Sub-4.7 Scaling Exponent of Polar Codes

Hsin-Po Wang®, Ting-Chun Lin®, Alexander Vardy®, Fellow, IEEE, and Ryan Gabrys®, Member, IEEE

Abstract—Polar codes approach channel capacity provably and empirically and are thereby a constituent code of the 5G standard. Compared to low-density parity-check codes, however, the performance of short-length polar codes have 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. The idea behind this improvement is that, instead of describing the relation between a channel W and its children $W_3$ and $W_5$, we describe the relation between W and its grandchildren $W^F_7$, $W^\circ_5$, $W^F_3$, and $W^{n\circ}$. By doing so, the evolution of channels becomes “less Markovian” and hence more tighter inequalities can be obtained.

Index Terms—Polar code, scaling exponent, diversity evolution, information combining, binary memoryless symmetric (BMS) channels.

I. INTRODUCTION

POLAR codes were proved to be capacity achieving over any binary memoryless symmetric (BMS) channel [1]. Polar codes also showed great potential in practice and were 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 codes, the other member of the 5G standard. Improving short-length polar codes further paves the way for applications such as Internet of Things, as some devices can only afford easily-decodable codes and others must reply very promptly.

Now that improving the performance of polar codes at finite block length is on the agenda, we first need to know how much can be said about the unmodified codes. There are two regimes that were considered in literature. In the error exponent regime, the code rate is fixed and the asymptotics of the error probability is evaluated. For polar codes, it was shown that the block error probability under the standard successive cancellation decoder scales as $\exp(-\sqrt{N})$, where N is the block length. For variations of polar codes that use different matrices as the polarizing kernel, the asymptotics 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 now straightforward thanks to a number of interesting and highly non-trivial contributions [2], [3], [4], [5].

In the scaling exponent regime, the second approach that characterizes the performance of polar codes under the successive cancellation decoder, the error probability is fixed and the asymptotics of the code rate is evaluated. It is observed that the gap to capacity, which is the difference between the channel capacity and the code rate, scales as $N^{-1/\mu}$. Called the scaling exponent, this number $\mu$ is difficult to pinpoint exactly. Here is a list of progresses made in prior works.

- [6] showed that $0.2786 \geq 1/\mu \geq 0.2669$ over binary erasure channels (BECs).
- [7] showed that $\mu \approx 3.626$ over BECs.
- [8] showed that $3.553 \leq \mu$ over BMS channels.
- [9] showed that $3.579 \leq \mu \leq 6$ over BMS channels.
- [10] showed that $\mu \leq 5.702$ over BMS channels.
- Mondelli, Hassani, and Urbanke showed in [11] that $\mu \leq 4.714$ over BMS channels.

The last record stood for seven years and is the one we intend to break.

The notion of scaling exponent generalizes to other scenarios. To name a few:

- Over additive white Gaussian noise channels, $\mu \leq 4.714$ [12].

Now that improving the performance of polar codes at finite block length is on the agenda, we first need to know how much can be said about the unmodified codes. There are two regimes that were considered in literature. In the error exponent regime, the code rate is fixed and the asymptotics of the error probability is evaluated. For polar codes, it was shown that the block error probability under the standard successive cancellation decoder scales as $\exp(-\sqrt{N})$, where N is the block length. For variations of polar codes that use different matrices as the polarizing kernel, the asymptotics 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 now straightforward thanks to a number of interesting and highly non-trivial contributions [2], [3], [4], [5].

In the scaling exponent regime, the second approach that characterizes the performance of polar codes under the successive cancellation decoder, the error probability is fixed and the asymptotics of the code rate is evaluated. It is observed that the gap to capacity, which is the difference between the channel capacity and the code rate, scales as $N^{-1/\mu}$. Called the scaling exponent, this number $\mu$ is difficult to pinpoint exactly. Here is a list of progresses made in prior works.

- [6] showed that $0.2786 \geq 1/\mu \geq 0.2669$ over binary erasure channels (BECs).
- [7] showed that $\mu \approx 3.626$ over BECs.
- [8] showed that $3.553 \leq \mu$ over BMS channels.
- [9] showed that $3.579 \leq \mu \leq 6$ over BMS channels.
- [10] showed that $\mu \leq 5.702$ over BMS channels.
- Mondelli, Hassani, and Urbanke showed in [11] that $\mu \leq 4.714$ over BMS channels.

The last record stood for seven years and is the one we intend to break.

The notion of scaling exponent generalizes to other scenarios. To name a few:

- Over additive white Gaussian noise channels, $\mu \leq 4.714$ [12].

0018-9448 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Over non-stationary BECs, $\mu \leq 7.34$; over non-stationary BMS channels, $\mu \leq 8.54$ [13].

Over (hereafter stationary) BECs, permuting the rows of the Kronecker powers of Arıkan’s kernel [19] improves the scaling from $\mu \approx 3.627$ to $\mu \approx 3.479$ with little complexity overhead [14].

Using larger kernel matrices improves scaling exponents even further: over BECs,
- $\mu \approx 3.627$ for a $2 \times 2$ kernel,
- $\mu \approx 3.577$ for an $8 \times 8$ kernel [15],
- $\mu \approx 3.346$ for a $16 \times 16$ kernel [16],
- $\mu \approx 3.122$ for a $32 \times 32$ kernel, and
- $\mu \approx 2.87$ for a $64 \times 64$ kernel [17].

See [18] for sizes between $9 \times 9$ and $31 \times 31$.

Using alphabet extension also improves scaling exponents: over BECs, $\mu \approx 3.328$ for a $2 \times 2$ kernel over a 4-ary alphabet [19].

In general, any nontrivial matrix kernel over any alphabet has a finite scaling exponent over any discrete memoryless channel [20, Sections 5.7–5.10] (which is heavily inspired by [21], [22]).

 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 $3 \times 3$ kernels over BECs [23].

Most challengingly, a series of works attempted to reach $\mu \approx 2$, the optimal scaling exponent, and succeeded.
- Pfister and Urbanke [24] showed that $\mu \approx 2$ can be reached using Reed–Solomon kernels over $q$-ary erasure channels as $q \to \infty$.
- Fazeli et al. [25] showed that $\mu \approx 2$ can be reached using random linear kernel over BECs.
- Guruswami et al. [26] 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 [27] 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 broader impacts.
- One can now describe the trade-off between gap to capacity and error probability; this is called the moderate deviation regime [20, Section 2.6].
- For simplified decoders, the scaling exponent dictates how much soft-decision can be pruned away and controls the complexity [28].
- For parallelized decoders, the scaling exponent dictates how much work still needs to be processed in serial and controls the latency [29].
- Polar code achieves the asymmetric capacity of any binary-input channel using the technique introduced in [30]; the corresponding scaling exponent assumes the same estimate as the BMS case. [20, Chapter 3]. In fact, polar codes achieve the same scaling exponent over discrete memoryless channels with constructions given in [31].
- For lossless [32], [33] and lossy [34] compression via polar coding, scaling exponent can be defined similarly and assumes the same estimate [20, Chapter 3].
- For multiple access channel, rate-splitting helps avoid time-sharing and achieve the same scaling exponent [35]; for distributed lossless compression, a similar technique applies [20, Chapter 8].
- Over wiretap channels, polar codes achieve the secrecy capacity but need to consume secret keys shared between Alice and Bob; the scaling exponent gives prediction on the length of the secret key [36], [37].
- For coded computation, scaling behavior is related to not only the code rate but also the waiting time [38].

The goal of this paper is to improve $\mu \leq 4.714$ to $\mu \leq 4.63$. 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: 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 automaton to remember channel’s history.

This paper is organized as follows. Section II reviews notations and preliminary results. Section III reiterates the definitions and preliminary results. Section IV explains our plan for the new bound $\mu \leq 4.63$. Section V introduces tri-variate channel transformation $(U \otimes V) \ast W$ and proves new inequalities. Section VI demonstrates how to use a power iteration “with memory” to take advantage of tri-variate channel transformation. Section VII wraps up the proof of the new result $\mu \leq 4.63$.

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 $\text{BSC}(p)$ and picture it in Figure 1.

A binary erasure channel (BEC) with erasure probability $\varepsilon$ 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 \( \text{BEC}(\varepsilon) \) 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 multiple uses of this channel do 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 a symbol \( \bar{y} \in \mathcal{Y} \) such that \( W(\bar{y}|0) = W(\bar{y}|1) \) and \( W(y|1) = W(y|0) \) and \( \bar{y} = y \).

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 channel. An equivalence relation on the class of BMS channels is thus defined to identify and distinguish channels.

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

\[
V(y|0) : V(y|1) = W(\pi(y)|0) : W(\pi(y)|1),
\]

\[
\sum_{v \in \pi^{-1}(z)} V(v|0) + V(v|1) = W(z|0) + W(z|1)
\]

for all \( y \in \mathcal{Y} \) and \( z \in \mathcal{Z} \). That is to say, \( y \in \mathcal{Y} \) and \( \pi(y) \in \mathcal{Z} \) should have the same likelihood ratio (the first equation) and when \( \pi \) maps several \( y \)'s to a \( z \), the probability mass of \( z \) is the sum of those \( y \)'s (the second equation). Two BMS channels \( U \) and \( V \) are said to be equivalent if they share a common symbol aggregation \( W \); that is, \( W \) is a symbol aggregation of \( U \) and also a symbol aggregation of \( V \).

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 a special type of 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 [40], how to deal with general alphabet [41], how to describe input-degradation [42], and how output-degradation is used to achieve \( \mu = 2 \) within polynomial complexity [26].

Let \( \mathcal{BMS}^* \) be the set of equivalence classes of BMS channels. Let \( \mathcal{BMS} \) be the set of equivalence classes of BMS channels excluding the completely noiseless channel \( W(y|0)W(y|1) = 0 \) for all \( y \) and the totally 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 \( \mathcal{BMS}^* \) are channels with \( Z(W) \notin \{0, 1\} \).

Every BMS channel \( W \) assumes a BSC-decomposition

\[
W = \sum_j \alpha_j \text{BSC}(p_j),
\]

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

- select \( \text{BSC}(p_j) \) with probability \( \alpha_j \),
- reveal \( p_j \), and
- feed the input into \( \text{BSC}(p_j) \) and reveal the BSC’s output.

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

In general, the BSC-decomposition of a BMS channel \( W : \{0, 1\} \to \mathcal{Y} \) 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 \( p_j \leq 1/2 \) be such that \( 1 - p_j \) = \( W(y_j|0) \) = \( W(y_j|1) \) for all \( y_j \) such that \( W(y_j|0) \geq W(y_j|1) \), and then let \( \alpha_j = W(y_j|0) + W(y_j|1) \).

So far, we have implicitly assumed that BMS channels has discrete output alphabet \( \mathcal{Y} \). There is also a notion of continuous BMS channels. Most (if not all) statements in this paper can be generalized to the continuous case; so we stick to the discrete notation for simplicity. For more on BMS channels, see [43], Information Combining [39, Section 2] and Modern Coding Theory [44, Chapter 4].

C. Bhattacharyya Parameter

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

\[
Z\left(\sum_j \alpha_j \text{BSC}(p_j)\right) := \sum_j \alpha_j Z(\text{BSC}(p_j)) = \sum_j \alpha_j \sqrt{p_jp_j}.
\]

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

- select \( \text{BSC}(p_j) \) with probability \( \alpha_j \), and
- reveal \( Z(\text{BSC}(p_j)) \), which is \( 2\sqrt{p_jp_j} \).
As an example, the Bhattacharyya parameter of \( \text{BEC}(\varepsilon) = \varepsilon \cdot \text{BSC}(0) + (1 - \varepsilon) \cdot \text{BSC}(1/2) = \varepsilon \cdot 0 + (1 - \varepsilon) \cdot 1 = \varepsilon \). This corresponding random number follows the Bernoulli distribution with mean \( \varepsilon \).

We remark that this definition is compatible with other definitions of \( Z(W) \), e.g., in [1, Section I.A]. Bhattacharyya parameter is always between 0 and 1. Higher Z means that the channel has more noises. The wholly jammed channel has \( Z = 1 \) and it is the only channel that has \( Z = 1 \). The perfectly noiseless channel has \( Z = 0 \) and it is the only channel that has \( Z = 0 \).

D. Channel Synthesis

We now define serial combinations and parallel combinations. Readers are referred to [44, Chapter 4], [1, Section II], and [43, Section V] for more details.

The serial combination of two BMS channels \( V \) and \( W \) is denoted by \( V \parallel W \). It is first defined for BSCs: \( \text{BSC}(p) \parallel \text{BSC}(q) := \text{BSC}(p \cdot q) \), where \( p \cdot q := p_\times q + \bar{p}_\times \bar{q} \). This new crossover probability satisfies the property that \( p - \bar{p} \) is multiplicative: \( (\bar{p} - \bar{q})(\bar{q} - \bar{q}) = p \times q - p \times q \), where \( p \times q := 1 - p \times q = p_\times q + \bar{p}_\times \bar{q} \). See Figure 4 for a picture. Now extend the definition of serial combination to the whole BMS via bi-linearity:

\[
\left( \sum_j \alpha_j \text{BSC}(p_j) \right) \star \left( \sum_k \beta_k \text{BSC}(q_k) \right) := \sum_{jk} \alpha_j \beta_k \text{BSC}(p_j) \oplus \text{BSC}(q_k) = \sum_{jk} \alpha_j \beta_k \text{BSC}(p_j \star q_k).
\]

When the two operands are equal, \( W \star W \) is also denoted by \( W^\star \).

The parallel combination of two BMS channels \( V \) and \( W \) is denoted by \( V \otimes W \). It is first defined for BSCs: \( \text{BSC}(p) \otimes \text{BSC}(q) := \text{BSC}(p_\parallel q) \), where \( p_\parallel q = \frac{p_\times q \pm \sqrt{p_\times q - p_\times q \cdot p_\times q}}{2} \). And then the definition is extended to the whole BMS via bi-linearity:

\[
\left( \sum_j \alpha_j \text{BSC}(p_j) \right) \circ \left( \sum_k \beta_k \text{BSC}(q_k) \right) := \sum_{jk} \alpha_j \beta_k \text{BSC}(p_j) \otimes \text{BSC}(q_k) = \sum_{jk} \alpha_j \beta_k (p_j \star q_k) \text{BSC}(p_j \parallel q_k).
\]

When the two operands are equal, \( W \otimes W \) is also denoted by \( W^\otimes \).

Channel synthesis is studied for both polar codes and low-density parity-check codes and, due to its popularity and versatility, has many alternative names: channel transformation, density evolution, information combining, to name a few. The serial combination \( W \parallel W = W^\parallel \) is sometimes called the check-node convolution and denoted as \( W^- \) or \( W^\parallel \) (because it is more noisy than \( W \)) or \( W^\parallel \) (because it is decoded earlier in the successive cancellation decoder). The parallel combination \( W \otimes W = W^\otimes \) is also called the variable-node (bit-node) convolution and denoted as \( W^+ \) or \( W^\parallel \) (because it is more reliable than \( W \)) or \( W^\otimes \) (because it is decoded later in the successive cancellation decoder).

Should readers forget what \( W^\circ \) and \( W^\circ \) mean, we recommend going back to page 1 and check Table I.

E. Scaling Exponent

One natural question to ask is how can we characterize the children \( W^\circ \) and \( W^\circ \), the grandchildren \( W^\circ \), \( W^\circ \), \( W^\circ \), \( W^\circ \), and so on. The most important fact regarding polar codes is that, if we keep applying random combinations, the channels will almost always converge to the fully noiseless channel or the entirely jammed channel.

Theorem 1 (Polarization [1, Theorem 1]): Let \( C_1, C_2, \ldots \in \{\parallel, \circ\} \) be a sequence of unbiasedly and independently randomly selected combinations. Then, with probability 1,

\[
\lim_{k \to \infty} Z(W^{C_1 C_2 \ldots C_k}) \in \{0, 1\}.
\]

The second question to ask is how fast is this polarization? There are many equivalent ways to define the speed of polarization. We find the following one handy.

Definition 2 (scaling exponent—the eigenvalue definition): The scaling exponent of polar codes over BMS channels is defined to be the least number \( \mu \) such that there exists a number \( \delta > 0 \) and a function \( h \) from BMS* to positive real numbers such that

\[
\sup_{W \in \text{BMS}*} \frac{h(W^\circ) + h(W^\circ)}{2h(W)} = 2^{-1/\mu} \tag{1}
\]

and that \( h(W) > \delta \) whenever \( 1/3 < Z(W) < 2/3 \).
The overall idea is that the $h$-value of the descendant channels of $W$ (that is, $W^{[e]}$, $W^{[e]}$, $W^{[e]_0}$, $W^{[e]_0}$, and so on) should decrease on average. The condition $h'(W) > δ$ is to encode the fact that we want to avoid (by penalizing) descendant channels whose Bhattacharyya parameters do not converge to 0 or 1.

We refer readers to [11] or [20, Sections 2.4–2.6] for how this definition of scaling exponent is related the performance of polar codes’ successive cancellation decoder. As an example, the following corollary characterizes how $μ$ controls the asymptotic behavior of $h(W^{[e]_1}C^{[e]_2}C^{[e]_3})$.

**Corollary 3 (scaling exponent—the asymptotic characterization):** (Supremum) (1) implies, for all $k ∈ N$,

$$\frac{1}{2^k h(Z)} \sum_{C_1 ∈ {[1, 0]} \atop C_2 ∈ {[1, 0]} \atop C_3 ∈ {[1, 0]}} h(W^{C_1}C_2⋯C_{k}) \leq 2^{-k/μ}. \tag{2}$$

**Proof:** For $k = 1$ this is just supremum (1). For larger $k$, suppose that inequality (2) holds for if $k$ is replaced by $k − 1$, then

$$\frac{1}{2^k h(Z)} \sum_{C_1 ∈ {[1, 0]} \atop C_2 ∈ {[1, 0]} \atop C_3 ∈ {[1, 0]}} h(W^{C_1}C_2⋯C_{k}) = \frac{1}{2^k h(Z)} \sum_{C_1 ∈ {[1, 0]} \atop C_2 ∈ {[1, 0]} \atop C_3 ∈ {[1, 0]}} h(W^{C_1}) \cdot 2^{-k(k−1)/μ}$$

apply the induction hypothesis

$$\leq \frac{1}{2^k h(Z)} \sum_{C_1 ∈ {[1, 0]} \atop C_2 ∈ {[1, 0]} \atop C_3 ∈ {[1, 0]}} 2^{k−1} h(W^{C_1}) \cdot 2^{−(k−1)/μ}$$

$$= \frac{h(W^{[1]}) + h(W^{[0]})}{2h(Z)} \cdot 2^{−(k−1)/μ}$$

apply supremum (1)

$$\leq 2^{−1/μ} \cdot 2^{−(k−1)/μ}$$

$$= 2^{−k/μ}.$$  

This finishes the proof. □

Due to the corollary above and other reasons, $h$ is usually called the eigenfunction and $2^{−1/μ}$ is usually called the eigenvalue.

The main contribution of this paper is to show that $μ ≤ 4.63$ using a cleverly constructed eigenfunction $h$. Our proof is largely inspired by [11]’s proof. Hence we devote the next section for reviewing the old proof.

### III. OLD PROOF OF $μ ≤ 4.714$

This section follows [11] and gives a self-contained proof of $μ ≤ 4.714$.

#### A. Bhattacharyya Equality

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

**Theorem 4:** For any BMS channel $W$,

$$Z(V ⊗ W) = Z(V)Z(W).$$

In particular, $Z(W^{[0]}) = Z(W^2)$.

**Proof:** We first show that the equality holds for $V$ and $W$ being BSCs. Assume $V = BSC(p)$ and $W = BSC(q)$. Then $V ⊗ W = (p ∗ q) BSC(\frac{pq}{pq}) + p ∗ q BSC(\frac{pq}{pq})$. The two summand BSCs have Bhattacharyya parameters

$$Z(BSC(\frac{pq}{pq})) = 2 \sqrt{pq/p * q} = 2 \sqrt{pq/p * q}$$

and

$$Z(BSC(\frac{pq}{pq})) = 2 \sqrt{pq/p * q} = 2 \sqrt{pq/p * q}.$$  

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

The rest of the proof follows from the linearity of $Z$ and the bi-linearity of $\circ$. To elaborate, let $V$ and $W$ have BSC-decompositions $V = \sum_j α_jV_j$ and $W = \sum_k β_kW_k$, where $V_j$ and $W_k$ are BSCs. Then $V ⊗ W$ has BSC-decomposition $\sum_{jk} α_j β_k V_j ⊗ W_k$ and Bhattacharyya parameter

$$\sum_{jk} α_j β_k Z(V_j ⊗ W_k) = \sum_{jk} α_j Z(V_j) \sum_k β_k Z(W_k)$$

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

This finishes the proof. □

#### B. Bhattacharyya Inequalities

This subsection follows [44, Exercise 4.62 (iv)] and proves an inequality concerning Bhattacharyya parameter and serial combination. This is basically Mrs. Gerber’s Lemma for Bhattacharyya parameter.

Define a function $f : [0, 1]^2 → [0, 1]$ by

$$f(x, y) := \sqrt{x^2 + y^2 − x^2 y^2}. \tag{3}$$

**Lemma 5:** For $0 ≤ p, q ≤ 1$ we have

$$f(Z(BSC(p)), Z(BSC(q))) = Z(BSC(p) ∅ BSC(q)).$$

**Proof:** The left-hand side is

$$f(2\sqrt{pq}, 2\sqrt{pq}) = \sqrt{4pq + 16pq - 16pq}$$

$$= 2\sqrt{(pq + q^2)} + (p + p^2q^2 - 4pq)q$$

$$= 2p^2q + p^2q$$

$$= Z(BSC(p), q).$$

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 6:** \( f(x, y) \) as defined in formula (3) 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 \( 0 < y \leq 1 \). 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 finishes the proof \( \blacksquare \)

**Theorem 7:** For \( V, W \in \text{BMS} \) we have

\[
Z(V \| W) \geq f(Z(V), Z(W)).
\]

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 \( V \parallel W \) has BSC-decomposition \( \sum_{jk} \alpha_j \beta_k V_j \parallel W_k \). According to the decomposition, \( Z(V \parallel W) \) is

\[
\sum_{jk} \alpha_j \beta_k Z(V_j \parallel W_k) = \sum_{jk} \alpha_j \beta_k f(Z(V_j), Z(W_k)).
\]

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

\[
Z(V \parallel W) = E f(X, Y) \geq f(E X, E Y) = f(Z(V), Z(W)).
\]

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

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

**Corollary 8:** For any \( V, W \in \text{BMS} \), we have

\[
Z(V \parallel W) \leq Z(\text{BEC}(Z(V)) \parallel \text{BEC}(Z(W))).
\]

Equality holds when \( V \) and \( W \) are BECs.

**Proof:** Continue the notation from the proof of Theorem 7. Now we vary the random variables \( X \) and \( Y \) but fix their expectations. Then \( E f(X, Y) \) varies while \( f(E X, E Y) \) remains unchanged. By Karamata’s inequality, a corollary of Jensen’s inequality, \( E f(X, E Y) \) becomes larger when \( X \) and \( Y \) becomes more majorized. The most majorized random variables taking values in \([0, 1]\) are those that can only be \( 0 \) or \( 1 \). Those correspond to the BSC-decompositions of BECs, because BECs consist of BSC(0) (with Bhattacharyya parameter 0) and BSC(1/2) (with Bhattacharyya parameter 1). Therefore, when \( Z(V) \) and \( Z(W) \) are fixed, \( Z(V \parallel W) \) is maximized when \( V \) and \( W \) are BECs. This finishes the proof. \( \blacksquare \)

The following corollary summarizes Theorem 4 and Theorem 7.

**Corollary 9:** For any BMS channel \( W \) with \( z = Z(W) \),

\[
\begin{align*}
Z(W^\circ) &= z^2, \\
Z(W^\bullet) &\leq 2z - z^2,
\end{align*}
\]

The first inequality assumes equality if \( W \) is a BSC. The second inequality assumes equality if \( W \) is a BEC.

**C. Eigenfunction and Eigenvalue**

According to Definition 2, the scaling exponent \( \mu \) is associated to the optimal choice of function \( h \). If we choose an arbitrary, suboptimal \( h \), supremum (1) will give rise to an upper bound on the scaling exponent

\[
\sup_{W \in \text{BMS}^*} \frac{h(W^\circ) + h(W^\bullet)}{2h(W)} \geq 2^{-1/\mu}.
\]

But where to find a good \( h \) to obtain any explicit bound? A simplification used in [11] is to let \( h \) depend on \( Z(W) \), and hence the problem becomes finding a good function \( h : [0, 1] \rightarrow \mathbb{R} \) such that

\[
\sup_{W \in \text{BMS}^*} \frac{h(Z(W^\circ)) + h(Z(W^\bullet))}{2h(Z(W))} = \lambda
\]

is as small as we can make it be.

Since we know \( Z(W^\circ) = Z(W)^2 \) and we know how to bound \( Z(W^\bullet) \) using \( Z(W) \) (Corollary 9), supremum (4) becomes an even simpler expression:

\[
\lambda := \sup_{0 < x < 1} \sup_{x \sqrt{2-x^2} \leq y \leq 2x-x^2} \frac{h(x^2) + h(y)}{2h(x)}.
\]

As an example, \( h(x) := x^{0.78}(1-x)^{0.78}(2x^2 + 3) \) leads to a supremum of \( \lambda \leq 0.87 \) and an upper bound of \( \mu \leq -\log_2(\lambda)^{-1} \leq 4.98 \). The function \( h \) is plotted in Figure 6. The terms in the numerator is plotted in Figure 7.

**D. Power Iteration**

To obtain a good function \( h \) that minimizes supremum (5)—and thereby minimizing the overestimate of \( \mu \)—consider the
following inductive assignment:

\[ h_0(x) := x^{0.78}(1 - x)^{0.78}(2x^2 + 3), \]
\[ h_{i+1}(x) := \sup_{x\sqrt{2-x^2} \leq y \leq 2x-x^2} \frac{h_i(x^2) + h_i(y)}{2 \max h_i}. \]

This is very similar to power iteration, an algorithm that approximates the eigenvalue with the largest modulus of a square matrix. For this reason \( h \) is analogously called the eigenfunction. And quotients of the form \( (h(\text{child}) + h(\text{child}))/2h(\text{parent}) \) are called the eigenvalues.

It is unlikely that \( h_i \) has a simple algebraic formula for large \( i \). In order to handle \( h_i \), [11]'s idea is to use linear interpolation: Put several nodes on \([0, 1]\)

\[ L := \left\{ 0, \frac{1}{\ell}, \ldots, \frac{\ell - 1}{\ell}, 1 \right\} \]

and let \( H \in \mathbb{R}^{\ell+1} \) be an array parametrized by \( L \). Now we can use Linear_Interp(\( L, H \)) as a substitute of \( h \) both during the power iteration and when we want to estimate \( \mu \).

One then asks a computer to execute the following power iteration program.

\[
\begin{align*}
\text{For all } x \in L: \\
H[x] &\leftarrow x^{0.78}(1 - x)^{0.78}(2x^2 + 3); \\
\text{Loop until } H \text{ converges: } \\
&h \leftarrow \text{Linear_Interp}(L, H); \\
&\text{For all } x \in L: \\
H'[x] &\leftarrow \frac{h(x^2) + h(y(H, x))}{2 \max H}; \\
H &\leftarrow H';
\end{align*}
\]

In the program,

- \( h \): \([0, 1] \rightarrow \mathbb{R} \) 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 \( x\sqrt{2-x^2} \leq y \leq 2x-x^2 \).

We remark that there is an easy implementation of \( y(H, x) \):

\[
y(H, x) := \begin{cases} 
  x\sqrt{2-x^2} & \text{if } x\sqrt{2-x^2} \geq \arg \max(H), \\
  2x-x^2 & \text{if } 2x-x^2 \leq \arg \max(H), \\
  \arg \max(H) & \text{otherwise}.
\end{cases}
\]

This implementation is correct if \( H \) is unimodal. This might not be the case halfway through the power iteration; that is, \( H \) might not be unimodal and \( y(x, H) \) might fail to maximize \( h(y) \). But it deals no damage; as long as \( H \) converges and the limit is a unimodal array, the implementation of \( y(x, H) \) will eventually be correct.

Running the program, we found that \( H \) converges reasonably fast. In 100 iterations, \( H \) and \( H' \) will differ by \( 10^{-9} \). About 200 iterations is enough to make \( H \) and \( H' \) differ by \( 10^{-15} \).

Now that we know \( H \) converges, let \( \hat{H} \) be the limit of \( H \) and let \( \hat{h} \) be Linear_Interp(\( L, \hat{H} \)). An empirical upper bound on \( \mu \) is obtained by

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

Per our computation, \( \ell = 2 \cdot 10^5 \) gives the first four digits \( (4.695) \) mentioned in [11] (wherein \( \ell = 10^6 \)).

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

\[
L := \left\{ \frac{1 - \cos(\theta)}{2} \mid \theta = 0, \frac{1}{\ell} \pi, \ldots, \frac{\ell - 1}{\ell} \pi, \pi \right\}.
\]

The motivation behind Chebyshev nodes is that they pay more attention 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 nodes. Later, we will use \( \tanh(\theta) \) as nodes because \( \tanh \) pays even more attention to the two ends.

**E. Foot of the Mountain—Problems With Interpolation**

Having an array \( \hat{H} \) of evaluations, one would ask if \( \hat{h} := \text{Linear_Interp}(L, \hat{H}) \) is a proper substitute of the eigenfunction in the manner of whether

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

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 \( (\hat{h}(x^2) + \hat{h}(2x - x^2))/2h(x) \) is constantly 1 (whereas we want the quotient to be strictly less than 1).

In [11, 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 \( \delta \) be a tiny number; and let \( \hat{h}(x) \) be \( x^{0.78} \) when \( x \leq \delta \) and be \((1 - x)^{0.78} \)

---

3For comparison, IEEE 754's double-precision floating-point format has 53 significant bits; the smallest number greater than 1 is \( 1 + 2^{22} \cdot 10^{-16} \).
when \( x \geq 1 - \delta \). This way, the quotients for \( x \in (0, \delta) \) and for \( x \in (1 - \delta, 1) \) are uniformly bounded from above. For \( x \in [\delta, 1 - \delta] \), 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 nodes.

This type of function surgery is limited to the tiny neighborhood \([0, \delta] \) of 0 and the tiny neighborhood \([1 - \delta, 1] \) of 1. Hence it shall not affect the eigenvalue too much. An evidence that support this is that while the number obtained by formula (7) is 4.695, the rigorous value reported in [11] is 4.714. These two numbers are only 0.4% apart.

While the 0.4% penalty is considered impressive, we slightly improve the surgery part and propose the following theorem.

**Theorem 10:** Let \( \alpha, \beta_0, \beta_1, \gamma_0, \gamma_1, \delta, \) and \( \mu \) be positive numbers. Let \( L \subseteq [\delta^2, 1 - \delta^2] \) be a set of nodes that contains \( \delta^2, \delta, 1 - \delta, \) and \( 1 - \delta^2 \). Let \( H \) be an array of positive numbers parametrized by \( L \). Let \( h := \text{Linear Interp}(L, H) \) be the linear interpolant. Suppose that \( H \) is unimodal, and its mode is at \( \alpha \in [2\delta, 1 - 2\delta] \). Suppose that \( 2^{1 - 1/\mu} \cdot h(\delta^2) \geq h(2\delta^2 - \delta^4) \) and \( 2^{1 - 1/\mu} \cdot h(1 - \delta^2) \geq h((1 - \delta)^2) \). For any pair of neighboring nodes \( u, v \in L \) (by neighboring nodes we mean \([u, v] \cap L = \{u, v\} \)), define

\[
y_0(H, u, v) := \begin{cases} 
  u^2 & \text{if } u^2 \geq \alpha, \\
  v^2 & \text{if } v^2 \leq \alpha, \\
  \alpha & \text{otherwise},
\end{cases}
\]

and

\[
y_1(H, u, v) := \begin{cases} 
  u\sqrt{2 - u^2} & \text{if } u\sqrt{2 - u^2} \geq \alpha, \\
  2v - v^2 & \text{if } 2v - v^2 \leq \alpha, \\
  \alpha & \text{otherwise}.
\end{cases}
\]

Suppose that, for any pair of neighboring nodes \( u, v \in L \cap [\delta^2, \delta] \),

\[
h(u) \geq \beta_0 v^{1 - 1/\mu},
\]

\[
h(v) \leq \gamma_0 u^{1 - 1/\mu},
\]

\[
y_0(u^2) \geq 1 - \frac{(2\gamma_0 u^{1 - 1/\mu})}{h(u)} \leq 2^{-1/\mu},
\]

Suppose that, for any pair of neighboring nodes \( u, v \in L \cap [\delta, 1 - \delta] \),

\[
\frac{h(y_0(H, u, v)) + h(y_1(H, u, v))}{2 \min\{h(u), h(v)\}} \leq 2^{-1/\mu},
\]

Suppose that, for any pair of neighboring nodes \( u, v \in L \cap [1 - \delta, 1 - \delta^2] \),

\[
h(v) \geq \beta_1 (1 - u)^{1 - 1/\mu},
\]

\[
h(u) \leq \gamma_1 (1 - v)^{1 - 1/\mu},
\]

\[
\frac{h(u^2) + (2v - u^2)^{1 - 1/\mu}}{2h(v)} \leq 2^{-1/\mu}.
\]

Then there exists a function \( \hat{h} : [0, 1] \to \mathbb{R} \) such that

\[
\sup_{0 < x < 1} \sup_{x^2 - \frac{1}{2} \leq y \leq x^2 - \frac{\delta^2}{2}} \frac{h(x^2) + h(y)}{2h(x)} \leq 2^{-1/\mu} + \max\left\{\frac{\gamma_0}{\beta_0}, \frac{\gamma_1}{\beta_1}\right\} \delta^2 - 2/\mu.
\]

The proof of Theorem 10 is presented in Appendix A. With this theorem we can rigorously prove \( \mu \leq 4.69531 \) with the following setting:

- \( (\beta_0, \gamma_0, \beta_1, \gamma_1, \delta) = (2.520, 2.532, 4.312, 4.325, 10^{-5}) \),
- \( 10^6 \) nodes in \([\delta^2, \delta] \) such that \( \{\text{artanh}(2x - 1) \mid x \in L \cap [\delta^2, \delta]\} \) are evenly spaced,
- \( 2 \cdot 10^6 \) nodes in \([\delta, 1 - \delta] \) such that \( \{\text{artanh}(2x - 1) \mid x \in L \cap [\delta, 1 - \delta]\} \) are evenly spaced, and
- \( 10^6 \) nodes in \([1 - \delta, 1 - \delta^2] \) such that \( \{\text{artanh}(2x - 1) \mid x \in L \cap [1 - \delta, 1 - \delta^2]\} \) are evenly spaced.

The value obtained from formula (7) is 4.69515. The relative difference is about 0.003%.

Here is why Theorem 10 can improve the number 4.714 reported in [11, Section III.C]: when applying the old surgery, the eigenfunction \( h(x) \) is forced to be \( \sim x^{0.78} \) for \( x \in [0, \delta] \). On the other hand, our approach allows \( h \) to be oscillating between \( \beta_0 x^{1 - 1/\mu} \) and \( \gamma_0 x^{1 - 1/\mu} \). As it turns out, the power iteration will naturally give rise to an \( h \) that does oscillate when \( x \) is close to 0 or 1, hence the necessity of such relaxation.

While the improved numeral 4.69564 already justifies the title of this article, we present in subsequent sections information-theoretical techniques that lower this further.

**IV. Road Map to a Better Bound**

At the end of the previous section, we see that careful manipulation of the eigenfunction \( h \) has the ability to tighten the upper bound on \( \mu \). However, those are “numerical” improvements. In this section, we present our blueprint for “conceptual” improvements.

**A. Check If All Inequalities Are Tight**

In the old proof of \( \mu \leq 4.714 \), one can see that supremum (5) lets \( y \) range over an interval \([x\sqrt{2 - x^2}, 2x - x^2]\). The right endpoint of this interval is tight because \( W \) might be a BEC. That is, \( Z(V^{\infty}) \) does coincide with \( Z(W) \cdot Z(W)^2 \) when \( W \) is a BEC. We also know that, if \( W \) is a BEC, all its descendants are BECs. Hence the right endpoint is always tight.

The left endpoint of said interval, at first glance, is also tight because \( W \) might be a BSC. Indeed, Corollary 9 tells us that when \( W \) is a BSC, \( Z(W^{\infty}) \) does coincide with \( Z(W)^{\sqrt{2} - Z(W)^2} \). However, the left endpoint is only tight “for now”. After one parallel combination, \( W^{\infty} \) will no longer be a BSC, and since \( W^{\infty} \) is not BSC, the inequality \( x\sqrt{2 - x^2} \leq y \) is not tight when we plug in \((x, y) = (Z(W^\infty), Z(W^\infty^*))\). There is always a tiny gap between \( Z(W^{\infty} \parallel W) \) and \( Z(W^{\infty})^{\sqrt{2} - Z(W^{\infty})^2} \).

**B. Found an Untight Inequality: What Happen If It Is Tighter?**

Suppose that there is a tighter lower bound \( Z(W^{\infty}) \geq \gamma(Z(W^{\infty})) \), where \( \gamma \) is some function such that \( \gamma(x) > x\sqrt{2 - x^2} \) for \( x \in (0, 1) \), then supremum (5) can be updated to

\[
\sup_{0 < x < 1} \sup_{\gamma(x) \leq y \leq x^2 - \frac{\delta^2}{2}} \frac{h(x^2) + h(y)}{2h(x)}.
\]
This time, the inner supremum is taken over a smaller interval \([\gamma(x), 2x - x^2]\). Just like that \(\max\{A, B, C\}\) is never less than \(\max\{A, B\}\), a supremum with a smaller domain could, in principle, be greater, at least when we are considering channels \(W^\circ\) that are themselves parallel combinations.

Suppose that the supremum over a smaller interval does lead to a smaller eigenvalue. It will give us a smaller upper bound on \(\mu\), updating the recode after seven years. At this point, the concern is not whether there will be an improvement but how much the improvement will be. Our claim is that we can lower 4.714 to 4.63. Even if we use the more precise value 4.695 as the baseline, the reduction from 4.695 to 4.63 is still moderately big (about 1.4%), given that the final destination is somewhere around 4.

### C. How to Tighten?

Following the argument above, it all boils down to how we can describe a function \(\gamma(x)\) such that \(Z(W^\circ_\gamma) \geq \gamma(Z(W^\circ))\). A strategy is that, since \(Z(W^\circ) = Z(W)^2\) is an equality that always holds, it suffices to find the relation between \(Z(W)\) and \(Z(W^\circ_\gamma)\). That is, it suffices to find a relation between \(Z(W)\) and \(Z(W^\circ) \oplus (W \ominus W)\).

In the old proof of \(\mu \leq 4.714\), the relation between \(W\) and \(W^\circ_\gamma\) is found by generalizing the problem to finding the relation between \(U, V, U \ominus V\). So here, one would naturally ask if we can preform similar tricks to \(U, V, X, Y\) and \((U \ominus V) \oplus (X \ominus Y)\). However, we deem that characterizing a quad-variate channel transformation is too large of a leap. So instead, we will look at the tri-variate channel transformation \((U \ominus V) \ast W\).

Once we are able prove that \(Z((U \ominus V) \cup W) \geq g(Z(U, Z(V)), Z(W))\) for some function \(g\), we can plug in \((U, V, W) = (V, W, V^\circ)\) for a new relation \(Z(V^\circ_\gamma) = Z(V^\circ) \ominus V^\circ = Z(V, W, V^\circ) = g(Z(V), Z(W))\). That is to say, \(g(\sqrt{x}, \sqrt{y}, x)\) will be the \(\gamma(x)\) that we want.

Since we expect that \(BSC\) is the most extremal case, we will begin the next section with \(g(Z(U), Z(V), Z(W))\) defined as \(Z((U \ominus V) \cup W)\) whenever \(U, V, W\) are BSCs. And then we attempt to prove that \(g\) is a lower bound for non-BSCs.

### V. TRI-VARIATE CHANNEL TRANSFORMATION

Consider the channel combination \((U \ominus V) \ast W\). See Figure 8 for a visualization. Define function \(g: [0, 1]^3 \rightarrow [0, 1]\) that satisfies

\[
g(Z(U), Z(V), Z(W)) = Z((U \ominus V) \ast W)
\]

for all \(U, V, W\) that are BSCs. We want write \(g\) more explicitly with the help of some lemmas.

#### A. Tri-Variate Bhattacharyya Function

**Lemma 11** (Trivariate Z): \((BSC(p) \oplus BSC(q)) \ast BSC(r)\) has Bhattacharyya parameter

\[
2\sqrt{(pq\bar{r} + p\bar{q}r)(p\bar{q}r + \bar{p}q\bar{r})} + 2\sqrt{(pq\bar{r} + \bar{p}q\bar{r})(p\bar{q}r + \bar{p}q\bar{r})}
\]

**Proof:** \(BSC(p) \oplus BSC(q)\) is, by definition, \((p * q) \cdot BSC(p) \oplus BSC(q)\). When this channel is serially-combined with a \(BSC(r)\), the first summand becomes

\[
(p * q) BSC\left(\frac{p\bar{q}r}{p \ast q} \ast r\right) = (p * q) BSC\left(\frac{p\bar{q}r + p\bar{q}\bar{r}}{p \ast q}\right)
\]

and contributes Bhattacharyya parameter

\[
2(p * q) \sqrt{pq\bar{r} + p\bar{q}r} = 2\sqrt{(pq\bar{r} + p\bar{q}r)(p\bar{q}r + \bar{p}q\bar{r})}
\]

The second summand becomes

\[
\frac{p \ast q}{p \ast q} BSC\left(\frac{p\bar{q}r + \bar{p}q\bar{r}}{p \ast q} \ast r\right) = \frac{p \ast q}{p \ast q} BSC\left(\frac{p\bar{q}r + \bar{p}q\bar{r}}{p \ast q}\right)
\]

and contributes Bhattacharyya parameter

\[
2\sqrt{(pq\bar{r} + p\bar{q}r)(p\bar{q}r + \bar{p}q\bar{r})}
\]

This finishes the proof. \(\square\)

**Lemma 12** (Z in terms of Z’s): \(g\) as defined in formula (9) satisfies

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

where

\[
C := \frac{1}{4}(x^2 + y^2 + z^2),
\]

\[
D := \frac{1}{2}\sqrt{x^2 + y^2 + z^2}.
\]

**Proof:** Let \(x, y, z\) be two \(\sqrt{pp}, \sqrt{qq}\), and \(\sqrt{rr}\), respectively, for some \(0 \leq p, q, r \leq 1/2\). From Lemma 11, \(g(x, y, z) = 2\sqrt{A + 2\sqrt{B}}\), where

\[
A := (pq\bar{r} + p\bar{q}r)(pq\bar{r} + \bar{p}q\bar{r})
\]

\[
= pq\bar{r}pqr + pq\bar{r}\bar{p}qr + \bar{p}q\bar{r}pqr + \bar{p}q\bar{r}\bar{p}q\bar{r}
\]

\[
= p^2q^2r^2 + pp\bar{q}\bar{r}q^2 + pp\bar{q}\bar{r}q^2 + p^2q^2r^2
\]

\[
= pp\bar{q}\bar{r}q^2 + (p^2q^2 + p^2q^2)r^2
\]

and

\[
B := (pq\bar{r} + p\bar{q}r)(pq\bar{r} + \bar{p}q\bar{r})
\]

\[
= pq\bar{r}pqr + pq\bar{r}\bar{p}qr + \bar{p}q\bar{r}pqr + \bar{p}q\bar{r}\bar{p}q\bar{r}
\]

\[
= p^2q^2r^2 + pp\bar{q}\bar{r}q^2 + pp\bar{q}\bar{r}q^2 + p^2q^2r^2
\]

\[
= pp\bar{q}\bar{r}q^2 + (p^2q^2 + p^2q^2)r^2
\]

To show \(C + D = 4A\) and \(C - D = 4B\), it suffices to show

\[
2(A + B) = C \land 2(A - B) = D.
\]

For the former,

\[
2(A + B) = 2\left(\frac{pq\bar{q}q^2 + q^2}{2} + \frac{(p^2q^2 + p^2q^2)r^2}{2} + pp\bar{q}\bar{r}q^2 + pp\bar{q}\bar{r}q^2 + p^2q^2r^2\right)
\]

\[
= 4pp\bar{q}\bar{r}q^2 + 2(q^2 + q^2)(q^2 + q^2)r^2
\]

\[
= \frac{1}{4}x^2y^2 + \frac{1}{2}(x^2)(1) = \frac{1}{4}\left(1\right)^2 = C.
\]
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,
\[
2(A - B) = 2 \left( p\bar{p}q\bar{q}(r^2 + \bar{r}^2) + (p^2q^2 + \bar{p}^2\bar{q}^2)r\bar{r} \right)
\]
\[
= 2(p^2 - p^2)(q^2 - q^2)r\bar{r}
\]
\[
= \frac{1}{2}\sqrt{1 - x^2} \cdot \sqrt{1 - y^2} \cdot z^2
\]
\[= D.\]

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 $\sqrt{4A + 4\bar{A}} = \sqrt{C + D} = \sqrt{(\sqrt{C + D} + \sqrt{C - D})^2} = \sqrt{2C + 2\sqrt{C^2 - D^2}}$. This finishes the proof. \hfill \Box

A tri-variation function is said to be tri-convex 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 $Z((U \oplus V) \sqsupset W)$ is lower bounded by $g(Z(U), Z(V), Z(W))$ by the same Jensen-argument as in Theorem 7. Unfortunately, $g$ is not tri-convex. The next subsection will find a workaround to this.

### B. Lower Tri-Convex Envelope

$g(x, y, z)$ as defined above is not convex in any of the three variables. We thus attempt to find a lower bound on $g$ that is tri-convex so that Jensen’s inequality applies. Consider a function $\tilde{g}: \{0, 1\}^3 \to [0, 1]$ that reads
\[
\tilde{g}(x, y, z) := \sup \left\{ \theta(x, y, z) \mid \theta \text{ is tri-convex and } \theta \leq g \text{ pointwise} \right\},
\]
where the supremum runs over all functions $\theta: \{0, 1\}^3 \to [0, 1]$ that are tri-convex and bounding $g$ from below. This is very similar to the definition of the lower convex envelope, the difference being that $\theta$ is not convex but tri-convex. (An example is that $xyz$ is tri-convex but not convex.) We will refer to $\tilde{g}$ as the envelope of $g$.

**Theorem 13 (Counterpart of Theorem 7):** For $U, V, W \in BMS$,
\[
Z((U \oplus V) \sqsupset W) \geq \tilde{g}(Z(U), Z(V), Z(W)).
\]

In particular, if $W = V^\ominus$ for some $V \in BMS$,
\[
Z(W^\ominus) \geq \tilde{g}(\sqrt{Z(W)}, \sqrt{Z(W)}, Z(W)).
\]

**Proof:** For the first statement, it suffices to prove $Z((U \oplus V) \sqsupset W) \geq \theta(Z(U), Z(V), Z(W))$ for all tri-convex $\theta$ that is also $\leq g$ pointwise. Fix a $\theta$. When $U, V, W$ are BSCs, the inequality we want to prove holds:
\[
Z((U \oplus V) \sqsupset W) = g(Z(U), Z(V), Z(W)) \geq \theta(Z(U), Z(V), Z(W)).
\]

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 $(U \oplus V) \sqsupset W$ becomes $\sum_{ijk} \alpha_i \beta_j \gamma_k (U_i \oplus V_j) \sqsupset W_k$, thereby having Bhattacharyya parameter
\[
Z((U \oplus V) \sqsupset W) = \sum_{ijk} \alpha_i \beta_j \gamma_k Z((U_i \oplus V_j) \sqsupset W_k)
\]
\[\geq \sum_{ijk} \alpha_i \beta_j \gamma_k \theta(Z(U_i), Z(V_j), Z(W_k))\]
\[\geq \sum_{ij} \alpha_i \beta_j \theta(Z(U_i), Z(V_j), Z(W))\]
\[\geq \sum_i \alpha_i \theta(Z(U_i), Z(V), Z(W))\]
\[\geq \theta(Z(U), Z(V), Z(W)).\]

This finishes the proof of the lower bound on $Z((U \oplus V) \sqsupset W)$. For the lower bound on $Z(W^\ominus)$, plug in $(U, V, W) = (V, V, V^\ominus)$ and use the fact that $Z(W) = Z(V)^2$. \hfill \Box

### C. Approximate the Envelope $\tilde{g}$

Computing the envelope $\tilde{g}$ algebraically does not seem plausible nor possible. Our approach is to approximate $\tilde{g}$ numerically over a mesh
\[
M := \left\{ \frac{1}{n}, \ldots, \frac{n - 1}{n}, 1 \right\}^3 \subseteq [0, 1]^3.
\]

Here, $n$ is the resolution; say $n = 300$. 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 \not\in \{0, 1\}:
\[
2G[x, y, z] \leq G\left[x - \frac{1}{n}, y, z\right] + G\left[x + \frac{1}{n}, y, z\right], \tag{10}
\]
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
\[
G[x, y, z] \leftarrow \frac{1}{2}G\left[x - \frac{1}{n}, y, z\right] + \frac{1}{2}G\left[x + \frac{1}{n}, y, z\right]. \tag{11}
\]
We also demand the convexity in the \(y\)-direction and the \(z\)-direction:
\[
2G[x, y, z] \leq G\left[x, y - \frac{1}{n}, z\right] + G\left[x, y + \frac{1}{n}, z\right], \tag{12}
\]
\[
2G[x, y, z] \leq G\left[x, y, z - \frac{1}{n}\right] + G\left[x, y, z + \frac{1}{n}\right]. \tag{13}
\]
If not, we update \(G[x, y, z]\) with a formula similar to formula \((11)\).

We synthesize a program that consistently 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 \text{Tri\_Convexify}:

For all \((x, y, z) \in M\):
\[
G[x, y, z] \leftarrow g(x, y, z);
\]
Loop until \(G\) converges:
For all \((x, y, z) \in M\):
If criterion \((10)\) fails:
\[
\text{Update via formula } (11);
\]
If criterion \((12)\) fails:
\[
\text{Update similarly};
\]
If criterion \((13)\) fails:
\[
\text{Update similarly};
\]
The program will stop when all three criteria are met modulo rounding errors. Empirically, \(G\) converges. Mathematically, we can also prove that \(G\) converges for all \(n \geq 2\). This means that we can always go for a larger \(n\) for a better approximation, still the convexify procedure will always converge.

**Proposition 14:** \text{Tri\_Convexify}(\(G\)) makes \(G\) converge. For any mesh point \((x, y, z) \in M\), the data point \(G[x, y, z]\) converges to
\[
G[x, y, z] := \sup \left\{ \Theta[x, y, z] \mid \Theta \text{ is tri-convex and } \Theta \leq G \text{ pointwise} \right\}.
\]
The supremum is over all arrays \(\Theta \in \mathbb{R}^{(n+1) \times (n+1) \times (n+1)}\) that satisfy the discrete convexity criteria \((10)\), \((12)\), and \((13)\) and \(\Theta \leq G\) 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 \(\hat{G}\) 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 \(\hat{\Theta} \leq G\) remains to be a lower bound on \(G\) after an update of \(G\). So any such \(\hat{\Theta}\) maintains to be a lower bound on \(\hat{G}\). This means that \(\hat{G}\) is greater than or equal to the supremum of all such \(\Theta\)'s. But \(\hat{G}\) is itself a tri-convex lower bound of \(G\) so \(\hat{G}\) is equal to the supremum; the supremum is a maximum.

Hereafter, let \(\hat{G}\) denote both the empirical end result of \text{Tri\_Convexify}(\(G\)) and the supremum defined in Proposition 14. We call \(\hat{G}\) the discrete envelop in contrast to the “continuous” envelop \(\hat{g}\).

**Lemma 15:** \text{Linear\_Interp}(\(M, \hat{G}\)) is tri-convex if the data points \(\hat{G}\) satisfy the discrete convexity criteria \((10)\), \((12)\), and \((13)\).

Here, \text{Linear\_Interp}(\(M, \hat{G}\)) : \([0, 1]^3 \rightarrow \mathbb{R}\) is a function that evaluates to \(\hat{G}[x, y, z]\) at \((x, y, z) \in M\), and is tri-linearly interpolated if \((x, y, z) \notin M\). A defining feature of multi-linear interpolation is that it is piecewise linear in any cardinal direction. We use this feature in the following proof.

**Proof of Lemma 15:** We shall prove an analog of the lemma for a two dimensional \(2 \times 3\) grid; the general statement follows by a generalization of the argument.

Let there be six numbers on a grid
\[
a \rightarrow b \rightarrow c \rightarrow d \rightarrow e \rightarrow f
\]
such that \(a + c \geq 2b\) and \(d + f \geq 2e\), i.e., the data points are convex. Let \(\hat{g}\) be obtained by bi-linear interpolation such that
\[
\hat{g}(1, 1) = \hat{g}(0, 1) = \hat{g}(1, 0) = \hat{g}(0, 0) = \hat{g}(1, 0)
\]
corresponds to grid \((14)\).

We claim that \(\hat{g}\) is convex at \((0, 0)\) in the \(x\)-direction, that is, \(\hat{g}(\varepsilon, 0) + \hat{g}(\varepsilon, 0) \geq 2\hat{g}(0, 0)\) for \(0 \leq \varepsilon \leq 1\). This is because
\[
\hat{g}(\varepsilon, 0) + \hat{g}(\varepsilon, 0) \geq \hat{g}(0, 0) + \hat{g}(0, 0) = 2\hat{g}(0, 0)\]
for \(0 \leq \varepsilon \leq 1\). This is because
\[
\hat{g}(\varepsilon, 0) + \hat{g}(\varepsilon, 0) \geq \hat{g}(0, 0) + \hat{g}(0, 0) = 2\hat{g}(0, 0).
\]
Similarly, \(\hat{g}\) is convex at \((0, 1)\) in the \(x\) direction, that is, \(\hat{g}(\varepsilon, 1) + \hat{g}(\varepsilon, 1) \geq 2\hat{g}(0, 1)\). This shows that \(\hat{g}\) is convex at the vertices of the grid.

Next, we claim that \(\hat{g}\) is convex at \((0, y)\), where \(0 \leq y \leq 1\), in the \(x\)-direction. That is to say, \(\hat{g}(\xi, y) + \hat{g}(\xi, y) \geq 2\hat{g}(0, y)\) for \(0 \leq \xi \leq 1\). This is because
\[
\hat{g}(\varepsilon, 0) + \hat{g}(\varepsilon, 0) \geq \hat{g}(0, 0) + \hat{g}(0, 0) = 2\hat{g}(0, 0).
\]
This shows that the convexity on the lines of the grid follows from the convexity of the data points.

For convexity within the squares, it trivially holds because the value within a square is by definition bi-linear. Hence the lemma is sound for this particular two-dimensional grid. For the same reason, the lemma holds over the three-dimensional array \(\hat{G}\).

We believe that \text{Linear\_Interp}(\(M, \hat{G}\)) \(\approx \hat{g}\). That is, the tri-linear interpolant of the discrete envelop is a good approximation of the continuous envelop. Together with Theorem 13, we can now compute a lower bound on \(Z(W^{\infty})\) (up to some interpolation errors) using a concrete object \(\hat{G}\) instead of an abstract object \(\hat{g}\).
Here is the motivation of this indirect setup: in Section III, $h(Z(W))$ 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 giving its children scores and summing them. And we want to score the parallel child and the serial child using two different functions $\varphi_\circ$ and $\varphi_\otimes$. As we will see later, $\varphi_\otimes(x)$ is greater than or equal to $\varphi_\circ(x)$ for all $x$. This means that, even if $U^*$ and $V^\circ$ have the same Bhattacharyya parameter, we might still give $V^\circ$ a lower score—because we think that a parallel combination is more polarized.

There is a reason to distinguish serial combination from parallel combination. Comparing Theorem 13 with Theorem 7, we see that parallel combination assumes better bounds on Bhattacharyya parameters. This implies that the domain of supremum (5) 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 $W \in \mathcal{BMS}$:

$$\frac{\psi(W^*) + \psi(W^\circ)}{2\psi(W)} = \frac{\psi_\delta(W^* \Delta) + \psi_\delta(W^\circ \Delta) + \psi_\lambda(W^* \Delta) + \psi_\lambda(W^\circ \Delta)}{2\psi_\delta(W^*) + 2\psi_\delta(W^\circ)}.$$  

Hence it suffices to bound

$$\max\left\{ \frac{\psi_\delta(W^* \Delta) + \psi_\delta(W^\circ \Delta)}{2\psi_\delta(W^*)}, \frac{\psi_\lambda(W^* \Delta) + \psi_\lambda(W^\circ \Delta)}{2\psi_\lambda(W^\circ)} \right\}$$  

from above. This is because, as far as only positive real numbers are concerned,

$$\min\left\{ \frac{A+B}{E}, \frac{C+D}{F} \right\} \leq \frac{A+B+C+D}{E+F} \leq \max\left\{ \frac{A+B}{E}, \frac{C+D}{F} \right\}$$

because a weighted average always lies between the minimum data point and the maximum data point. Moreover, we can also infer that in order to get the tightest possible bound, $(A+B)/E$ and $(C+D)/F$ probably need be as close to each other as possible. This sums up why we can use formula (16) to upper bound formula (15).

B. A Deterministic Automaton

But why do we want to use (16) to upper bound (15)? As it turns out, there is an automaton structure behind (16) that can help us formulate a power iteration strategy that finds the best functions $\varphi_\Delta$ and $\varphi_\circ$.

Let $\mathcal{A}$ be an automaton whose state is a pair of a channel $W \in \mathcal{BMS}$ and a combination $C \in \{\Delta, \otimes\}$. The accepted inputs are $\psi_\sharp$ and $\otimes$. If we input $\Delta$, $\mathcal{A}$ updates its state to $(W^\Delta, \sharp)$ and outputs $\psi_\Delta(W^\sharp)$. Likewise, if we input $\Delta$, $\mathcal{A}$ updates its state to $(W^\Delta, \otimes)$ and outputs $\psi_\Delta(W^\otimes)/\psi(C(W))$. Now, suppose that $\mathcal{A}$ begins with state $(W, C)$ and we input $C_1, C_2, \ldots, C_k$, then it will output a series of numbers

$$\frac{\psi(C_1(W^C_1))}{\psi(C(W))}, \frac{\psi(C_2(W^C_2))}{\psi(C_1(W))}, \ldots, \frac{\psi(C_k(W^C_1 \cdots C_k))}{\psi(C_{k-1}(W^C_1 \cdots C_{k-1}))}.$$  

The product of these numbers is $\psi(C_k(W^C_1 \cdots C_k))/\psi(C(W))$. 

---

A. A Biased Scoring Function

To begin, suppose that there are two unimodal continuous functions $\varphi_\sharp, \varphi_\otimes : [0, 1] \rightarrow \mathbb{R}$ that satisfy $\varphi_\Delta(0) = \varphi_\Delta(1) = \varphi_\otimes(0) = \varphi_\otimes(1) = 0$ but are positive elsewhere. Define shorthands $\psi_\lambda, \psi_\delta, \psi : \mathcal{BMS} \rightarrow \mathbb{R}$ by

$$\psi_\lambda(W) := \varphi_\otimes(Z(W)),$$
$$\psi_\delta(W) := \varphi_\Delta(Z(W)),$$
$$\psi(W) := \psi_\otimes(W^\otimes) + \psi_\Delta(W^\Delta) := \varphi_\otimes(Z(W^\Delta)) + \varphi_\Delta(Z(W^\Delta)).$$

$\psi$ will be the counterpart of $h$ in our new bound.

That being said, the claimed upper bound on the scaling exponent, $\mu \leq 4.63$, is derived by a rigorous lower bound on $Z(W^\Delta)$. To obtain a rigorous bound, the data points in $G$ are lowered by certain amounts to form $G_\otimes$. This is to make sure that Linear_Interp $(M, G_\otimes) \leq g$ and hence Linear_Interp $(M, G_\otimes) \leq \tilde{g}$ is a mathematically sound bound. Readers are referred to Appendix B for more details.

Bibliographical remark: some of the arguments presented in this section share common elements with [45], in which Witsenhausen talked about using lower convex envelopes of certain functions to bound entropies.

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

VI. FUTURE STATE POWER ITERATION

For this section, recall the lesson that a finite state automaton 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.
From the previous paragraph, we learn that the product of all the numbers output by \( A \) controls the asymptotic behavior of \( W^{E_1 \cdots E_n} \) in the same way Corollary 3 does. If the input \( C_1 \) is a random combination like in Theorem 1, then number output by \( A \) become a random variable whose expectation is

\[
\frac{\psi_\ast (W \Xi) + \psi_\circ (W \circ)}{2\psi_C (W)},
\]

which is controlled by formula (16).

Next, we want to eliminate the dependence of \( A \) on \( W \) to obtain a universal bound over \( BMS^* \). We take the supremum of the first quotient in (16) and simplify it as below:

\[
\sup_{W \in BMS^*} \frac{\psi_\Xi (W^{m \ast}) + \psi_\circ (W^{\ast \circ})}{2 \psi_\ast (W \Xi)} = \sup_{U = W^\ast} \frac{\varphi_\Xi (Z(U^\ast)) + \varphi_\circ (Z(U^\circ))}{2 \varphi_\ast (Z(U))} \leq \sup_{U \in BMS^*} \frac{\varphi_\Xi (Z(U^\Xi)) + \varphi_\circ (Z(U^\circ))}{2 \varphi_\Xi (Z(U))} = \sup_{0 < x < 1} \sup_{f(x) \leq y \leq 2x - x^2} \frac{\varphi_\ast (x^2) + \varphi_\circ (y)}{2 \varphi_\Xi (x)}.
\]

Here, the supremum on the second line is taken over those \( U \) that are themselves serial combinations of some channels \( W \in BMS^* \). We then “forget” that \( U \) is a serial combination and treat \( U \) as an usual BMS channel and apply the classic lower bound (Theorem 7). Because of that, \( y \) ranges over \([f(x), 2x - x^2]\).

Similarly but not identically, we consider the supremum of the second quotient in (16) with \( \psi_\circ \) in the denominator:

\[
\sup_{W \in BMS^*} \frac{\psi_\Xi (W^{m \circ}) + \psi_\circ (W^{\circ \circ})}{2 \psi_\circ (W \circ)} = \sup_{V = W^\circ} \frac{\varphi_\Xi (Z(V^\Xi)) + \varphi_\circ (Z(V^\circ))}{2 \varphi_\circ (Z(V))} \leq \sup_{0 < x < 1} \sup_{\hat{g}(\sqrt{x}, \sqrt{x}, x) \leq z \leq 2x - x^2} \frac{\varphi_\ast (x^2) + \varphi_\circ (z)}{2 \varphi_\circ (x)}.
\]

Here, the supremum on the second line is taken over those \( V \) that are themselves parallel combinations of some channels \( W \in BMS^* \). We invoke Theorem 13 and let \( z \) range over \([\hat{g}(\sqrt{x}, \sqrt{x}, x), 2x - x^2]\). The inner supremum is now running over a strictly smaller region than in the previous work—see Figure 9—so a smaller supremum is expected.

C. A Duplex Power Iteration

It remains to use linear interpolation to represent \( \varphi_\Xi \) and \( \varphi_\circ \), and apply some power iteration machinery to minimize the eigenvalues. To facilitate that, we reverse-engineer \( A \) and come up with the following.

Let \( L \) be defined by formula (8) with, say, \( \ell = 10^6 \). Let \( \Phi_\tau, \Phi_\circ \in \mathbb{R}^{\ell + 1} \) be arrays parametrized by \( L \). We execute the following power iteration program:

For all \( x \in L \):

\[
\Phi_\Xi [x] \leftarrow \varphi_\ast (x^2) + \varphi_\circ (y(\Phi_\Pi, x)) / \max \Phi_\Xi;
\]
\[
\Phi_\circ [x] \leftarrow \varphi_\circ (x^2) + \varphi_\circ (z(\Phi_\Xi, x)) / \max \Phi_\circ;
\]

Loop until \( \Phi_\Xi \) and \( \Phi_\circ \) converge:

\[
\Phi_\ast \leftarrow \text{Linear Interp}(L, \Phi_\Pi);
\]
\[
\Phi_\circ \leftarrow \text{Linear Interp}(L, \Phi_\circ);
\]

For all \( x \in L \):

\[
\Phi_\Pi [x] \leftarrow \varphi_\ast (x^2) + \varphi_\circ (y(\Phi_\Pi, x)) / \max \Phi_\Xi;
\]
\[
\Phi_\circ [x] \leftarrow \varphi_\circ (x^2) + \varphi_\circ (z(\Phi_\Xi, x)) / \max \Phi_\circ;
\]

% Not a typo; the denominators coincide;
\[
\Phi_\ast \leftarrow \Phi_\ast';
\]
\[
\Phi_\circ \leftarrow \Phi_\circ';
\]

In the program,

- \( \Phi_\Xi \) and \( \Phi_\Pi \) are temporary memory spaces that store the updated content for the next round;
- \( y(\Phi_\Pi, x) \) and \( z(\Phi_\Xi, x) \) are meant to be the arguments that maximize \( \varphi_\Pi (y) \) and \( \varphi_\ast (z) \) over the ranges \( f(x, x) \leq y \leq 2x - x^2 \) and \( \hat{g}(\sqrt{x}, \sqrt{x}, x) \leq z \leq 2x - x^2 \), respectively;
- \( \hat{g} \) is \( \text{Linear Interp}(M, \hat{G}) \), which is \( \approx \hat{g} \). If a rigorous lower bound on \( \hat{g} \) is desired, see Appendix B for the recipe of \( \hat{g} \).

We can reuse the implementation of \( y(H, x) \) in formula (6); and implement \( z(H, x) \) as

\[
z(H, x) := \begin{cases} \hat{g}(\sqrt{x}, \sqrt{x}, x) & \text{if } \hat{g}(\sqrt{x}, \sqrt{x}, x) > \arg \max H, \\ 2x - x^2 & \text{if } 2x - x^2 \leq \arg \max H, \\ \arg \max H & \text{otherwise.} \end{cases}
\]

This program is more complicated than the one in Section III-D; thankfully \( \Phi_\Xi \) and \( \Phi_\circ \) converge. Let \( \Phi_\Pi \) and
\( \hat{\Phi}_\circ \) be the end results of power iteration. We can now use
\[
\hat{\varphi}_m := \text{Linear\_Interp}(L, \hat{\Phi}_m), \\
\hat{\varphi}_\circ := \text{Linear\_Interp}(L, \hat{\Phi}_\circ)
\]
as the scoring functions. See Figure 10 for their plots; notice that \( \hat{\varphi}_m \geq \hat{\varphi}_\circ \) for all \( x \in [0, 1] \) and \( \hat{\varphi}_\circ > \hat{\varphi}_m \) for \( x \in [0.42, 0.97] \). With the scoring functions, we can finally prove \( \mu \leq 4.63 \).

VII. NEW PROOF OF \( \mu \leq 4.63 \)

This section gathers the materials and proves the main theorem.

**Theorem 16 (Main theorem):** \( \mu \leq 4.63 \), where \( \mu \) is the scaling exponent of polar coding using Arıkan’s kernel \([11]\) over BMS channels.

**Proof:** We have seen that Linear\_Interp\((M, \hat{G}) \approx \hat{g} \) and \( \hat{g}(\sqrt{x}, \sqrt{x}, x) \leq Z(\hat{W}^\circ) \), where \( W \) is a parallel combination of another BMS channel and \( x = Z(W) \). To obtain a practical yet rigorous lower bound on \( Z(\hat{W}) \), see Appendix B for how to define \( \hat{G} \) and \( \hat{G} \). By Theorem 26 therein, we have \( \hat{g}(\sqrt{x}, \sqrt{x}, x) \leq Z(\hat{W}^\circ) \) where \( \hat{g} := \text{Linear\_Interp}(M, \hat{G}) \).

Next, apply power iteration to optimize for the eigenvalues
\[
\lambda_m := \sup_{x \in L \setminus \{0, 1\}} \sup_{f(x, x) \leq z \geq 2x - x^2} \frac{\hat{\varphi}_m(x^2) + \hat{\varphi}_\circ(z)}{2\hat{\varphi}_m(x)}, \\
\lambda_\circ := \sup_{x \in L \setminus \{0, 1\}} \hat{g}(\sqrt{x}, \sqrt{x}, x) \leq 2x - x^2 \frac{2\hat{\varphi}_\circ(x)}{2\hat{\varphi}_\circ(x)}
\]
Per our execution, both suprema are about 0.860698. Therefore,
\[
\frac{\psi(W^\circ) + \psi(W^\circ)}{2\psi(W)}
\]
has \( \max\{\lambda_\circ, \lambda_\circ\} \approx 0.860698 \) as an empirical upper bound.

Finally, we use a generalization of Theorem 10 to compute a rigorous upper bound on \( \mu \). The final value is 4.62065 obtained by the following setup:
- \( G \) has shape \( 301 \times 301 \times 301 \),
- \( (\beta_0, \gamma_0, \beta_1, \gamma_1, \delta) = (2.552, 2.563, 4.163, 1.474, 10^{-5}) \),
- \( 10^6 \) nodes in \( [\delta^2, \delta] \) such that \( \{\text{artanh}(2x - 1) \mid x \in L \setminus [\delta^2, \delta]\} \) are evenly spaced,
- \( 2 \cdot 10^6 \) nodes in \( [\delta, 1 - \delta] \) such that \( \{\text{artanh}(2x - 1) \mid x \in L \setminus [\delta, 1 - \delta]\} \) are evenly spaced, and
- \( 10^6 \) nodes in \( [1 - \delta, 1 - \delta^2] \) such that \( \{\text{artanh}(2x - 1) \mid x \in L \setminus [1 - \delta, 1 - \delta^2]\} \) are evenly spaced.

Hence it is mathematically sound to say \( \mu \leq 4.63 \). \( \square \)

VIII. CONCLUSION

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

The limit of this method—analyzing \( (U \otimes V) \ast W \) to gain better control on \( Z \)—is 4.61126. This number is obtained by assuming \( g \) tri-convex and using \( g(\sqrt{x}, \sqrt{x}, x) = (1 + \sqrt{5 - 4x^2})/2 \) as the lower bound on the \( Z(W^\circ) \) in terms of \( x = Z(W^\circ) \).

For potential future work, we expect that analyzing the quad-variate channel transformation \( (U \otimes V) \ast (X \otimes Y) \) leads to a better bound. We said so because \( X \otimes Y \) is “less BSC” than the \( W \) in the tri-variate approach. More generally, one may consider octa-variate channel transformations
\[
(U \otimes U' \otimes V \otimes V') \ast (X \otimes X' \otimes Y \otimes Y'),
\]
which corresponds to \( W^{8\times8} \), and
\[
(U \otimes U') \ast (V \otimes V') \ast (X \otimes X') \ast (Y \otimes Y'),
\]
which corresponds to \( W^{8\times8} \). By considering larger and larger transformations, each being “less BSC” than the previous one, we can, in principle, obtain tighter and tighter bounds on \( \mu \). Except that there are two hurdles. (A) For a \( v \)-variate channel transformation, the size of the data array \( G \) grows exponentially in \( v \), which means that we probably will have to stop around \( v = 8 \). (B) Even if the size of \( G \) is admissible, we still need to convexify \( g \), which will penalize the overestimate of \( \mu \).

In general, it is not clear how far this line of research could go. We look forward to any work that characterizes density evolution better, as such work will be useful for both polar codes’ scaling exponent and low-density parity-check codes in general.

APPENDIX A

COPY-AND-PASTE SURGERY

This appendix proves Theorem 10. Let us recall the statement of the theorem.

**Theorem (Restating Theorem 10 for Convenience):** Let \( \alpha, \beta_0, \beta_1, \gamma_0, \gamma_1, \delta, \) and \( \mu \) be positive numbers. Let \( L \subseteq [\delta^2, 1 - \delta^2] \) be a set of nodes that contains \( \delta^2, 1 - \delta, \) and \( 1 - \delta^2 \). Let \( H \) be an array of positive numbers parametrized by \( L \). Let \( h := \text{Linear\_Interp}(L, H) \) be the linear interpolant. Suppose that \( H \) is unimodal, and its mode is at \( \alpha \in [\delta^2, \delta, 1 - \delta^2] \). Suppose that \( 21^{-1/\mu} \cdot h(\delta^2) \geq h(2\delta^2 - \delta^4) \) and \( 21^{-1/\mu} \cdot h(1 - \delta^2) \geq h((1 - \delta)^2) \). For any pair of neighboring nodes \( u, v \in L \) (by **neighboring nodes** we mean \( [u, v] \cap L = \{u, v\} \)), define
\[
y_0(H, u, v) := \begin{cases} 
  u^2 & \text{if } u^2 \geq \alpha, \\
  v^2 & \text{if } v^2 \leq \alpha, \\
  \alpha & \text{otherwise}.
\end{cases}
\]
and
\[
y_1(H, u, v) := \begin{cases} 
  u\sqrt{2 - u^2} & \text{if } u\sqrt{2 - u^2} \geq \alpha, \\
  v\sqrt{2 - v^2} & \text{if } v\sqrt{2 - v^2} \leq \alpha, \\
  \alpha & \text{otherwise}.
\end{cases}
\]
Suppose that, for any pair of neighboring nodes \( u, v \in L \cap [\delta^2, \delta] \),
\[
h(u) \geq \beta_0 u^{1-1/\mu}, \\
h(v) \leq \gamma_0 u^{1-1/\mu}, \\
\gamma_0 \cdot (v^2)^{1-1/\mu} + h(2v - v^2) \leq 2^{-1/\mu},
\]
and
\[
\frac{h(y_0(H, u, v)) + h(y_1(H, u, v))}{2 \min\{h(u), h(v)\}} \leq 2^{-1/\mu},
\]
Suppose that, for any pair of neighboring nodes \( u, v \in L \cap [1-\delta, 1-\delta^2] \),
\[
    h(v) \geq \beta_1 (1-u)^{1-1/\mu},
\]
\[
    h(u) \leq \gamma_1 (1-v)^{1-1/\mu},
\]
\[
    \frac{h(u^2) + \gamma_1 \cdot (u\sqrt{2-u^2})^{1-1/\mu}}{2h(v)} \leq 2^{-1/\mu}.
\]

Then there exists a function \( \tilde{h} : [0, 1] \to \mathbb{R} \) such that
\[
    \sup_{0 \leq x < 1} \sup_{0 \leq y \leq 2x - x^2} \tilde{h}(x^2) + \tilde{h}(y) \leq 2^{-1/\mu} + \max\{\gamma_0, \gamma_1 \} \delta^{2-2/\mu}.
\]

The overall strategy for the proof is as follows. We will define an eigenfunction function \( \tilde{h} \) and bound the eigenvalue \( (h(x^2) + h(y)) / 2h(x) \) in three sub-appendices, one for \( x \in [\delta, 1 - \delta] \), another for \( x \in [\delta^2, \delta] \), and the last one for \( x \in (0, \delta^2) \). For \((1-\delta, \delta^2]\) and \((1-\delta^2, 0] \), the argument is symmetric and omitted.

### A. The Eigenfunction \( \tilde{h} \)

Define \( \tilde{h} : [0, 1] \to \mathbb{R} \) as follows
\[
    \tilde{h}(x) := \begin{cases} 
    2^{-1+1/\mu} \cdot \tilde{h}(2x - x^2) & \text{if } x < \delta^2, \\
    h(x) & \text{if } \delta^2 \leq x \leq 1 - \delta^2, \\
    2^{-1+1/\mu} \cdot \tilde{h}(x^2) & \text{if } 1 - \delta^2 < x.
    \end{cases}
\]

Note that this is a recursive definition. For example, let’s try to evaluate \( \tilde{h}(\delta^2/2) \). Clearly \( \delta^2/2 \) belongs to the first case and so the evaluation is defined to be \( 2^{-1+1/\mu} \cdot \tilde{h}(\delta^2 - \delta^2/2) \). But \( \delta^2 - \delta^2/2 \) is still less than \( \delta^2 \) so the first case still applies. Now we are at \( 2(\delta^2 - \delta^2/2) - (\delta^2 - \delta^2/2)^2 \) which finally belongs to the second case. So the final evaluation is \( 2^{-2+2/\mu} \cdot h(2\delta^2 - \delta^2 - (\delta^2 - \delta^2/2)^2/2) \).

From the condition \( \alpha \in [2\delta, 1-\delta^2] \) we infer that \( \delta \leq 1/4 \).

So it is not possible that there is some \( x < \delta^2 \) such that \( 2x - x^2 > 1 - \delta^2 \). Therefore, the evaluation of \( \tilde{h} \) at any point \( x \in (0, 1) \) will eventually hit the second case after finitely many recursive calls.

For \( x = 0 \) or \( 1 \), it is also clear that the definition forces \( \tilde{h}(x) = 0 \). The evaluations at 0 and 1 are not used elsewhere in the proof but they signal the fact that channels having \( Z(W) \in \{0, 1\} \) are perfectly polarized channels.

To facilitate the proof in the upcoming sub-appendices, we bake two lemmas.

**Lemma 17:** \( \tilde{h} \) is unimodal, and its mode is at \( \alpha \).

**Proof:** We first show that \( \tilde{h} \) is increasing over \( [\delta^2, \alpha] \).
Note that this is within the range where \( \tilde{h} \) and \( h \) coincide, so it suffices to check if \( \tilde{h} \) is increasing. But \( \tilde{h} \) is just a linear interpolant so it remains to check if the data points \( H \) are ordered. By the assumption that \( H \) is unimodal and the mode is at \( \alpha \), we know that \( H \) is indeed increasing before \( \alpha \). This proves that \( \tilde{h} \) and \( h \) are increasing over \([\delta^2, \alpha]\).

Next, we see that the same reasoning implies that \( \tilde{h} \) is decreasing over \([\alpha, 1 - \delta^2]\). It remains to check if \( \tilde{h} \) is increasing over \([0, \delta^2]\) and decreasing over \([1 - \delta^2, 1]\).

Let \( u, v \) be two points, \( 0 \leq u < v \leq \delta^2 \). We want to prove \( \tilde{h}(u) \leq \tilde{h}(v) \). If \( u = 0 \), then \( \tilde{h}(u) = 0 < \tilde{h}(v) \) and there is nothing to prove. If \( 0 < u < v = \delta^2 \), then \( \tilde{h}(u) = 2^{-1+1/\mu} \cdot \tilde{h}(2u - u^2) \) by the recursive definition of \( \tilde{h} \). Depending on whether \( 2u - u^2 \geq \delta^2 \), we have two cases. On the one hand, if \( 2u - u^2 \) is indeed \( \geq \delta^2 \), then
\[
\begin{align*}
    \tilde{h}(u) &= 2^{-1+1/\mu} \cdot \tilde{h}(2u - u^2) \\
    &= 2^{-1+1/\mu} \cdot h(2u - u^2) \\
    &\leq 2^{-1+1/\mu} \cdot h(2\delta^2 - \delta^4) \\
    &\leq \tilde{h}(\delta^2) \\
    &= \tilde{h}(v).
\end{align*}
\]

Here, the first inequality is by that \( \tilde{h} \) is unimodal; the second inequality is given as an assumption in the theorem statement. On the other hand, if \( 2u - u^2 \) is still \( < \delta^2 \), we then need to run the recursive definition couple more times before the argument of \( \tilde{h} \) is \( \geq \delta \). But every time we apply the recursive definition, there is an additional \( 2^{-1+1/\mu} \) term. Hence \( h(u) \) will be less than \( h(\delta^2) = \tilde{h}(v) \). To summarize this paragraph, we now know that \( \tilde{h}(u) \leq \tilde{h}(v) \) whenever \( 0 \leq u < v \leq \delta^2 \).

We now assume \( 0 < u < v < \delta^2 \), and attempt to prove \( \tilde{h}(u) \leq \tilde{h}(v) \). By the recursive definition, the problem reduces to whether \( h(2u - u^2) \leq \tilde{h}(2u - u^2) \). This means that we can keep reassigning \((u, v) \leftarrow (2u - u^2, 2u - u^2)\) until one of the following stops us: (a) both \( u \) and \( v \) are \( \geq \delta^2 \); or (b) \( u < \delta^2 \leq v \). For case (a), \( \tilde{h}(u) \leq \tilde{h}(v) \) is equivalent to \( h(u) \leq h(v) \), which we know is true by that \( \tilde{h} \) is unimodal. For case (b), we know \( \tilde{h}(u) \leq \tilde{h}(\delta^2) \leq \tilde{h}(v) = \tilde{h}(u) \). Either way, \( \tilde{h}(u) \) is less than or equal to \( h(v) \). This finishes the proof of monotonocity of \( \tilde{h} \) over \([0, \delta^2]\).

Finally, we are left with proving that \( \tilde{h} \) is decreasing over \([1 - \delta^2, 1]\). This can be done by an argument similar to the \([0, \delta^2]\) case and is omitted. This concludes the proof of the unimodality of \( \tilde{h} \).

**Lemma 18:** For \( x \in [0, \delta] \),
\[
    \tilde{h}(x) \leq \gamma_0 x^{1-1/\mu}.
\]

For \( x \in [1 - \delta, 1] \),
\[
    \tilde{h}(x) \leq \gamma_1 (1 - x)^{1-1/\mu}.
\]

**Proof:** For \( x \in [\delta, 1-\delta] \), let \( u, v \in L \) be a pair of neighboring nodes that sandwich \( x \). Then
\[
    \tilde{h}(x) = h(x) \leq h(v) \leq \gamma_0 u^{1-1/\mu} \leq \gamma_0 x^{1-1/\mu}.
\]

The first inequality is by unimodality; the second inequality is an assumption in the theorem statement.

Next we want to show the inequality for \( x \in [0, \delta^2] \). For that, we apply mathematical induction on \(-\log_2(x)\) Suppose that \( \tilde{h}(x) \leq \gamma_0 x^{1-1/\mu} \) holds for \( x \in [\delta^2/2, \delta] \). Then, for \( x \in [\delta^2/2^{j+1}, \delta^2/2^j] \),
\[
    \tilde{h}(x) = 2^{-j+1/\mu} \cdot \tilde{h}(2x - x^2) \\
    \leq 2^{-j+1/\mu} \cdot h(2x) \\
    \leq 2^{-j+1/\mu} \cdot 2^{-\log_2(\delta^2/2^j)} \\
    = \gamma_0 x^{1-1/\mu}.
\]
The first inequality is by unimodality; the second inequality is by the induction hypothesis. This completes the induction and completes the proof of $\widehat{h}(x) \leq \gamma_0 (x^{2^{-1/\mu}})$ for $x \in [0, \delta]$. For the $x \in (1 - \delta, 1]$, a mirrored argument applies.

Lemma 17 makes it easy to bound the eigenvalue $\langle \widehat{h}(x^2) + \widehat{h}(y)/2\widehat{h}(x) \rangle$ when $x$ is not in $L$; Lemma 18 makes it easy when $x$ is not in $[\delta^2, 1 - \delta^2]$.\hfill\Box

**B. The Eigenvalue Over $[\delta, 1 - \delta]$**

In this sub-appendix, we want to upper bound $\langle \widehat{h}(x^2) + \widehat{h}(y)/2\widehat{h}(x) \rangle$ for $\delta \leq x \leq 1 - \delta$ and $x\sqrt{2} - x^2 \leq y \leq 2x - x^2$.

In order to do so, let $u, v \in L$ be a pair of neighboring nodes that sandwich $x$. The denominator $2\widehat{h}(x)$ is at least $2\min\{h(u), h(v)\}$ because $h(x)$, which is just $h(x)$, is defined through linear interpolation.

For the numerator, $\widehat{h}(x^2)$ is at most $h(u^2)$ if $h$ is increasing over $[u^2, v^2]$, and is at most $h(v^2)$ if $h$ is decreasing over $[u^2, v^2]$. By that $h$ is unimodal, if $h$ is neither increasing or decreasing over $[u^2, v^2]$, it must be the case that the mode is between $u^2$ and $v^2$; in this case, $\widehat{h}(x)$ is at most $h(\alpha)$. So we end up with $\widehat{h}(x^2) \leq h(y_0(H, u, v)).$

Similarly, using unimodality we can argue that $\widehat{h}(y)$, for $x\sqrt{2} - x^2 \leq y \leq 2x - x^2$, is at most $h(2v - v^2)$ if $h$ is increasing, $h(u/2 - u^2)$ if $h$ is decreasing, or $h(\alpha)$ if the mode is contained in the interval. We therefore conclude that $\sup_{x\sqrt{2} - x^2 \leq y \leq 2x - x^2} \frac{\widehat{h}(x^2) + \widehat{h}(y)}{2\widehat{h}(x)} \leq \frac{h(y_0(H, u, v)) + h(y_1(H, u, v))}{2\min\{h(u), h(v)\}}$.

The right-hand side is upper bounded by $2^{1-1/\mu}$ as part of the premises in the theorem statement. This completes the bound on the eigenvalue over the interval $[\delta^2, 1 - \delta^2]$.

**C. The Eigenvalue Over $[\delta^2, \delta)$**

In this sub-appendix, we want to upper bound $\langle \widehat{h}(x^2) + \widehat{h}(y)/2\widehat{h}(x) \rangle$ for $\delta^2 \leq x < \delta$ and $x\sqrt{2} - x^2 \leq y \leq 2x - x^2$.

Again, let $u, v \in L$ be a pair of neighboring nodes that sandwich $x$. For the denominator, $2\widehat{h}(x) \geq 2h(u)$ because the mode $\alpha$ is greater than $u$, $x$, and $v$ and we are at a place where $h$ is increasing. Similarly, for the second term in the numerator, $\widehat{h}(y) \leq h(2v - v^2)$.

The problem is with the evaluation of $\widehat{h}(x^2)$. As $x < \delta$, the square of $x$ is below the domain of $h$. So we apply Lemma 18 and obtain that $\widehat{h}(x^2) \leq \gamma_0 \cdot (x^{2^{-1/\mu}}) \leq \gamma_0 \cdot (v^{2})^{1-1/\mu}$. Hence, we have the following proposition.

**Proposition 20:** For $\delta^2 \leq x < \delta$,
\[
\sup_{x\sqrt{2} - x^2 \leq y \leq 2x - x^2} \frac{\widehat{h}(x^2) + \widehat{h}(y)}{2\widehat{h}(x)} \leq \frac{\gamma_0 \cdot (v^{2})^{1-1/\mu} + h(2v - v^2)}{2h(v)}.
\]

The right-hand side in upper bounded by $2^{1-1/\mu}$ as part of the premises in the theorem statement. This completes the bound on the eigenvalue over the interval $[\delta^2, \delta)$.

**D. The Eigenvalue Over $[0, \delta^2)$**

In this sub-appendix, we want to upper bound $\langle \widehat{h}(x^2) + \widehat{h}(y)/2\widehat{h}(x) \rangle$ for $0 < x < \delta^2$ and $x\sqrt{2} - x^2 \leq y \leq 2x - x^2$.

Unlike the previous cases, here, even $x$ is outside the domain of $h$. So we have to check the recursive definition and deduce that $\widehat{h}(x) = 2^{1-1/\mu} \cdot \widehat{h}(2x - x^2)$. Now we see a convenient fact that
\[
\frac{\widehat{h}(y)}{2h(x)} \leq \frac{\widehat{h}(2x - x^2)}{2\gamma_0} \leq 2^{1-1/\mu}.
\]

It remains to upper bound $\widehat{h}(x^2)$. We hope to prove that $\widehat{h}(x^2)/2\widehat{h}(x) \leq (\gamma_0/\beta_0)^{2^{-2/\mu}}$. But again, we need to first look up the recursive definition because both $x^2$ and $x$ are less than $\delta^2$.

After looking up, we deduce that $\widehat{h}(x^2)/2\widehat{h}(x) = \widehat{h}(2x^2 - x^4)/2\widehat{h}(2x - x^2)$. There are two possibilities: either $2x - x^2$ is $\geq \delta^2$ or $< \delta^2$. If $\geq \delta^2$, the case then the denominator is $\widehat{h}(2x - x^2) = \widehat{h}(2x^2 - x^4) > \beta_0 \cdot (2x^2 - x^4)^{1-1/\mu}$.

(2x^2 - x^4) is the last inequality by the assumption $h(u) \geq \beta_0 v^{1-1/\mu}$ in the theorem statement.) Meanwhile, the numerator is $\widehat{h}(2x^2 - x^4) \leq \gamma_0 \cdot (2x^2 - x^4)^{1-1/\mu}$ by Lemma 18. So we can bound their quotient as
\[
\frac{\widehat{h}(x^2)}{2h(x)} = \frac{\widehat{h}(2x^2 - x^4)}{2h(2x - x^2)} \leq \frac{\gamma_0 \cdot (2x^2 - x^4)^{1-1/\mu}}{2\beta_0 \cdot (2x^2 - x^4)^{1-1/\mu}} \leq \frac{\gamma_0 \cdot (2x^2 - x^4)^{1-1/\mu}}{\beta_0 \cdot (2x^2 - x^4)^{1-1/\mu}} \leq \frac{\gamma_0 \cdot (2x^2 - x^4)^{1-1/\mu}}{\beta_0 \cdot (2x^2 - x^4)^{1-1/\mu}} \leq \frac{\gamma_0 \cdot (2x^2 - x^4)^{1-1/\mu}}{\beta_0 \cdot (2x^2 - x^4)^{1-1/\mu}}.
\]

The second inequality uses $2 - x^2 \leq 2 - 4x - x^2$; the third inequality uses $x \leq \delta^2$.

It remains to bound the case where after looking up once, $2x - x^2$ is still $< \delta^2$. For such case, we deduce that
\[
\frac{\widehat{h}(x^2)}{2h(x)} = \frac{\widehat{h}(2x^2 - x^4)}{2h(2x - x^2)} \leq \frac{\widehat{h}(2x^2 - x^4)}{2h(2x - x^2)} \leq \frac{\widehat{h}(2x^2 - x^4)}{2h(2x - x^2)}.
\]

The inequality uses $(2x - x^2)^2 - (2x^2 - x^4) = 2x^2(1-x^2) \geq 0$. With this, we reduce a very small argument $x$ to a not-so-small argument $2x - x^2$. After finitely many reductions, $2x - x^2$ will eventually $\geq \delta^2$ and the previous paragraph will apply.

We therefore conclude that following proposition.

**Proposition 21:** For $0 < x < \delta^2$,
\[
\sup_{x\sqrt{2} - x^2 \leq y \leq 2x - x^2} \frac{\widehat{h}(x^2) + \widehat{h}(y)}{2h(x)} \leq 2^{1-1/\mu} + \frac{\gamma_0 \cdot \delta^2 - 2/\mu}{\beta_0}.
\]
Fig. 11. Piecewise linear interpolation (brown) of an arbitrary function (blue) is an approximation but not a valid lower bound.

Fig. 12. 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.

E. Concluding the Proof

We can finally prove Theorem 10.

Proof of Theorem 10: By Propositions 19–21, we deduce that

\[
\sup_{x^2 < y, x^2 + 2x^2 \leq y < 2x^2} \frac{\hat{h}(x^2) + \hat{h}(y)}{2h(x)} \leq 2^{-1/\mu} + \frac{\gamma_0 \delta^2 - 2/\mu}{\beta_0},
\]

for \( x \) from 0 to \( 1 - \delta \). For \( x \) from \( 1 - \delta \) to \( 1 - \delta^2 \), an argument similar to Proposition 20 applies and is omitted. For \( x \) from \( 1 - \delta^2 \) to 1, an argument similar to Proposition 21 applies and is omitted. This finishes the proof for all \( x \in (0, 1) \), hence

\[
\sup_{0 < x < 1} \sup_{x^2 < y, x^2 + 2x^2 \leq y < 2x^2} \frac{\hat{h}(x^2) + \hat{h}(y)}{2h(x)} \leq 2^{-1/\mu} + \frac{\gamma_0 \delta^2 - 2/\mu}{\beta_0}.
\]

\[\square\]

APPENDIX B
LINEAR INTERPOLATION MADE A PROPER LOWER BOUND

There is a caveat when approximating \( \hat{g} \) using \( \hat{G} \): the mesh is coarse. For one-dimensional interpolation (i.e., \( H \) and \( \Phi \)), 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 \( 100 \) is already \( 10^6 \) but the error is of the order of 1/100. See Figure 11 for an illustration of the caveat.

In this appendix, we will demonstrate how to find an array \( G_{\rightarrow} \) such that \( \text{Linear}_\text{Interp}(M,G_{\rightarrow}) \leq g \) pointwise. With \( G_{\rightarrow} \), we can run the iterative algorithm \( \text{Tri}_\text{Convexity} \) and the resulting array \( \hat{G}_{\rightarrow} \) will satisfy \( \text{Linear}_\text{Interp}(M,\hat{G}_{\rightarrow}) \leq \hat{g} \) pointwise. This will give us a mathematically rigorous control on \( Z(W \otimes \Xi) \).

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 \( g(a, b, c) \leq g(x, y, z) \) for all \( (x, y, z) \) in \( (a, b, c) + [0, 1/n]^3 \). Here, \( (a, b, c) + [0, 1/n]^3 \) is a small cube, called 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 \( G_{\rightarrow} \in R^{(n+1) \times (n+1) \times (n+1)} \) that is parametrized by \( M \) and populated by

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

Here, \( a - 1/n \vee 0 \) means \( \max\{a - 1/n, 0\} \). We call this the monotonic increasing approach and illustrate it in Figure 12.

The following lemma shows that linearly interpolating this array serves as a lower bound on \( g \).

Lemma 22: \( \text{Linear}_\text{Interp}(M,G_{\rightarrow}) \leq g \) 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, \( \text{Linear}_\text{Interp}(M,G_{\rightarrow})(x, y, z) \) is a convex combination of these eight numbers

\[
\begin{align*}
G_{\rightarrow}[a + \frac{0}{n}, b + \frac{1}{n}, c + \frac{1}{n}] & , \quad G_{\rightarrow}[a + \frac{1}{n}, b + \frac{1}{n}, c + \frac{1}{n}] , \\
G_{\rightarrow}[a + \frac{0}{n}, b + \frac{1}{n}, c + \frac{0}{n}] & , \quad G_{\rightarrow}[a + \frac{1}{n}, b + \frac{1}{n}, c + \frac{0}{n}] , \\
G_{\rightarrow}[a + \frac{0}{n}, b + \frac{0}{n}, c + \frac{1}{n}] & , \quad G_{\rightarrow}[a + \frac{1}{n}, b + \frac{0}{n}, c + \frac{1}{n}] , \\
G_{\rightarrow}[a + \frac{0}{n}, b + \frac{0}{n}, c + \frac{0}{n}] & , \quad G_{\rightarrow}[a + \frac{1}{n}, b + \frac{0}{n}, c + \frac{0}{n}] ,
\end{align*}
\]

By the definition of \( G_{\rightarrow} \), all eight numbers are less than or equal to \( g(a, b, c) \), so \( \text{Linear}_\text{Interp}(M,G_{\rightarrow})(x, y, z) \leq g(a, b, c) \leq g(x, y, z) \).

If we apply the monotonic increasing approach to a \( 301 \times 301 \times 301 \) mesh, we get about \( \mu \leq 4.65 \). To go below 4.63, we have to combine this with a second approach introduced in the next sub-appendix.
B. Smoothness Approach

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

Lemma 23: Let \( f : [0, 1] \to \mathbb{R} \) be doubly-differentiable on \([0, 1]\). Suppose \( f(0) = f(1) = 0 \) and \( f''(x) \leq m \) for some \( m \geq 0 \). Then for any \( x \in [0, 1] \),
\[
    f(x) \geq -\frac{m}{8}.
\]

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

Lemma 24: Let \( n \) be a positive integer. Let \( g : [0, 1/n]^3 \to \mathbb{R} \) be doubly-differentiable on \([0, 1/n]^3\). Suppose \( g = 0 \) at the eight corners of the cube \([0, 1/n]^3\). Suppose \( g_{xx} \leq m_1 \) and \( g_{yy} \leq m_2 \) as well as \( g_{zz} \leq m_3 \) for some \( m_1, m_2, m_3 \geq 0 \). Then for any \((x, y, z) \in [0, 1/n]^3\),
\[
    g(x, y, z) \geq -\frac{m_1 + m_2 + m_3}{8n^2}.
\]

Proof: First apply Lemma 23 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 23 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 23 in the \( z \)-direction to lower bound \( g(x, y, z) \) by \(-m_1 + m_2 + m_3)/8n^2\).

Lemma 24 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 the error was \( O(g'(n)) \).

Let us declare a new array \( G_1 : \mathbb{R}^{(n+1)\times(n+1)\times(n+1)} \) that is parametrized by \( M \) and populated by
\[
    G_1[a, b, c] := g(a, b, c) - \frac{m_1 + m_2 + m_3}{8n^3},
\]
where
\[
    m_1 = \sup_{((a,b,c)+[-1/n,1/n]^3)\cap[0,1]^3} \max\{g_{xx}, 0\},
    m_2 = \sup_{((a,b,c)+[-1/n,1/n]^3)\cap[0,1]^3} \max\{g_{yy}, 0\},
    m_3 = \sup_{((a,b,c)+[-1/n,1/n]^3)\cap[0,1]^3} \max\{g_{zz}, 0\}.
\]
The suprema are taken over all mesh cells that touch \((a, b, c)\). The following lemma confirms that linearly interpolating \( G_1 \) serves as a valid lower bound of \( g \). See also Figure 13 for an illustration.

Lemma 25: \( \text{Linear\_Interp}(M, G_1) \leq g \) 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 \( \tilde{g} := g - \text{Linear\_Interp}(M, G) \) is a function that is zero at the eight corners. Its second derivatives \( g_{xx}, g_{yy}, \) and \( g_{zz} \) are nothing but \( g_{xx}, g_{yy}, \) and \( g_{zz} \), respectively. Now apply Lemma 24: \( \tilde{g} \geq -(m_1 + m_2 + m_3)/8n^3 \), where \( m_1, m_2, m_3 \) are the suprema of the second derivatives over the concerned cell. Hence
\[
    g \geq \text{Linear\_Interp}(M, G) - \frac{m_1 + m_2 + m_3}{8n^3} \geq \text{Linear\_Interp}(M, G_1).
\]
This finishes the proof.

C. Interval Arithmetic for Derivatives

In the previous sub-appendix, we see how to initialize \( G_1 \) 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 \( G_1 \).

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 \textit{provably} contains the true value of every mathematical expression.

In our case, MPFI is the C-library that 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, \( G_\prec \) and \( G_\succ \), we see that \( G_\prec \) is tighter at places where \( g' \) is small but \( g'' \) is large; and \( G_\succ \) is tighter whenever \( g' \) is big and \( g'' \) is far less than \( 8n^2 \). In the sequel, we will compute \( G_\prec \) to be the array that uses values from \( G_\prec \) or \( G_\succ \) 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 \(G_{-}\) by default, but if \(G_{-}\) is worse than \(G_{1}\) at all eight corners \((a, b, c) + [0, 1]^3\), we switch to \(G_{1}\).

Now that we have specified which approach to use for every cell, we can initialize \(G_{-}\). Intuitively speaking, \(G_{-}[a, b, c]\) will be \(G_{-}[a, b, c]\) if any cell that touches \((a, b, c)\) decides to go for the increasing approach, but will be \(G_{1}[a, b, c]\) if all cells that touch \((a, b, c)\) decide to go for the smoothness bound. A formal description is as follows,

- \(G_{-}[a, b, c] = G_{-}[a, b, c]\) 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: \(G_{-}[x, y, z] \geq G_{1}[x, y, z]\).
- \(G_{-}[a, b, c] = G_{1}[a, b, c]\) 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: \(G_{1}[x, y, z] \geq G_{-}[x, y, z]\).

The following theorem concludes this appendix.

**Theorem 26:** With \(G_{-}\) defined as above, we have

\[
\text{Linear}_\text{Interp}(M, G_{-}) \leq g.
\]

With \(G_{-}\) being the result of performing \(\text{Tri}_\text{Convexify}(G_{-})\), we have

\[
\hat{g}_{-} := \text{Linear}_\text{Interp}(M, G_{-}) \leq \hat{g}.
\]

In particular, with \(x := Z(W)\) we have

\[
Z(W E) \geq \hat{g}_{-}(\sqrt{x}, \sqrt{x}, x).
\]

**Proof:** The first statement is by how \(G_{-}\) merges data points from \(G_{-}\) and \(G_{1}\). The second statement is by the first statement and Lemma 15. The last statement is by the second statement and Theorem 13.

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

**APPENDIX C**

**CONVEXIFY FASTER**

Earlier in this paper, we use formula (11) to tri-convexify a three-dimensional array \(G\). In this appendix, we describe a new strategy that converges faster.

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 \(\text{Linear}_\text{Interp}(L, A)\) 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 the hull, 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 three-dimensional array \(G\). Here, an axis of \(G\) is data points where two coordinates are fixed and the other coordinate is varying; for example \(\{(a, l, 2) \mid l \in L\}\).

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**

[1] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] E. Arikan and E. Telatar, “On the rate of channel polarization,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2009, pp. 1493–1495.

[3] S. Korada, E. Şaşoğlu, and R. Urbanke, “Polar codes: Characterization of exponent, bounds, and constructions,” *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6253–6264, Dec. 2010.

[4] A. Fazeli and A. Vardy, “On the scaling exponent of binary polarization codes,” *IEEE Trans. Inf. Theory*, vol. 59, no. 4, pp. 2267–2276, Apr. 2013.

[5] R. Mori and T. Tanaka, “Source and channel polarization over finite fields and Reed–Solomon matrices,” *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 7270–7276, May 2014.

[6] S. H. Hassani, K. Alishahi, and R. Urbanke, “On the scaling of polar codes: II. The behavior of un-polarized channels,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 879–883.

[7] S. B. Korada, A. Montanari, E. Telatar, and R. Urbanke, “An empirical scaling law for polar codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2010, pp. 884–888.

[8] A. Goli, S. H. Hassani, and R. Urbanke, “Universal bounds on the scaling behavior of polar codes,” in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 1957–1961.

[9] S. H. Hassani, K. Alishahi, and R. L. Urbanke, “Finite-length scaling for polar codes,” *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5875–5898, Oct. 2014.

[10] D. Golden and D. Burshtein, “Improved bounds on the finite length scaling of polar codes,” *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6966–6978, Nov. 2014.

[11] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors,” *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6698–6712, Dec. 2016.

[12] S. Fong and V. Tan, “Scaling exponent and moderate deviations asymptotics for polar codes for the AWGN channel,” *Entropy*, vol. 19, no. 7, p. 364, Jul. 2017. [Online]. Available: https://www.mdpi.com/1099-4300/19/7/364

[13] H. Mahdavifar, “Polar coding for non-stationary channels,” *IEEE Trans. Inf. Theory*, vol. 66, no. 11, pp. 6920–6938, Nov. 2020.

[14] S. Buzaglo, A. Fazeli, P. H. Siegel, V. Taranalli, and A. Vardy, “Perturbed successive cancellation decoding for polar codes,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 2618–2622.

[15] A. Fazeli and A. Vardy, “On the scaling exponent of binary polarization kernels,” in *Proc. 52nd Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2014, pp. 797–804.

[16] G. Trofimuk and P. Trifonov, “Window processing of binary polarization kernels,” *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4294–4305, Jul. 2021.

[17] H. Yao, A. Fazeli, and A. Vardy, “Explicit polar codes with small scaling exponent,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1757–1761.

[18] G. Trofimuk, “Shortened polarization kernels,” in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2021, pp. 1–6. [Online]. Available: https://drops.dagstuhl.de/opus/volltexte/2022/17139
Hsin-Po Wang received the B.S. degree in mathematics from National Taiwan University in 2015 and the Ph.D. degree in mathematics from the University of Illinois Urbana-Champaign in 2021. After spending a year with the University of California San Diego, he is currently a Post-Doctoral Scholar with the University of California, Berkeley. He won the Gold Medal at International Mathematical Olympiad in 2009 and 2011.

Ting-Chun Lin received the B.S. degree in physics and in mathematics from the Massachusetts Institute of Technology in 2020. He is currently pursuing the Ph.D. degree with the Department of Physics, University of California San Diego. Since 2021, he has been an Intern with the Hor Hai Quantum Computing Research Center. He won an Honorable Mentions and Top-25 Place at William Lowell Putnam Mathematical Competition in 2016, 2017, and 2019, the Gold Medal at International Physics Olympiad in 2014, the Silver Medal at Asian Physics Olympiad in 2014, and the Gold Medal at International Mathematical Olympiad in 2013.

Alexander Vardy (Fellow, IEEE) was born in Moscow, Russia, in 1963. He received the B.Sc. degree (summa cum laude) from the Technion—Israel Institute of Technology, Israel, in 1985, and the Ph.D. degree from Tel Aviv University, Israel, in 1991.

From 1985 to 1990, he was with the Israeli Air Force, where he worked on electronic counter measures systems and algorithms. From 1992 to 1993, he was a Visiting Scientist with the IBM Almaden Research Center, San Jose, CA, USA. From 1993 to 1998, he was with the University of Illinois Urbana-Champaign, first as an Assistant Professor and then as an Associate Professor. From 1998 to 2022, he was with the University of California at San Diego (UCSD), where he was the Jack Keil Wolf Chair Professor with the Department of Electrical and Computer Engineering and the Department of Computer Science. While on sabbatical from UCSD, he held long-term visiting appointments with CNRS, France, EPFL, Switzerland, the Technion—Israel Institute of Technology, and Nanyang Technological University, Singapore. His research interests include error-correcting codes, algebraic and iterative decoding algorithms, lattices and sphere packings, coding for storage systems, cryptography, computational complexity theory, and fun math problems. He was a member of the Board of Governors of the IEEE Information Theory Society from 1998 to 2006 and from 2011 to 2017. In 1996, he became a fellow of the David and Lucile Packard Foundation. In 1996, he was appointed as a fellow of the Center for Advanced Study, University of Illinois, and received the Xerox Award for faculty research. He received the IBM Invention Achievement Award in 1993 and the NSF Research Initiation and CAREER Awards in 1994 and 1995. He received the IEEE Information Theory Society Paper Award (jointly with Ralf Koetter) in 2004. In 2005, he also received the Fulbright Senior Scholar Fellowship and the Bus. Paper Award at the IEEE Symposium on Foundations of Computer Science (FOCS). In 2017, his work on polar codes was recognized by the IEEE Communications and Information Theory Societies Joint Paper Award. From 1995 to 1998, he was an Associate Editor of Coding Theory. From 1998 to 2001, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON INFORMATION THEORY.

Ryan Gabrys (Member, IEEE) received the B.S. degree in mathematics and computer science from the University of Illinois Urbana-Champaign in 2005, the M.Eng. degree in electrical and computer engineering from the University of California San Diego in 2010, and the Ph.D. degree in electrical engineering from the University of California, Los Angeles, in 2014. He is currently a Principal Investigator jointly affiliated with the Naval Information Warfare Center and the California Institute for Telecommunications and Information Technology (Calit2), University of California San Diego. His research interests include information theory and applied mathematics with applications to communications systems, computer security, and storage systems. He regularly serves as a technical program committee member for numerous conferences related to information and storage. He is a former SMART Scholar and a recipient of the IEEE/ACM George Michael High Performance Computing Award, the IEEE Data Storage Technical Committee Student Best Paper Award, and the NWMM Persistent Impact Award. He is currently an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY.