Polar Codes with Balanced Codewords

Utkarsh Gupta*, Han Mao Kiah†, Alexander Vardy‡, and Hanwen Yao‡
*Department of Mathematics, Indian Institute of Technology, Hauz Khas, New Delhi, India
†School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
‡Department of Electrical & Computer Engineering, University of California San Diego, LA Jolla, CA, USA
Emails: utkarshgupta99@gmail.com, hmkiah@ntu.edu.sg, avardy@ucsd.edu, hay125@eng.ucsd.edu

***Abstract*—The *imbalance* of a binary word refers to the absolute difference between the number of ones and zeros in the word. Motivated by applications in DNA-based data storage and the success of polar codes, we study the problem of reducing imbalance in the codewords of a polar code. To this end, we adapt the technique of Mazumdar, Roth, and Vontobel by considering balancing sets that correspond to low-order Reed-Muller (RM) codes. Such balancing sets are likely to be included as subcodes in polar codes.**

Specifically, using the first-order RM code, we show that any message can be encoded into a length-n polar codeword with imbalance at most $o(n)$ in $O(n \log n)$-time. We then reduce the imbalance even further using two methods. First, we constrain the ambient space $\mathbb{X}$ and analyze the imbalance that the first-order RM code can achieve for words in $\mathbb{X}$. We demonstrate that for codelengths up to 128, the first-order RM code achieves zero imbalance for appropriate choices of $\mathbb{X}$ that sacrifice only a few message bits. Second, we augment the balancing set by considering higher order RM codes. We give a simple recursive upper bound for the guaranteed imbalance of RM codes. We also prove that the second-order RM code $\mathbb{RM}(2, m)$ balances all even-weight words for $m \leqslant 5$, while the RM code of order $m-3$ balances all even-weight words for $m \geqslant 5$.

I. Introduction

The *imbalance* of a binary word $\boldsymbol{x}$ refers to the absolute difference between the number of ones and the number of zeros in $\boldsymbol{x}$. A word is *balanced* if its imbalance is at most one and a code is *balanced* if all its codewords are balanced. Due to their applications in various recording systems, balanced codes have been extensively studied (see [1] for a survey).

Coupled with recent progress in the biotechnology industry, DNA macromolecules are emerging as a next-generation data storage medium with its unprecedented density, durability and replication efficiency [2]. This has rekindled interest in balanced codes. Specifically, a DNA string comprises four bases or letters: `A`, `C`, `T` and `G`, and a string is `GC`-*rich (or* `GC`-*poor)* if a high (or low) proportion of the bases corresponds to either `G` or `C`. Since `GC`-rich or `GC`-poor DNA strings are prone to both synthesis and sequencing errors [3], [4], we aim to reduce the difference with the number of `G` and `C` and the number of `A` and `T` on every DNA codeword. This requirement is equivalent to reducing the imbalance of a related binary word.

To further reduce errors, we equip our balanced codes with error-correcting capabilities and previous constructions of balanced error-correcting codes can be found in [5]–[10]. We highlight a few techniques that transform known linear error-correcting codes into balanced ones. In [8], Weber *et al.* take two input codes of distance d: a linear code and a *short* balanced codebook, and constructs a long balanced code of distance d. Later, Chee *et al.* [9] remove the need of a short codebook and construct balanced error-correcting codes from cyclic codes.

Of significance to our work is the technique of Mazumdar *et al.* [7]. At a high level, Mazumdar *et al.* consider a linear code $\mathbb{C}$ of length n and write it as a direct sum of two linear subspaces $\mathbb{C}'$ and $\mathbb{B}$, in other words, $\mathbb{C} = \mathbb{C}' \oplus \mathbb{B}$. Here, $\mathbb{B}$ is chosen so that it is a *balancing set*. In other words, for any binary word $\boldsymbol{x}$, there exists a balancing word $\boldsymbol{b} \in \mathbb{B}$ so that $\boldsymbol{x} + \boldsymbol{b}$ is balanced. In this coding scheme, the codewords are of the form $\boldsymbol{x} + \boldsymbol{b}$, where $\boldsymbol{x}$ belongs to $\mathbb{C}'$ and $\boldsymbol{b}$ is its corresponding balancing word. In the same paper, the authors demonstrated that a random subcode $\mathbb{B}$ of dimension $(3/2)\log n + o(\log n)$ is a balancing set with high probability. However, verifying that $\mathbb{B}$ is indeed a balancing set remains computationally difficult.

In this paper, we apply the technique of Mazumdar *et al.* [7] to a beautiful and important class of codes, the *polar codes*. Invented by Arıkan [11], polar codes achieve capacity for many channels with low encoding and decoding complexities. Recently, they are being adopted in 5G standard [12] and have been adapted for insertion and deletion channels [13], [14]. Given this appeal, we study efficient means of transforming messages into balanced polar codewords while retaining the low complexities of the polar encoding and decoding algorithms.

We remark that techniques to adapt polar codes for processes with memory or constrained systems were studied by Şaşoğlu [15] and Shuval and Tal [16]. However, it is unclear whether the framework in these works can be used efficiently here.

II. Preliminaries

Let $\boldsymbol{x}$ be a binary word of length n. We use $\mathrm{wt}(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ to denote its weight and imbalance, respectively. We have that

$$\mu(\boldsymbol{x}) = |2\mathrm{wt}(\boldsymbol{x}) - n|. \tag{1}$$

We also regard $\boldsymbol{x}$ as a vector in $\mathbb{F}_2^n$. For a subset $\mathcal{X} \subset \mathbb{F}_2^n$, we use $\mathrm{span}(\mathcal{X})$ to denote the *linear span of vectors in* $\mathcal{X}$. Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we let $\mathrm{wt}(\boldsymbol{x} \cap \boldsymbol{y})$ denote the number of indices where both $\boldsymbol{x}$ and $\boldsymbol{y}$ are one, and we have the observation:

$$\mathrm{wt}(\boldsymbol{x} + \boldsymbol{y}) = \mathrm{wt}(\boldsymbol{x}) + \mathrm{wt}(\boldsymbol{y}) - 2\mathrm{wt}(\boldsymbol{x} \cap \boldsymbol{y}). \tag{2}$$

We use $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ to denote the *dot product* of $\boldsymbol{x}$ and $\boldsymbol{y}$ over $\mathbb{F}_2$ and we have that $\mathrm{wt}(\boldsymbol{x} \cap \boldsymbol{y}) \equiv \langle \boldsymbol{x}, \boldsymbol{y} \rangle \pmod 2$.

A. Polar Codes

The *polarization kernel matrix* $\boldsymbol{G}_1$ is given by $\left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. For $m \geqslant 2$, set $n \triangleq 2^m$. Then the *polar transformation matrix* $\boldsymbol{G}_m$ is an $n \times n$ matrix, recursively defined by

$$\boldsymbol{G}_m \triangleq \begin{bmatrix} \boldsymbol{G}_{m-1} & \boldsymbol{0} \\ \boldsymbol{G}_{m-1} & \boldsymbol{G}_{m-1} \end{bmatrix}.$$

We index the rows of $\boldsymbol{G}_m$ with words in $\mathcal{Z} \triangleq \mathbb{F}_2^m$ and list them in bit-reversal order. To define a polar code, we pick a set $\mathcal{F} \subset \mathcal{Z}$ of *frozen indices* and a *frozen syndrome* $\boldsymbol{f} \in \mathbb{F}_2^{|\mathcal{F}|}$. Then the *polar code* defined by $\mathcal{F}$ and $\boldsymbol{f}$ is the linear code $\mathbb{C}(\mathcal{F}, \boldsymbol{f}) \triangleq \{\boldsymbol{x}\boldsymbol{G}_m : \boldsymbol{x}|_{\mathcal{F}} = \boldsymbol{f}\}$. The index set $\mathcal{I} \triangleq \mathcal{Z} \setminus \mathcal{F}$ corresponds to the *information indices* and $k \triangleq |\mathcal{I}|$ measures the bits of information that the polar code $\mathbb{C}(\mathcal{F}, \boldsymbol{f})$ can transmit.

Specifically, given a message $\boldsymbol{m} \in \mathbb{F}_2^k$, we can efficiently encode $\boldsymbol{m}$ to a polar codeword $\boldsymbol{c} \triangleq \mathrm{enc}(\boldsymbol{m})$ in $O(n \log n)$ time. On the other hand, suppose that the noisy word $\widetilde{\boldsymbol{c}}$ is received. Using successive cancellation decoding, we compute $\mathrm{dec}(\widetilde{\boldsymbol{c}})$ in $O(n \log n)$ time and recover the message $\boldsymbol{m}$ with high probability.

978-1-7281-6432-8/20/$31.00 ©2020 IEEE ISIT 2020

Now, the choice of $\mathcal{F}$ is channel dependent and typically, the frozen syndrome $\boldsymbol{f}$ is chosen to be the all zero word. When $\boldsymbol{f} = \mathbf{0}$, the polar code $\mathbb{C}(\mathcal{F}, \mathbf{0})$ is simply the linear span of the rows with indices in $\mathcal{I}$. In this case, we simply write this linear span as $\mathrm{span}(\mathcal{I})$. For certain choices of frozen indices, the polar code $\mathbb{C}(\mathcal{F}, \mathbf{0})$ corresponds to the ubiquitous class of *Reed-Muller* (RM) codes [17, Ch. 13]. Specifically, if $\mathcal{F}_r$ is the set of indices whose weight are at most r, then $\mathbb{C}(\mathcal{F}_r, \mathbf{0})$ defines the class of Reed-Muller code of order $m - r - 1$, or $\mathbb{RM}(m - r - 1, m)$ in short. Equivalently, if $\mathcal{J}_{m-r}$ denote the set of indices whose weight is at least $m - r$, we have that $\mathrm{span}(\mathcal{J}_{m-r}) = \mathbb{RM}(r, m)$. RM codes have been studied extensively and recently, Abbe and Ye demonstrated that the RM codes share similar polarization behaviour as polar codes [18].

In the next subsection, we pick a subset $\mathcal{B}$ from the information set $\mathcal{I}$ for the purposes of reducing the imbalance of codewords. Specifically, we reduce the number of information bits to $k' = n - |\mathcal{F}| - |\mathcal{B}|$ so that a k'-bit message $\boldsymbol{m}'$ is encoded to a polar codeword $\boldsymbol{c}$ with small imbalance $\mu(\boldsymbol{c})$.

B. Linear Balancing Sets

We discuss the technique *ala* Mazumdar *et al.* [7] in the context of polar codes. As before, we fix a set of frozen indices $\mathcal{F}$ and a frozen syndrome $\boldsymbol{f}$. Let $k = n - |\mathcal{F}|$ and we have the polar code $\mathcal{C}(\mathcal{F}, \boldsymbol{f})$ with a corresponding pair of encoding $\mathrm{enc} : \mathbb{F}_2^k \to \mathbb{F}_2^n$ and decoding $\mathrm{dec} : \mathbb{F}_2^n \to \mathbb{F}_2^k$ algorithms.

We pick a set of *balancing indices* $\mathcal{B}$ so that $\mathcal{F} \cap \mathcal{B} = \varnothing$ and let $\mathbb{B}$ be the linear span of the rows corresponding to $\mathcal{B}$. We then set $k' \triangleq k - |\mathcal{B}|$ and in this *balancing encoding scheme*, we transmit k'-bit messages (instead of k-bit). Consider $\boldsymbol{m}' \in \mathbb{F}_2^{k'}$. We first insert $|\mathcal{B}| = k - k'$ zeros to $\boldsymbol{m}'$ at positions corresponding to $\mathcal{B}$ and compute the corresponding encoding $\boldsymbol{c}'$. Next, we find a balancing vector $\boldsymbol{b} \in \mathbb{B}$ so that the corresponding imbalance $\mu(\boldsymbol{c}' + \boldsymbol{b})$ is minimized. Then we transmit the word $\boldsymbol{c} \triangleq \boldsymbol{c}' + \boldsymbol{b}$.

To decode a noisy word $\widetilde{\boldsymbol{c}}$, we simply apply the polar-decoding algorithm and find the k-bit vector $\boldsymbol{m} = \mathrm{dec}(\widetilde{\boldsymbol{c}})$. The desired message $\widehat{\boldsymbol{m}}'$ is then the k'-prefix of $\boldsymbol{m}$. So, if $\boldsymbol{m}$ is successfully decoded under the original polar coding scheme, we also successfully recover our message $\boldsymbol{m}'$. Therefore, in this encoding / decoding scheme, provided that we can find $\boldsymbol{b}$ efficiently, we retain the low encoding and decoding complexities and more importantly, the performance guarantees of polar codes.

It remains to understand how small the imbalance $\mu(\boldsymbol{c}' + \boldsymbol{b})$ is. To do so, we define the *guaranteed imbalance* as the quantity

$$\mu(\mathbb{X}, \mathbb{B}) \triangleq \max_{\boldsymbol{c}' \in \mathbb{X}} \left(\min_{\boldsymbol{b} \in \mathbb{B}} \mu(\boldsymbol{c}' + \boldsymbol{b}) \right),$$

where $\mathbb{X}$ is some subset $\mathbb{F}_2^n$. When $\mathbb{X} = \mathbb{F}_2^n$, we simply write $\mu(\mathbb{B})$, instead of $\mu(\mathbb{F}_2^n, \mathbb{B})$.

In this paper, we are interested in imbalance guarantees, or equivalently, determining upper bounds for $\mu(\mathbb{X}, \mathbb{B})$ for various choices of $\mathbb{X}$ and $\mathbb{B}$. Before stating our results, we discuss differences with the original scheme of Mazumdar *et al.* [7].

(a) $\mathbb{B}$ *need not be a* balancing set *in strictest sense*. As mentioned in the introduction, to reduce errors in DNA data storage, it suffices to have DNA strings with small GC-imbalance. Therefore, instead of requiring $\mu(\mathbb{B}) = 0$, we focus on finding a set $\mathbb{B}$ so that $\mu(\mathbb{B}) = o(n)$. We loosely refer to $\mathbb{B}$ as a balancing set.

(b) $\mathbb{X}$ *need not be the whole space* $\mathbb{F}_2^n$. In the balancing encoding scheme, we are only required to reduce the imbalance of the words in the polar code $\mathbb{C}(\mathcal{F}, \boldsymbol{f})$. In other words, it suffices to determine $\mu(\mathbb{C}(\mathcal{F}, \boldsymbol{f}), \mathbb{B})$. However, as the choice of $\mathcal{F}$ and $\boldsymbol{f}$ is channel dependent, it is tedious to determine the guaranteed imbalance for all possibilities. Instead, we determine $\mu(\mathbb{X}, \mathbb{B})$ for a large space $\mathbb{X}$ that is likely to contain $\mathbb{C}(\mathcal{F}, \boldsymbol{f})$. Then the guaranteed imbalance $\mu(\mathbb{C}(\mathcal{F}, \boldsymbol{f}), \mathbb{B})$ is simply upper bounded by $\mu(\mathbb{X}, \mathbb{B})$. Furthermore, we consider spaces $\mathbb{X}$ that are easily described by the polar transformation matrix. Specifically, for the spaces $\mathbb{X}$ considered in this paper, either we have $\mathbb{X} = \mathbb{C}(\mathcal{F}^*, \boldsymbol{f}^*)$ for some choice of frozen indices or we provide an efficient mapping of messages into $\mathbb{X} \subseteq \mathbb{C}(\mathcal{F}^*, \boldsymbol{f}^*)$ using the polar transformation matrix $\boldsymbol{G}_m$. These descriptions allow one to incorporate the balancing technique into the polar encoding. In all cases, the indices in set $\mathcal{F}^*$ have weight at most two and thus, are very likely to belong to an arbitrary frozen set $\mathcal{F}$.

(c) $\mathbb{B}$ *corresponds to the linear space spanned by rows of* $\boldsymbol{G}_m$. This requirement is specific to polar codes and allows one to simply read off the original message $\boldsymbol{m}'$ from the vector obtained from polar decoding. Also, as with the point (b), we study balancing sets that are likely to be *not frozen*.

m	n	$\log \lvert\mathbb{X}_1\rvert$	$\mu(\mathbb{X}_1, \mathbb{B}_m)$	$\log \lvert\mathbb{X}_2\rvert$	$\mu(\mathbb{X}_2, \mathbb{B}_m)$	$\log \lvert\mathbb{X}_3\rvert$	$\mu(\mathbb{X}_3, \mathbb{B}_m)$
2	4	3	0	–	–	–	–
3	8	7	0	–	–	–	–
4	16	15	4	14	0	–	–
5	32	31	$\leqslant 4$	30	0	–	–
6	64	63	8	62	$\leqslant 4$	≈ 50	0
7	128	127	$\leqslant 8$	126	$\leqslant 8$	≈ 112	0
8	256	255	16	254	$\leqslant 12$	≈ 238	$\leqslant 8$
9	512	511	$\leqslant 20$	510	$\leqslant 20$	≈ 492	$\leqslant 16$
10	1024	1023	32	1022	$\leqslant 28$	≈ 1002	$\leqslant 24$

TABLE I: Guaranteed imbalance $\mu(\mathbb{X}, \mathbb{B}_m)$ for various choices of $\mathbb{X}$.

C. Our Contributions

In this paper, we focus on a family of balancing sets. For $m \geqslant 2$ and $1 \leqslant r \leqslant m - 1$, we let $\mathcal{B}_{r,m}$ be the set of indices with weight between $m - r$ and $m - 1$ (inclusive), and set $\mathbb{B}_{r,m} = \mathrm{span}(\mathcal{B}_{r,m})$. Since $\mathcal{J}_{m-r} = \mathcal{B}_{r,m} \cup 1^m$, we have that $\mathbb{RM}(r, m) = \mathbb{B}_{r,m} \oplus \mathrm{span}(1^n)$. In other words, $\mathbb{B}_{r,m}$ is closely related to the Reed-Muller codes and we make use of certain combinatorial properties of the latter to derive estimates on guaranteed imbalance. Furthermore, when r is fixed, it can be shown that the rows or channels corresponding to $\mathcal{B}_{r,m}$ are "good" asymptotically; thus they are likely to be part of the information indices. We omit the details for lack of space.

In Section III, we study the special case where $r = 1$. Here, we simply write $\mathcal{B}_m$ and $\mathbb{B}_m$, instead of $\mathcal{B}_{1,m}$ and $\mathbb{B}_{1,m}$, respectively. Our first result states that $\mu(\mathbb{B}_m) \leqslant \sqrt{n} = o(n)$. Since $|\mathcal{B}_m| = m = \log n$, this means that we can significantly reduce the imbalance of *all codewords* by sacrificing only $\log n$ information bits. Also, we show that we can find the corresponding codeword with the smallest imbalance in $O(n \log n)$ time.

Even though the proportion of imbalance vanishes to zero with codelength, the imbalance is relatively high for small code lengths. Hence, in Section IV, we focus on attention of certain constrained spaces $\mathbb{X}$ and analyse the corresponding guaranteed imbalances. Specifically, we show that for appropriate choices of $\mathbb{X}$, we can achieve *zero imbalance* for $m \leqslant 7$ or $n \leqslant 128$. We summarize our results in Table I.

In Section V, we reduce the guaranteed imbalance by consider $\mathbb{B}_{r,m}$ for $r > 1$. We derive a simple recursive upper bound for $\mu(\mathbb{B}_{r,m})$ and demonstrate the $\mathbb{B}_{m-3,m}$ is able to balance all even-weight codewords for $m \geqslant 5$.

Example 1. Let $m = 4$. So, $n = 16$ and the polar transformation matrix $\boldsymbol{G}_4$ is given by

$$\begin{array}{l}0000\\1000\\0100\\1100\\0010\\1010\\0110\\1110\\0001\\1001\\0101\\1101\\0011\\1011\\0111\\1111\end{array}\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\1&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0\\1&0&1&0&1&0&1&0&0&0&0&0&0&0&0&0\\1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\1&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\1&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0\\1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0\\1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0\\1&1&0&0&1&1&0&0&1&1&0&0&1&1&0&0\\1&0&1&0&1&0&1&0&1&0&1&0&1&0&1&0\\1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\end{bmatrix}$$

The row indices in $\mathcal{B}_4$ are highlighted in blue. In Section III, we show that the guaranteed imbalance $\mu(\mathbb{B}_4)$ is four, i.e. for any word $\boldsymbol{x} \in \mathbb{F}_2^n$, we have $\mu(\boldsymbol{x}+\boldsymbol{b}) \leqslant 4$ for some $\boldsymbol{b}$ in $\mathbb{B}_4$. For example, if $\boldsymbol{x} = 0001000100011110$, we can check that $\mu(\boldsymbol{x}+\boldsymbol{b}) = 4$ for all $\boldsymbol{b}$ in $\mathbb{B}_4$.

On the other hand, we consider the space $\mathbb{X}_2$ whose words are of the form $\boldsymbol{x}_1\boldsymbol{x}_2$ with both weights $\mathrm{wt}(\boldsymbol{x}_1)$ and $\mathrm{wt}(\boldsymbol{x}_2)$ odd. In other words, $\mathbb{X}_2 = \mathbb{C}(\{0000, 0001\}, (0, 1))$. Now, $\boldsymbol{x}' \triangleq 0^7 1 0^7 1$ belongs to $\mathbb{X}_2$ and it is not difficult to see that $\mu(\boldsymbol{x}' + 1^8 0^8) = 0$. Furthermore, in Section IV, for all $\boldsymbol{x} \in \mathbb{X}_2$, we show that it is always possible to find $\boldsymbol{b} \in \mathbb{B}_m$ so that $\mu(\boldsymbol{x}+\boldsymbol{b}) = 0$.

Finally, the row indices in $\mathcal{B}_{2,4}$ (which include $\mathcal{B}_4$) are highlighted in either blue or magenta. In Section V, we show that $\mathbb{B}_{2,4}$ balances all even-weight words, That is, if $\boldsymbol{x}$ has even weight, then we can find $\boldsymbol{b} \in \mathbb{B}_{2,4}$ such that $\mu(\boldsymbol{x}+\boldsymbol{b}) = 0$. For instance, when $\boldsymbol{x} = 0001000100011110$ as above, we can choose $\boldsymbol{b}' = 1^4 0^{12} \in \mathbb{B}_{2,4}$ and check that $\mu(\boldsymbol{x}+\boldsymbol{b}') = 0$.

III. $\mu(\mathbb{B}_m)$ and the Walsh Spectrum

For $m \geqslant 2$, set $n = 2^m$. Recall that $\mathcal{B}_m$ is the set of indices with weight exactly $m-1$ and $\mathbb{B}_m = \mathrm{span}(\mathcal{B}_m)$. In this section, we derive an upper bound for $\mu(\mathbb{B}_m)$.

As mentioned earlier, $\mathrm{span}(\mathcal{B}_m \cup \{1^n\})$ is equal to the first order RM codes. Hence, since the covering radius of the latter is at most $(n - \sqrt{n})/2$ [19], it can be argued that $\mu(\mathbb{B}_m)$ is at most $\sqrt{n}$. There are many derivations of the upper bound for the covering radius of $\mathbb{RM}(1, m)$ and we briefly outline one proof that relies on the *Hadamard transform* and *Walsh spectrum*. The proof illustrates certain combinatorial properties that we exploit to reduce the imbalance in later sections.

Definition 2. For $m \geqslant 2$, set $n = 2^m$. The *Hadamard matrix of order m* is an integer-valued symmetric $n \times n$ matrix defined by the following recursion. Set $\boldsymbol{H}_2 = \left[\begin{smallmatrix}1&1\\1&-1\end{smallmatrix}\right]$ and for $m \geqslant 3$, $\boldsymbol{H}_m \triangleq \begin{bmatrix}\boldsymbol{H}_{m-1} & \boldsymbol{H}_{m-1}\\ \boldsymbol{H}_{m-1} & -\boldsymbol{H}_{m-1}\end{bmatrix}$. For $\boldsymbol{x} = x_1 x_2 \ldots x_n \in \mathbb{F}_2^n$, we define the *Walsh spectrum of* $\boldsymbol{x}$, denoted by $\boldsymbol{W}(\boldsymbol{x})$, to be the integer vector $\boldsymbol{H}_m \boldsymbol{u}$, where $\boldsymbol{u} = ((-1)^{x_i})_{i=1}^n$.

From the definition of Hadamard matrices, we observe that

$$\boldsymbol{H}_m \begin{bmatrix}\boldsymbol{u}_1^T\\ \boldsymbol{u}_2^T\end{bmatrix} = \begin{bmatrix}\boldsymbol{H}_m\boldsymbol{u}_1^T + \boldsymbol{H}_m\boldsymbol{u}_2^T\\ \boldsymbol{H}_m\boldsymbol{u}_1^T - \boldsymbol{H}_m\boldsymbol{u}_2^T\end{bmatrix} \tag{3}$$

Using (3) and standard divide-and-conquer techniques, we can compute the Walsh spectrum efficiently.

Theorem 3 ([17, Ch. 14])**.** *Let $\boldsymbol{x} \in \mathbb{F}_2^n$. Then the Walsh spectrum of $\boldsymbol{x}$ can be computed in $O(n \log n)$ time.*

The next theorem provides the connection between Walsh spectrum and imbalances.

Theorem 4 ([17, Ch. 14])**.** *Let $\boldsymbol{x} \in \mathbb{F}_2^n$ and write the Walsh spectrum of $\boldsymbol{x}$ as $\boldsymbol{W}(\boldsymbol{x}) = (W_1, W_2, \ldots, W_n)$. Then the n absolute values in the Walsh spectrum $|W_i|$, $i = 1, 2, \ldots, n$, correspond to the n imbalances $\mu(\boldsymbol{x}+\boldsymbol{b})$, $\boldsymbol{b} \in \mathbb{B}_m$.*

Therefore, finding the smallest imbalance is equivalent to determining the smallest absolute value in the Walsh spectrum of $\boldsymbol{x}$. To provide an upper bound for the smallest value, we consider the square of the norm of $\boldsymbol{W}(\boldsymbol{x})$, in other words, $||\boldsymbol{W}(\boldsymbol{x})||^2 \triangleq \sum_{i=1}^n W_i^2$. Since $\boldsymbol{H}_m^2 = n\boldsymbol{I}_n$, we have that

$$||\boldsymbol{W}(\boldsymbol{x})||^2 = \boldsymbol{W}(\boldsymbol{x})^T \boldsymbol{W}(\boldsymbol{x}) = \boldsymbol{u}^T \boldsymbol{H}_m^T \boldsymbol{H}_m \boldsymbol{u} = n\boldsymbol{u}^T\boldsymbol{u} = n^2. \tag{4}$$

Therefore, the average squared value in the Walsh spectrum is n and so, the smallest absolute value is at most $\sqrt{n}$. When m is even, it can be shown that there exist words $\boldsymbol{x}$ with $\boldsymbol{W}(\boldsymbol{x}) = (\sqrt{n})_{i=1}^n$. Thus the upper bound is essentially sharp.

We summarize our discussion in the following theorem.

Theorem 5. *$\mu(\mathbb{B}_m) \leqslant \lfloor\sqrt{n}\rfloor = \lfloor 2^{m/2}\rfloor$. Furthermore, we have equality when m is even.*

Remark 6. Here, we point out that the reduction in imbalance is nontrivial. Assuming a symmetric channel, if a polar code is properly designed, then each bit in a random codeword is either 0 or 1 with probability $1/2$. Thus, Hoeffding's inequality guarantees that the probability that the imbalance of a random codeword exceeding tn is at most $2e^{-2nt^2}$. Now, taking $t = 1/\sqrt{n}$, we have the probability that a random codeword has imbalance at least $\sqrt{n}$ is about about $2e^{-2} \approx 0.27$. In contrast, the method in this section ensures all transmitted codewords have imbalance at most $\sqrt{n}$ and this comes with a small rate penalty of $\log n$ bits.

IV. Reducing the Imbalance for Certain Spaces

In this section, we demonstrate that if we constrain the space $\mathbb{X}$, we are able to reduce the guaranteed imbalance. Our first result makes use of the following lemma which is implied by (1).

Lemma 7. *Let $n \equiv 0 \pmod 4$. Then*

$$\mu(\boldsymbol{x}) \equiv \begin{cases} 0 \pmod 4, & \text{when } \mathrm{wt}(\boldsymbol{x}) \text{ is even},\\ 2 \pmod 4, & \text{when } \mathrm{wt}(\boldsymbol{x}) \text{ is odd}.\end{cases}$$

Now, for $m \geqslant 2$, n is always divisible by four. Hence, if $\boldsymbol{x}$ has even weight, it is necessary that its imbalance is divisible by four and the following result is straightforward.

Theorem 8. *Set $\mathbb{X}_1 \triangleq \mathbb{C}(\{\boldsymbol{0}\}, (0))$. Then $\mu(\mathbb{X}_1, \mathbb{B}_m) \leqslant 4\lfloor\sqrt{n}/4\rfloor$ for $m \geqslant 2$.*

Next, we consider a smaller space $\mathbb{X}_2 \subset \mathbb{X}_1$ and show that the guaranteed imbalance can be reduced further.

Theorem 9. *Set $\mathbb{X}_2 \triangleq \mathbb{C}(\{\boldsymbol{0}, 0^{m-1}1\}, (0, 1))$. If $\delta(n) = \lfloor(\sqrt{n}-2)/4\rfloor$, then $\mu(\mathbb{X}_2, \mathbb{B}_m) \leqslant 4\delta(n)$ for $m \geqslant 3$.*

Proof. Let $\boldsymbol{x} = x_1 x_2 \ldots x_n \in \mathbb{X}_2$. Set $\boldsymbol{u}_1 = ((-1)^{x_i})_{i=1}^{n/2}$, $\boldsymbol{u}_2 = ((-1)^{x_i})_{i=n/2+1}^n$ and $\boldsymbol{W}^{(j)} = \boldsymbol{H}_{m-1}\boldsymbol{u}_j$ for $j \in \{1, 2\}$. It follows from the definition of $\mathbb{X}_2$ that both $x_1 x_2 \ldots x_{n/2}$ and $x_{n/2+1} x_{n/2+2} \ldots x_n$ have odd weight. Therefore, Lemma 7 states that both Walsh spectra $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ have entries congruent to two modulo four.

Now, let $j \in \{1, 2\}$. We claim that strictly more than half the entries in $\boldsymbol{W}^{(j)}$ have absolute values at most $4\delta(n) + 2$. We

prove by contradiction. Suppose otherwise that half of the entries in $\boldsymbol{W}^{(j)}$ have absolute values strictly greater than $4\delta(n)+2$. Then

$$||\boldsymbol{W}^{(j)}||^2 \geqslant \frac{n}{4}(4\delta(n)+6)^2 > \frac{n}{4}\left(4\left(\frac{\sqrt{n}-2}{4}-1\right)+6\right)^2 = \frac{n^2}{4}.$$

However, (4) states that $||\boldsymbol{W}^{(j)}||^2 = (n/2)^2 = n^2/4$ and hence, we obtain a contradiction.

As such, we have a common index i where the absolute values of both $W_i^{(1)}$ and $W_i^{(2)}$ are at most $4\delta(n)+2$. Since $\boldsymbol{W}(\boldsymbol{x}) = \left(\boldsymbol{W}^{(1)}+\boldsymbol{W}^{(2)}, \boldsymbol{W}^{(1)}-\boldsymbol{W}^{(2)}\right)$, either $W_i^{(1)}+W_i^{(2)}$ or $W_i^{(1)}-W_i^{(2)}$ has absolute value at most $4\delta(n)+2-2 = 4\delta(n)$. □

Theorem 9 guarantees that $\mu(\mathbb{X}_2, \mathbb{B}_m) = 0$ for $m \in \{4, 5\}$. For larger values of m, we consider another choice of constrained space to reduce the guaranteed imbalance. For $m \geqslant 5$, we consider the extended Hamming code $\mathbb{H}_m \triangleq \mathbb{RM}(m-2, m)$. The constrained space $\mathbb{X}_3$ is defined as follows:

$$\mathbb{X}_3 \triangleq \left\{\boldsymbol{x}_1\boldsymbol{x}_2 : \boldsymbol{x}_j \in \mathbb{H}_{m-1}, \mathrm{wt}(\boldsymbol{x}_j) \equiv 2 \pmod 4 \text{ for } j \in \{1,2\}\right\}. \quad (5)$$

To analyse $\mu(\mathbb{X}_3, \mathbb{B}_m)$, we consider the dual relation of $\mathbb{H}_m$ with $\mathbb{B}_m$ and consider the imbalances modulo eight.

Lemma 10. *Let $m \geqslant 5$. If $\boldsymbol{x} \in \mathbb{H}_m$ and $\boldsymbol{b} \in \mathbb{B}_m$, then*

$$\mu(\boldsymbol{x}+\boldsymbol{b}) \equiv \begin{cases} 4 \pmod 8, & \text{when } \mathrm{wt}(\boldsymbol{x}) \equiv 2 \pmod 4, \\ 0 \pmod 8, & \text{when } \mathrm{wt}(\boldsymbol{x}) \equiv 0 \pmod 4. \end{cases}$$

Proof. From (2), we have that $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{b}) = \mathrm{wt}(\boldsymbol{x}) + \mathrm{wt}(\boldsymbol{b}) - 2\mathrm{wt}(\boldsymbol{x}\cap\boldsymbol{b})$. Since $\boldsymbol{b} \in \mathbb{B}_m \subset \mathbb{RM}(1,m)$ and $\mathbb{RM}(1,m)$ is the dual code for $\mathbb{H}_m$, we have that $\langle \boldsymbol{x}, \boldsymbol{b}\rangle = 0$ and so, $2\mathrm{wt}(\boldsymbol{x}\cap\boldsymbol{b}) \equiv 0 \pmod 4$. For $m \geqslant 5$, since $\mathrm{wt}(\boldsymbol{b}) \equiv 0 \pmod 4$, we have $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{b}) \equiv \mathrm{wt}(\boldsymbol{x}) \pmod 4$. The lemma follows from (1). □

Therefore, for $\boldsymbol{x}_1\boldsymbol{x}_2 \in \mathbb{X}_3$, the values in the Walsh spectra of both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are congruent to four modulo eight. Proceeding in a similar manner as the proof of Theorem 9, we have the following result on the guaranteed imbalance.

Theorem 11. *Define $\mathbb{X}_3$ as in (5). If $\epsilon(n) = \lfloor(\sqrt{n}-4)/8\rfloor$, then $\mu(\mathbb{X}_3, \mathbb{B}_m) \leqslant 8\epsilon(n)$ for $m \geqslant 6$.*

It remains to estimate the size of $|\mathbb{X}_3|$.

Proposition 12. $|\mathbb{H}_m| \geqslant 2^{n-m-2} - 2^{n/2-1}$. *Therefore,*

$$|\mathbb{X}_3| \geqslant \left(2^{n/2-m-1} - 2^{n/4-1}\right)^2 \quad (6)$$

The proof of Proposition 12 is technical and relies on certain known properties of RM codes. First, we consider the span of certain rows in the polar matrix $\boldsymbol{G}_m$. For $m \geqslant 2$, define

$$\mathcal{D}_m \triangleq \begin{cases} \mathcal{J}_{(m+1)/2}, & \text{if } m \text{ is odd}, \\ \mathcal{J}_{m/2+1} \cup \{\boldsymbol{z}'1 : \mathrm{wt}(\boldsymbol{z}') = m/2-1\}, & \text{if } m \text{ is even}. \end{cases} \quad (7)$$

It is straightforward to verify that $|\mathcal{D}_m| = 2^{m-1} = n/2$. Let $\mathbb{D}_m \triangleq \mathrm{span}(\mathcal{D}_m)$. The next lemma states that $\mathbb{D}_m$ is self-dual.

Lemma 13. *$\mathbb{D}_m$ is a self-dual code.*

Proof. When m is odd, $\mathbb{D}_m$ is $\mathbb{RM}((m-1)/2, m)$ and the latter is known to be self-dual. When m is even, $\mathbb{D}_m$ is the direct sum $\mathbb{RM}(m/2-1, m) \oplus \mathbb{D}'$, where $\mathbb{D}' \subset \mathbb{RM}(m/2, m)$. It is also known that the dual of $\mathbb{RM}(m/2-1, m)$ is $\mathbb{RM}(m/2, m)$ [17, Ch. 13].

Now, to show that $\mathbb{D}_m$ is self-dual, we need to demonstrate that $\langle \boldsymbol{x}, \boldsymbol{x}'\rangle = 0$ for any vectors $\boldsymbol{x}$, $\boldsymbol{x}'$ in $\mathbb{D}_m$. We write $\boldsymbol{x} = \boldsymbol{u}+\boldsymbol{v}$ and $\boldsymbol{x}' = \boldsymbol{u}'+\boldsymbol{v}'$ with $\boldsymbol{u}, \boldsymbol{u}' \in \mathbb{RM}(m/2-1, m)$ and $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{D}'$. Then $\langle \boldsymbol{x}, \boldsymbol{x}'\rangle = \langle \boldsymbol{u}, \boldsymbol{u}'\rangle + \langle \boldsymbol{u}, \boldsymbol{v}'\rangle + \langle \boldsymbol{v}, \boldsymbol{u}'\rangle + \langle \boldsymbol{v}, \boldsymbol{v}'\rangle = \langle \boldsymbol{v}, \boldsymbol{v}'\rangle$. Note that $\langle \boldsymbol{u}, \boldsymbol{u}'\rangle = \langle \boldsymbol{u}, \boldsymbol{v}'\rangle = \langle \boldsymbol{v}, \boldsymbol{u}'\rangle = 0$ because of the duality of $\mathbb{RM}(m/2-1, m)$ and $\mathbb{RM}(m/2, m)$. Next, since $\boldsymbol{v} = \boldsymbol{y}\boldsymbol{y}$ and $\boldsymbol{v}' = \boldsymbol{y}'\boldsymbol{y}'$, we have that $\langle \boldsymbol{v}, \boldsymbol{v}'\rangle = 2\langle \boldsymbol{y}, \boldsymbol{y}'\rangle = 0$. □

Proof of Proposition 12. Consider the set of binary words:

$$\mathcal{M} \triangleq \left\{\boldsymbol{m}_1\boldsymbol{m}_2 : \boldsymbol{m}_1 \in \mathbb{F}_2^{n/2-m-1} \setminus \{\boldsymbol{0}\}, \boldsymbol{m}_2 \in \mathbb{F}_2^{n/2-1}\right\}.$$

For our proof, we present an injective map / encoding $\phi : \mathcal{M} \to \mathbb{H}_m$ such that $\mathrm{wt}(\phi(\boldsymbol{m}_1\boldsymbol{m}_2)) \equiv 2 \pmod 4$. This then implies the lower bound as $|\mathcal{M}| = 2^{n/2-1}(2^{n/2-m-1}-1) = 2^{n-m-2} - 2^{n/2-1}$. To compute $\phi(\boldsymbol{m}_1\boldsymbol{m}_2)$, we proceed as follows. Recall $\mathcal{F}_1$ is the set of indices with weight $\leqslant 1$ and we set $\mathcal{I} = \mathcal{Z}\setminus(\mathcal{F}_1 \cup \mathcal{D}_m)$. Hence, $|\mathcal{F}_1| = m+1$, $|\mathcal{I}| = n/2-m-1$, $|\mathcal{D}_m| = n/2$.

- Let $\boldsymbol{x}_1$ be such that $\boldsymbol{x}_1|_{\mathcal{I}} = \boldsymbol{m}_1$ and $\boldsymbol{x}_1|_{\mathcal{F}_1\cup\mathcal{D}_m} = \boldsymbol{0}$. Set $\boldsymbol{c}_1 \triangleq \boldsymbol{x}_1\boldsymbol{G}_m$.
- Since $\boldsymbol{x}_1|_{\mathcal{I}} \neq \boldsymbol{0}$, we have that $\boldsymbol{c}_1 \notin \mathbb{D}_m$. Now, as $\mathbb{D}_m$ is self-dual, there exists a row $\boldsymbol{v}$ with index in $\mathcal{D}_m$ such that $\langle \boldsymbol{c}_1, \boldsymbol{v}\rangle = 1$. Set j^* be the smallest such index.
- Let $\boldsymbol{x}_2$ be such that $\boldsymbol{x}_2|_{\mathcal{D}_m\setminus\{j^*\}} = \boldsymbol{m}_2$ and $\boldsymbol{x}_2|_{\mathcal{F}_1\cup\mathcal{I}\cup\{j^*\}} = \boldsymbol{0}$. Set $\boldsymbol{c}_2 \triangleq \boldsymbol{x}_2\boldsymbol{G}_m$.
- If $\mathrm{wt}(\boldsymbol{c}_1+\boldsymbol{c}_2) \equiv 2 \pmod 4$, we return $\boldsymbol{c}_1+\boldsymbol{c}_2$. Note that $\boldsymbol{c}_1+\boldsymbol{c}_2$ belongs to $\mathbb{H}_m$ as $\boldsymbol{x}_1|_{\mathcal{F}_1} = \boldsymbol{x}_2|_{\mathcal{F}_1} = \boldsymbol{0}$.
- If $\mathrm{wt}(\boldsymbol{c}_1+\boldsymbol{c}_2) \equiv 0 \pmod 4$, we return $\boldsymbol{c}_1+\boldsymbol{c}_2+\boldsymbol{v}$. Since $\boldsymbol{v} \in \mathbb{D}_m \subseteq \mathbb{H}_m$, the word $\boldsymbol{c}_1+\boldsymbol{c}_2+\boldsymbol{v}$ belongs to $\mathbb{H}_m$ too. To check its weight, recall that $\mathrm{wt}(\boldsymbol{c}_1+\boldsymbol{c}_2+\boldsymbol{v}) = \mathrm{wt}(\boldsymbol{c}_1+\boldsymbol{c}_2)+\mathrm{wt}(\boldsymbol{v})-2\mathrm{wt}((\boldsymbol{c}_1+\boldsymbol{c}_2)\cap\boldsymbol{v})$. Now, $\mathrm{wt}(\boldsymbol{c}_1+\boldsymbol{c}_2) \equiv \mathrm{wt}(\boldsymbol{v}) \equiv 0 \pmod 4$. Also, $\mathrm{wt}((\boldsymbol{c}_1+\boldsymbol{c}_2)\cap\boldsymbol{v}) \equiv \langle \boldsymbol{c}_1+\boldsymbol{c}_2, \boldsymbol{v}\rangle = \langle \boldsymbol{c}_1, \boldsymbol{v}\rangle + \langle \boldsymbol{c}_2, \boldsymbol{v}\rangle \equiv 1 \pmod 2$. Therefore, $\mathrm{wt}(\boldsymbol{c}_1+\boldsymbol{c}_2+\boldsymbol{v}) \equiv 2 \pmod 4$.

To show injectivity, we consider $\phi(\boldsymbol{m}_1\boldsymbol{m}_2) = \boldsymbol{c}$ and demonstrate how to recover $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ from $\boldsymbol{c}$. First, compute $\boldsymbol{x} = \boldsymbol{c}\boldsymbol{G}_m$. Since $\boldsymbol{G}_m\boldsymbol{G}_m = \boldsymbol{I}_n$, we have that $\boldsymbol{x} = \boldsymbol{c}_1+\boldsymbol{c}_2$ or $\boldsymbol{x} = \boldsymbol{c}_1+\boldsymbol{c}_2+\boldsymbol{v}$. In either case, $\boldsymbol{x}|_{\mathcal{I}} = \boldsymbol{m}_1$ and we can then figure out j^*. Subsequently, we have that $\boldsymbol{x}|_{\mathcal{D}_m\setminus\{j^*\}} = \boldsymbol{m}_2$. □

The lower bound (6) can be improved as long as we have a code $\mathbb{D}^*$ satisfying certain properties. Specifically, we have the following proposition and its proof follows from the proof of Proposition 12 by replacing $\mathbb{D}_m$ with $\mathbb{D}^*$. However, while $|\mathbb{D}^*| \leqslant 2^{n/2}$, it is unclear how to find $\mathbb{D}^*$ efficiently.

Proposition 14. *Suppose that $\mathbb{D}^*$ is a maximal linear subcode of $\mathbb{H}_m$ whose words have weight divisible by four. That is, if $\boldsymbol{c} \in \mathbb{H}_m \setminus \mathbb{D}^*$, then there is a word in $\mathrm{span}(\mathbb{D}^* \cup \{\boldsymbol{c}\})$ whose weight is not divisible by four. Then the number of words in $\mathbb{H}_m$ is at least $2^{n-m-2} - |\mathbb{D}^*|/2$.*

V. Reducing the Imbalance using $\mathbb{B}_{r,m}$ for $r > 1$

In this section, instead of constraining the space $\mathbb{X}$, we reduce the guaranteed imbalance by augmenting the balancing set. Specifically, we provide upper bounds for the guaranteed imbalance for $\mathbb{B}_{r,m}$ when $r > 1$. Now, since $\mathbb{RM}(r,m) = \mathbb{B}_{r,m} \oplus \mathrm{span}(1^n)$ and $\mu(\boldsymbol{x}) = \mu(\boldsymbol{x}+1^n)$, we have that $\mu(\mathbb{B}_{r,m}) = \mu(\mathbb{RM}(r,m))$. Hence, it suffices to compute the guaranteed imbalance for RM codes and this is convenient as RM codes have many well-known properties.

First, we have the following recursive upper bound.

Proposition 15. $\mu(\mathbb{B}_{r+1,m+1}) \leqslant \mu(\mathbb{B}_{r,m})$.

Proof. We show that $\mu(\mathbb{RM}(r+1, m+1)) \leqslant \mu(\mathbb{RM}(r,m))$. Let $\boldsymbol{x}$ be a binary word of length n whose imbalance is to be reduced.

In other words, if we write $\mu(\mathbb{RM}(r,m)) = \mu_0$, our task is to find $\boldsymbol{b} \in \mathbb{RM}(r+1, m+1)$ such that $\mu(\boldsymbol{x}+\boldsymbol{b}) \leqslant \mu_0$.

If $\mathrm{wt}(\boldsymbol{x}) = n/2$, then $\mu(\boldsymbol{x}) = 0 \leqslant \mu_0$ and we choose $\boldsymbol{b} = \boldsymbol{0}$. Otherwise, we have that $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{b}) \notin \{0, n\}$ for all $\boldsymbol{b} \in \mathbb{B}_{1,m+1} \subseteq \mathbb{RM}(r+1, m+1)$. We compute the Walsh spectrum of $\boldsymbol{x}$. Again, if $\mu(\boldsymbol{x}+\boldsymbol{b}) = 0$ for some $\boldsymbol{b} \in \mathbb{B}_{1,m+1}$, we simply choose the corresponding $\boldsymbol{b}$.

Hence, in the worst case, $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{b}) \notin \{0, n/2, n\}$ for all $\boldsymbol{b} \in \mathbb{B}_{1,m+1}$. Since there are $(n+1)-3 = n-2$ possible weights and the number of words in $\mathbb{B}_{1,m+1}$ is $2^{m+1} = n$, we must have $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{a}) = \mathrm{wt}(\boldsymbol{x}+\boldsymbol{b})$ for some $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{B}_{1,m+1}$. Using the automorphism group of the RM codes [17, Ch. 13], we can assume that

$$\boldsymbol{a} = 0^{n/4}1^{n/4}1^{n/4}0^{n/4}, \text{ and,} \boldsymbol{b} = 1^{n/4}0^{n/4}1^{n/4}0^{n/4}.$$

Now, we divide $\boldsymbol{x}$ into four blocks of equal length (i.e. $n/4$) and let w_i denote the weight of the ith block for $i \in \{1,2,3,4\}$. Since $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{a}) = w_1 + w_4 + (n - w_2 - w_3)$, and $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{b}) = w_2 + w_4 + (n - w_1 - w_3)$, we have that $w_1 = w_2$.

We then consider $\boldsymbol{c} = 1^{n/4}0^{3n/4} \in \mathbb{B}_{2,m+1} \subseteq \mathbb{RM}(r+1, m+1)$ and check that the number of ones in the first $n/2$ coordinates of $\boldsymbol{x}+\boldsymbol{c}$ is exactly $(n/4 - w_1) + w_2 = n/4$. In other words, the prefix of length $n/2$ of $\boldsymbol{x}+\boldsymbol{c}$ has zero imbalance.

Let $\boldsymbol{x}'$ be the suffix of $\boldsymbol{x}$ of length $n/2$. Since $\mu(\mathbb{RM}(r,m)) = \mu_0$, we can find $\boldsymbol{d}$ in $\mathbb{RM}(r,m)$ such that $\mu(\boldsymbol{x}'+\boldsymbol{d}) \leqslant \mu_0$. Since $\mathbb{RM}(r+1, m+1) = \{(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v}) : \boldsymbol{u} \in \mathbb{RM}(r+1, m), \boldsymbol{v} \in \mathbb{RM}(r,m)\}$ [17, Ch. 13], the word $\boldsymbol{0d}$ belongs to $\mathbb{RM}(r+1, m+1)$. Therefore, if we choose the balancing word to be $\boldsymbol{c}+\boldsymbol{0d} \in \mathbb{RM}(r+1, m+1)$, then the resulting imbalance $\mu(\boldsymbol{x}+\boldsymbol{c}+\boldsymbol{0d}) = \mu(\boldsymbol{x}+\boldsymbol{c}) + (\boldsymbol{x}'+\boldsymbol{d}) \leqslant \mu_0$, as desired. □

We apply Proposition 15 recursively to obtain the next result.

Corollary 16. *$\mu(\mathbb{B}_{r,m}) \leqslant \left\lfloor 2^{(m-r+1)/2} \right\rfloor$ for $1 \leqslant r \leqslant m$.*

An immediate consequence of Corollary 16 is that $\mu(\mathbb{B}_{2,4}) \leqslant 2$. Recall that $\mathbb{X}_1$ is the set of all words with even weight and it is now immediate that $\mu(\mathbb{X}_1, \mathbb{B}_{2,4}) = 0$. In other words, $\mathbb{B}_{2,4}$ or $\mathbb{RM}(2,4)$ balances all words of even weight. The next proposition states that $\mathbb{RM}(2,5)$ also balances all words of even weight.

Proposition 17. *$\mu(\mathbb{X}_1, \mathbb{B}_{2,4}) = \mu(\mathbb{X}_1, \mathbb{B}_{2,5}) = 0$.*

We defer the proof to the next subsection where we allude to certain geometric properties of RM codes. Nevertheless, a direct consequence of Propositions 15 and 17 is the following corollary.

Corollary 18. *$\mu(\mathbb{X}_1, \mathbb{B}_{m-3,m}) = 0$ for $m \geqslant 5$.*

Now, for all $m \geqslant 2$, define the quantity

$$\rho(m) \triangleq \min\{r : \mu(\mathbb{X}_1, \mathbb{B}_{r,m}) = 0\}. \tag{8}$$

That is, $\rho(m)$ is the smallest value r such that $\mathbb{B}_{r,m}$ balances all even-weight words. The results of the paper state that

$$\rho(2) = \rho(3) = 1,\ \rho(4) = \rho(5) = 2,\ 1 < \rho(m) \leqslant m-3 \text{ for } m \geqslant 6.$$

For $m \geqslant 4$, we ran computer simulations and it seems that $\mathbb{B}_{2,m}$ is always able to balance an even-weight word. Hence, we make the following conjecture.

Conjecture. $\rho(m) = 2$ for all $m \geqslant 4$.

A. Proof of Proposition 17

In this subsection, we show that $\mathbb{RM}(2,5)$ balances all words of even weight. To do so, we recall certain classical characterizations of RM codewords. Fix $m \geqslant 2$ and index the codewords in $\mathbb{RM}(r,m)$ with $\mathbb{F}_2^m$. For $X \subseteq \mathbb{F}_2^m$, we use $\boldsymbol{c}(X)$ to denote the binary word whose support is exactly X. In other words, if $\boldsymbol{c}(X) = (c_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha} \in \mathbb{F}_2^m}$, we have that $c_{\boldsymbol{\alpha}} = 1$ if and only if $\boldsymbol{\alpha} \in X$.

Lemma 19 ([17, Ch. 13])**.** *If $X \subseteq \mathbb{F}_2^m$ is an affine space with dimension at least $m-r$, then $\boldsymbol{c}(X)$ belongs to $\mathbb{RM}(r,m)$.*

Using this characterization, the next lemma tells us when a word in $\mathbb{RM}(1,m)$ is able to balance a certain even-weight word.

Lemma 20. *Let $X \subseteq \mathbb{F}_2^m$ with $|X|$ even. If U is a linear space of dimension $m-1$ such that $|U \cap X| = |X|/2$, then $\boldsymbol{c}(U) \in \mathbb{RM}(1,m)$ balances $\boldsymbol{c}(X)$.*

The next lemma follows from the covering radius of $\mathbb{RM}(2,5)$.

Lemma 21 ([19])**.** *For $\boldsymbol{x} \in \mathbb{X}_1$, there exists $\boldsymbol{c} \in \mathbb{RM}(2,5)$ such that $\mathrm{wt}(\boldsymbol{x}+\boldsymbol{c}) \in \{0,2,4,6\}$.*

We are now ready to show that $\mathbb{RM}(2,5)$ balances all words of even weight. Our broad strategy is as follows.

- Let X be the support of the word to be balanced. From Lemma 21, it suffices to assume that $|X| \in \{0,2,4,6\}$.
- To apply Lemma 20, we find a four-dimensional linear space U such that $|U \cap X| = |X|/2$. In most cases, this is always possible and in what follows, we simply state the space U and omit the detailed verification of the intersection size. When it is not possible to find U, we modify X so that Lemma 20 is applicable.
- We assume that X contains $\boldsymbol{0}$. Otherwise, we can shift it using the automorphism group of $\mathbb{RM}(2,5)$. Also, the vectors $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, and $\boldsymbol{\epsilon}$ are assumed to be linearly independent.

We consider the following cases according to the size of X.

- If $|X| = 0$, we pick any four-dimensional space U.
- If $|X| = 2$, then $X = \{\boldsymbol{0}, \boldsymbol{\alpha}\}$. Pick the three-dimensional space U such that $\boldsymbol{\alpha} \notin U$. For example, we can set $U = \mathrm{span}(\{\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\})$.
- When $|X| = 4$, we have the following two cases. If the rank of X is two, then $X = \{\boldsymbol{0}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\alpha}+\boldsymbol{\beta}\}$ and we pick $U = \mathrm{span}(\{\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\})$. If the rank of X is three, then $X = \{\boldsymbol{0}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\}$ and we pick $U = \mathrm{span}(\{\boldsymbol{\alpha}+\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\})$.
- When $|X| = 6$, we have the following four cases.

(i) If the rank of X is three, we pick a three-dimensional space V that contains X. So, $\boldsymbol{c}(V)$ belongs to $\mathbb{RM}(2,5)$ and $\boldsymbol{c}(V) + \boldsymbol{c}(U)$ has weight two. We then proceed as before to balance the word $\boldsymbol{c}(V) + \boldsymbol{c}(U)$.

(ii) Suppose that the rank of X is four and $X = \{\boldsymbol{0}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma}+\boldsymbol{\delta}\}$. Here, we set $V = \mathrm{span}(\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\epsilon}\})$ and so, $\boldsymbol{c}(V)$ belongs to $\mathbb{RM}(2,5)$. Let X' be the support of $\boldsymbol{c}(X) + \boldsymbol{c}(X)$ and here, $|X'| = 8$. Now, if we pick the four-dimensional space $U = \mathrm{span}(\{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}\})$, then $|X' \cap U| = 4$. Therefore, Lemma 20 states that $\boldsymbol{c}(X') + \boldsymbol{c}(U) = \boldsymbol{c}(X) + \boldsymbol{c}(V) + \boldsymbol{c}(U)$ is balanced.

(iii) Suppose that the rank of X is four and X contains $\{\boldsymbol{0}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}\}$ with the sixth vector not equal to the sum of the five nonzero vectors. Then we may assume that the sixth vector is $\boldsymbol{\beta} + f(\boldsymbol{\gamma}, \boldsymbol{\delta})\}$ where $f(\boldsymbol{\gamma}, \boldsymbol{\delta})$ is a linear combination of $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$, not involving $\boldsymbol{\alpha}$. In this case, we pick $U = \mathrm{span}(\{\boldsymbol{\alpha}+\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\})$.

(iv) If the rank of X is five, then $X = \{\boldsymbol{0}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\}$ and we pick $U = \mathrm{span}(\{\boldsymbol{\alpha}+\boldsymbol{\beta}, \boldsymbol{\alpha}+\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\epsilon}\})$.

ACKNOWLEDGEMENT

We thank the reviewers for the insightful comments. In particular, Remark 6 was highlighted by an anonymous reviewer.

REFERENCES

[1] K. A. S. Immink, *Codes for Mass Data Storage Systems*. Shannon Foundation Publisher, 2004.

[2] S. Yazdi, H. M. Kiah, E. R. Garcia, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: Trends and methods," *IEEE Trans. Molecular, Biological, Multi-Scale Commun.*, vol. 1, no. 3, pp. 230–248, 2015.

[3] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty, C. Nusbaum, D. B. Jaffe, "Characterizing and Measuring Bias in Sequence Data", *Genome Biol. 14*, R51, 2013.

[4] K. A. S. Immink and K. Cai, "Properties and constructions of constrained codes for DNA-based data storage," *arXiv preprint arXiv:1812.06798*, 2018.

[5] H. Van Tilborg and M. Blaum, "On error-correcting balanced codes," *IEEE Trans. Inform. Theory*, vol. 35, no. 5, pp. 1091–1095, 1989.

[6] S. Al-Bassam and B. Bose, "Design of efficient error-correcting balanced codes," *IEEE Trans. Computers*, vol. 42, no. 10, pp. 1261–1266, 1993.

[7] A. Mazumdar, R. M. Roth, and P. O. Vontobel, "On linear balancing sets," *Adv. Math. Commun.* vol. 4, no. 3,pp. 345–361, 2010.

[8] J. H. Weber, K. A. S. Immink, and H. C. Ferreira, "Error-correcting balanced knuth codes," *IEEE Trans. Inform. Theory*, vol. 58, no. 1, pp. 82–89, 2012.

[9] Y. M. Chee, H. M. Kiah, and H. J. Wei, "Efficient and Explicit Balanced Primer Codes," *arXiv preprint arXiv:1901.01023*, 2019.

[10] K. Cai, Y. M. Chee, R. Gabrys, H. M. Kiah, and T. T. Nguyen, "Optimal Codes Correcting a Single Indel / Edit for DNA-Based Data Storage", *arXiv preprint arXiv:1910.06501*, 2019.

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

[12] D. Kane, *Samsung Licenses 5G Polar Coding Technology Developed by UC San Diego Engineers*, UC San Diego Press Release, October 2018. [Online]. Available as: https://ucsdnews.ucsd.edu/pressrelease/samsung_licenses_5g_polar_coding_technology_developed_by_uc_san_diego_engineers

[13] K. Tian, A. Fazeli, and A. Vardy, "Polar coding for deletion channels: Theory and implementation", in *Proc. IEEE Int. Symp. Inform. Theory*, 2018, pp. 1869–1873.

[14] I. Tal, H. D. Pfister, A. Fazeli, and A. Vardy, "Polar Codes for the Deletion Channel: Weak and Strong Polarization", *arXiv preprint arXiv:1904.13385*, 2019.

[15] E. Şaşoğlu, "Polarization in the presence of memory," in *Proc. IEEE Symp. Inform. Theory*, 2011, pp. 189–193.

[16] B. Shuval and I. Tal. "Fast Polarization for Processes with Memory." *IEEE Trans. Inf. Theory*, vol. 65, no. 4, pp. 2004–2020, 2019.

[17] F. J. MacWilliams, and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[18] E. Abbe and M. Ye, "Reed-Muller codes polarize," [Online]. *arXiv preprint arXiv:1901.11533*, 2019.

[19] G. D. Cohen, and S. N. Litsyn, "On the covering radius of Reed-Muller codes," *Discrete Mathematics*, 106, pp. 147–155, 1992.