Explicit Polar Codes with Small Scaling Exponent

Hanwen Yao
University of California San Diego
9500 Gilman Drive, La Jolla, CA 92093
hwyao@ucsd.edu

Arman Fazeli
University of California San Diego
9500 Gilman Drive, La Jolla, CA 92093
afazelic@ucsd.edu

Alexander Vardy
University of California San Diego
9500 Gilman Drive, La Jolla, CA 92093
avardy@ucsd.edu

February 27, 2019

Abstract

Polar coding gives rise to the first explicit family of codes that provably achieve capacity for a wide range of channels with efficient encoding and decoding. But how fast can polar coding approach capacity as a function of the code length? In finite-length analysis, the scaling between code length and the gap to capacity is usually measured in terms of the scaling exponent $\mu$. It is well known that the optimal scaling exponent, achieved by random binary codes, is $\mu = 2$. It is also well known that the scaling exponent of conventional polar codes on the binary erasure channel (BEC) is $\mu = 3.627$, which falls far short of the optimal value. On the other hand, it was recently shown that polar codes derived from $\ell \times \ell$ binary polarization kernels approach the optimal scaling exponent $\mu = 2$ on the BEC as $\ell \to \infty$, with high probability over a random choice of the kernel.

Herein, we focus on explicit constructions of $\ell \times \ell$ binary kernels with small scaling exponent for $\ell \leq 64$. In particular, we exhibit a sequence of binary linear codes that approaches capacity on the BEC with quasi-linear complexity and scaling exponent $\mu < 3$. To the best of our knowledge, such a sequence of codes was not previously known to exist. The principal challenges in establishing our results are twofold: how to construct such kernels and how to evaluate their scaling exponent.

In a single polarization step, an $\ell \times \ell$ kernel $K_\ell$ transforms an underlying BEC into $\ell$ bit-channels $W_1, W_2, \ldots, W_\ell$. The erasure probabilities of $W_1, W_2, \ldots, W_\ell$, known as the polarization behavior of $K_\ell$, determine the resulting scaling exponent $\mu(K_\ell)$. We first introduce a class of self-dual binary kernels and prove that their polarization behavior satisfies a strong symmetry property. This reduces the problem of constructing $K_\ell$ to that of producing a certain nested chain of only $\ell/2$ self-orthogonal codes. We use nested cyclic codes, whose distance is as high as possible subject to the orthogonality constraint, to construct the kernels $K_{32}$ and $K_{64}$. In order to evaluate the polarization behavior of $K_{32}$ and $K_{64}$, two alternative trellis representations (which may be of independent interest) are proposed. Using the resulting trellises, we show that $\mu(K_{32}) = 3.122$ and explicitly compute over half of the polarization-behavior coefficients for $K_{64}$, at which point the complexity becomes prohibitive. To complete the computation, we introduce a Monte-Carlo interpolation method, which produces the estimate $\mu(K_{64}) \approx 2.87$. We augment this estimate with a rigorous proof that $\mu(K_{64}) < 2.97$. 
1 Introduction

Polar coding, pioneered by Arıkan in [1], gives rise to the first explicit family of codes that provably achieve capacity for a wide range of channels with efficient encoding and decoding. This paper is concerned with how fast can polar coding approach capacity as a function of the code length? In finite-length analysis [3, 5, 8, 10, 11], the scaling between code length \( n \) and the gap to capacity is usually measured in terms of the scaling exponent \( \mu \). It is well known that the scaling exponent of conventional polar codes on the BEC is 3.627, which falls far short of the optimal value \( \mu = 2 \). However, it was recently shown [3] that polar codes derived from \( \ell \times \ell \) polarization kernels approach optimal scaling on the BEC as \( \ell \to \infty \), with high probability over a random choice of the kernel.

Korada, Şaşoğlu, and Urbanke [6] were the first to show that polarization theorems still hold if one replaces the conventional \( 2 \times 2 \) kernel \( K_2 \) of Arıkan [1] with an \( \ell \times \ell \) binary matrix, provided that this matrix is nonsingular and not upper triangular under any column permutation. Moreover, [6] establishes a simple formula for the error exponent of the resulting polar codes in terms of the partial distances of certain nested kernel codes. However, an explicit formulation for the scaling exponent is at present unknown, even for the simple case of the BEC. Just like Arıkan’s \( 2 \times 2 \) kernel \( K_2 \), which transforms the underlying channel \( W \) into two synthesized bit-channels \( \{W^+, W^-\} \), an \( \ell \times \ell \) kernel \( K_\ell \) transforms \( W \) into \( \ell \) synthesized bit-channels \( W_1, W_2, \ldots, W_\ell \). If \( W \) is a BEC with erasure probability \( z \), the bit-channels \( W_1, W_2, \ldots, W_\ell \) are also BECs and their erasure probabilities are given by integer polynomials \( f_i(z) \) for \( i = 1, 2, \ldots, \ell \). The set \( \{f_1(z), f_2(z), \ldots, f_\ell(z)\} \) is known [3, 4] as the polarization behavior of \( K_\ell \) and completely determines its scaling exponent \( \mu(K_\ell) \).

While smaller scaling exponents translate into better finite-length performance, the complexity of decoding can grow exponentially with the kernel size. There have been attempts to reduce the decoding complexity of large kernels [2, 9], however this problem remains unsolved in general. We note that, although our constructions are explicit, issues such as decoding the kernel are beyond the scope of this work. Rather, our goal is to study the following simple question: what is the smallest scaling exponent one can
get with an $\ell \times \ell$ binary kernel? In particular, we construct a kernel $K_{64}$ with $\mu(K_{64}) \simeq 2.87$. This gives rise to a sequence of binary linear codes that approaches capacity on the BEC with quasilinear complexity and scaling exponent strictly less than 3. To the best of our knowledge such a sequence of codes was not previously known to exist.

1.1 Related Prior Work

Scaling exponents of error-correcting codes have been subject to an extensive amount of research. It was known since the work of Strassen [12] that random codes attain the optimal scaling exponent $\mu = 2$. It was furthermore shown in [11] that random linear codes also achieve this optimal value. For polar codes, the first attempts at bounding their scaling exponents were given in [5], where the scaling exponent of polar codes for arbitrary channels were shown to be bounded by $3.579 \leq \mu \leq 6$. The upper bound was improved to $\mu \leq 4.714$ in [8]. An upper bound on the scaling exponent of polar codes for non-stationary channels was also presented in [7] as $\mu \leq 10.78$.

Authors in [5] also introduced a method to explicitly calculate the scaling exponent of polar codes over BEC based on its polarization behavior. They showed that for the Arıkan’s kernel $K_2$, $\mu = 3.627$. Later on, an $8 \times 8$ kernel $K_8$ was found with $\mu = 3.577$ for BEC, which is optimal among all kernels with $\ell \leq 8$ [4]. It was accompanied with a heuristic construction to design larger polarizing kernels with smaller scaling exponents, which gave rise to a $16 \times 16$ kernel with $\mu = 3.356$. In [9], a $32 \times 32$ kernel $F_{32}$ and a $64 \times 64$ kernel was constructed, which was shown (via simulations) to have a better frame error rate than the Arıkan’s kernel. They have also introduced an algorithm based on the binary decision diagram (BDD) to efficiently calculate the polarization behavior of larger kernels. Attempts to achieve the optimal scaling exponent of 2 were first seen in [10], where it was shown that polar codes can achieve the near-optimal scaling exponent of $\mu = 2 + \epsilon$ by using explicit large kernels over large alphabets. The conjecture was just recently solved in [3], where it was shown that one can achieve the near-optimal scaling exponent via almost any binary $\ell \times \ell$ kernel given that $\ell$ is sufficiently large enough. Now it remains to find the explicit constructions of such optimal kernels. Our results in this paper can be viewed as another step towards the derandomization of the proof in [3].

1.2 Our Contributions

In this paper, a more comprehensive kernel construction approach is proposed. We first introduce a special class of polarizing kernels called the self-dual kernels. For those self-dual kernels, we prove a duality theorem showing that their polarization behaviors are symmetric, which enables us to construct the kernel by only designing its bottom half. In our construction, we use a greedy approach for the bottom half of the kernel, where we push the values of $f_i(z)$ as close to 0 as possible in the order of $i = \ell, \ell - 1, \cdots$, which intuitively gives us small scaling exponents. This construction gives the best previously found $16 \times 16$ kernel $K_{16}$ provided in [13] with scaling exponent 3.346, a new $32 \times 32$ kernel $K_{32}$ with $\mu(K_{32}) = 3.122$, and a new $64 \times 64$ kernel $K_{64}$ with $\mu(K_{64}) \simeq 2.87$ as depicted in Figure 1. We utilize the partial distances of nested Reed-Muller (RM) codes and cyclic codes to implement the proposed construction approach.

To calculate the scaling exponent of our constructed kernels, we first calculate their polarization behaviors, and then invoke the method introduced in [5]. For a specific bit-channel, its polarization behavior polynomial $f_i(z)$ can be described by the weight distribution of its uncorrectable erasure patterns. To calculate this weight distribution, we introduce a new trellis-based algorithm. Our algorithm is significantly faster than the BDD based algorithm proposed in [9]. It first builds a proper trellis for those uncorrectable erasure patterns, and then applies the Viterbi algorithm to calculate its weight distribution. We also propose
an alternative approach that builds a *stitching trellis*, which we believe is of independent interest. However, for a very large kernel ($K_{64}$ in our case), the complexity of our trellis algorithm gets prohibitively high for intermediate bit-channels. As a fix, we introduce an alternative Monte Carlo interpolation-based method to numerically estimate those polynomials of the intermediate bit-channels, which we use to estimate the scaling exponent of $K_{64}$ as $\mu(K_{64}) \simeq 2.87$. We further give a rigorous proof that $\mu(K_{64}) < 2.97$.

2 Preliminary Discussions

Let $K_{\ell}$ be a $\ell \times \ell$ kernel $K_{\ell} = [s_1^T, s_2^T, \cdots, s_{2\ell}^T]^T$ and $x = uK_{\ell}$ be a codeword that is transmitted over $\ell$ i.i.d. BEC channels $W = \text{BEC}(z)$. We define an *erasure pattern* to be a vector $e \in \{0, 1\}^\ell$, where 1 corresponds to the erased positions of $x$ and 0 corresponds to the unerased positions. The probability of occurrence of a specific erasure pattern $e$ will be $z^{\text{wt}(e)}(1-z)^{\ell - \text{wt}(e)}$, where $\text{wt}(e)$ is the Hamming weight of $e$.

**Definition 1** (Uncorrectable Erasure Patterns). We say the erasure pattern $e$ is *uncorrectable* for a bit-channel $W_i$ if and only if there exists two information vectors $u', u''$ such that $u'_j = u''_j$ for $j < i$, $u'_i \neq u''_i$ and $(u'K_{\ell})_j = (u''K_{\ell})_j$ for all unerased positions $j \in \{k : e_k = 0\}$.

For the $i$-th bit-channel $W_i$, let $E_{i,w}$ be the number of its uncorrectable erasure patterns of weight $w$, then its erasure probability $f_i(z)$ can be represented as the polynomial

$$f_i(z) = \sum_{w=0}^{\ell} E_{i,w} z^w (1-z)^{\ell - w}.$$ 

Therefore if we can calculate the weight distribution of its uncorrectable erasure patterns $E_{i,0}, E_{i,1}, \cdots, E_{i,\ell}$, we can get the polynomial $f_i(z)$. We call the entire set $\{f_1(z), \cdots, f_{\ell}(z)\}$ as the *polarization behavior* of $K_{\ell}$. One can utilize the techniques in [5] to estimate the scaling exponent of polar codes with large kernels by replacing the transformation polynomials $\{z^2, 1 - (1-z)^2\}$ in the traditional polar codes with the polarization behavior of $K_{\ell}$ defined above.

3 Construction of Large Self-dual Kernels

3.1 Kernel Codes

Before we find out what those uncorrectable erasure patterns are, we give the following definitions. Given two vectors $v_1, v_2$, we say $v_2$ *covers* $v_1$ if $\text{supp}(v_1) \subseteq \text{supp}(v_2)$. Given a set $S \subseteq \mathbb{F}_2^n$, we define its *cover set* $\Delta(S)$ as the set of vectors that covers at least one vector in $S$. It will be shown later, that the set of those uncorrectable erasure patterns are the cover set of a coset.
Definition 2 (Kernel Codes). Given an $\ell \times \ell$ kernel $K_\ell$, we define the kernel codes $C_i = \langle g_{i+1}, \ldots, g_{\ell} \rangle$ for $i = 0, \ldots, \ell$, and $C_\ell = \{0\}$.  

Theorem 1. An erasure pattern $e$ is uncorrectable for $W_i$ if and only if $e \in \Delta(C_{i-1} \setminus C_i)$.  

Proof. Here, we prove the “only if” direction. The other direction follows similarly. If $e$ is uncorrectable, then there exists $u', u''$ as described in Definition[1] of every uncorrectable erasure pattern $e$. Assume $e$ is uncorrectable for $W_i$, while its complement $\overline{e}$ is also uncorrectable for $W_{i+1-i}$, then $e$ covers a codeword $c_1$ in $(C_{i-1} \setminus C_i)$ and $\overline{e}$ covers a codeword $c_2$ in $(C_{\ell-i} \setminus C_{\ell+1-i}) = (C_{i-1}^\perp \setminus C_i^\perp)$. Since $\text{supp}(e)$ and $\text{supp}(\overline{e})$ are disjoint, we have $c_1 \perp c_2$. But since $C_{i-1}$ only has one more dimension than $C_i$, $c_1 \perp c_2$ and $c_1 \perp v$ would imply $c_2 \perp C_{i-1}$, which is a contradiction. Therefore the complement $\overline{e}$ of every uncorrectable erasure pattern $e$ for $W_i$ is correctable for $W_{i+1-i}$, which yields in the proof. □  

3.2 Self-dual Kernels and Duality Theorem  

We first introduce a special type of self-dual kernels. We call an $\ell \times \ell$ kernel self-dual if $C_i = C_{\ell-i}^\perp$ for all $i = 0, \ldots, \ell$. Then we prove the duality theorem, which shows that the polarization behavior of a self-dual kernel is symmetric.  

Lemma 1. If $K_\ell$ is self-dual, then  

$$\forall i \forall w : E_{i,w} + E_{\ell+1-i, \ell-w} \leq \binom{\ell}{w} \tag{2}$$  

Proof. Let $e$ be an uncorretable erasure pattern for $W_i$. Assume $e$ is uncorrectable for $W_i$ while its complement $\overline{e}$ is also uncorrectable for $W_{i+1-i}$, then $e$ covers a codeword $c_1$ in $(C_{i-1} \setminus C_i)$ and $\overline{e}$ covers a codeword $c_2$ in $(C_{\ell-i} \setminus C_{\ell+1-i}) = (C_{i-1}^\perp \setminus C_i^\perp)$. Since $\text{supp}(e)$ and $\text{supp}(\overline{e})$ are disjoint, we have $c_1 \perp c_2$. But since $C_{i-1}$ only has one more dimension than $C_i$, $c_1 \perp c_2$ and $c_1 \perp v$ would imply $c_2 \perp C_{i-1}$, which is a contradiction. Therefore the complement $\overline{e}$ of every uncorrectable erasure pattern $e$ for $W_i$ is correctable for $W_{i+1-i}$, which yields in the proof. □  

Theorem 2 (Duality Theorem). If $K_\ell$ is self-dual, then for $i = 1, \ldots, \ell$  

$$f_{\ell+1-i}(z) = 1 - f_i(1 - z) \tag{3}$$  

Proof. For all $i = 1, \ldots, \ell$ we have  

$$f_i(z) + f_{\ell+1-i}(1 - z) = \sum_{w=0}^\ell (E_{i,w} + E_{\ell+1-i, \ell-w})z^w(1 - z)^{\ell-w} \leq 1. \tag{4}$$  

Therefore, $\sum_{i=1}^\ell (f_i(z) + f_{\ell+1-i}(1 - z)) \leq \ell$. But a polarization step is a capacity preserving transformation, which means  

$$\sum_{i=1}^\ell f_i(z) + \sum_{i=1}^\ell f_{\ell+1-i}(1 - z) = \ell z + \ell(1 - z) = \ell. \tag{5}$$  

So all the previous inequalities must hold with equality. □
### Table 1: Kernel codes of $K_{32}$ at the bottom half

| rows | kernel codes                  | partial distances |
|------|-------------------------------|-------------------|
| 32   | $C_{32} = \{0, 1\}$           | 32                |
| 28-31| subcodes of $C_{27}$          | 16                |
| 27   | $C_{27} = \text{RM}(1,5)$     | 16                |
| 23-26| subcodes of $C_{12}$          | 12                |
| 22   | $C_{12} = \text{extended BCH}(31,11,11)$ | 12              |
| 18-21| subcodes of $C_{17}$          | 8                 |
| 17   | $C_{17} = \text{RM}(2,5)$     | 8                 |

### Table 2: Kernel codes of $K_{64}$ at the bottom half

| rows | kernel codes                  | partial distances |
|------|-------------------------------|-------------------|
| 64   | $C_{64} = \{0, 1\}$           | 64                |
| 59-33| subcodes of $C_{58}$          | 32                |
| 58   | $C_{58} = \text{RM}(1,6)$     | 32                |
| 56-57| subcodes of $C_{55}$          | 28                |
| 55   | $C_{55} = \text{extended BCH}(63,10,27)$ | 28            |
| 50-54| subcodes of $C_{49}$          | 24                |
| 49   | $C_{49} = \text{extended BCH}(63,16,23)$ | 24          |
| 44-48| subcodes of $C_{43}$          | 16                |
| 43   | $C_{43} = \text{RM}(2,6)$     | 16                |
| 38-42| subcodes of $C_{37}$          | 16                |
| 37   | $C_{37} = \text{extended cyclic}(63,28,15)$ | 16        |
| 36   | $C_{36} = (64,29,14)$ linear code | 14          |
| 35   | $C_{35} = (64,30,12)$ linear code | 12        |
| 34   | $C_{34} = (64,31,12)$ linear code | 12        |
| 33   | $C_{33} = (64,32,12)$ linear code | 12        |

### 3.3 Kernel Construction

The intuition behind our kernel construction is to a) mimic the polarization behavior of random kernels by making $f_i(z)$’s jump from $f_i(z) \in (0, \epsilon)$ to $(1 - \epsilon, 1)$ as sharp as possible (see Figure 3). b) provide a symmetry property in which half of the polynomials are polarizing to the value of 0 and the other half are polarizing to the value of 1 as depicted in Figure 2. In each step of our construction algorithm, we make sure that the constructed kernel is self-dual to design a symmetric polarization behavior according to the duality theorem. This allows us to focus on constructing only one half of the kernel. Here, we pick the bottom half. The strategy behind constructing the bottom half is to construct the rows in kernel one by one, while maximizing the partial distance, defined below, in each step.

**Definition 3 (Partial Distances).** Given an $\ell \times \ell$ kernel $K_\ell$, we define the partial distances $d_i = d_H(g_i, C_i)$ for $i = 1, \ldots, \ell - 1$, and $d_\ell = d_H(g_\ell, 0)$.

When $z$ is close to 0, the polynomial $f_i(z)$ will be dominated by the first non-zero term $E_{i,w}z^{\ell}(1-z)^{\ell-w}$. By Theorem[1] the first non-zero coefficients of $f_i(z)$ is $E_{i,d_i}$. So, we aim to maximize the partial distance $d_i$ to make $f_i(z)$ polarize towards 0.
The construction algorithm in a nutshell is described in the following. Start by setting $C_\ell = \{0\}$. Then from the bottom upwards, construct the bottom half of the kernel row by row greedily with maximum possible partial distances, while maintaining the kernel’s self-dual property. Namely for $i$ from $\ell$ to $\ell/2 + 1$, pick $v \in (C_\ell^\perp \setminus C_i)$ with the maximum partial distance $d_i = d(v, C_i)$ to be the $i$-th row of the kernel. The construction of the other half follows immediately by preserving the self-duality in each step.

Let us implement the algorithm for $\ell = 32$. We first pick the bottom row $g_{32}$ of $K_{32}$ to be the all 1 vector $\mathbf{1}$. Then for row 27-31, we pick codewords in RM(1,5) with maximum partial distance 16. After that, we carefully select codewords in the extended BCH codes and the RM(2,5), that both have maximum partial distances, and preserve the self-dual property of the kernel. The kernel code $C_{17}$ happens to be exactly the self-dual code RM(2,5). We finish our construction by filling up the top half and get the self-dual kernel $K_{32}$ as shown in Fig. 7. We construct $K_{64}$ as shown in Fig. 8 similarly, except that row 33-36 are picked through computer search. The kernel codes at the bottom half of $K_{32}$ and $K_{64}$ are shown in Table. 12.

4 Calculate the Polarization Behaviors

So far, we presented an algorithm to construct large binary kernels with intuitively good scaling exponents. In this section, we address the last challenge, which is to efficiently derive the polarization behavior of a given kernel. The NP hardness of this problem was previously established in [4]. In this section, we propose a few methods to reduce the computation complexity just enough so we can implement it. To this end, we present two trellis-based algorithms that can explicitly calculate the polarization behavior of $K_{32}$. Sadly, even these improved algorithms are beyond implementation for $K_{64}$. So, we present an alternative approach of “estimating” the polarization behavior of $K_{64}$ with high precision using a large but limited number of samples from the set of all erasure patterns. One can plug the estimated polarization behavior into the methods described in [5] and get $\mu(K_{64}) \approx 2.87$. We also provide a more careful analysis to show that $\mu(K_{64}) \leq 2.9603$ rigorously.
4.1 Trellis Algorithms

A trellis is a graphical representation of a block code, in which every path represents a codeword. This representation allows us to do ML decoding with reduced complexity using the famous Viterbi algorithm. The Viterbi algorithm allows one to find the most likely path in a trellis. Besides decoding, it can also be generalized to find the weight distribution of the block code, given that the trellis is one-to-one. A trellis is called one-to-one if all of its paths are labeled distinctly. We refer the readers to [14] for the known facts about trellises we use in this section.

In this work, we develop new theory for trellis representation for the cover sets, which are both nonlinear and non-rectangular. We introduced two different algorithms that both can construct a one-to-one trellis for the cover set $\Delta(C_{i-1} \setminus C_i)$. By efficiently representing the cover sets using trellises, we can use the Viterbi algorithm to calculate its weight distribution. A brief description of these algorithms together are given in the following. An example is also provided in Figure 5 for interested readers to track the steps in both algorithms.

**Proper Trellis Algorithm**

A trellis is called proper if edges beginning at any given vertex are labeled distinctly. It is known that if a trellis is proper, then it is one-to-one. So, one way of constructing a one-to-one trellis for $\Delta(C_{i-1} \setminus C_i)$ is to construct a proper trellis. The proper trellis algorithm has the following steps. **Step 1:** Construct a minimal trellis for the linear code $C_i$. For every edges in $E_i$ where $i \in \text{supp}(g_i)$, flip its label. We can then get a trellis for the coset $(C_{i-1} \setminus C_i)$. **Step 2:** For every label-0 edges, add a parallel label-1 edge. Then we get a trellis representing the cover set $\Delta(C_{i-1} \setminus C_i)$. But it is not a one-to-one trellis. **Step 3:** Let $T = (V, E, A)$ be the trellis we just constructed, use algorithm 1 to convert it into a proper trellis $T^* = (V^*, E^*, A)$, where for $i = 0, 1, 2, \ldots, \ell$, vertices in $V^*_i$ are labeled uniquely by the subsets of $V_i$. $T^*$ will thus be a one-to-one trellis representing the same cover set $\Delta(C_{i-1} \setminus C_i)$.

The proper trellis algorithm allows us to calculate the full polarization behavior of $K_{32}$, as shown in Figure 3. Unfortunately, the computational complexity is still too high for $K_{64}$, in which we were able to explicitly calculate the erasure probability polynomials associated with the last and first 15 rows in the kernel, as shown in Figure 4.

**Stitching Trellis Algorithm**

The complexity of proper trellis algorithm depends on the number of vertices in the trellis. It’s difficult

---

**Algorithm 1:** Construct a proper trellis $T^*$ from $T$

1. **for** $i = 0$ to $(\ell - 1)$ **do**
2.   **for** every vertex $v^*_i \in V^*_i$ **do**
3.     **for** $a \in \{0, 1\}$ **do**
4.       calculate $s = \{v_{i+1} \in V_{i+1} : \exists v_i \in L(v^*_i), (v_i, v_{i+1}, a) \in E\}$
5.       **if** $\exists v^*_i \in V^*_{i+1}$ with $L(v^*_i) = s$ **then**
6.         add an edge $(v^*_i, v^*_{i+1}, a)$ in $E^*$
7.       **else**
8.         add a vertex $v^*_{i+1} \in V^*_{i+1}$ with $L(v^*_{i+1}) = s$
9.         add an edge $(v^*_i, v^*_{i+1}, a)$ in $E^*$

---
to predict the number of vertices for general kernels, which could be significantly large. Hence, we also propose an alternative approach which also constructs a one-to-one trellis for $\Delta(C_{i-1}\setminus C_{j})$, but has far less vertices. The stitching trellis algorithm differs from the proper trellis algorithm only by Step 3: Let $T = (V, E, A)$ be the trellis we just constructed, use algorithm 1 only for $i$ from 0 to $(\ell/2 - 1)$ to convert the first half of $T$ into a proper trellis $T_1$. Reverse algorithm 1 to convert the second half of $T$ into a co-proper trellis $T_2$. Let $V_1, V_2$ be the vertex class of $T_1, T_2$ at time $\ell/2$. Connect $T_1$ and $T_2$ by adding an edge $(v_1, v_2)$ with label 0 for every pair of vertices $v_1 \in V_1, v_2 \in V_2$ where $L(v_1) \cap L(v_2) \neq 0$. Then the combined
Figure 6: An example for the trellis algorithms. Top left: minimal trellis for $C_1 \setminus C_2$ in $K_4 = K_2 \otimes 2$; top right: step 2 of trellis algorithm; bottom left: step 3 of proper trellis algorithm; bottom right: step 3 of the stitching algorithm.

trellis, called *stitching trellis*, will be a one-to-one trellis representing the same cover set $\Delta(C_{i-1} \setminus C_i)$.

The first and second half of the stitching trellis are proper and coproper respectively. Therefore, its number of vertices is bounded by $2^{\ell/2} + 1$, which is far less than a proper trellis. Unfortunately, the naive way of stitching the middle segment requires a large amount of computation. We are still searching for a method to reduce its complexity and believe this can be of independent interest to other researchers as well. Assuming such an efficient stitching is in place, the stitching trellis will be much more efficient than the proper trellis, which can also be used in other applications.

### 4.2 Monte Carlo Interpolation Method

As discussed earlier, the complexity of the trellis-based algorithms grow too high for the intermediate bit-channels of $K_{64}$. We present a Monte Carlo algorithm to estimate the values of polynomials $f_i(z)$ for any given $z \in (0, 1)$. We recall again that $f_i(z)$ denotes the erasure probability of the $i$-th bit-channel $W_i$ given that the communication is taking place over a BEC($z$). A naive yet explicit approach to formulate $f_i(z)$ is to cross check all $2^{\ell}$ erasure patterns to discover the exact ratio of which become uncorrectable from $W_i$’s point of view. Instead, we propose to take $N$ samples of such erasure patterns and estimate the ratio accordingly. We recall that the computational complexity of determining “correctability” is no more than the complexity of a MAP decoder for the BEC, which is bounded by $O(\ell^\omega)$, where $\omega$ is the exponent of matrix multiplication. Therefore, the overall complexity of the proposed approximation method can be bounded by $O(N\ell^\omega)$. While this approach adds some uncertainty to our derivations, the numerical simulations suggest that $\hat{f}_i(z)$’s for $\forall i$ become visibly smooth and stable at $N = 10^6$, as shown in Figure 5. The estimated value of $\mu(K_{64}) \simeq 2.87$ is generated by invoking the recursive methods in [5] initialized with $\hat{f}_i(z)$’s for $\forall i$.

If the accurate values of $f_i(z)$ were known, one could use the bounding techniques in [5] to show that
\[ \mu(K_{64}) \leq -\left( \log_{64} \left( \sup_{z \in (0,1)} \frac{1}{64} \sum_{i=1}^{64} g(f_i(z)) \right) \right)^{-1} \]  

(6)

where \( g(z) \) is a positive test function on \((0,1)\). However for kernel \( K_{64} \), due to high computational complexity, 34 intermediate polarization behavior polynomials are unknown. But we can still derive the strict upperbounds and lowerbounds for those unknown \( f_i(z) \)s’ to get the following theorem, with the proof in Appendix B.

**Theorem 3.**

\[ \mu(K_{64}) \leq 2.9603 \]  

(7)

**Acknowledgment**

We are grateful to Hamed Hassani and Peter Trifonov for very helpful discussions. We are also indebted to Peter Trifonov for sharing the source code of his BDD program.

**Appendices**

**Appendix A  Kernels \( K_{32} \) and \( K_{64} \)**

![Figure 7: Kernel \( K_{32} \)](image)
Figure 8: Kernel $K_{64}$
Appendix B  Proof for Theorem 3

Given an \( \ell \times \ell \) kernel with polarization behavior \( \{f_1, \cdots, f_\ell\} \), for a fixed \( z \in [0,1] \), we can define the process

\[
Z_0 = z, \quad Z_{n+1} = \begin{cases} 
  f_1(Z_n) & \text{w.p. 1/\ell} \\
  f_2(Z_n) & \text{w.p. 1/\ell} \\
  \vdots & \vdots \\
  f_1(Z_n) & \text{w.p. 1/\ell}
\end{cases}
\] (8)

First let’s recall the scaling assumption

**Assumption 1.** There exists \( \mu \in (0, \infty) \) such that, for any \( z, a, b \in (0,1) \) such that \( a < b \), The limit \( \lim_{n \to \infty} \ell^n \Pr(Z_n \in [a, b]) \) exists in \((0, \infty)\).

For a generic test function \( g : [0,1] \to [0,1] \), define the sequence of functions \( \{g_n\}_{n \in \mathbb{N}} \) as \( g_n : [0,1] \to [0,1] \) that

\[
g_n(z) = \mathbb{E}[g(Z_n) \mid Z_0 = z]
\] (9)

Then this sequence of functions satisfies the recursive relation

\[
g_0(z) = g(z), \quad g_{n+1}(z) = \frac{1}{\ell} \sum_{i=1}^{\ell} g_n(f_i(z))
\] (10)

Our approach of bounding \( \mu(K_{64}) \) has the following steps: (1) Find a suitable test function \( g(z) \); (2) provide an upper bound on the polarizing speed of the sequence \( \{g_n(z)\}_{n \in \mathbb{N}} \) and (3) turn this upper bound into bound for \( \mu(K_{64}) \). We here define the sequence \( \{b_n\}_{n \in \mathbb{N}} \) to measure the polarizing speed of \( \{g_n(z)\}_{n \in \mathbb{N}} \).

**Definition 4.**

\[
b_n(z) = \frac{g_{n+1}(z)}{g_n(z)}, \quad b_n = \sup_{z \in (0,1)} b_n(z)
\]

We can prove that \( \{b_n\}_{n \in \mathbb{N}} \) is a decreasing sequence.

**Lemma 2.** \( \{b_n\}_{n \in \mathbb{N}} \) is a decreasing sequence.

**Proof.** Since for any fixed \( z \)

\[
g_{n+1}(z) = \frac{1}{\ell} \sum_{i=1}^{\ell} g_n(f_i(z))
\]

\[
\leq b_{n-1} \left( \frac{1}{\ell} \sum_{i=1}^{\ell} g_{n-1}(f_i(z)) \right)
\]

\[
= b_{n-1} \cdot g_n(z)
\]

We have \( b_n(z) = \frac{g_{n+1}(z)}{g_n(z)} \leq b_{n-1} \) for any \( z \). Therefore \( b_n \leq b_{n-1} \) and \( \{b_n\}_{n \in \mathbb{N}} \) is a decreasing sequence. \(\square\)
From the above lemma we have:

\[
\begin{align*}
  g_n(z) & \leq b_{n-1}g_{n-1}(z) \\
  & \leq b_0g_{n-1}(z) \\
  & \leq \cdots \\
  & \leq b_0^ng(z)
\end{align*}
\]

Next we use \(b_0\) to give an upperbound for the scaling exponent of the kernel.

**Lemma 3.** For \(a, b \in (0, 1)\) and \(n \in \mathbb{N}\) we have:

\[
\frac{1}{n} \log_{\ell} \Pr(Z_n \in [a, b]) \leq \log_{\ell} b_0 + O \left( \frac{1}{n} \right)
\]

**Proof.** By Markov inequality

\[
\Pr(Z_n \in [a, b]) \leq \Pr(g(Z_n) \geq \min_{z \in [a, b]} g(z)) \leq \frac{\mathbb{E}[g(Z_n)]}{\min_{z \in [a, b]} g(z)}
\]

So

\[
\frac{1}{n} \log_{\ell} \Pr(Z_n \in [a, b]) \leq \frac{1}{n} \log_{\ell} \frac{(b_0)^n g(z)}{\min_{z \in [a, b]} g(z)} \leq \log_{\ell} b_0 + \frac{1}{n} \left( \log_{\ell} \frac{g(z)}{\min_{z \in [a, b]} g(z)} \right)
\]

\(\square\)

Since by scaling assumption

\[
-\frac{1}{\mu} = \lim_{n \to \infty} \frac{1}{n} \log_{\ell} (\Pr(Z_n \in [a, b]))
\]

We have \(\mu \leq -\frac{1}{\log_{\ell} b_0}\). We next pick the appropriate test function \(g\) and use \(b_0\) to obtain a valid upperbound for \(\mu(K_{64})\). First we explain in detail how we construct this test function \(g(z)\).

By the trellis algorithm, we get the explicit polynomials \(f_{50}, \cdots, f_{64}\). By the duality theorem, we also get to know the explicit formulas for \(f_{1}, \cdots, f_{15}\). And there are 34 polarization behavior polynomials left unknown. But for those unknown coefficients, we can calculate their upperbound as follows:

**Lemma 4.** Let \(A_0, A_1, \cdots, A_\ell\) be the weight enumerators for the coset \((C_{i-1} \setminus C_i)\), then

\[
E_{i,w} \leq \min \left( \sum_{j=1}^{i} \binom{\ell - i}{j - i} A_j \binom{\ell}{i} \right)
\]

**Proof.** By theorem[1] any erasure pattern is uncorrectable iff it covers a codeword in \((C_{i-1} \setminus C_i)\). For each codeword in \((C_{i-1} \setminus C_i)\) of weight \(j\), there are \(\binom{n-j}{w-j}\) erasure patterns with weight \(w\) that covers it. So \(E_{i,w} \leq \sum_{j=1}^{i} \binom{\ell - i}{j - i} A_j\). On the other hand, \(E_{i,w}\) is at most \(\binom{\ell}{i}\).  

\(\square\)
Figure 9: For kernel $K_{64}$, on the left, from left to right we have $\tilde{f}_{40}(z)$, $f_{40}(z)$ simulated by Monte Carlo interpolation method and $\tilde{f}_{40}(z)$. On the right we have, $\tilde{f}_{40}(z)$.

For $i = 16, 17, \cdots, 49$ of $K_{64}$. Define

$$E_{i,w} = \min \left( \sum_{j=1}^{i} \left( \ell - i \right) A_{i, \ell} \right), \quad E_{i,w} = \left( \frac{\ell}{i} \right) - E_{\ell+1-i, \ell-w}$$

And define

$$\tilde{f}_{i}(z) = \sum_{i=0}^{\ell} E_{i,w} z^{i} (1-z)^{\ell-i}, \quad f_{i}(z) = \sum_{i=0}^{\ell} E_{i,w} z^{i} (1-z)^{\ell-i}$$

Then for $i = 16, 17, \cdots, 49$ and any fixed $z$, $f_{i}(z) \in \left[ \tilde{f}_{i}(z), f_{i}(z) \right]$. An example for $\tilde{f}_{i}(z)$ and $f_{i}(z)$ are shown in Fig 9.

We define our test function $g$ as follows

$$g(z) = \frac{1}{64} \left( \sum_{i=1}^{15} g^*(f_{i}(z)) + \sum_{i=50}^{64} g^*(f_{i}(z)) + \sum_{i=16}^{49} g^*(\tilde{f}_{i}(z)) \right), \quad g^*(z) = z^{1/2} (1-z)^{1/2}$$

where

$$\tilde{f}_{i}(z) = \begin{cases} f_{i}(z) & \tilde{f}_{i}(z) \leq 0.5 \\ 0.5 & 0.5 \in (\tilde{f}_{i}(z), f_{i}(z)) \\ f_{i}(z) & \tilde{f}_{i}(z) \geq 0.5 \end{cases}$$

An example of $\tilde{f}_{i}(z)$ is shown in Fig 9. And a plot of the test function is shown in Fig 10. Since $K_{64}$ is self-dual, by duality theorem we can shown that $g(z)$ increases on $[0, 0.5]$, decreases on $[0.5, 1]$, and reach its maximum when $z = 0.5$. Therefore for $i = 16, 17, \cdots, 49$:

$$g(f_{i}(z)) \leq \begin{cases} g(\tilde{f}_{i}(z)) & \tilde{f}_{i}(z) \leq 0.5 \\ g(0.5) & 0.5 \in (\tilde{f}_{i}(z), f_{i}(z)) \\ g(f_{i}(z)) & \tilde{f}_{i}(z) \geq 0.5 \end{cases}$$
Figure 10: Left: test function \( g(z) \). Right: upperbound \( \overline{b}_0(z) \).

And this gives us a strict upper bound \( \overline{\mathcal{g}}_1(z) \) for \( g_1(z) \):

\[
\begin{align*}
g_1(z) = \frac{1}{64} \left( \sum_{i=1}^{64} s(f_i(z)) \right) & \leq \overline{\mathcal{g}}_1(z) = \frac{1}{64} \left( \sum_{i=1}^{15} s(f_i(z)) + \sum_{i=50}^{64} s(f_i(z)) + \sum_{i=16}^{49} s(\tilde{f}_i(z)) \right) \\
\end{align*}
\]

which provide a strict upper bound \( \overline{b}_0(z) \) for \( b_0(z) \), as shown in Fig 10:

\[
\begin{align*}
b_0(z) = \frac{g_1(z)}{g(z)} & \leq \overline{b}_0(z) = \frac{\overline{\mathcal{g}}_1(z)}{g(z)} \\
\end{align*}
\]

And the maximum value of \( \overline{b}_0(z) \) can be calculated analytically up to any desired precision. Our calculation shows that:

\[
b_0 = \sup_{z \in (0,1)} b_0(z) \leq \max_{z \in (0,1)} \overline{b}_0(z) = 0.2454
\]

which provides an upperbound \( \mu(K_{64}) \leq -\frac{1}{\log_{64} 0.2454} = 2.9603 \).
References

[1] E. Arıkan, “Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–73, 2009.

[2] S. Buzaglo, A. Fazeli, P. H. Siegel, V. Taranalli, and A. Vardy, “Permuted successive cancellation decoding for polar codes,” *In Proc. of IEEE International Symposium on Information Theory*, pp. 2618–22, 2017.

[3] A. Fazeli, S. H. Hassani, M. Mondelli, and A. Vardy, “Binary linear codes with optimal scaling and quasi-linear complexity,” [Online] arXiv preprint arXiv:1711.01339, 2017.

[4] A. Fazeli and A. Vardy. “On the scaling exponent of binary polarization kernels”, *In Proceedings of Allerton Conference on Communication, Control, and Computing*, pp. 797–804, 2014.

[5] H. S. Hassani, K. Alishahi, and R. L. Urbanke, “Finite-length scaling of polar codes,” *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5875–98, 2014.

[6] S. B. Korada, E. Şaşoğlu, and R. Urbanke, “Polar codes: Characterization of exponent, bounds, and constructions,” *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6253–64, 2010.

[7] H. Mahdavifar, “Fast polarization and finite-length scaling for non-stationary channels,” [Online] arXiv preprint arXiv:1611.04203, 2016.

[8] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors,” *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 6698-712, 2016.

[9] V. Miloslavskaya and P. Trifonov, “Design of binary polar codes with arbitrary kernel,” *In Proceedings of IEEE Information Theory Workshop*, pp. 119–123, 2012.

[10] H. D. Pfister and R. Urbanke, “Near-optimal finite-length scaling for polar codes over large alphabets,” *In Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pp. 215–219, 2016.

[11] Y. Polyanskiy, H. V. Poor, and S. Verdu, “Channel coding rate in the finite blocklength regime,” *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–59, 2010.

[12] V. Strassen, “Asymptotische Abschatzungen in Shannon’s Informationstheorie,” *Prague Conference on Information Theory, Statistical Decision Functions, and Random Processes*, pp. 689–723, 1962.

[13] G. Trofimiuk, P. Trifonov, “Efficient decoding of polar codes with some 16 × 16 kernels,” *2018 IEEE Information Theory Workshop (ITW)*. IEEE, 2018.

[14] A. Vardy, “Trellis structure of codes,” in *HANDBOOK OF CODING THEORY*, Elsevier, 1998.