri-variate channel transformation $(U \circledast V) \boxleftarrow W$ and the corresponding Bhattacharyya parameter inequalities. Section V demonstrates how to use power iterations with memory to utilize the new Bhattacharyya parameter inequalities. Section VI wraps up the proof of the new result $\mu \leqslant 4.63$.

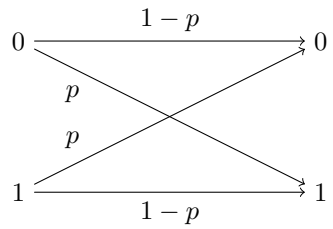

Fig. 1. $\mathrm{BSC}(p)$, binary symmetric channel with crossover probability p.

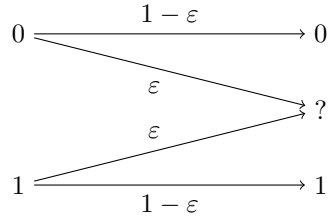

Fig. 2. $\mathrm{BEC}(\varepsilon)$, binary erasure channel with erasure probability ε.

II. Preliminary

A. Binary memoryless symmetric channels

A *binary symmetric channel* (BSC) with crossover probability p is a channel where a user feeds in a 0 or a 1 and it outputs what is fed with probability $1-p$ or flips the bit with probability p. We denote it by $\mathrm{BSC}(p)$ and picture it in Figure 1.

A *binary erasure channel* (BEC) with erasure probability ε is a channel where a user feeds in a 0 or a 1 and it outputs what is fed with probability $1-\varepsilon$ or outputs a question mark with probability ε. We denote it by $\mathrm{BEC}(\varepsilon)$ and picture it in Figure 2.

A *binary memoryless symmetric* (BMS) generalizes BSC and BEC. It is a channel where a user feeds in a 0 or a 1 and it outputs a symbol randomly selected from an alphabet set $\mathcal{Y}$. For a BMS channel W, the conditional probabilities of outputting $y \in \mathcal{Y}$ conditioning on inputs 0 and 1 are denoted by $W(y|0)$ and $W(y|1)$, respectively. A BMS channel is memoryless in the sense that repeated uses of this channel does not alter the conditional distribution. A BMS channel W is symmetric in the sense that for any output symbol $y \in \mathcal{Y}$, there is another symbol $\bar{y} \in \mathcal{Y}$ such that $W(\bar{y}|0) = W(y|1)$ and $W(\bar{y}|1) = W(y|0)$.

B. Channel equivalence and channel decomposition

Channels can have arbitrary output alphabets, but those that pose the same coding challenge are usually treated as the same. An equivalence relation on the class of BMS channels is thus defined to identify and distinguish channels.

We say that a BMS channel $W\colon \{0,1\} \to \mathcal{Z}$ is a *symbol aggregation* of another BMS channel $V\colon \{0,1\} \to \mathcal{Y}$ if there exists a map $\pi\colon \mathcal{Y} \to \mathcal{Z}$ such that

$$V(y|0) : V(y|1) = W(\pi(y)|0) : W(\pi(y)|1),$$
$$\sum_{z \in \pi^{-1}(z)} V(v|0) + V(v|1) = W(z|0) + W(z|1)$$

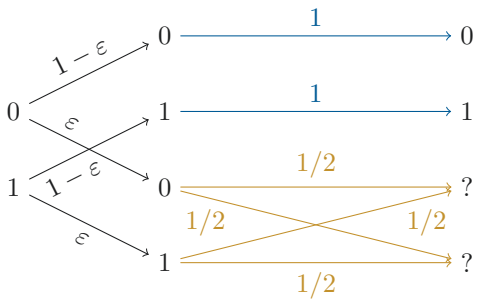

Fig. 3. Illustration of $\mathrm{BEC}(\varepsilon)$ as undergoing $\mathrm{BSC}(0)$ with frequency $1-\varepsilon$ and undergoing $\mathrm{BSC}(1/2)$ with frequency ε. Cf. [LH06, Fig. 2.1].

for all $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$. One sees that the purpose of π is to identify symbols sharing the same likelihood ratio. Two BMS channels are said to be *equivalent* if they share a common symbol aggregation.

This equivalence relation on BMS channels extends to a partial ordering. A BMS W is said to be a *degradation* of V if W can be obtained by post-processing the output of V. (For instance, symbol aggregation counts as post-processing.) It can be shown that V and W are equivalent iff V is a degradation of W and W is a degradation of V. For more on this viewpoint, see how to construct polar codes [TV13], how to deal with general alphabet [GYB18], how to describe input-degradation [Nas18], and how output-degradation is used to achieve $\mu = 2$ within polynomial complexity [GRY22].

Let $\mathcal{BMS}$ be the set of equivalence classes of BMS channels. Let $\mathcal{BMS}_\diamondsuit$ be the set of equivalence classes excluding the noiseless channel ($W(y|0)W(y|1) = 0$ for all y) and the jammed channel ($W(y|0) = W(y|1)$ for all y). What remain are the nontrivial channels where coding is meaningful. Later when Bhattacharyya parameter Z is defined, one will see that $\mathcal{BMS}_\diamondsuit$ are channels with $Z(W) \notin \{0, 1\}$.

Every BMS channel W assumes a *BSC-decomposition*

$$W = \sum_j \alpha_j \,\mathrm{BSC}(p_j),$$

where $\sum_j \alpha_j = 1$ and $0 \leqslant p_j \leqslant 1/2$. This notation means that W can be simulated by (is equivalent to) the following procedure:

- select $\mathrm{BSC}(p_j)$ with probability α_j,
- reveal p_j, and
- feed the input into $\mathrm{BSC}(p_j)$ and reveal the BSC's output.

As an example, Figure 3 pictures the decomposition of $\mathrm{BEC}(\varepsilon)$ into $(1-\varepsilon)\,\mathrm{BSC}(0) + \varepsilon\,\mathrm{BSC}(1/2)$.

In general, the BSC-decomposition of a BMS channel $W\colon \{0,1\} \to \mathcal{Y}$ can be obtained by the following procedure: First, aggregate all output symbols that share the same likelihood ratio. Now that $W(y|0) : W(y|1)$ are all distinct for all $y \in \mathcal{Y}$, enumerate the output alphabet $\mathcal{Y} = \{y_1, \ldots, y_{|\mathcal{Y}|}\}$, let $p_j \leqslant 1/2$ be such that $1 - p_j : p_j = W(y_j|0) : W(y_j|1)$ for all y_j such that $W(y_j|0) \geqslant W(y_j|1)$, and then let α_j be $W(y_j|0) + W(y_j|1)$. For more on this topic, see [GR20] and Modern Coding Theory [RU08, Chapter 4].

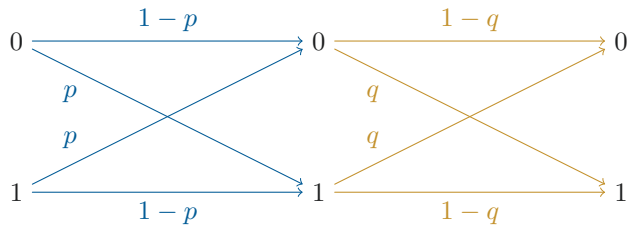

Fig. 4. Illustration of $\mathrm{BSC}(p) \boxplus \mathrm{BSC}(q)$, the serial combination of two BSCs. A 0 ends up flipped to 1 with probability $p(1-q) + (1-p)q = p \star q$. Cf. [LH06, Fig. 1.2].

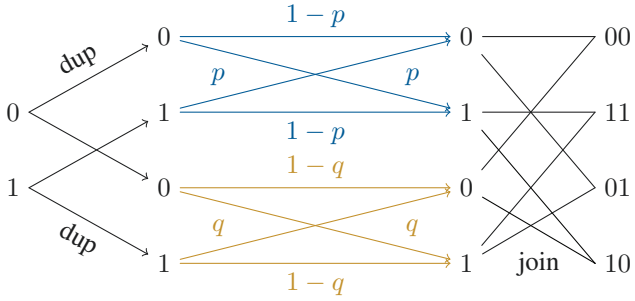

Fig. 5. Illustration of $\mathrm{BSC}(p) \circledast \mathrm{BSC}(q)$, the parallel combination of two BSCs. The output is conflictive (01 or 10) with probability $p \star q$. The output is consistent (00 or 11) with probability $\overline{p \star q}$. Cf. [LH06, Fig. 1.3].

C. Bhattacharyya parameter

The *Bhattacharyya parameter* of a BMS channel W is denoted by $Z(W)$. It is defined to be $Z(\mathrm{BSC}(p)) \coloneqq 2\sqrt{p\bar{p}}$ for BSCs, where $\bar{p}$ means $1-p$. And the definition extends to the entire $\mathcal{BMS}$ via linearity:

$$Z\Big(\sum_j \alpha_j \,\mathrm{BSC}(p_j)\Big) \coloneqq \sum_j \alpha_j Z(\mathrm{BSC}(p_j)) = \sum_j \alpha_j \sqrt{p_j \bar{p}_j}.$$

This quantity can be seen as the expectation of the following random variable:

- select $\mathrm{BSC}(p_j)$ with probability α_j, and
- reveal $Z(\mathrm{BSC}(p_j))$, which is $2\sqrt{p_j\bar{p}_j}$.

As an example, the Bhattacharyya parameter of $\mathrm{BEC}(\varepsilon) = \bar{\varepsilon}\,\mathrm{BSC}(0) + \varepsilon\,\mathrm{BSC}(1/2)$ is $\bar{\varepsilon}Z(\mathrm{BSC}(0)) + \varepsilon Z(\mathrm{BSC}(1/2)) = \bar{\varepsilon}\cdot 0 + \varepsilon \cdot 1 = \varepsilon$. The corresponding random variable follows the Bernoulli distribution with mean ε.

D. Channel synthesis

We now define serial combinations and parallel combinations. Readers are referred to [RU08, Chapter 4], [Ari09], and [GR20] for more details.

The *serial combination* of two BMS channels V and W is denoted by $V \boxplus W$. It is first defined for BSCs: $\mathrm{BSC}(p) \boxplus \mathrm{BSC}(q) \coloneqq \mathrm{BSC}(p \star q)$, where $p \star q \coloneqq p\bar{q} + \bar{p}q$. This new crossover probability satisfies $(\bar{p} - p)(\bar{q} - q) = \overline{p \star q} - p \star q$, where $\overline{p \star q} \coloneqq 1 - p \star q = pq + \bar{p}\bar{q}$. See Figure 4 for a picture. Now extend the definition of serial combination to the whole $\mathcal{BMS}$ via bi-linearity:

$$\Big(\sum_j \alpha_j \,\mathrm{BSC}(p)\Big) \boxplus \Big(\sum_k \beta_k \,\mathrm{BSC}(q_k)\Big)$$

$$\coloneqq \sum_{jk} \alpha_j \beta_k \,\mathrm{BSC}(p_j) \boxplus \mathrm{BSC}(q_k)$$
$$= \sum_{jk} \alpha_j \beta_k \,\mathrm{BSC}(p_j \star q_k).$$

When the two operands are equal, $W \boxplus W$ is also denoted by $W^{\boxplus}$.

The *parallel combination* of two BMS channels V and W is denoted by $V \circledast W$. It is first defined for BSCs: $\mathrm{BSC}(p) \circledast \mathrm{BSC}(q) \coloneqq p \star q \,\mathrm{BSC}(\frac{p\bar{q}}{p\star q}) + \overline{p \star q}\,\mathrm{BSC}(\frac{pq}{\overline{p\star q}})$. And then the definition is extended to the whole $\mathcal{BMS}$ via bi-linearity:

$$\Big(\sum_j \alpha_j \,\mathrm{BSC}(p)\Big) \circledast \Big(\sum_k \beta_k \,\mathrm{BSC}(q_k)\Big)$$
$$\coloneqq \sum_{jk} \alpha_j \beta_k \,\mathrm{BSC}(p_j) \circledast \mathrm{BSC}(q_k)$$
$$= \sum_{jk} \alpha_j \beta_k (p_j \star q_k) \,\mathrm{BSC}\Big(\frac{p_j \bar{q}_k}{p_j \star q_k}\Big)$$
$$+ \sum_{jk} \alpha_j \beta_k (\overline{p_j \star q_k}) \,\mathrm{BSC}\Big(\frac{p_j q_k}{\overline{p_j \star q_k}}\Big).$$

When the two operands are equal, $W \circledast W$ is also denoted by $W^{\bigcirc}$.

E. Bhattacharyya equality

Bhattacharyya parameter is a special parameter in that parallel combination of channels translates to multiplication of Z's.

Theorem 1. *For any BMS channel W,*

$$Z(V \circledast W) = Z(V)Z(W).$$

In particular, $Z(W^{\bigcirc}) = Z(W)^2$.

Proof. We first show that equality holds for V and W being BSCs. Assume $V = \mathrm{BSC}(p)$ and $W = \mathrm{BSC}(q)$. Then $V \circledast W = p \star q \,\mathrm{BSC}(\frac{p\bar{q}}{p\star q}) + \overline{p \star q}\,\mathrm{BSC}(\frac{pq}{\overline{p\star q}})$. The two component BSCs have Bhattacharyya parameters

$$Z\Big(\mathrm{BSC}\Big(\frac{p\bar{q}}{p\star q}\Big)\Big) = 2\sqrt{\frac{p\bar{q}}{p\star q}\,\overline{\Big(\frac{p\bar{q}}{p\star q}\Big)}} = \frac{2\sqrt{p\bar{q}\bar{p}q}}{p\star q}$$

and

$$Z\Big(\mathrm{BSC}\Big(\frac{pq}{\overline{p\star q}}\Big)\Big) = 2\sqrt{\frac{pq}{\overline{p\star q}}\,\overline{\Big(\frac{pq}{\overline{p\star q}}\Big)}} = \frac{2\sqrt{pq\bar{p}\bar{q}}}{\overline{p\star q}}.$$

Overall, $\mathrm{BSC}(\frac{p\bar{q}}{p\star q})$ is with weight $p \star q$ so it contributes $2\sqrt{p\bar{p}q\bar{q}}$ to the Bhattacharyya parameter; $\mathrm{BSC}(\frac{pq}{\overline{p\star q}})$ is with weight $\overline{p\star q}$ so it also contributes $2\sqrt{p\bar{p}q\bar{q}}$ to the Bhattacharyya parameter. In sum, $Z(V \circledast W) = 4\sqrt{p\bar{p}q\bar{q}} = Z(V)Z(W)$.

The rest follows from the linearity of Z and the bi-linearity of $\circledast$ in the BSC-decomposition. More precisely, let V and W have BSC-decompositions $V = \sum_j \alpha_j V_j$ and $W = \sum_k \beta_k W_k$, where V_j and W_k are BSCs. Then $V \circledast W$ has BSC-decomposition $\sum_{jk} \alpha_j \beta_k V_j \circledast W_k$ and Bhattacharyya parameter

$$\sum_{jk} \alpha_j \beta_k Z(V_j \circledast W_k) = \sum_{jk} \alpha_j \beta_k Z(V_j) Z(W_k)$$
$$= \sum_j \alpha_j Z(V_j) \sum_k \beta_k Z(W_k)$$
$$= Z(V)Z(W).$$

This finishes the proof. □

III. Old Proof of $\mu \leqslant 4.714$

This section follows [MHU16] and gives a self-contained proof of $\mu \leqslant 4.174$.

A. Bhattacharyya inequalities

This subsection follows [RU08, Exercise 4.62 (iv)] and proves an inequality concerning Bhattacharyya parameters.

Define a function $f\colon [0,1]^2 \to [0,1]$ by

$$f(x,y) \coloneqq \sqrt{x^2 + y^2 - x^2y^2}.$$

Lemma 2. *For $0 \leqslant p, q \leqslant 1$ we have*

$$f\big(Z(\mathrm{BSC}(p)), Z(\mathrm{BSC}(q))\big) = Z(\mathrm{BSC}(p) \boxplus \mathrm{BSC}(q)).$$

Proof. The left-hand side is

$$\begin{aligned} f(2\sqrt{p\bar{p}}, 2\sqrt{q\bar{q}}) &= \sqrt{4p\bar{p} + 4q\bar{q} - 16p\bar{p}q\bar{q}} \\ &= 2\sqrt{p\bar{p}(q+\bar{q})^2 + (p+\bar{p})^2 q\bar{q} - 4p\bar{p}q\bar{q}} \\ &= 2\sqrt{(p\bar{q}+\bar{p}q)(pq+\bar{p}\bar{q})} \\ &= 2\sqrt{(p\star q)(\overline{p\star q})} \\ &= Z(\mathrm{BSC}(p\star q)), \end{aligned}$$

which is equal to the right-hand side. □

A bi-variate function $f(x,y)$ is said to be *bi-convex* if the function is convex in x for any fixed y and convex in y for any fixed x.

Lemma 3. *$f(x,y)$ is bi-convex.*

Proof. Take the second derivative of f in x:

$$\frac{\partial^2 f}{\partial x^2}(x,y) = \frac{y^2(1-y^2)}{f(x,y)^3}.$$

This fraction is well-defined and nonnegative when $0 < y \leqslant 1$. Along the $y = 0$ segment, f evaluates to $\sqrt{x^2}$ and this is convex in x. Therefore f is convex in x for any fixed y. For convexity in the y-direction we invoke symmetry. This finished the proof □

Theorem 4. *For $V, W \in \mathcal{BMS}$ we have*

$$Z(V \boxminus W) \geqslant f(Z(V), Z(W)).$$

Equality holds when V and W are BSCs.

Proof. Let V and W have BSC-decompositions $\sum_j \alpha_j V_j$ and $\sum_k \beta_k W_k$, respectively, where V_j and W_k are BSCs. Then $V \boxminus W$ has BSC-decomposition $\sum_{jk} \alpha_j \beta_k V_j \boxminus W_k$ and Bhattacharyya parameter

$$\sum_{jk} \alpha_j \beta_k Z(V_j \boxplus W_k) = \sum_{jk} \alpha_j \beta_k f(Z(V_j), Z(W_k)).$$

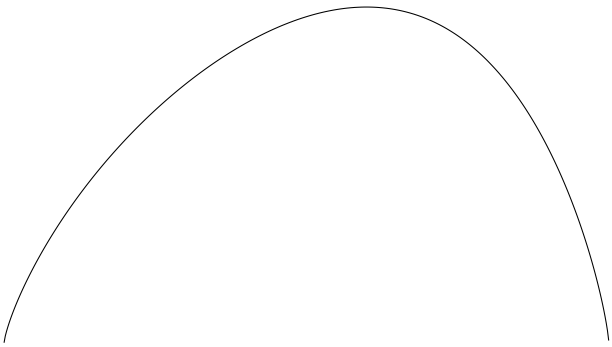

Fig. 6. $x^{0.78}(1-x)^{0.78}(2x^2+3)$, a smooth instance of eigenfunction that induces supremum of ratios 0.87 and overestimate $\mu < 5$.

Let X be a random variable that takes value $Z(V_j)$ with probability α_j. Let Y be an independent random variable that takes value $Z(W_k)$ with probability β_k. Now we want to show

$$Z(V \boxplus W) = \mathbb{E}f(X,Y) \geqslant f(\mathbb{E}X, \mathbb{E}Y) = f(Z(V), Z(W)).$$

The left-hand side is greater than or equal to $\mathbb{E}f(X, \mathbb{E}Y)$ because f is convex in y for each $x = Z(V_j)$. The right-hand side is less than or equal to $\mathbb{E}f(X, \mathbb{E}Y)$ because f is convex in x for a fixed $y = \mathbb{E}Y$. This finishes the proof. □

An interesting consequence of the preceding argument is that the upper bound on $Z(V \boxminus W)$ follows consequently.

Corollary 5. *For any $V, W \in \mathcal{BMS}$, we have*

$$Z(V \boxminus W) \leqslant Z\big(\mathrm{BEC}(Z(V)) \boxminus \mathrm{BEC}(Z(W))\big).$$

Equality holds when V and W are BECs.

Proof. Continue the notation from the previous proof. Now we vary the random variables X and Y but fix their expectations. Then $\mathbb{E}f(X,Y)$ varies while $f(\mathbb{E}X, \mathbb{E}Y)$ remains unchanged. By Karamata's inequality, a corollary of Jensen's inequality, $\mathbb{E}f(X,Y)$ becomes larger when X and Y becomes more marjorized. The most marjorized random variables taking values in $[0,1]$ are those that can only be 0 or 1. Those correspond to the BSC-decompositions of BECs, which consist of $\mathrm{BSC}(0) = \mathrm{BSC}(1)$ (with Bhattacharyya parameter 0) and $\mathrm{BSC}(1/2)$ (with Bhattacharyya parameter 1). Therefore, $Z(V \boxminus W)$ is maximized when V and W are BECs. This finishes the proof. □

Corollary 6. *For any BMS channel W with $z = Z(W)$,*

$$z\sqrt{2-z^2} \leqslant Z(W^{-}) \leqslant 2z - z^2,$$
$$Z(W^{\circ}) = z^2.$$

B. Eigenfunction and eigenvalue

Let $h\colon [0,1] \to \mathbb{R}$ be a concave function such that $h(0) = h(1) = 0$ but positive elsewhere. An overestimate of the scaling exponent can be obtained via the following relation

$$\lambda \coloneqq \sup_{W \in \mathcal{BMS}_\diamond} \frac{h(Z(W^{\circ})) + h(Z(W^{-}))}{2h(Z(W))} \geqslant 2^{-1/\mu}. \tag{1}$$

Recall that $\mathcal{BMS}_\diamond$ is the collection of all equivalence classes of BMS channels where $0 < Z(W) < 1$.

To see why the quotient governs the scaling behavior, note that the "eigenvalue" λ is accumulative when we consider W's children, grandchildren, grand-grandchildren, and so on. To be more precise, we have

$$h(Z(W^{+})) + h(Z(W^{\circ})) \leqslant 2\lambda h(Z(W))$$

and

$$\begin{aligned} &h(Z(W^{+-})) + h(Z(W^{-\circ}) + h(Z(W^{\circ+})) + h(Z(W^{\circ\circ})) \\ &\quad\leqslant 2\lambda h(Z(W^{-})) + 2\lambda h(Z(W^{\circ})) \\ &\quad\leqslant 4\lambda h(Z(W)). \end{aligned}$$

And it is not hard to imagine

$$\begin{aligned} &h(Z(W^{-+-})) + \cdots + h(Z(W^{\circ\circ\circ})) \\ &\quad\leqslant 2\lambda h(Z(W^{+-})) + \cdots + 2\lambda h(Z(W^{\circ\circ})) \\ &\quad\leqslant 4\lambda^2 h(Z(W^{-})) + 4\lambda^2 h(Z(W^{\circ})) \\ &\quad\leqslant 8\lambda^3 h(Z(W)). \end{aligned}$$

In general, when we consider all descendants $W^{?_1?_2\ldots?_n}$ at the nth generation, the average of $h(Z(W^{?_1?_2\ldots?_n}))$ cannot exceed $\lambda^n h(Z(W))$. This quantity is exponentially small. This implies that the Z of deep enough descendants are generally very close to 0 or to 1, hence the polarization phenomenon.

In our proof of $\mu \leqslant 4.63$, we will use eigenvalue to infer the scaling exponent without elaborating on the gap to capacity of an actual polar code. For the machinery that translates the eigenvalue into the asymptotic behavior of polar codes, see [MHU16] or [Wan21, Sections 2.4–2.6].

Since we know $Z(W^{\circ}) = Z(W)^2$ and we know how to bound $Z(W^{-})$ using functions in $Z(W)$, supremum (1) assumes a simpler expression:

$$\sup_{0<x<1} \sup_{x\sqrt{2-x^2} \leqslant y \leqslant 2x-x^2} \frac{h(x^2) + h(y)}{2h(x)}. \tag{2}$$

As an example, $h(x) \coloneqq x^{0.78}(1-x)^{0.78}(2x^2+3)$ leads to a supremum of 0.87 and an upper bound of $\mu \leqslant 4.98$. This eigenfunction is plotted in Figure 6.

C. Power iteration

To obtain a good function h that minimizes suprema (1) and (2)—and thereby minimizing the overestimate of μ—consider the following inductive assignment:

$$h_0(x) \coloneqq x^{0.78}(1-x)^{0.78}(2x^2+3),$$
$$h_{n+1}(x) \coloneqq \sup_{x\sqrt{2-x^2} \leqslant y \leqslant 2x-x^2} \frac{h_n(x^2) + h_n(y)}{2\max h_n}.$$

This is very similar to power iteration, an algorithm that approximates the longest eigenvalue of a square matrix. For this reason h is analogously called an *eigenfunction* and quotients of the form $(h+h)/2h$ are called *eigenvalues*.

It is unlikely that h_n has a simple algebraic formula for large n. To proceed, one puts several ticks on $[0,1]$

$$L \coloneqq \Big\{0, \frac{1}{\ell}, \ldots, \frac{\ell-1}{\ell}, 1\Big\}$$

and let $H \in \mathbb{R}^{\ell+1}$ be an array parametrized by L. The idea is to use $\mathrm{Linear\_Interp}(L, H)$ as a substitute of h both during power iteration and when we want to overestimate μ.

So we let a computer execute the following program.

> For all $x \in L$:
> $H[x] \leftarrow x^{0.78}(1-x)^{0.78}(2x^2+3)$;
> Loop until H converges:
> $h \leftarrow \mathrm{Linear\_Interp}(L, H)$;
> For all $x \in L$:
> $H'[x] \leftarrow \dfrac{h(x^2) + h(y(H,x))}{2h(x)\max H}$;
> $H \leftarrow H'$;

Here,

- H' is an auxiliary array that holds the new content of H;
- $h\colon [0,1] \to \mathbb{R}$ is a function such that $h(x) = H[x]$ for $x \in L$ and linearly interpolated for $x \notin L$;
- $y(H,x)$ is the argument y that maximizes $h(y)$ over the range $x\sqrt{2-x^2} \leqslant y \leqslant 2x - x^2$.

We remark that there is an easy, i.e., $O(1)$, implementation of $y(H,x)$:

$$y(H,x) \coloneqq \begin{cases} x\sqrt{2-x^2} & \text{if } x\sqrt{2-x^2} \geqslant \arg\max H, \\ 2x - x^2 & \text{if } 2x - x^2 \leqslant \arg\max H, \\ \arg\max H & \text{otherwise.} \end{cases} \tag{3}$$

This implementation is sound if h is unimodal. This might not be the case halfway the power iteration; but it deals no damage as long as H converges and induces a good bound.

Empirically, H converges fast. About 200 iterations is enough to make H and H' differ by 10^{-15}. As a comparison, IEEE 754's double-precision floating-point format has 53 significant bits (including the implicit leading 1) and a relative precision of $2.22 \cdot 10^{-16}$.

Now that H converges, let $\hat{H}$ be the limit of H and let $\hat{h}$ be $\mathrm{Linear\_Interp}(L, \hat{H})$. An empirical upper bound of μ is obtained by

$$\left(-\log_2 \max_{x \in L \setminus \{0,1\}} \frac{\hat{h}(x^2) + \hat{h}(y(\hat{H},x))}{2h(x)}\right)^{-1}. \tag{4}$$

Per our computation, $\ell = 2 \cdot 10^5$ gives the first four digits (4.695) mentioned in [MHU16] (wherein $\ell = 10^6$).

We also tested using a variant of Chebyshev nodes as L:

$$L \coloneqq \left\{ \frac{1-\cos(\theta)}{2} \;\middle|\; \theta = 0, \frac{1}{\ell}\pi, \ldots, \frac{\ell-1}{\ell}\pi, \pi \right\}. \tag{5}$$

The motivation behind Chebyshev nodes is that they pay more attentions to the two ends of the interval, the places where $h(x)$ becomes small and more precisions are needed. We found that $\ell = 2 \cdot 10^3$ gives the first four digits (4.695), which indicates that Chebyshev nodes is superior than evenly spaced ticks.

D. Foot of the mountain

Having an array $\hat{H}$ of evaluations, one would ask if $\hat{h} \coloneqq \mathrm{Linear\_Interp}(L, H)$ is a proper substitute of the eigenfunction in the manner of whether

$$\mu \leqslant \left(-\log_2 \max_{0<x<1} \frac{\hat{h}(x^2) + \hat{h}(y(\hat{H},x))}{2h(x)}\right)^{-1}$$

gives a finite upper bound. Unfortunately, no. When x is in $[0, 1/2\ell]$ or in $[1 - 1/2\ell, 1]$, the interpolant is locally linear and the quotient $(\hat{h}(x^2) + \hat{h}(2x - x^2))/2\hat{h}(x)$ is constantly 1 (whereas we want it to be strictly less than 1).

In [MHU16, Section III.C], it is explained how to manipulate $\hat{h}$ to obtain a proper eigenfunction that gives a more rigorous bound on the eigenvalue. The strategy is to let δ be a tiny number; and let $\tilde{h}(x)$ be $x^{0.78}$ when $x \leqslant \delta$ and be $(1-x)^{0.78}$ when $x \geqslant 1 - \delta$. This way, the quotients for $0 < x < \delta$ and for $1 - \delta < x < 1$ are uniformly bounded from above. For $\delta \leqslant x \leqslant 1 - \delta$, since the denominator $2h(x)$ is far away from 0, rounding error and sampling error can be controlled if we evaluate the quotient at a sufficiently fine set of points.

This type of function surgery is limited to very tiny neighborhoods $[0, \delta]$ and $[1-\delta, 1]$ of 0 and 1, respectively. Hence it shall not affect the eigenvalue too much. As an example, the empirical estimate obtained by formula (4) is 4.695; and the rigorous value reported in [MHU16] is $\mu \leqslant 4.714$. These two numbers are only 0.4% apart.

For our new overestimate of μ, we will skip the surgery step and use formula (4), the maximum over a discrete but very fine lattice, as an upper bound on the scaling exponent.

E. Road map to a better bound

While taking suprema (1) and (2), y ranges over an interval $[x\sqrt{2-x^2}, 2x - x^2]$ where the left endpoint is tight if W is a BSC and the right endpoint is tight if W is a BEC. If W is a BEC, then all descendants of W are BECs and $2x - x^2$ is always tight.

On the contrary, if W is a BSC, the left endpoint is only tight for now. After one parallel combination, $W^{\bigcirc}$ will no longer be a BSC, and $x\sqrt{2-x^2}$ will not be tight anymore. That is to say, there is always a tiny gap between $Z(W^{\bigcirc\llcorner})$ and $Z(W^{\bigcirc})\sqrt{2 - Z(W^{\bigcirc})^2}$. If we can come up with a better lower bound than $x\sqrt{2-x^2}$, then supremum (2) will be taken over a smaller region, which makes it smaller.

The next section finds the better bound.

IV. Tri-variate Channel Transformation

Consider the channel combination $(U \circledast V) \boxplus W$. See Figure 7 for a visualization. Define a function $g\colon [0,1]^3 \to [0,1]$ that satisfies

$$g(Z(U), Z(V), Z(W)) = Z((U \circledast V) \boxplus W)$$

for all U, V, W that are BSCs. We can write g more explicitly with the help of the following lemmas.

A. Tri-variate Bhattacharyya function

Lemma 7 (Trivariate Z)**.** $(\mathrm{BSC}(p) \circledast \mathrm{BSC}(q)) \boxplus \mathrm{BSC}(r)$ *has Bhattacharyya parameter*

$$2\sqrt{(p\bar{q}\bar{r} + \bar{p}qr)(p\bar{q}r + \bar{p}q\bar{r})} + 2\sqrt{(pq\bar{r} + \bar{p}\bar{q}r)(pqr + \bar{p}\bar{q}\bar{r})}.$$

Proof. $\mathrm{BSC}(p) \circledast \mathrm{BSC}(q)$ is, by definition, $p \star q\,\mathrm{BSC}(\frac{p\bar{q}}{p\star q}) + \overline{p \star q}\,\mathrm{BSC}(\frac{pq}{p\star q})$. When this channel is serially-combined with a $\mathrm{BSC}(r)$, the first summand becomes

$$p \star q\,\mathrm{BSC}\Big(\frac{p\bar{q}}{p\star q} \star r\Big) = p \star q\,\mathrm{BSC}\Big(\frac{p\bar{q}\bar{r} + \bar{p}qr}{p \star q}\Big)$$

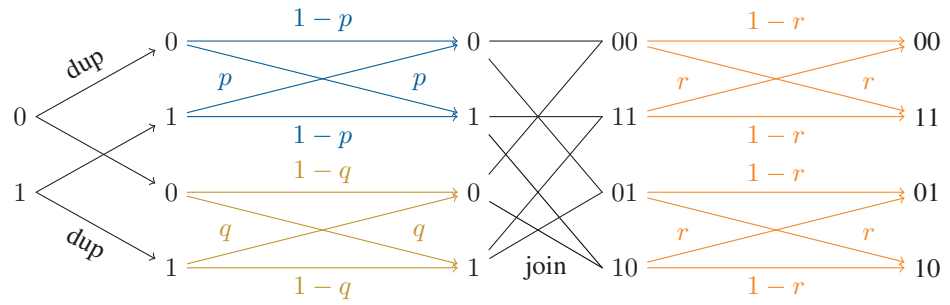

Fig. 7. Illustration of $(\mathrm{BSC}(p) \circledast \mathrm{BSC}(q)) \boxplus \mathrm{BSC}(r)$, The output is conflictive (01 or 10) with probability $p \star q$. The output is consistent (00 or 11) with probability $\overline{p \star q}$.

and contributes Bhattacharyya parameter

$$2(p \star q)\sqrt{\tfrac{p\bar{q}\bar{r}+\bar{p}qr}{p\star q}\left(\overline{\tfrac{p\bar{q}\bar{r}+\bar{p}qr}{p\star q}}\right)} = 2\sqrt{(p\bar{q}\bar{r}+\bar{p}qr)(p\bar{q}r+\bar{p}q\bar{r})}.$$

The second summand becomes

$$\overline{p\star q}\,\mathrm{BSC}\Big(\frac{pq}{\overline{p\star q}} \star r\Big) = \overline{p\star q}\,\mathrm{BSC}\Big(\frac{pq\bar{r}+\bar{p}\bar{q}r}{\overline{p\star q}}\Big)$$

and contributes Bhattacharyya parameter

$$2\overline{p\star q}\sqrt{\tfrac{pq\bar{r}+\bar{p}\bar{q}r}{\overline{p\star q}}\left(\overline{\tfrac{pq\bar{r}+\bar{p}\bar{q}r}{\overline{p\star q}}}\right)} = 2\sqrt{(pq\bar{r}+\bar{p}\bar{q}r)(pqr+\bar{p}\bar{q}\bar{r})}.$$

This finishes the proof. □

Lemma 8 (Z in terms of Z's)**.**

$$g(x,y,z) = \sqrt{C+D}+\sqrt{C-D} = \sqrt{2C+\sqrt{C^2-D^2}},$$

where

$$C \coloneqq \frac{1}{4}(x^2y^2 + \overline{x^2}z^2 + \overline{y^2}z^2),$$
$$D \coloneqq \frac{1}{2}\sqrt{\overline{x^2}}\cdot\sqrt{\overline{y^2}}\cdot z^2.$$

Proof. Let x, y, and z be $2\sqrt{p\bar{p}}$, $2\sqrt{q\bar{q}}$, and $2\sqrt{r\bar{r}}$, respectively, for some $0 \leqslant p,q,r \leqslant 1/2$. From Lemma 7, $g(x,y,z)$ is $2\sqrt{A}+2\sqrt{B}$, where

$$\begin{aligned} A &\coloneqq (p\bar{q}\bar{r}+\bar{p}qr)(p\bar{q}r+\bar{p}q\bar{r}) \\ &= p\bar{q}\bar{r}p\bar{q}r + p\bar{q}\bar{r}\bar{p}q\bar{r} + \bar{p}qrp\bar{q}r + \bar{p}qr\bar{p}q\bar{r} \\ &= p^2\bar{q}^2r\bar{r} + p\bar{p}q\bar{q}\bar{r}^2 + p\bar{p}q\bar{q}r^2 + \bar{p}^2q^2r\bar{r}, \\ &= p\bar{p}q\bar{q}(r^2+\bar{r}^2) + (p^2\bar{q}^2+\bar{p}^2q^2)r\bar{r} \end{aligned}$$

and

$$\begin{aligned} B &\coloneqq (pq\bar{r}+\bar{p}\bar{q}r)(pqr+\bar{p}\bar{q}\bar{r}) \\ &= pq\bar{r}pqr + pq\bar{r}\bar{p}\bar{q}\bar{r} + \bar{p}\bar{q}rpqr + \bar{p}\bar{q}r\bar{p}\bar{q}\bar{r} \\ &= p^2q^2r\bar{r} + p\bar{p}q\bar{q}\bar{r}^2 + p\bar{p}q\bar{q}r^2 + \bar{p}^2\bar{q}^2r\bar{r} \\ &= p\bar{p}q\bar{q}(r^2+\bar{r}^2) + (p^2q^2+\bar{p}^2\bar{q}^2)r\bar{r}. \end{aligned}$$

To show $C+D = 4A$ and $C-D = 4B$, it suffices to show $2(A+B) = C$ and $2(A-B) = D$. For the former,

$$\begin{aligned} 2(A+B) &= 2\begin{pmatrix} p\bar{p}q\bar{q}(r^2+\bar{r}^2) + (p^2\bar{q}^2+\bar{p}^2q^2)r\bar{r} \\ + p\bar{p}q\bar{q}(r^2+\bar{r}^2) + (p^2q^2+\bar{p}^2\bar{q}^2)r\bar{r} \end{pmatrix} \\ &= 4p\bar{p}q\bar{q}(r^2+\bar{r}^2) + 2(p^2+\bar{p}^2)(q^2+\bar{q}^2)r\bar{r} \\ &= \frac{1}{4}x^2y^2\Big(1-\frac{z^2}{2}\Big) + \frac{1}{2}\Big(1-\frac{x^2}{2}\Big)\Big(1-\frac{y^2}{2}\Big)z^2 \\ &= C. \end{aligned}$$

The third equality makes use of the rewriting rules $4r\bar{r} = z^2$ and $r^2+\bar{r}^2 = (r+\bar{r})^2 - 2r\bar{r} = 1 - z^2/2$. For the latter,

$$\begin{aligned} 2(A-B) &= 2\begin{pmatrix} p\bar{p}q\bar{q}(r^2+\bar{r}^2) + (p^2q^2+\bar{p}^2\bar{q}^2)r\bar{r} \\ - p\bar{p}q\bar{q}(r^2+\bar{r}^2) - (p^2\bar{q}^2+\bar{p}^2q^2)r\bar{r} \end{pmatrix} \\ &= 2(\bar{p}^2-p^2)(\bar{q}^2-q^2)r\bar{r} \\ &= 2(\bar{p}-p)(\bar{q}-q)r\bar{r} \\ &= \frac{1}{2}\sqrt{1-x^2}\cdot\sqrt{1-y^2}\cdot z^2 \\ &= D. \end{aligned}$$

The fourth equality makes use of the rewriting rule $(\bar{p}-p)^2 = (\bar{p}+p)^2 - 4\bar{p}p = 1-x^2$. In conclusion, we have $\sqrt{4A}+\sqrt{4B} = \sqrt{C+D}+\sqrt{C-D} = \sqrt{\left(\sqrt{C+D}+\sqrt{C-D}\right)^2} = \sqrt{2C+2\sqrt{C^2-D^2}}$. This finishes the proof. □

A tri-variate function is said to be *tri-convexity* if it is convex whenever any two arguments are fixed and the other argument is varying. If g happens to be tri-convex, we will be able to show that $Z((U \circledast V) \boxplus W)$ is lower bounded by $g(Z(U),Z(V),Z(W))$ by the same Jensen-argument as in Theorem 4. Unfortunately, g is not tri-convex. The next subsection will find a workaround to this.

B. Lower tri-convex envelope

g as defined above is not convex in any of the three variables. We thus attempt to find a lower bound of g that is tri-convex so that Jensen's inequality applies. Consider a function $\breve{g}\colon [0,1]^3 \to [0,1]$ that reads

$$\breve{g}(x,y,z) \coloneqq \sup\{\theta(x,y,z) \mid \theta \leqslant g \text{ and is tri-convex}\},$$

where the supremum runs over all functions $\theta\colon [0,1]^3 \to [0,1]$ that are tri-convex and pointwise bound g from below. This is very similar to the definition of the lower convex envelope, the difference being that θ is not convex but tri-convex. (An example is that xyz is tri-convex but not convex.) We will refer to $\breve{g}$ as the *envelope* of g.

Theorem 9 (Counterpart of Theorem 4)**.** *For $U,V,W \in \mathcal{BMS}$,*

$$Z((U \circledast V) \boxplus W) \geqslant \breve{g}\big(Z(U),Z(V),Z(W)\big).$$

In particular, if $W = V^{\bigcirc}$ for some $V \in \mathcal{BMS}$,

$$Z(W^{\urcorner}) \geqslant \breve{g}\big(\sqrt{Z(W)}, \sqrt{Z(W)}, Z(W)\big).$$

Proof. For the former, it suffices to prove $Z((U \circledast V) \mathbin{\bar{\boxplus}} W) \geqslant \theta\big(Z(U), Z(V), Z(W)\big)$ for all tri-convex θ that is also $\leqslant g$ pointwise. Fix a θ. When U, V, W are BSCs, the inequality we want to prove holds:

$$\begin{aligned} Z((U \circledast V) \mathbin{\bar{\boxplus}} W) &= g(Z(U), Z(V), Z(W)) \\ &\geqslant \theta(Z(U), Z(V), Z(W)). \end{aligned}$$

Now consider BSC-decompositions $U = \sum_i \alpha_i u_i$ and $V = \sum_j \beta_j V_j$ and $W = \sum_k \gamma_k W_k$, where U_i, V_j, W_k are BSCs. Then $(U \circledast V) \mathbin{\bar{\boxplus}} W$ becomes $\sum_{ijk} \alpha_i\beta_j\gamma_k (U_i \circledast V_j) \mathbin{\bar{\boxplus}} W_k$, thereby having Bhattacharyya parameter

$$\begin{aligned} Z((U \circledast V) \mathbin{\bar{\boxplus}} W) &= \sum_{ijk} \alpha_i\beta_j\gamma_k Z((U_i \circledast V_j) \mathbin{\bar{\boxplus}} W_k)) \\ &= \sum_{ijk} \alpha_i\beta_j\gamma_k g(Z(U_i), Z(V_j), Z(W_k)) \\ &\geqslant \sum_{ijk} \alpha_i\beta_j\gamma_k \theta(Z(U_i), Z(V_j), Z(W_k)) \\ &\geqslant \sum_{ij} \alpha_i\beta_j \theta(Z(U_i), Z(V_j), Z(W)) \\ &\geqslant \sum_{i} \alpha_i \theta(Z(U_i), Z(V), Z(W)) \\ &\geqslant \theta(Z(U), Z(V), Z(W)). \end{aligned}$$

This finishes the proof of the lower bound on $Z((U \circledast V) \mathbin{\bar{\boxplus}} W)$. For the lower bound on $Z(W^{\lrcorner})$, plug in $U = V$ and $W = V^{\bigcirc}$, and use the fact that $Z(W) = Z(V)^2$. □

C. Approximate the envelop $\breve{g}$

Computing the envelop $\breve{g}$ algebraically does not seem plausible nor possible. Our approach is to approximate $\breve{g}$ numerically over a mesh

$$M \coloneqq \left\{0, \frac{1}{n}, \dots, \frac{n-1}{n}, 1\right\}^3 \subseteq [0,1]^3.$$

Here, n is the resolution; say $n = 200$. We next evaluate g at this mesh and run a program that iteratively lowers any evaluation that breaks tri-convexity.

In detail, let $G \in \mathbb{R}^{(n+1)\times(n+1)\times(n+1)}$ be an $(n+1)\times(n+1)\times(n+1)$ array indexed by M. Initialize G as

$$G[x,y,z] \leftarrow g(x,y,z)$$

for all $(x,y,z) \in M$. We call G the data points. If the following does not hold for some $(x,y,z) \in M$ and $x \neq 0, 1$:

$$2G[x,y,z] \leqslant G\Big[x - \frac{1}{n}, y, z\Big] + G\Big[x + \frac{1}{n}, y, z\Big], \tag{6}$$

we say that the data point at (x,y,z) is *breaking the convexity along the x-direction*. To correct that, we update this data point as follows

$$G[x,y,z] \leftarrow \frac{1}{2}G\Big[x - \frac{1}{n}, y, z\Big] + \frac{1}{2}G\Big[x + \frac{1}{n}, y, z\Big]. \tag{7}$$

We also demand the convexity in y-direction and z-direction:

$$2G[x,y,z] \leqslant G\Big[x, y - \frac{1}{n}, z\Big] + G\Big[x, y + \frac{1}{n}, z\Big], \tag{8}$$

$$2G[x,y,z] \leqslant G\Big[x, y, z - \frac{1}{n}\Big] + G\Big[x, y, z + \frac{1}{n}\Big] \tag{9}$$

If not, we update $G[x,y,z]$ similarly.

We synthesize a program that constantly searches for instances of data points that break the convexity in any of the three directions and keeps lowering data points. Below is the program; let us call it Tri_Convexify:

```
For all (x, y, z) ∈ M:
    G[x] ← g(x, y, z);
Loop until G converges:
    For all (x, y, z) ∈ M:
        If criterion (6) fails:
            Update via formula (7)
        If criterion (8) fails:
            Update similarly;
        If criterion (9) fails:
            Update similarly;
```

It will stop when all three criteria are met modulo rounding error. Empirically, G converges; mathematically, we can also prove that G converges.

Proposition 10. Tri_Convexify *makes G converge. For any mesh point $(x,y,z) \in M$, the data point $G[x,y,z]$ converges to*

$$\check{G}[x,y,z] \coloneqq \sup\{\Theta[x,y,z] \mid \Theta \leqslant G \text{ and is tri-convex}\}.$$

The supremum is over all arrays $\Theta \in \mathbb{R}^{(n+1)\times(n+1)\times(n+1)}$ that satisfy the discrete convexity criteria criteria (6) to (9) and $\Theta \leqslant G$ entry-wise.

Proof. $\Theta \equiv 0$ is a lower bound on G; it remains to be a lower bound after an update of data point. Thus G keeps decreasing but stays nonnegative. By the monotone convergence theorem, G converges. Let $\check{G}$ be the limit of G after any order of updates. It must be tri-convex because any data point that violates convexity should have been updated.

Now notice that any tri-convex lower bound $\Theta \leqslant G$ remains to be a lower bound on G after an update of G. So any such Θ maintains to be a lower bound on $\check{G}$. This means that $\check{G}$ is greater than or equal to the supremum of all such Θ's. But $\check{G}$ is itself a tri-convex lower bound of G so $\check{G}$ is equal to the supremum; the supremum is a maximum. □

Hereafter, $\check{G}$ denotes both the empirical end result of Tri_Convexify and the supremum defined in Proposition 10. We call $\check{G}$ the *discrete envelop* in contrast to the "continuous" envelop $\breve{g}$.

Lemma 11. Linear_Interp$(M, \check{G})$ *is tri-convex if the data points $\check{G}$ satisfy the discrete convexity criteria (6) to (9).*

Here, Linear_Interp$(M, \check{G})\colon [0,1]^3 \to \mathbb{R}$ is a function that evaluates to $\check{G}[x,y,z]$ at $(x,y,z) \in M$, and is tri-linearly interpolated if $(x,y,z) \notin M$. A defining feature of multilinear interpolation is that it is piecewise linear in any cardinal direction.

Proof of the lemma. We shall prove this for a two dimensional 2×3 grid; the general statement follows by a generalization of this argument.

Let there be six numbers on a grid

$$\begin{array}{ccc} a & - & b & - & c \\ | & & | & & | \\ d & - & e & - & f \end{array} \tag{10}$$

such that $a + c \geqslant 2b$ and $d + f \geqslant 2e$, i.e., the data points are convex. Let $\check{g}$ be obtained by bi-linear interpolation such that

$$\begin{array}{ccc} \check{g}(-1,1) & - & \check{g}(0,1) & - & \check{g}(1,1) \\ | & & | & & | \\ \check{g}(-1,0) & - & \check{g}(0,0) & - & \check{g}(1,0) \end{array}$$

corresponds to grid (10).

We claim that $\check{g}$ is convex at $(0,0)$ in the x-direction, that is, $\check{g}(-\varepsilon,0) + \check{g}(\varepsilon,0) \geqslant 2g(0,0)$ for $0 \leqslant \varepsilon \leqslant 1$. This is because

$$\check{g}(-\varepsilon,0) + \check{g}(\varepsilon,0) = \varepsilon d + \bar{\varepsilon} e + \varepsilon f + \bar{\varepsilon} e \geqslant 2e.$$

Similarly, $\check{g}$ is convex at $(0,1)$ in the x direction, that is, $\check{g}(-\varepsilon,1) + \check{g}(\varepsilon,1) \geqslant 2g(0,1)$.

Now we claim that $\check{g}$ is convex at $(0,y)$, where $0 \leqslant y \leqslant 1$, in the x-direction. That is to say, $\check{g}(-\xi,y)+\check{g}(\xi,y) \geqslant 2g(0,y)$ for $0 \leqslant \xi \leqslant 1$. This is because

$$\begin{aligned} &\check{g}(-\xi,y) + \check{g}(\xi,y) \\ &\quad = y\check{g}(-\xi,1) + \bar{y}\check{g}(-\xi,0) + y\check{g}(\xi,1) + \bar{y}\check{g}(\xi,0) \\ &\quad \geqslant 2y\check{g}(0,1) + 2\bar{y}\check{g}(0,0) = 2\check{g}(0,y). \end{aligned}$$

This shows that the convexity on the boundary of the interpolation cells follows from the convexity of the data points. For convexity within a cell it trivially holds because the value within a cell is defined through interpolation. Hence the lemma is sound. □

We conclude that $\mathrm{Linear\_Interp}(M, \check{G})$, the tri-linear interpolant of the discrete envelop, can be used as a substitute of $\check{g}$, the continuous envelop. Together with Theorem 9, we can now lower bound $Z(W^{\bigcirc\urcorner})$ with a concrete object $\check{G}$ in place of the abstract object $\check{g}$.

Bibliographical remark: some of the arguments presented in this section share common elements with [Wit74].

In the next section, we will demonstrate how to utilize this new lower bound in power iteration.

V. Finite State Power Iteration

For this section, recall the lesson that finite state automata has some memory when digesting the input stream. We develop a variant of power iteration that keeps track of whether a synthetic channel is obtained by serial or parallel combination.

A. Finite state automata

To begin, suppose that there are two concave functions $\varphi_{\urcorner}, \varphi_{\bigcirc} \colon [0,1] \to \mathbb{R}$ that satisfy $\varphi_{\urcorner}(0) = \varphi_{\urcorner}(1) = \varphi_{\bigcirc}(0) = \varphi_{\bigcirc}(1) = 0$ but are positive elsewhere. Define shorthands $\psi_{\urcorner}, \psi_{\bigcirc}, \psi \colon \mathcal{BMS} \to \mathbb{R}$ by

$$\begin{aligned} \psi_{\lrcorner}(W) &\coloneqq \varphi_{\lrcorner}(Z(W)), \\ \psi_{\bigcirc}(W) &\coloneqq \varphi_{\bigcirc}(Z(W)), \\ \psi(W) &\coloneqq \psi_{\urcorner}(W^{\urcorner}) + \psi_{\bigcirc}(W^{\bigcirc}) \\ &\coloneqq \varphi_{\neg}(Z(W^{\urcorner})) + \varphi_{\bigcirc}(Z(W^{\bigcirc})). \end{aligned}$$

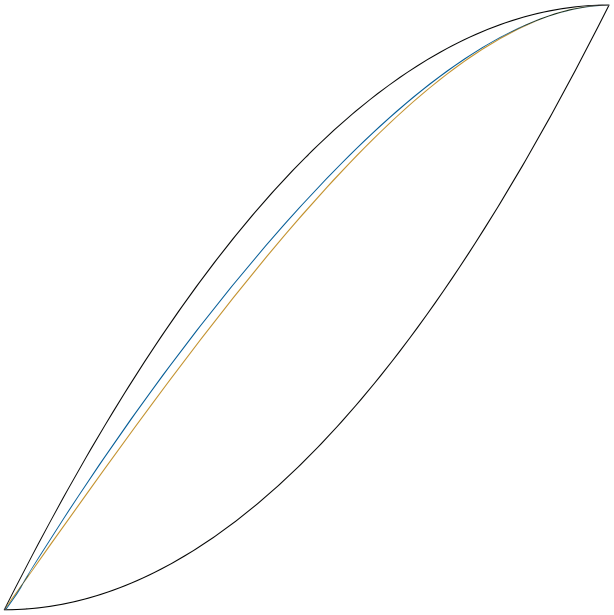

Fig. 8. From top-left to bottom-right: old upper bound of $2x - x^2$, new lower bound of $\check{g}(\sqrt{x}, \sqrt{x}, x)$, old lower bound of $x\sqrt{2 - x^2}$, and parallel combination's Bhattacharyya parameter x^2.

ψ will be the counterpart of h in our new bound.

Here is the motivation of this indirect setup: in [MHU16], $h(Z(W))$ is a score that measures the extent of polarization—a smaller $h(Z(W))$ means that W is more polarized. Now we measure the extent of polarization of W by first giving its children scores and sum them, except that we are biased. As we will see later, $\varphi_{\llcorner}(x)$ is greater than or equal to $\varphi_{\bigcirc}(x)$ for all x. This means that, if $U^{\urcorner}$ and $V^{\bigcirc}$ have the same Bhattacharyya parameter, we will give $V^{\bigcirc}$, a parallel combination, a lower score—because we think that $V^{\bigcirc}$ is more polarized.

There is a reason to distinguish serial combination from parallel combination. Comparing Theorem 9 with Theorem 4, we see that parallel combination assumes better bounds on Bhattacharyya parameters. This implies that the domain of supremum (2) can be made smaller, which potentially makes the quotient corresponding to parallel combination smaller.

Given the motivation, now we want a uniform upper bound on this ratio for all $W \in \mathcal{BMS}$:

$$\begin{aligned} &\frac{\psi(W^{\llcorner}) + \psi(W^{\bigcirc})}{2\psi(W)} \\ &= \frac{\psi_{\neg}(W^{\lrcorner\llcorner}) + \psi_{\bigcirc}(W^{\llcorner\bigcirc}) + \psi_{\neg}(W^{\bigcirc\lrcorner}) + \psi_{\bigcirc}(W^{\bigcirc\bigcirc})}{2\psi_{\lrcorner}(W^{\lrcorner}) + 2\psi_{\bigcirc}(W^{\bigcirc})}. \end{aligned}$$

Hence it suffices to bound

$$\frac{\psi_{\llcorner}(W^{\urcorner\urcorner}) + \psi_{\bigcirc}(W^{\urcorner\bigcirc})}{2\psi_{\llcorner}(W^{\urcorner})} \quad \text{and} \quad \frac{\psi_{\llcorner}(W^{\bigcirc\urcorner}) + \psi_{\bigcirc}(W^{\bigcirc\bigcirc})}{2\psi_{\bigcirc}(W^{\bigcirc})}$$

from above. One can now see the automata: channels that are serial combinations are always scored by $\psi_{\llcorner}$, and channels that are parallel combinations are always scored by $\psi_{\bigcirc}$. The subscript of ψ indicates the current state of the automata; it remembers how the concerned channel was synthesized.

We simplify the supremum of the first quotient as below:

$$\begin{aligned}
&\sup_{W\in\mathcal{BMS}_\diamond} \frac{\psi_{\urcorner}(W^{\lrcorner\llcorner}) + \psi_{\bigcirc}(W^{\llcorner\bigcirc})}{2\psi_{\llcorner}(W^{\urcorner})} \\
&\quad= \sup_{U=W^{\urcorner}} \frac{\varphi_{\urcorner}(Z(U^{\ulcorner})) + \varphi_{\bigcirc}(Z(U^{\bigcirc}))}{2\varphi_{\ulcorner}(Z(U))} \\
&\quad\leqslant \sup_{U\in\mathcal{BMS}_\diamond} \frac{\varphi_{\ulcorner}(Z(U^{\llcorner})) + \varphi_{\bigcirc}(Z(U^{\bigcirc}))}{2\varphi_{\ulcorner}(Z(U))} \\
&\quad= \sup_{0<x<1} \sup_{f(x,x)\leqslant y\leqslant 2x-x^2} \frac{\varphi_{\ulcorner}(x^2) + \varphi_{\bigcirc}(y)}{2\varphi_{\lrcorner}(x)}.
\end{aligned}$$

Here, the second supremum is taken over those U that are themselves serial combinations. We then treat U as an usual BMS channel and apply the classic lower bound (Theorem 4). Because of that, y ranges over $[f(x,x), 2x-x^2]$.

Similarly but not identically, the other quotient with $\psi_{\bigcirc}$ in the denominator can be simplified as below:

$$\begin{aligned}
&\sup_{W\in\mathcal{BMS}_\diamond} \frac{\psi_{\lrcorner}(W^{\bigcirc\urcorner}) + \psi_{\bigcirc}(W^{\bigcirc\bigcirc})}{2\psi_{\bigcirc}(W^{\bigcirc})} \\
&\quad= \sup_{V=W^{\bigcirc}} \frac{\varphi_{\bigcirc}(Z(V^{\lrcorner})) + \varphi_{\bigcirc}(Z(V^{\bigcirc}))}{2\varphi_{\bigcirc}Z(V)} \\
&\quad\leqslant \sup_{0<x<1} \sup_{\breve{g}(\sqrt{x},\sqrt{x},x)\leqslant y\leqslant 2x-x^2} \frac{\varphi_{\bigcirc}(x^2) + \varphi_{\lrcorner}(z)}{2\varphi_{\bigcirc}(x)}.
\end{aligned}$$

Here, the second supremum is taken over those V that are themselves parallel combinations. We invoke Theorem 9 and let z range over $[\breve{g}(\sqrt{x},\sqrt{x},x), 2x-x^2]$. The new supremum is taken over a strictly smaller region than in the previous work—see Figure 8—so a smaller supremum is expected.

B. Power Iteration

It remains to use linear interpolation to represent $\varphi_{\ulcorner}$ and $\varphi_{\bigcirc}$, and apply power iteration to minimize the eigenvalues.

Let L be formula (5); say $\ell = 10^6$. Let $\Phi^{\lrcorner}, \Phi^{\bigcirc} \in \mathbb{R}^{\ell+1}$ be arrays parametrized by L. We execute this program:

For all $x \in L$:
  $\Phi_{\lrcorner}[x] \leftarrow x^{0.78}(1-x)^{0.78}(2x^2+3)$;
  $\Phi_{\bigcirc}[x] \leftarrow x^{0.78}(1-x)^{0.78}(2x^2+3)$;
Loop until $\Phi_{\urcorner}$ and $\Phi_{\bigcirc}$ converge:
  $\varphi_{\urcorner} \leftarrow \mathrm{Linear\_Interp}(L, \Phi_{\ulcorner})$;
  $\varphi_{\bigcirc} \leftarrow \mathrm{Linear\_Interp}(L, \Phi_{\bigcirc})$;
  For all $x \in L$:
    $\Phi'_{\lrcorner}[x] \leftarrow \dfrac{\varphi_{\bigcirc}(x^2) + \varphi_{\ulcorner}(y(\Phi_{\urcorner}, x))}{2\varphi_{\ulcorner}(x)\max\Phi_{\ulcorner}}$;
    $\Phi'_{\bigcirc}[x] \leftarrow \dfrac{\varphi_{\bigcirc}(x^2) + \varphi_{\llcorner}(z(\Phi_{\llcorner}, x))}{2\varphi_{\bigcirc}(x)\max\Phi_{\llcorner}}$;
  $\Phi_{\lrcorner} \leftarrow \Phi'_{\ulcorner}$;
  $\Phi_{\bigcirc} \leftarrow \Phi'_{\bigcirc}$;

Here,

- $\Phi'_{\ulcorner}$ and $\Phi'_{\bigcirc}$ are temporary memory spaces that store the updated content for the next round.
- $y(\Phi_{\lrcorner}, x)$ and $z(\Phi_{\lrcorner}, x)$ are meant to be the arguments that maximize $\varphi_{\llcorner}(y)$ and $\varphi_{\llcorner}(z)$ over the ranges $f(x,x) \leqslant y \leqslant 2x-x^2$ and $\breve{g}(\sqrt{x},\sqrt{x},x) \leqslant z \leqslant 2x-x^2$, respectively.
- $\breve{g}$ is $\mathrm{Linear\_Interp}(M, \check{G})$, which is $\approx \breve{g}$. If a rigorous lower bound of $\breve{g}$ is desired, see Appendix A.

We can reuse the implementation of $y(H,x)$ in formula (3); and implement $z(H,x)$ as

$$z(H,x) \coloneqq \begin{cases} \breve{g}(\sqrt{x},\sqrt{x},x) & \text{if } \breve{g}(\sqrt{x},\sqrt{x},x) \geqslant \arg\max H, \\ 2x-x^2 & \text{if } 2x-x^2 \leqslant \arg\max H, \\ \arg\max H & \text{otherwise.} \end{cases}$$

Empirically, $\Phi_{\lrcorner}$ and $\Phi_{\bigcirc}$ converge. Let $\hat{\Phi}_{\lrcorner}$ and $\hat{\Phi}_{\bigcirc}$ be the end results of power iteration. We can now use

$$\hat{\varphi}_{\ulcorner} \coloneqq \mathrm{Linear\_Interp}(L, \hat{\Phi}_{\urcorner})$$
$$\hat{\varphi}_{\bigcirc} \coloneqq \mathrm{Linear\_Interp}(L, \hat{\Phi}_{\bigcirc})$$

as the scoring functions. See Figure 9 for their plots; notice that $\hat{\varphi}_{\urcorner} \geqslant \hat{\varphi}_{\bigcirc}$.

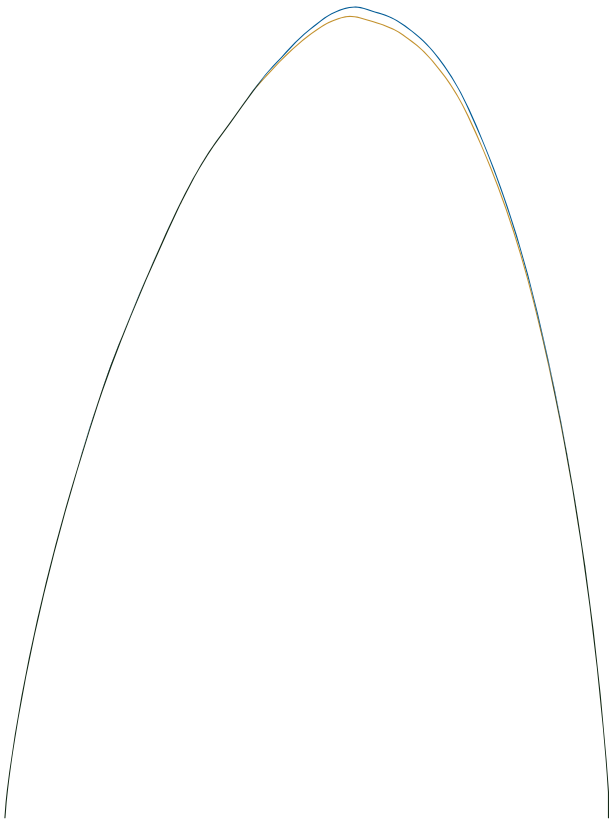

Fig. 9. Eigenfunction pair $\hat{\varphi}_{\urcorner}$ (blue) and $\hat{\varphi}_{\bigcirc}$ (brown). The former is greater for $x > 0.4$—this is the place where $z(x, \hat{\Phi}_{\ulcorner}) > y(x, \hat{\Phi}_{\ulcorner})$ due to $\breve{g}(\sqrt{x},\sqrt{x},x) > f(x,x)$.

VI. New Proof of $\mu \leqslant 4.63$

This section gathers the materials and proves the main theorem.

Theorem 12 (Main theorem)**.** *$\mu \leqslant 4.63$, where μ is the scaling exponent of polar coding using Arıkan's kernel $\left[\begin{smallmatrix}1&0\\1&1\end{smallmatrix}\right]$ over BMS channels.*

Proof. We have seen that $\mathrm{Linear\_Interp}(M, \check{G}) \approx \breve{g}$ and $\breve{g}(\sqrt{x},\sqrt{x},x) \leqslant Z(W^{\llcorner})$, where W is a parallel combination of another BMS channel and $x = Z(W)$. To obtain

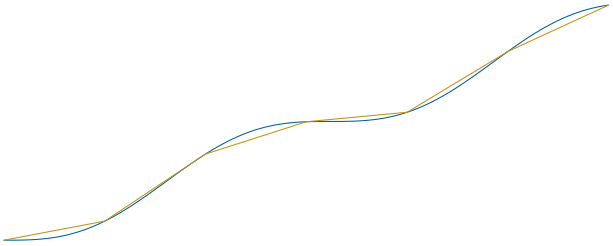

Fig. 10. Piecewise linear interpolation (brown) of an arbitrary function (blue) is an approximation but not a valid lower bound.

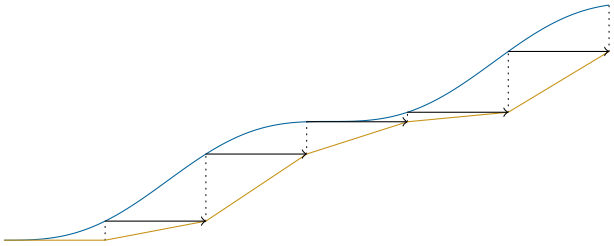

Fig. 11. If the target function is monotonically increasing, the evaluation at an interval's left end is a lower bound over the interval. Thus, shifting the interpolant δ units right makes it a lower bound, where δ is the width of the intervals.

a practical yet rigorous lower bound on $Z(W^{\urcorner})$, see Appendix A for how to define $G_{\searrow}$ and $\check{G}_{\searrow}$. By Theorem 17 therein, we have $\check{g}_{\searrow}(\sqrt{x},\sqrt{x},x) \leqslant Z(W^{\llcorner})$ where $\check{g}_{\searrow} \coloneqq \mathrm{Linear\_Interp}(M, \check{G}_{\searrow})$.

Next, apply power iteration to optimize for the eigenvalues

$$\lambda_{\urcorner} \coloneqq \sup_{x\in L\setminus\{0,1\}} \sup_{f(x,x)\leqslant z\leqslant 2x-x^2} \frac{\hat{\varphi}_{\urcorner}(x^2)+\hat{\varphi}_{\bigcirc}(z)}{2\hat{\varphi}_{\urcorner}(x)},$$

$$\lambda_{\bigcirc} \coloneqq \sup_{x\in L\setminus\{0,1\}} \sup_{\check{g}_{\searrow}(\sqrt{x},\sqrt{x},x)\leqslant z\leqslant 2x-x^2} \frac{\hat{\varphi}_{\bigcirc}(x^2)+\hat{\varphi}_{\urcorner}(z)}{2\hat{\varphi}_{\bigcirc}(x)}.$$

Per our execution, both suprema are about 0.860714.

Finally, we conclude that

$$\sup_{W\in\mathcal{BMS}_{\diamond}} \frac{\psi(W^{\urcorner})+\psi(W^{\bigcirc})}{2\psi(W)}$$

has $\max(\lambda_{\urcorner},\lambda_{\bigcirc}) \approx 0.860715$ as an empirical upper bound. And μ has $\log_2(\max(\lambda_{\llcorner},\lambda_{\bigcirc}))^{-1} \approx 4.62125$ as an empirical upper bound. Hence it is safe to say $\mu \leqslant 4.63$. □

VII. Conclusions

In this paper, we argue that the scaling exponent is an essential constant characterizing the scaling behavior of polar coding, of which very little is known. We then lower the overestimate of the scaling exponent from 4.714 to 4.63.

The limit of this method—analyzing $(U \circledast V) \boxplus W$ to gain better control on Z—is 4.61126. This number is obtained by assuming g tri-convex and using $g(\sqrt{x},\sqrt{x},x) = x(1+\sqrt{5-4x^2})/2$ as the lower bound on the $Z(W^{\bigcirc\lrcorner})$ in terms of $x = Z(W^{\bigcirc})$. Futhermore, we expect that analyzing $(U \circledast V) \boxplus (W \circledast X)$ leads to a better bound.

Appendix A
Linear Interpolation Made a Proper Lower Bound

There is a caveat when approximating $\check{g}$ using $\check{G}$: the mesh is coarse. For one-dimensional interpolation (i.e., H and $\Phi_{\llcorner}$ and $\Phi_{\bigcirc}$), we can afford arrays of size 10^6 and the error is negligible as we only cares about the first three digits of the scaling exponent. Unlike the one-dimensional case, for a three-dimensional mesh, the cube of 200 is already $8\cdot 10^6$ but the error is of the order of $1/200$. See Figure 10 for an illustration of the caveat.

In this appendix, we will demonstrate how to find an array $G_{\searrow}$ such that $\mathrm{Linear\_Interp}(M, G_{\searrow}) \leqslant g$ pointwise. With $G_{\searrow}$, we can run the iterative algorithm Tri_Convexify and the resulting array $\check{G}_{\searrow}$ will satisfy $\mathrm{Linear\_Interp}(M, \check{G}_{\searrow}) \leqslant \check{g}$ pointwise. This will give us a mathematically rigorous control on $Z(W^{\bigcirc\llcorner})$.

A. Monotonic increasing approach

Observe that $g(x,y,z)$ is a monotonic increasing function in x, y, and z. This is a consequence of x, y, z, and g being the Bhattacharyya parameters of certain BSCs. In particular, we know $g(a,b,c) \leqslant g(x,y,z)$ for all $(x,y,z) \in (a,b,c) + [0,1/n]^3$. Here, the right-hand side is the mesh cell whose lower-left-near corner is (a,b,c) and upper-right-far corner is $(a+1/n, b+1/n, c+1/n)$.

Inspired by the observation, we declare a new array $G_{\rightarrow} \in \mathbb{R}^{(n+1)\times(n+1)\times(n+1)}$ that is parametrized by M and populated by

$$G_{\rightarrow}[a,b,c] \leftarrow g\Big(a-\frac{1}{n}\vee 0, b-\frac{1}{n}\vee 0, c-\frac{1}{n}\vee 0\Big).$$

Here, $a - 1/n \vee 0$ means $\max(a-1/n, 0)$. We call this the *monotonic increasing approach* and illustrate it in Figure 11. The following lemma shows that linearly interpolating this array serves as a lower bound.

Lemma 13. $\mathrm{Linear\_Interp}(M, G_{\rightarrow}) \leqslant g$ *pointwise.*

Proof. It suffices to check the inequality cell-by-cell. Fix an $(a,b,c) \in M$; we shall prove the inequality on the cell $(a,b,c)+[0,1/n]^3$. Now for any (x,y,z) in this cell, $\mathrm{Linear\_Interp}(M, G_{\rightarrow})(x,y,z)$ is a convex combination of these eight numbers

$$\begin{aligned} &G_{\rightarrow}[a+\tfrac{0}{n}, b+\tfrac{1}{n}, c+\tfrac{1}{n}] \quad G_{\rightarrow}[a+\tfrac{1}{n}, b+\tfrac{1}{n}, c+\tfrac{1}{n}],\\ &G_{\rightarrow}[a+\tfrac{0}{n}, b+\tfrac{0}{n}, c+\tfrac{1}{n}], \quad G_{\rightarrow}[a+\tfrac{1}{n}, b+\tfrac{0}{n}, c+\tfrac{1}{n}],\\ &G_{\rightarrow}[a+\tfrac{0}{n}, b+\tfrac{1}{n}, c+\tfrac{0}{n}], \quad G_{\rightarrow}[a+\tfrac{1}{n}, b+\tfrac{1}{n}, c+\tfrac{0}{n}],\\ &G_{\rightarrow}[a+\tfrac{0}{n}, b+\tfrac{0}{n}, c+\tfrac{0}{n}], \quad G_{\rightarrow}[a+\tfrac{1}{n}, b+\tfrac{0}{n}, c+\tfrac{0}{n}], \end{aligned}$$

By the definition of $G_{\rightarrow}$, all eight numbers are less than or equal to $g(a,b,c)$, so $\mathrm{Linear\_Interp}(M, G_{\rightarrow})(x,y,z) \leqslant g(a,b,c) \leqslant g(x,y,z)$. □

If we apply the monotonic increasing approach to a $200\times 200\times 200$ mesh, we get $\mu \leqslant 4.66359$. To go below 4.63, we have to combine this with a second approach introduced in the next subsection.

B. Smoothness approach

Idea: if we control two end points and the second derivative, we control the evaluations in between.

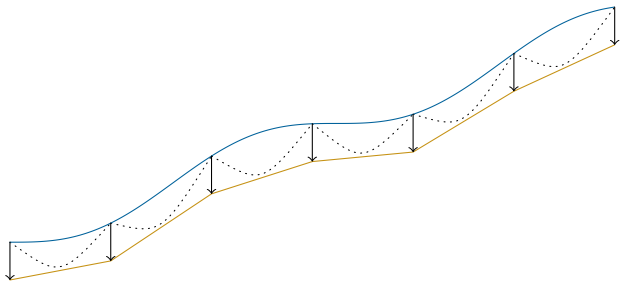

Fig. 12. If the target function is smooth (the second derivative has an upper bound, $\sup f'' \leqslant m$), it can be lower bounded by parabolas. Thus, shifting the linear interpolant $m\delta^2/8$ units down makes it a lower bound, where δ is the width of the intervals.

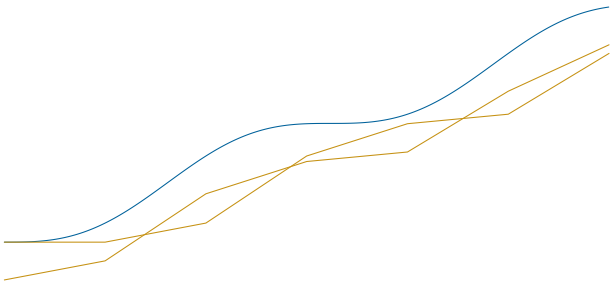

Fig. 13. Both Figure 11 and 12 are proper lower bounds. Now we have the freedom to choose tighter bound on an interval-by-interval basis.

Lemma 14. *Let $f\colon [0,1] \to \mathbb{R}$ be doubly-differentiable on $[0,1]$. Suppose $f(0) = f(1) = 0$ and $f''(x) \leqslant m$ for some $m \geqslant 0$. Then for any $x \in [0,1]$,*

$$f(x) \geqslant -\frac{m}{8}.$$

Proof. As a special case of Lagrange interpolation, consider a linear interpolation using $(0, f(0))$ and $(0, f(1))$ as reference points. Its error term is $(0-x)(1-x)f''(y)/2$ for some $y \in [0,1]$. Clearly $x(1-x) \leqslant 1/4$ and this finishes the proof. □

Lemma 15. *Let n be a positive integer. Let $g\colon [0,1/n]^3 \to \mathbb{R}$ be doubly-differentiable on $[0,1/n]^3$. Suppose $g = 0$ at the eight corners of the cube $[0,1/n]^3$. Suppose $g_{xx} \leqslant m_1$ and $g_{yy} \leqslant m_2$ as well as $g_{zz} \leqslant m_3$ for some $m_1, m_2, m_3 \geqslant 0$. Then for any $(x,y,z) \in [0,1/n]^3$,*

$$g(x,y,z) \geqslant -\frac{m_1+m_2+m_3}{8n^2}.$$

Proof. First apply Lemma 14 in the x-direction to lower bound $g(x,0,0)$, $g(x,0,1/n)$, $g(x,1/n,0)$, and $g(x,1/n,1/n)$ by $-m_1/8n^2$. Then apply Lemma 14 in the y-direction to lower bound $g(x,y,0)$ and $g(x,y,1/n)$ by $-(m_1+m_2)/8n^2$. Finally, apply Lemma 14 in the z-direction to lower bound $g(x,y,z)$ by $-(m_1+m_2+m_3)/8n^2$. □

Lemma 15 provides an excellent way to lower bound g on a mesh as the denominator $8n^2$ keeps up with the memory usage $O(n^3)$ better than the monotonic increasing approach did, in which case the error was $O(g'/n)$,

Let us declare a new array $G_\downarrow \in \mathbb{R}^{(n+1)\times(n+1)\times(n+1)}$ that is parametrized by M and populated by

$$G_\downarrow[a,b,c] \leftarrow g(a,b,c) - \frac{m_1+m_2+m_3}{8n^3},$$

where

$$\begin{aligned} m_1 &= \sup_{((a,b,c)+[-1/n,1/n]^3)\cap[0,1]^3} \max(g_{xx},0),\\ m_2 &= \sup_{((a,b,c)+[-1/n,1/n]^3)\cap[0,1]^3} \max(g_{yy},0),\\ m_3 &= \sup_{((a,b,c)+[-1/n,1/n]^3)\cap[0,1]^3} \max(g_{zz},0). \end{aligned}$$

The suprema are taken over all mesh cells that touch (a,b,c). The following lemma confirms that linearly interpolating $G_\downarrow$ serves as a valid lower bound of g. See also Figure 12 for an illustration.

Lemma 16. $\text{Linear\_Interp}(M, G_\downarrow) \leqslant g$ *pointwise.*

Proof. It suffices to check the inequality cell-by-cell. Fix an $(a,b,c) \in M$; we shall prove that the inequality holds on the cell $(a,b,c) + [0,1/n]^3$. At the eight corners of this cell, g and $\text{Linear\_Interp}(M,G)$ coincide. Hence $\tilde{g} \coloneqq g - \text{Linear\_Interp}(M,G)$ is a function that is zero at the eight corners. Its second derivatives $\tilde{g}_{xx}$, $\tilde{g}_{yy}$, and $\tilde{g}_{zz}$ are nothing but g_{xx}, g_{yy}, and g_{zz}, respectively. Now apply Lemma 15: $\tilde{g} \geqslant -(m_1+m_2+m_3)/8n^3$, where m_1, m_2, m_3 are the suprema of the second derivatives over the concerned cell. Hence

$$\begin{aligned} g &\geqslant \text{Linear\_Interp}(M,G) - \frac{m_1+m_2+m_3}{8n^3}\\ &\geqslant \text{Linear\_Interp}(M, G_\downarrow). \end{aligned}$$

This finishes the proof. □

C. Interval arithmetic for derivatives

In the previous subsection, we see how to initialize $G_\downarrow$ in principle—evaluate g at every mesh point and subtract by $1/8n^2$ times the local suprema of second derivatives. It remains to actually compute the second derivatives.

The first shortcut we take is that m_1, m_2, m_3 do not have to be the exact suprema; any upper bounds serve the same purpose. So it remains to bound the second derivatives from above for every cell. In fact, since we have $8n^2$ in the denominator, there is nearly no precision requirement; any m_1, m_2, m_3 that are < 10 will end up giving a better bound than $G_\rightarrow$.

The second shortcut we take is that there are softwares that can take care of differentiation. Given the formula of g, SageMath, an open-source mathematical software system, computes its symbolic derivatives by passing the queries to Maxima, a classical open-source software that excels at algebra.

Once the symbolic expressions of g_{xx}, g_{yy}, and g_{zz} are obtained, the third—perhaps the biggest—shortcut we take is treating each cell as a fuzzy triple of real numbers and evaluating the expressions using interval arithmetic. For example, the cell $(0.1, 0.4, 0.7) + [0, 0.1]^3$ can be seen as an imperfect representation of three real numbers x, y, and z that are approximately 0.15, 0.45, and 0.75 with error radius 0.05. When evaluating, say, $xy - z$, all we know is that the true value must lie in the set

$$\{xy - z \mid (x, y, z) \in (0.1, 0.4, 0.7) + [0, 0.1]^3\} \\ = [0.1 \cdot 0.4 - 0.8, 0.2 \cdot 0.5 - 0.7].$$

An interval arithmetic package takes cares of the tedious edge cases and returns an interval that *provably* contains the true value of every mathematical expression.

In our case, MPFI is the C-library SageMath calls behind the scene. The abbreviation stands for multiple-precision floating-point interval. A defining feature of the MPFI library is that it temporarily increases the precision during the evaluation process to narrow down the output interval. As an example, evaluating $x - x$ without simplification first will double the error radius. But by cutting the interval into smaller pieces the result will be the union of smaller intervals surrounding 0, hence improving the output precision.

D. The better-of-the-two approach

Given two approaches, $G_\rightarrow$ and $G_\downarrow$, we see that $G_\rightarrow$ is tighter at places where g' is small but g'' is large; and $G_\downarrow$ is tighter whenever g' is big and g'' is far less than $8n^2$. In the sequel, we will let $G_\searrow$ be the array that uses values from $G_\rightarrow$ or $G_\downarrow$ depending on which is tighter.

Consider a cell $(a, b, c) + [0, 1/n]^3$ whose lower-left-near corner is at (a, b, c) and upper-right-far corner is at $(a + 1/n, b + 1/n, c + 1/n)$. For every such cell, we want to decide whether to use the monotonic increasing approach or the smoothness approach. We set a rule: we will use $G_\rightarrow$ by default, but if $G_\rightarrow$ is worse than $G_\downarrow$ at all eight corners $(a, b, c) + \{0, 1\}^3$, we switch to $G_\downarrow$.

Now that we have specified which approach to use for every cell, we can initialize $G_\searrow$. Intuitively speaking, $G_\searrow[a, b, c]$ will be $G_\rightarrow[a, b, c]$ if any cell that touches (a, b, c) decides to go for the increasing approach, but will be $G_\downarrow[a, b, c]$ if all cells that touch (a, b, c) decide to go for the smoothness bound. A formal summary is as below,

- $G_\searrow[a, b, c] = G_\rightarrow[a, b, c]$ iff for some mesh point $(x, y, z) \in (a, b, c) + \{-1/n, 0, 1/n\}^3$ that shares a common cell with (a, b, c), the monotonic increasing approach is better: $G_\rightarrow[x, y, z] \geqslant G_\downarrow[x, y, z]$.
- $G_\searrow[a, b, c] = G_\downarrow[a, b, c]$ iff for all mesh points $(x, y, z) \in (a, b, c) + \{-1/n, 0, 1/n\}^3$ that share a common cell with (a, b, c), the smoothness approach is better: $G_\downarrow[x, y, z] \geqslant G_\rightarrow[x, y, z]$.

The following theorem concludes this appendix.

Theorem 17. *With $G_\searrow$ defined as above, we have*

$$\mathrm{Linear\_Interp}(M, G_\searrow) \leqslant g.$$

With $\check{G}_\searrow$ being the result of performing Tri_Convexify *on $G_\searrow$, we have*

$$\check{g}_\searrow \coloneqq \mathrm{Linear\_Interp}(M, \check{G}_\searrow) \leqslant \check{g}.$$

In particular, with $x \coloneqq Z(W)$ we have

$$Z(W^{\llcorner}) \geqslant \check{g}_\searrow(\sqrt{x}, \sqrt{x}, x).$$

Proof. The first statement is by how $G_\searrow$ merges data points from $G_\rightarrow$ and $G_\downarrow$. The second statement is by the first statement and Lemma 11. The last statement is by the second statement and Theorem 9. □

For a faster way to convexify an array, see the next appendix.

Appendix B
Convexify Faster

In this appendix, we describe a strategy to tri-convexify a three-dimensional array G. This strategy converges faster than repeated uses of formula (7).

Consider a one dimensional array $A = \{a_0, \ldots, a_n\}$ that is parametrized by $L = \{l_0, \ldots, l_n\}$. We want to lower some entries of A so that $\mathrm{Linear\_Interp}(L, A)$ becomes convex. This is equivalent to finding the convex hull of points

$$(l_0, \max(A)), (l_0, a_0), \ldots, (l_n, a_n), (l_n, \max(A)).$$

We next apply Graham's scan. Since the l-coordinates are already sorted, the time complexity of one scan is $O(n)$. The output of Graham's scan is a list of points that support the convex hull. For points that lie strictly inside, we update their a-values using linear interpolation. This step also costs time complexity $O(n)$.

Now that we know how to convexify one dimensional arrays, we iteratively apply this to the axes of the thee-dimensional array G. Here, an axis of G is data points where two coordinates are fixed and the other coordinate is varying.

Since convexifying one axis only lowers the data points, G is ever decreasing. But since G stays non-negative, it converges by monotone convergence theorem.

References

[Ari09] Erdal Arikan. "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels". In: *IEEE Transactions on Information Theory* 55.7 (July 2009), pp. 3051–3073. ISSN: 1557-9654. DOI: 10.1109/TIT.2009.2021379.

[Ari10] Erdal Arikan. "Source polarization". In: *2010 IEEE International Symposium on Information Theory*. June 2010, pp. 899–903. DOI: 10.1109/ISIT.2010.5513567.

[AT09] Erdal Arikan and Emre Telatar. "On the rate of channel polarization". In: *2009 IEEE International Symposium on Information Theory*. June 2009, pp. 1493–1495. DOI: 10.1109/ISIT.2009.5205856.

[BFS+17] Sarit Buzaglo, Arman Fazeli, Paul H. Siegel, Veeresh Taranalli, and Alexander Vardy. "Permuted successive cancellation decoding for polar codes". In: *2017 IEEE International Symposium on Information Theory (ISIT)*. June 2017, pp. 2618–2622. DOI: 10.1109/ISIT.2017.8007003.

[CK10] Harm S. Cronie and Satish Babu Korada. "Lossless source coding with polar codes". In: *2010 IEEE International Symposium on Information Theory*. June 2010, pp. 904–908. DOI: 10.1109/ISIT.2010.5513561.

[CY18] Rémi A. Chou and Aylin Yener. “Polar Coding for the Multiple Access Wiretap Channel via Rate-Splitting and Cooperative Jamming”. In: *IEEE Transactions on Information Theory* 64.12 (Dec. 2018), pp. 7903–7921. ISSN: 1557-9654. DOI: 10.1109/TIT.2018.2865741.

[FHMV21] Arman Fazeli, Hamed Hassani, Marco Mondelli, and Alexander Vardy. “Binary Linear Codes With Optimal Scaling: Polar Codes With Large Kernels”. In: *IEEE Transactions on Information Theory* 67.9 (Sept. 2021), pp. 5693–5710. ISSN: 1557-9654. DOI: 10.1109/TIT.2020.3038806.

[FM22] Dorsa Fathollahi and Marco Mondelli. *Polar Coded Computing: The Role of the Scaling Exponent*. 2022. arXiv: 2201.10082 [cs.IT].

[FT17] Silas L. Fong and Vincent Y. F. Tan. “Scaling Exponent and Moderate Deviations Asymptotics of Polar Codes for the AWGN Channel”. In: *Entropy* 19.7 (2017). ISSN: 1099-4300. DOI: 10.3390/e19070364. URL: https://www.mdpi.com/1099-4300/19/7/364.

[FV14] Arman Fazeli and Alexander Vardy. “On the scaling exponent of binary polarization kernels”. In: *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Sept. 2014, pp. 797–804. DOI: 10.1109/ALLERTON.2014.7028536.

[GB14] Dina Goldin and David Burshtein. “Improved Bounds on the Finite Length Scaling of Polar Codes”. In: *IEEE Transactions on Information Theory* 60.11 (Nov. 2014), pp. 6966–6978. ISSN: 1557-9654. DOI: 10.1109/TIT.2014.2359197.

[GB17] Talha Cihad Gulcu and Alexander Barg. “Achieving Secrecy Capacity of the Wiretap Channel and Broadcast Channel With a Confidential Component”. In: *IEEE Transactions on Information Theory* 63.2 (Feb. 2017), pp. 1311–1324. ISSN: 1557-9654. DOI: 10.1109/TIT.2016.2631223.

[GHU12] Ali Goli, S. Hamed Hassani, and Rüdiger Urbanke. “Universal bounds on the scaling behavior of polar codes”. In: *2012 IEEE International Symposium on Information Theory Proceedings*. July 2012, pp. 1957–1961. DOI: 10.1109/ISIT.2012.6283641.

[GR20] Naveen Goela and Maxim Raginsky. “Channel Polarization Through the Lens of Blackwell Measures”. In: *IEEE Transactions on Information Theory* 66.10 (Oct. 2020), pp. 6222–6241. ISSN: 1557-9654. DOI: 10.1109/TIT.2020.3016605.

[GRY22] Venkatesan Guruswami, Andrii Riazanov, and Min Ye. “Arıkan meets Shannon: Polar codes with near-optimal convergence to channel capacity”. In: *IEEE Transactions on Information Theory* (2022), pp. 1–1. ISSN: 1557-9654. DOI: 10.1109/TIT.2022.3146786.

[GYB18] Talha Cihad Gulcu, Min Ye, and Alexander Barg. “Construction of Polar Codes for Arbitrary Discrete Memoryless Channels”. In: *IEEE Transactions on Information Theory* 64.1 (2018), pp. 309–321. DOI: 10.1109/TIT.2017.2765663.

[HAU10] S. Hamed Hassani, Kasra Alishahi, and Rudiger Urbanke. “On the scaling of polar codes: II. The behavior of un-polarized channels”. In: *2010 IEEE International Symposium on Information Theory*. June 2010, pp. 879–883. DOI: 10.1109/ISIT.2010.5513585.

[HAU14] Seyed Hamed Hassani, Kasra Alishahi, and Rüdiger L. Urbanke. “Finite-Length Scaling for Polar Codes”. In: *IEEE Transactions on Information Theory* 60.10 (Oct. 2014), pp. 5875–5898. ISSN: 1557-9654. DOI: 10.1109/TIT.2014.2341919.

[HMF+21] Seyyed Ali Hashemi, Marco Mondelli, Arman Fazeli, Alexander Vardy, John Cioffi, and Andrea Goldsmith. “Parallelism versus Latency in Simplified Successive-Cancellation Decoding of Polar Codes”. In: *IEEE Transactions on Wireless Communications* (2021), pp. 1–1. ISSN: 1558-2248. DOI: 10.1109/TWC.2021.3125626.

[HMTU13] S. Hamed Hassani, Ryuhei Mori, Toshiyuki Tanaka, and Rüdiger L. Urbanke. “Rate-Dependent Analysis of the Asymptotic Behavior of Channel Polarization”. In: *IEEE Transactions on Information Theory* 59.4 (Apr. 2013), pp. 2267–2276. ISSN: 1557-9654. DOI: 10.1109/TIT.2012.2228295.

[HY13] Junya Honda and Hirosuke Yamamoto. “Polar Coding Without Alphabet Extension for Asymmetric Models”. In: *IEEE Transactions on Information Theory* 59.12 (Dec. 2013), pp. 7829–7838. ISSN: 1557-9654. DOI: 10.1109/TIT.2013.2282305.

[KMTU10] Satish Babu Korada, Andrea Montanari, Emre Telatar, and Rüdiger Urbanke. “An empirical scaling law for polar codes”. In: *2010 IEEE International Symposium on Information Theory*. June 2010, pp. 884–888. DOI: 10.1109/ISIT.2010.5513579.

[KŞU10] Satish Babu Korada, Eren Şaşoğlu, and Rüdiger Urbanke. “Polar Codes: Characterization of Exponent, Bounds, and Constructions”. In: *IEEE Transactions on Information Theory* 56.12 (Dec. 2010), pp. 6253–6264. ISSN: 1557-9654. DOI: 10.1109/TIT.2010.2080990.

[KU10] Satish Babu Korada and Rüdiger L. Urbanke. “Polar Codes are Optimal for Lossy Source Coding”. In: *IEEE Transactions on Information Theory* 56.4 (Apr. 2010), pp. 1751–1768. ISSN: 1557-9654. DOI: 10.1109/TIT.2010.2040961.

[LH06] Ingmar Land and Johannes Huber. “Information Combining”. In: *Foundations and Trends® in Communications and Information Theory* 3.3 (2006), pp. 227–330. ISSN: 1567-2190. DOI: 10.1561/0100000013. URL: http://dx.doi.org/10.1561/0100000013.

[Mah20] Hessam Mahdavifar. “Polar Coding for Non-Stationary Channels”. In: *IEEE Transactions on Information Theory* 66.11 (Nov. 2020), pp. 6920–

6938. ISSN: 1557-9654. DOI: 10.1109/TIT.2020.3020929.

[MHU16] Marco Mondelli, S. Hamed Hassani, and Rüdiger L. Urbanke. "Unified Scaling of Polar Codes: Error Exponent, Scaling Exponent, Moderate Deviations, and Error Floors". In: *IEEE Transactions on Information Theory* 62.12 (Dec. 2016), pp. 6698–6712. ISSN: 1557-9654. DOI: 10.1109/TIT.2016.2616117.

[MT14] Ryuhei Mori and Toshiyuki Tanaka. "Source and Channel Polarization Over Finite Fields and Reed–Solomon Matrices". In: *IEEE Transactions on Information Theory* 60.5 (May 2014), pp. 2720–2736. ISSN: 1557-9654. DOI: 10.1109/TIT.2014.2312181.

[Nas18] Rajai Nasser. "Characterizations of Two Channel Orderings: Input-Degradedness and the Shannon Ordering". In: *IEEE Transactions on Information Theory* 64.10 (Oct. 2018), pp. 6759–6770. ISSN: 1557-9654. DOI: 10.1109/TIT.2018.2859252.

[PU19] Henry D. Pfister and Rüdiger L. Urbanke. "Near-Optimal Finite-Length Scaling for Polar Codes Over Large Alphabets". In: *IEEE Transactions on Information Theory* 65.9 (Sept. 2019), pp. 5643–5655. ISSN: 1557-9654. DOI: 10.1109/TIT.2019.2915595.

[RU08] Tom Richardson and Rüdiger Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008. DOI: 10.1017/CBO9780511791338.

[RWLP22] Constantin Runge, Thomas Wiegart, Diego Lentner, and Tobias Prinz. *Multilevel Binary Polar-Coded Modulation Achieving the Capacity of Asymmetric Channels*. 2022. arXiv: 2202.04010 [cs.IT].

[Tro21] Grigorii Trofimiuk. "Shortened Polarization Kernels". In: *2021 IEEE Globecom Workshops (GC Wkshps)*. Dec. 2021, pp. 1–6. DOI: 10.1109/GCWkshps52748.2021.9681982.

[TT21] Grigorii Trofimiuk and Peter Trifonov. "Window Processing of Binary Polarization Kernels". In: *IEEE Transactions on Communications* 69.7 (July 2021), pp. 4294–4305. ISSN: 1558-0857. DOI: 10.1109/TCOMM.2021.3072730.

[TV13] Ido Tal and Alexander Vardy. "How to Construct Polar Codes". In: *IEEE Transactions on Information Theory* 59.10 (Oct. 2013), pp. 6562–6582. ISSN: 1557-9654. DOI: 10.1109/TIT.2013.2272694.

[Wan21] Hsin-Po Wang. *Complexity and Second Moment of the Mathematical Theory of Communication*. 2021. arXiv: 2107.06420 [cs.IT].

[WD21a] Hsin-Po Wang and Iwan M. Duursma. "Log-Logarithmic Time Pruned Polar Coding". In: *IEEE Transactions on Information Theory* 67.3 (Mar. 2021), pp. 1509–1521. ISSN: 1557-9654. DOI: 10.1109/TIT.2020.3041523.

[WD21b] Hsin-Po Wang and Iwan M. Duursma. "Polar Codes' Simplicity, Random Codes' Durability". In: *IEEE Transactions on Information Theory* 67.3 (2021), pp. 1478–1508.

[Wit74] H. Witsenhausen. "Entropy inequalities for discrete channels". In: *IEEE Transactions on Information Theory* 20.5 (Sept. 1974), pp. 610–616. ISSN: 1557-9654. DOI: 10.1109/TIT.1974.1055285.

[WU16] Yi-Peng Wei and Sennur Ulukus. "Polar Coding for the General Wiretap Channel With Extensions to Multiuser Scenarios". In: *IEEE Journal on Selected Areas in Communications* 34.2 (Feb. 2016), pp. 278–291. ISSN: 1558-0008. DOI: 10.1109/JSAC.2015.2504275.

[YB15] Min Ye and Alexander Barg. "Polar codes using dynamic kernels". In: *2015 IEEE International Symposium on Information Theory (ISIT)*. June 2015, pp. 231–235. DOI: 10.1109/ISIT.2015.7282451.

[YFV19] Hanwen Yao, Arman Fazeli, and Alexander Vardy. "Explicit Polar Codes with Small Scaling Exponent". In: *2019 IEEE International Symposium on Information Theory (ISIT)*. 2019, pp. 1757–1761. DOI: 10.1109/ISIT.2019.8849741.