Optimizing Linear Correctors: A Tight Output Min-Entropy Bound and Selection Technique

Miloš Grujić and Ingrid Verbauwhede

***Abstract*—Post-processing of the raw bits produced by a true random number generator (TRNG) is always necessary when the entropy per bit is insufficient for security applications. In this paper, we derive a tight bound on the output min-entropy of the algorithmic post-processing module based on linear codes, known as linear correctors. Our bound is based on the codes' weight distributions, and we prove that it holds even for the real-world noise sources that produce independent but not identically distributed bits. Additionally, we present a method for identifying the optimal linear corrector for a given input min-entropy rate that maximizes the throughput of the post-processed bits while simultaneously achieving the needed security level. Our findings show that for an output min-entropy rate of 0.999, the extraction efficiency of the linear correctors with the new bound can be up to $130.56\,\%$ higher when compared to the old bound, with an average improvement of $41.2\,\%$ over the entire input min-entropy range. On the other hand, the required min-entropy of the raw bits for the individual correctors can be reduced by up to $61.62\,\%$.**

***Index Terms*—Entropy, true random number generator, post-processing, linear correctors.**

I. Introduction

RANDOM numbers produced directly by a noise source of a true random number generator (TRNG) – raw random numbers, are rarely ideal. In order to be considered ideal and possess full entropy, random numbers should be independent, identically and uniformly distributed. However, raw random numbers often display dependencies, biases, and a lack of identical distribution. Therefore, before using them for critical security and cryptographic applications, these numbers should be subjugated to entropy extraction (post-processing) to increase the entropy content per random bit to an acceptable level. An important figure-of-merit of the post-processing algorithms is the extraction efficiency, which represents the ratio of the output to the input entropy. According to the US standard for TRNGs, referred to as entropy sources in the standard, NIST SP 800-90B [1], the raw random numbers can be post-processed (conditioned) by either using one of the six vetted conditioning algorithms or by using custom algorithms with appropriate entropy estimation. On the other hand, German AIS-31 [2], [3], which has emerged as the leading TRNG standard and evaluation methodology within the European Union [4], categorizes post-processing methods into two main types: cryptographic and algorithmic post-processing.

While the main role of cryptographic post-processing is to ensure computational security [2], [3], it is also used to increase the entropy rate (entropy per bit) of the random numbers. To achieve this enhancement, it is crucial for the cryptographic post-processing to be compressive. The well-understood and widely used cryptographic hash functions and block ciphers, as building blocks of one-way compression functions, can be used for this purpose. The security and entropy of the output from the cryptographic post-processing can be derived by modeling it as a random mapping, as discussed in [2], [3]. Since the random mapping behavior is a theoretical idealization, the entropy estimation of the output relies on the computational security of the used underlying cryptographic primitive. Cryptographic post-processing is not tailored to any specific distribution family of the raw random numbers. It can often be attractive from a practical perspective in security systems that already have software or dedicated hardware implementations of cryptographic primitives. However, using cryptographic primitives for the sole purpose of post-processing can also be prohibitively expensive. Most noise sources produce raw numbers at rates significantly lower than the operating frequencies of modern CPUs [5]–[8]. Consequently, the cryptographic post-processing tasks would require a considerable amount of processing time due to the resulting latency. In digital platforms with dedicated cryptographic accelerators, all non-TRNG applications that require their use would be precluded from employing them during the post-processing. Further, performing cryptographic operations can be power- or energy-expensive, thereby increasing the overall cost of randomness.

Algorithmic post-processing entails using straightforward and lightweight functions often adapted to the stochastic model of the noise source and the family of raw bit distributions [2], [3]. Unlike cryptographic post-processing, the algorithmic methods provide information-theoretical security and the output entropy can often be precisely determined. This post-processing is inherently future-proof when used appropriately, as new and improved cryptanalytic techniques cannot compromise its security. For the noise sources that produce independent and identically distributed (IID) bits, the well-known Von Neumann unbiasing [9] can be used as algorithmic post-processing to obtain the full entropy output. While Von Neumann's procedure's maximum extraction efficiency of only

The authors are with COSIC, KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven – Heverlee, Belgium (e-mail: milos.grujic@esat.kuleuven.be; ingrid.verbauwhede@esat.kuleuven.be).

This work was partially supported by CyberSecurity Research Flanders with reference number VR20192203, the European Commission through Horizon 2020 research and innovation program under Belfort ERC Advanced Grant 101020005 and Twinning Grant SAFEST 952252, through Horizon Europe program under grant agreement No.101114043 (QSNP) and through Digital Europe Program together with the Belgian Federal Science Policy Office (Belspo) through the Federal restart and transition plan under grant agreement No.101091625 (BE-QCI).

0.25 can be increased by its generalizations – Peres' [10] and Elias' [11] unbiasing methods, this comes at a much greater computational cost. Additional practical disadvantages of these constructions are their variable output rate and the strict IID requirement, which might be impossible to achieve with real-world TRNGs. Another commonly used algorithmic post-processing method is the simple XOR function of n consecutive bits, which reduces the bias of independent but not necessarily identically distributed raw bits at the cost of n-fold throughput reduction [12]. While this post-processing can never achieve full entropy of the output bits, it can increase the entropy rate to the desired amount, has a fixed output rate and very low implementation costs.

In [13], Dichtl proposed several XOR-based post-processing constructions for IID bits with higher extraction efficiency than the basic XOR function due to the reuse of input bits. These constructions were later formalized as *linear correctors* by Lacharme in [14], [15], who also gave a lower bound on the min-entropy of their output. Linear correctors are represented by the mappings of the form:

$$\boldsymbol{Y}^{k\times 1} = \boldsymbol{G}^{k\times n}\boldsymbol{X}^{n\times 1}, \tag{1}$$

where $\boldsymbol{X}^{n\times 1}$ and $\boldsymbol{Y}^{k\times 1}$ are column vectors of n input and k output bits, respectively, $\boldsymbol{G}^{k\times n}$ is a generator matrix of a binary linear code with minimum distance d and multiplication is performed in the Galois field of size 2. If all input bits have bias δ, then the lower bound on the min-entropy of the output of the linear corrector can be derived as [14]:

$$\mathrm{H}_{\infty}^{out,\,tot} \geq k - \log_2\left(1+\delta^d 2^{k+d}\right). \tag{2}$$

In subsequent works [16], [17], it was shown that the linear correctors could also be used on the independent raw bits that are not identically distributed. A slightly modified version of Lacharme's bound, which includes a lower bound on min-entropy of independent raw bits H_{∞}^{in}, was given in [17]:

$$\mathrm{H}_{\infty}^{out,\,tot} \geq k - \log_2\left(1+\left(2^{1-\mathrm{H}_{\infty}^{in}}-1\right)^d\cdot 2^k\right). \tag{3}$$

Linear correctors are recognized by the RISC-V consortium [18], [19] as a form of admissible non-cryptographic post-processing and are recommended to be used in several recent TRNG designs [17], [20]–[24]. They represent an attractive post-processing method due to a significantly smaller hardware footprint compared to cryptographic post-processing [25], the ability to deal with not identically distributed raw random bits and higher extraction efficiency than simple XOR function [13], [14]. Refining the output min-entropy bound of the corrector can prevent the unnecessary dissipation of entropy from raw bits during the post-processing stage, thereby enhancing the performance of TRNG designs that incorporate linear correctors.

A. Our Contributions

In this work, we noticeably improve Lacharme's previously established min-entropy bound of the linear corrector's output. The improvement is achieved by first establishing new relations between the probabilities of a linear code and its cosets. These relations are then used to gain new insights into the connection between the weight distribution of a binary linear code and the linear corrector's output probabilities. We show that our new bound is also suitable for TRNGs whose noise sources produce independent and non-identically distributed raw bits. To demonstrate the applicability of this newly established result, we devise an optimization procedure to select linear correctors that achieve the best trade-off between the necessary input min-entropy rate and the throughput reduction to obtain the desired output min-entropy rate. We leverage the existing knowledge of the best known linear codes and known weight distributions to find the optimal performing linear correctors. Our newly introduced bound enables us to find linear correctors that are up to 130.56% more efficient in entropy extraction compared to those derived from the previous bound for an equivalent input min-entropy. Across the entire examined input min-entropy range, the new bound averages an enhancement in extraction efficiency by 41.2%. We have made the list of optimal performing correctors according to the new bound available at [26], along with the weight distributions of their corresponding codes and the input min-entropies required to use them. This resource is intended to help TRNG designers in selecting appropriate post-processing techniques and to facilitate the reproduction of our work.

II. Preliminaries

In this section, we introduce notation, basic definitions and necessary background in coding theory. For a more in-depth treatment of the coding theory fundamentals, we recommend referring to [27] and [28] along with their respective references.

A. Notations and Definitions

We denote binary vectors with bold lowercase italic letters and matrices with bold uppercase italic letters. Calligraphic uppercase letters represent random variables, while the uppercase italic letters are reserved for denoting sets. The i-th bit from the left of an n-bit vector $\boldsymbol{x}$ is denoted as $\boldsymbol{x}[i]$ and is referred to as the i coordinate of $\boldsymbol{x}$. The Hamming weight of a binary vector $\boldsymbol{x}$ is the number of coordinates of $\boldsymbol{x}$ equal to 1 and we denote it by $\mathrm{HW}(\boldsymbol{x})$. We use $\mathbf{1}_{l_0}$ to denote a bit vector characterized by having a value of 1 exclusively at the l_0 coordinate and zeros elsewhere. The probability of an event is denoted with $\mathbb{P}[\cdot]$. Let S be some set of n-bit vectors $\boldsymbol{x}$, which are realizations of an n-bit discrete random variable $\mathcal{X}$ with independent coordinates. The probability of set S is then defined as the sum of the occurrence probabilities of its element vectors, i.e.,

$$\mathbb{P}[S] = \sum_{\boldsymbol{x}\in S}\prod_{i=0}^{n-1}\left(\boldsymbol{x}[i]\,p_i + (1-\boldsymbol{x}[i])\,(1-p_i)\right), \tag{4}$$

where $p_i = \mathbb{P}[\mathcal{X}[i]=1]$, $0\leq i\leq n-1$, and p_i is called the 1-probability of bit in coordinate i. $\mathcal{X}$ is an independent and identically distributed (IID) random variable (source) only when p_i is identical for all n bits of $\mathcal{X}$. In this work, we use

min-entropy as a post-processing performance measure, as it is the most conservative uncertainty quantity and is used by both NIST SP 800-90B [1] and the latest version of AIS-31 standards [3]. The min-entropy of a discrete random variable $\mathcal{R}$, with the outcomes from the set R, is defined as

$$\mathrm{H}_\infty = -\log_2\left(\max_{r\in R}\mathbb{P}\left[\mathcal{R}=r\right]\right). \tag{5}$$

In this work, we formally define the extraction efficiency of the post-processing algorithm as

$$\eta = \frac{\mathrm{H}_\infty^{out,\,tot}}{n\,\mathrm{H}_\infty^{in}}, \tag{6}$$

where $\mathrm{H}_\infty^{out,\,tot}$ is the total entropy at the output, n is the number of input raw bits and H_∞^{in} is the lower bound on the min-entropy rate of the raw bits. We also define post-processing throughput reduction as the ratio of the number of input bits versus the number of output bits.

B. Coding Theory

A binary linear code C_0 of length n and dimension k is a k-dimensional subspace of the vector space $\mathbb{F}_2^n$. Hence, C_0 is a set of order 2^k of n-bit row vectors called codewords that form a group under the operation of bitwise modulo 2 addition ($\oplus$). A minimum distance of a binary linear code is the smallest Hamming weight of the non-zero codewords. A binary linear code C_0 of length n, dimension k and minimum distance d is called $[n,k,d]$-code or just $[n,k]$-code when properties of a code can be generalized independently of d. Quantity k/n is called the code rate.

Example: Consider a $[3,2]$-code. Here, $n=3$ and $k=2$. A potential code could be $C_0^A=\{000,110,101,011\}$, which forms a 2-dimensional subspace in $\mathbb{F}_2^3$. The minimum distance of this code is 2, as that is the smallest Hamming weight among the non-zero codewords 110, 101, and 011. Another potential $[3,2]$-code could be $C_0^B=\{000,110,100,010\}$. The minimum distance of this code is 1, as that is the smallest Hamming weight among the non-zero codewords 110, 100, and 010.

The list of non-negative integers $(\mathfrak{A}_i)_{i=0}^n$, where $\mathfrak{A}_i$ is the number of codewords of Hamming weight i in a $[n,k]$-code C_0, is called the weight distribution of the code.

Example: For the $[3,2]$-code C_0^A provided earlier, the weight distribution is $\mathfrak{A}_0=1$, $\mathfrak{A}_1=0$, $\mathfrak{A}_2=3$ and $\mathfrak{A}_3=0$ since there is one codeword of weight 0, zero codewords of weight 1, and three codewords of weight 2.

For any binary linear code and for any given coordinate, either all codewords have a 0 at that coordinate or exactly half of them [28]. A generator matrix $\boldsymbol{G}$ of an $[n,k]$-code C_0 is a binary $k\times n$ full rank matrix whose rows are k linearly independent codewords of C_0.

Example: Let us consider our $[3,2]$-code C_0^A again. When we look at the first coordinate, two codewords have a 1 $(110,101)$ and the other two have a 0 $(000,011)$. A possible generator matrix $\boldsymbol{G}$ for this code could be: $\left(\begin{smallmatrix}1&0&1\\0&1&1\end{smallmatrix}\right)$. This matrix represents two linearly independent codewords from C_0^A. If we consider $[3,2]$-code C_0^B and look at the third coordinate, we see that all codewords have a 0 at this coordinate.

A full rank $(n-k)\times n$ binary matrix $\boldsymbol{H}$ such that for all codewords $\boldsymbol{c}$ of an $[n,k]$-code C_0 it holds $\boldsymbol{H}\boldsymbol{c}^\intercal=\boldsymbol{0}$ is called a parity-check matrix of C_0. For any n-bit vector $\boldsymbol{x}$, the parity-check matrix determines the syndrome of $\boldsymbol{x}$ as $\boldsymbol{s}=\boldsymbol{H}\boldsymbol{x}^\intercal$. A binary linear $[n,n-k]$-code $C_0^\perp$ whose generator matrix is the parity-check matrix of C_0 is called the dual code of C_0. $C_0^\perp$ is the null space of C_0, i.e., for any codeword $\boldsymbol{c}$ of C_0 and any codeword $\boldsymbol{c}^\perp$ of its dual code $C_0^\perp$ it holds $\sum_{i=0}^{n-1}\boldsymbol{c}[i]\boldsymbol{c}^\perp[i]=0$, where additions and multiplications are in $\mathbb{F}_2$. The weight distribution of the dual code $\left(\mathfrak{A}_i^\perp\right)_{i=0}^n$ is called the dual weight distribution and it is related to the weight distribution $(\mathfrak{A}_i)_{i=0}^n$ of the C_0 code by the MacWilliams identity [28], [29]:

$$\sum_{i=0}^{n}\mathfrak{A}_i z^i = 2^{k-n}\left(1+z\right)^n\sum_{i=0}^{n}\mathfrak{A}_i^\perp\left(\frac{1-z}{1+z}\right)^i. \tag{7}$$

Example: Assuming the generator matrix $\boldsymbol{G}$ mentioned above, a parity-check matrix $\boldsymbol{H}$ for our $[3,2]$-code C_0^A is: $\left(\begin{smallmatrix}1&1&1\end{smallmatrix}\right)$. This matrix ensures that for all codewords $\boldsymbol{c}$ in C_0^A, $\boldsymbol{H}\boldsymbol{c}^\intercal=\boldsymbol{0}$. Matrix $\boldsymbol{H}$ is at the same time generator matrix of the dual $[3,3-2]$ code $C_0^{A,\perp}=\{000,111\}$ with weight distribution $\mathfrak{A}_0^\perp=1$, $\mathfrak{A}_1^\perp=0$, $\mathfrak{A}_2^\perp=0$ and $\mathfrak{A}_3^\perp=1$.

For a binary linear $[n,k]$-code C_0 and an n-bit vector $\boldsymbol{a}$, the set $\{\boldsymbol{a}\oplus\boldsymbol{c}\mid\boldsymbol{c}\in C_0\}$ is called a coset of C_0. Two n-bit vectors are in the same coset if and only if they have an identical syndrome. Hence, a syndrome uniquely determines a coset. A coset leader is the element with the smallest Hamming weight in its coset. If there are multiple elements with the same minimal Hamming weight, any of them can be selected to be the coset leader. We will also sometimes refer to the set of codewords C_0 as a coset, with the all-zero vector being its unique coset leader. The total number of cosets of an $[n,k]$-code is 2^{n-k}, including the set of codewords.

Example: Let us continue with our $[3,2]$-code C_0^A and consider the vector $\boldsymbol{a}=100$. The coset for this vector will be: $\{100,010,001,111\}$. This is the result of 100 xored with each codeword in C_0^A. The coset leader can be any of the vectors with the smallest Hamming weight. In this case, any of the weight 1 vectors 100, 010, or 001 could be chosen. The total number of cosets of our $[3,2]$-code C_0^A would be $2^{3-2}=2$, meaning that no other 3-bit vector $\boldsymbol{a}$ produces a new coset.

Since there is an equivalence between binary linear codes and linear correctors [14], we will sometimes interchangeably use the terms corrector and code.

III. Previous Work

The relationship between a code's weight distribution and the output of a linear corrector was first noted by Lacharme in [15], although the previously established min-entropy bound in [14] was not improved. Zhou et al. [30], [31] studied the exact, average, and asymptotic performance of linear correctors and more general random binary matrices, but only in terms of their statistical distance from the uniform distribution, without considering the entropy rate. In [25], Kwok et al. compared the performance of Von Neumann unbiasing, XOR function, and linear correctors with respect to throughput reduction, post-processed bit bias, and adversarial bias reduction. However,

their study did not consider the performance of these post-processing techniques for non-identically distributed input bits, nor did it account for the correlation between the output bits of a linear corrector and, therefore, the total entropy of the output. In contrast, Meneghetti et al. [32] and Tomasi et al. [16] provided a bound on the statistical distance of linear correctors' output from the uniform distribution based on the code's weight distribution, and they also determined a lower bound on the Shannon entropy using Sason's theorem [33], which relates statistical distance and entropy. However, this bound is loose because it relies on the statistical distance bound and does not apply to the min-entropy, which is always lower than the Shannon entropy.

In the following section, we will use and expand on two older results from coding theory to improve Lacharme's bound: Sullivan's subgroup-coset inequality [34] and its generalization by Živković [35]. Sullivan showed in [34] that when all coordinate 1-probabilities of n-bit vectors are smaller than 0.5, the probability of the set of codewords is the highest among all coset probabilities. Živković later demonstrated in [35] that this relation also holds for any q-ary linear code, where q is a prime power, even when individual coordinate 1-probabilities are different but all smaller than 0.5.

IV. Improving the Min-entropy Bound

We improve the min-entropy bound for linear correctors by first generalizing Sullivan's subgroup-coset inequality [34] for binary linear codes and cases when the coordinate 1-probabilities are different and not upper limited to 0.5. First, we recall a lemma from [34] that will also be used in our proofs.

Lemma 1 (adapted from [34])**.** Let C_0 be a binary linear $[n, k]$-code, and let $\boldsymbol{e}$, $\mathrm{HW}(\boldsymbol{e}) = l$, be a coset leader in some coset of C_0. Then the code C_0', obtained by deleting l coordinates in which $\boldsymbol{e}$ is 1, is a binary linear $[n-l, k]$-code.

We now introduce our first inequality theorem, named the *coset-coset inequality*. This theorem establishes a relationship between the probabilities of two distinct cosets belonging to a specific binary linear code. It offers a distinctive perspective when compared to the subgroup-coset inequalities proposed by Sullivan and Živković. The proof of this theorem builds upon the foundations laid out in [34] and [35].

Theorem 1. Let C_i, $0 \leq i \leq 2^{n-k}-1$, denote sets of n-bit element vectors $\boldsymbol{x}$, which are realizations of the n-bit row vector random variable $\mathcal{X}$ with independent coordinates. Let C_0 be a binary linear $[n, k]$-code, and all other C_i, $i \neq 0$, are cosets of C_0. Let $C_{i_{max}}$ denote the set that contains the most probable element vector $\boldsymbol{x_{max}}$ with coordinates

$$\boldsymbol{x_{max}}[j] = \begin{cases} 1, \text{ if } 0.5 \leq p_j \leq 1, \\ 0, \text{ if } 0 \leq p_j < 0.5, \end{cases}$$

where p_j is the 1-probability of bit in j coordinate, $0 \leq j \leq n-1$. Then it holds $\mathbb{P}[C_{i_{max}}] \geq \mathbb{P}[C_i]$, and we call $C_{i_{max}}$ the most probable set.

Proof. First, we arrange all possible 2^n vectors in the standard array such that the i-th row contains elements of the set C_i. The first entry in each row $\boldsymbol{c_{i,0}}$ is a coset leader $\boldsymbol{e_i}$, i.e., a vector with the lowest weight in the corresponding set, while all other row entries $\boldsymbol{c_{i,j}}$ are obtained by adding $\boldsymbol{e_i}$ and the corresponding entry in the 0-th row: $\boldsymbol{c_{i,j}} = \boldsymbol{e_i} \oplus \boldsymbol{c_{0,j}}$, $1 \leq j \leq 2^k - 1$. Consider now the set that contains the most probable vector $C_{i_{max}}$ with coset leader $\boldsymbol{e_{i_{max}}}$ and some arbitrary but fixed set C_i, $C_i \neq C_{i_{max}}$, with coset leader $\boldsymbol{e_i}$, as well as their corresponding rows in the standard array. If $\boldsymbol{e_i} \oplus \boldsymbol{e_{i_{max}}}$ is in some set C_l, but is not equal to its coset leader $\boldsymbol{e_l}$, we rearrange the entries in the i_{max}-th row so that for the first entry $\boldsymbol{c_{i_{max},0}}$ we select an element of $C_{i_{max}}$ that is equal to $\boldsymbol{e_i} \oplus \boldsymbol{e_l}$. All other row entries are rearranged so that the j-th element is equal to $\boldsymbol{c_{i_{max},j}} = \boldsymbol{c_{i_{max},0}} \oplus \boldsymbol{c_{0,j}} = \boldsymbol{e_i} \oplus \boldsymbol{e_l} \oplus \boldsymbol{c_{0,j}}$. On the other hand, no rearrangements are made if $\boldsymbol{e_i} \oplus \boldsymbol{e_{i_{max}}} = \boldsymbol{e_l}$ already holds. After possible rearrangement, any entry in the i-th row $\boldsymbol{c_{i,j}}$ is related to the entry $\boldsymbol{c_{i_{max},j}}$ in the i_{max}-th row by relation $\boldsymbol{c_{i,j}} = \boldsymbol{c_{i_{max},j}} \oplus \boldsymbol{e_l}$. Since all entries in the i-th and i_{max}-th row are also elements of the sets C_i and $C_{i_{max}}$, respectively, this shows that every element in $C_{i_{max}}$ has exactly one corresponding element in C_i from which it differs only in coordinates in which $\boldsymbol{e_l}$ is 1. We will prove the theorem by double induction over the code dimension k, $0 \leq k < n$, and the Hamming weight of the coset leaders $\mathrm{HW}(\boldsymbol{e_l}) \geq 1$.

Base case. For $k = 0$ and $\mathrm{HW}(\boldsymbol{e_l}) = 1$, we have $C_0 = \{\mathbf{0}\}$, where $\mathbf{0}$ is the all-zero vector. Since, in this case, each set contains only one n-bit vector, it is clear that the set that includes the most probable vector $\boldsymbol{x_{max}}$ will have probability $\mathbb{P}[C_{i_{max}}] = \mathbb{P}[\boldsymbol{x_{max}}]$ and that $\mathbb{P}[C_{i_{max}}] \geq \mathbb{P}[C_i]$ always holds.

Outer induction hypothesis. Assume that the theorem is true for all binary linear codes of dimension $k \leq k'$ and $\mathrm{HW}(\boldsymbol{e_l}) = 1$.

Outer induction step. We will show that the outer induction hypothesis implies that the theorem also holds for all binary linear codes of dimension $k = k' + 1$ and $\mathrm{HW}(\boldsymbol{e_l}) = 1$. Suppose that $\boldsymbol{e_l}$ has 1 in coordinate l_0 and let $I_{l_0} = \{0, \ldots, n-1\} \setminus \{l_0\}$. We now partition sets C_i and $C_{i_{max}}$ into two subsets, depending on the value in coordinate l_0 of their element vectors: $C_{i_{max}}^{l_0,b} = \{\boldsymbol{x} \in C_{i_{max}} \mid \boldsymbol{x}[l_0] = b\}$ and $C_i^{l_0,b} = \{\boldsymbol{x} \in C_i \mid \boldsymbol{x}[l_0] = b\}$, $b \in \{0, 1\}$.

Case 1a: Suppose first that $\boldsymbol{x}[l_0] = \hat{b}$ holds for all $\boldsymbol{x} \in C_{i_{max}}$, where $\hat{b}$ is fixed to either 0 or 1. Then the order of $C_{i_{max}}^{l_0,\hat{b}}$ is $2^{k'+1}$ since $C_{i_{max}}^{l_0,\hat{b}} = C_{i_{max}}$ and $C_{i_{max}}^{l_0,1-\hat{b}} = \varnothing$. Given that the elements in $C_i^{l_0,1-\hat{b}}$ differ from the elements in $C_{i_{max}}^{l_0,\hat{b}}$ only in the l_0 coordinate, we have that the order of $C_i^{l_0,1-\hat{b}}$ is also $2^{k'+1}$ and $C_i^{l_0,1-\hat{b}} = C_i$, while $C_i^{l_0,\hat{b}} = \varnothing$. Therefore, we can express the probabilities of the sets $C_{i_{max}}$ and C_i as

$$\mathbb{P}[C_{i_{max}}] = \mathbb{P}\left[C_{i_{max}}^{l_0,\hat{b}}\right] = \left(\hat{b}p_{l_0} + \left(1-\hat{b}\right)(1-p_{l_0})\right) \cdot \sum_{\boldsymbol{x} \in C_{i_{max}}} \prod_{i \in I_{l_0}} \left(\boldsymbol{x}[i]\, p_i + (1-\boldsymbol{x}[i])(1-p_i)\right), \quad (8)$$

and

$$\mathbb{P}\left[C_i\right] = \mathbb{P}\left[C_i^{l_0,1-\hat{b}}\right] = \left(\left(1-\hat{b}\right)p_{l_0} + \hat{b}\left(1-p_{l_0}\right)\right) \cdot \sum_{\boldsymbol{x}\in C_i}\prod_{i\in I_{l_0}}\left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right). \quad (9)$$

Note that

$$\sum_{\boldsymbol{x}\in C_i}\prod_{i\in I_{l_0}}\left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right) = \sum_{\boldsymbol{x}\in C_{i_{max}}}\prod_{i\in I_{l_0}}\left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right), \quad (10)$$

holds since the elements in C_i and $C_{i_{max}}$ differ only in the l_0 coordinate.

Subcase 1.1a: For $\hat{b}=0$, it holds $1-p_{l_0} > p_{l_0}$ since $0 \leq p_{l_0} < 0.5$, which follows from the fact that $\boldsymbol{x_{max}} \in C_{i_{max}}$ and all vectors in $C_{i_{max}}$ have 0 in coordinate l_0 for $\hat{b}=0$. Therefore, from (8) and (9), we have the inequality

$$\mathbb{P}\left[C_{i_{max}}\right] = (1-p_{l_0})\sum_{\boldsymbol{x}\in C_{i_{max}}}\prod_{i\in I_{l_0}}\left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right) >$$
$$\mathbb{P}\left[C_i\right] = p_{l_0}\sum_{\boldsymbol{x}\in C_i}\prod_{i\in I_{l_0}}\left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right). \quad (11)$$

Subcase 1.2a: For $\hat{b}=1$, all vectors in $C_{i_{max}}$ have 1 in coordinate l_0 and $\boldsymbol{x_{max}} \in C_{i_{max}}$. Thus, $p_{l_0} \geq 1-p_{l_0}$, since $0.5 \leq p_{l_0} \leq 1$. Hence, $\mathbb{P}\left[C_{i_{max}}\right] \geq \mathbb{P}\left[C_i\right]$ holds in this case as well, which can be seen by substituting $\hat{b}=1$ in (8) and (9):

$$\mathbb{P}\left[C_{i_{max}}\right] = p_{l_0}\sum_{\boldsymbol{x}\in C_{i_{max}}}\prod_{i\in I_{l_0}}\left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right) \geq$$
$$\mathbb{P}\left[C_i\right] = (1-p_{l_0})\sum_{\boldsymbol{x}\in C_i}\prod_{i\in I_{l_0}}\left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right). \quad (12)$$

Case 2a: Suppose the values in coordinate l_0 are not identical for all vectors in $C_{i_{max}}$. The orders of $C_{i_{max}}^{l_0,0}$, $C_{i_{max}}^{l_0,1}$, $C_i^{l_0,0}$ and $C_i^{l_0,1}$ are all equal to $2^{k'}$. We now delete component in coordinate l_0 of every element in both C_i and $C_{i_{max}}$ and denote the resulting sets by $C_i^{\overline{l_0}}$ and $C_{i_{max}}^{\overline{l_0}}$, and the corresponding partitioning subsets by $C_i^{\overline{l_0},0}$, $C_i^{\overline{l_0},1}$, $C_{i_{max}}^{\overline{l_0},0}$ and $C_{i_{max}}^{\overline{l_0},1}$. Since C_i and $C_{i_{max}}$ are either equivalent to C_0 or are its proper cosets, from Lemma 1, we have that the orders of $C_i^{\overline{l_0}}$ and $C_{i_{max}}^{\overline{l_0}}$ are $2^{k'+1}$. Consequently, the orders of $C_i^{\overline{l_0},0}$, $C_i^{\overline{l_0},1}$, $C_{i_{max}}^{\overline{l_0},0}$ and $C_{i_{max}}^{\overline{l_0},1}$ will be $2^{k'}$. Since the elements in C_i differ from the elements in $C_{i_{max}}$ only in the coordinate l_0, it follows that $C_i^{\overline{l_0},0} = C_{i_{max}}^{\overline{l_0},1}$ and $C_i^{\overline{l_0},1} = C_{i_{max}}^{\overline{l_0},0}$. The set probabilities $\mathbb{P}\left[C_{i_{max}}\right]$ and $\mathbb{P}\left[C_i\right]$ can be expressed as

$$\mathbb{P}\left[C_{i_{max}}\right] = \mathbb{P}\left[C_{i_{max}}^{l_0,1}\right] + \mathbb{P}\left[C_{i_{max}}^{l_0,0}\right] = p_{l_0}\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},1}\right] + \left(1-p_{l_0}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},0}\right] \quad (13)$$

and

$$\mathbb{P}\left[C_i\right] = \mathbb{P}\left[C_i^{l_0,1}\right] + \mathbb{P}\left[C_i^{l_0,0}\right] = p_{l_0}\mathbb{P}\left[C_i^{\overline{l_0},1}\right] + \left(1-p_{l_0}\right)\mathbb{P}\left[C_i^{\overline{l_0},0}\right] = p_{l_0}\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},0}\right] + \left(1-p_{l_0}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},1}\right]. \quad (14)$$

Thus, we obtain

$$\mathbb{P}\left[C_{i_{max}}\right] - \mathbb{P}\left[C_i\right] = p_{l_0}\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},1}\right] - \left(1-p_{l_0}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},1}\right] + \left(1-p_{l_0}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},0}\right] - p_{l_0}\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},0}\right] = \left(1-2p_{l_0}\right)\left(\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},0}\right] - \mathbb{P}\left[C_{i_{max}}^{\overline{l_0},1}\right]\right). \quad (15)$$

Subcase 2.1a: If $\boldsymbol{x_{max}} \in C_{i_{max}}^{l_0,0}$, then the most probable $(n-1)$-bit vector obtained from $\boldsymbol{x_{max}}$ by deleting its l_0 coordinate $\boldsymbol{x_{max}^{\overline{l_0}}} = (\boldsymbol{x_{max}}[0] \dots \boldsymbol{x_{max}}[l_0-1]\,\boldsymbol{x_{max}}[l_0+1] \dots \boldsymbol{x_{max}}[n-1])$ will be in the subset $C_{i_{max}}^{\overline{l_0},0}$. Hence, from the induction hypothesis $\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},0}\right] \geq \mathbb{P}\left[C_{i_{max}}^{\overline{l_0},1}\right]$. We note that since $\boldsymbol{x_{max}} \in C_{i_{max}}^{l_0,0}$, we have $0 \leq p_{l_0} < 0.5$, and thus, $1-2p_{l_0} > 0$. Based on this observation and the outer induction hypothesis, we have that both multiplication terms in the last line of (15) are non-negative, implying that $\mathbb{P}\left[C_{i_{max}}\right] \geq \mathbb{P}\left[C_i\right]$.

Subcase 2.2a: If $\boldsymbol{x_{max}} \in C_{i_{max}}^{l_0,1}$, then $\boldsymbol{x_{max}^{\overline{l_0}}}$ will be an element of the subset $C_{i_{max}}^{\overline{l_0},1}$. From the induction hypothesis, in this case, we have $\mathbb{P}\left[C_{i_{max}}^{\overline{l_0},1}\right] \geq \mathbb{P}\left[C_{i_{max}}^{\overline{l_0},0}\right]$. Furthermore, since $\boldsymbol{x_{max}} \in C_{i_{max}}^{l_0,1}$, we have $0.5 \leq p_{l_0} \leq 1$, and thus, $1-2p_{l_0} \leq 0$. Therefore, both terms in the last line of (15) are non-positive, implying that their product is non-negative, and $\mathbb{P}\left[C_{i_{max}}\right] \geq \mathbb{P}\left[C_i\right]$ holds in this case as well.

By induction, the theorem is true for all binary linear codes' dimensions k, $0 \leq k < n$, and $\mathrm{HW}(\boldsymbol{e_l}) = 1$.

Inner induction hypothesis. Assume that the theorem holds for all binary linear codes of dimension k and $\mathrm{HW}(\boldsymbol{e_l})$ values not greater than m.

Inner induction step. We proceed with the second induction step by showing that the inner induction hypothesis implies that the theorem holds for $\mathrm{HW}(\boldsymbol{e_l}) = m+1$ and all binary linear codes of dimension k. Let l_m be one of the $m+1$ possible positions in which $\boldsymbol{e_l}$ has 1, and let $I_{l_m} = \{0,\dots,n-1\} \setminus \{l_m\}$. We separate all elements in both C_i and $C_{i_{max}}$ into two subsets according to their coordinate value in coordinate l_m: $C_{i_{max}}^{l_m,b} = \{\boldsymbol{x} \in C_{i_{max}} \mid \boldsymbol{x}[l_m] = b\}$ and $C_i^{l_m,b} = \{\boldsymbol{x} \in C_i \mid \boldsymbol{x}[l_m] = b\}$, $b \in \{0,1\}$. Let $C_{i_{max}}^{\overline{l_m}}$ and $C_i^{\overline{l_m}}$ be sets obtained from $C_{i_{max}}$ and C_i by removing the component in coordinate l_m in all vectors in both sets. According to Lemma 1, the orders of $C_{i_{max}}^{\overline{l_m}}$ and $C_i^{\overline{l_m}}$ will remain 2^k and the elements in $C_i^{\overline{l_m}}$ will differ from the elements in $C_{i_{max}}^{\overline{l_m}}$ in coordinates in which vector $\boldsymbol{e_l^{\overline{l_m}}} = (\boldsymbol{e_l}[0] \dots \boldsymbol{e_l}[l_m-1], \boldsymbol{e_l}[l_m+1] \dots \boldsymbol{e_l}[n-1])$ is 1. Since the most probable $(n-1)$-bit vector $\boldsymbol{x_{max}^{\overline{l_m}}} = (\boldsymbol{x_{max}}[0], \dots \boldsymbol{x_{max}}[l_m-$

$1], \boldsymbol{x_{max}}[l_m+1], \ldots \boldsymbol{x_{max}}[n-1])$ will be in set $C_{i_{max}}^{\overline{l_m}}$ and $\mathrm{HW}\left(\boldsymbol{e_l^{\overline{lm}}}\right) = m$, by the inner induction hypothesis, we obtain

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] &= \sum_{\boldsymbol{x}\in C_{i_{max}}} \prod_{i\in I_{l_m}} \left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right) \\ &\geq \mathbb{P}\left[C_i^{\overline{l_m}}\right] = \sum_{\boldsymbol{x}\in C_i} \prod_{i\in I_{l_m}} \left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right).\end{aligned} \tag{16}$$

Case 1b: Suppose $\boldsymbol{x}[l_m] = \hat{b}$ holds for all $\boldsymbol{x} \in C_{i_{max}}$, where $\hat{b}$ is fixed to either a 0 or a 1. The order of $C_{i_{max}}^{l_m,\hat{b}}$ is then 2^k and $C_{i_{max}}^{l_m,1-\hat{b}} = \varnothing$. Since $\boldsymbol{e_l}$ has 1 in coordinate l_m, all vectors in C_i will have $1-\hat{b}$ in coordinate l_m. Hence, the order of $C_i^{l_m,1-\hat{b}}$ is also 2^k and $C_i^{l_m,\hat{b}} = \varnothing$. For the probabilities of sets $C_{i_{max}}$ and C_i, we have

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}\right] = \mathbb{P}\left[C_{i_{max}}^{l_m,\hat{b}}\right] = \\ \left(\hat{b}p_{l_m} + \left(1-\hat{b}\right)\left(1-p_{l_m}\right)\right)\cdot \\ \sum_{\boldsymbol{x}\in C_{i_{max}}} \prod_{i\in I_{l_m}} \left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right),\end{aligned} \tag{17}$$

and

$$\begin{aligned}\mathbb{P}\left[C_i\right] = \mathbb{P}\left[C_i^{l_m,1-\hat{b}}\right] = \\ \left(\left(1-\hat{b}\right)p_{l_m} + \hat{b}\left(1-p_{l_m}\right)\right)\cdot \\ \sum_{\boldsymbol{x}\in C_i} \prod_{i\in I_{l_m}} \left(\boldsymbol{x}\left[i\right]p_i + \left(1-\boldsymbol{x}\left[i\right]\right)\left(1-p_i\right)\right).\end{aligned} \tag{18}$$

By substituting $\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right]$ and $\mathbb{P}\left[C_i^{\overline{l_m}}\right]$ from (16) in (17) and (18), and then subtracting $\mathbb{P}\left[C_i\right]$ from $\mathbb{P}\left[C_{i_{max}}\right]$, we obtain

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}\right] - \mathbb{P}\left[C_i\right] = \\ \hat{b}\left(p_{l_m}\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] - \left(1-p_{l_m}\right)\mathbb{P}\left[C_i^{\overline{l_m}}\right]\right) + \\ \left(1-\hat{b}\right)\left(\left(1-p_{l_m}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] - p_{l_m}\mathbb{P}\left[C_i^{\overline{l_m}}\right]\right).\end{aligned} \tag{19}$$

Subcase 1.1b: For $\hat{b} = 0$, since $\boldsymbol{x_{max}} \in C_{i_{max}}$, we have $0 \leq p_{l_m} < 0.5$, thus, $(1-p_{l_m}) > p_{l_m}$. Equation (19) then becomes

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}\right] - \mathbb{P}\left[C_i\right] = \\ \left(1-p_{l_m}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] - p_{l_m}\mathbb{P}\left[C_i^{\overline{l_m}}\right].\end{aligned} \tag{20}$$

By multiplying both sides of (16) by $(1-p_{l_m})$ and combining this result with $(1-p_{l_m}) > p_{l_m}$, we have the inequality

$$\left(1-p_{l_m}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] \geq \left(1-p_{l_m}\right)\mathbb{P}\left[C_i^{\overline{l_m}}\right] > p_{l_m}\mathbb{P}\left[C_i^{\overline{l_m}}\right]. \tag{21}$$

From the preceding inequality and (20), it holds $\mathbb{P}\left[C_{i_{max}}\right] > \mathbb{P}\left[C_i\right]$.

Subcase 1.2b: Similarly, for $\hat{b} = 1$, we have $0.5 \leq p_{l_m} \leq 1$, thus, $p_{l_m} \geq (1-p_{l_m})$ and (19) becomes

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}\right] - \mathbb{P}\left[C_i\right] = \\ p_{l_m}\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] - \left(1-p_{l_m}\right)\mathbb{P}\left[C_i^{\overline{l_m}}\right].\end{aligned} \tag{22}$$

By multiplying both sides of (16) by $(1-p_{l_m})$ and combining this result with the inequality $p_{l_m} \geq (1-p_{l_m})$, we obtain

$$p_{l_m}\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] \geq \left(1-p_{l_m}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] \geq \left(1-p_{l_m}\right)\mathbb{P}\left[C_i^{\overline{l_m}}\right]. \tag{23}$$

From (22) and (23), it follows that $\mathbb{P}\left[C_{i_{max}}\right] \geq \mathbb{P}\left[C_i\right]$ holds in this case as well.

Case 2b: Suppose that $\boldsymbol{x}\left[l_m\right]$ is not identical for all $\boldsymbol{x} \in C_{i_{max}}$. Let $C_{i_{max}}^{\overline{l_m},b}$ and $C_i^{\overline{l_m},b}$, $b \in \{0,1\}$, be subsets of $C_{i_{max}}^{\overline{l_m}}$ and $C_i^{\overline{l_m}}$, respectively, obtained from $C_{i_{max}}^{l_m,b}$ and $C_i^{l_m,b}$ by deleting the l_0 coordinate in the element vectors. We can express the probabilities of sets $C_{i_{max}}^{\overline{l_m}}$ and $C_i^{\overline{l_m}}$ as $\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] = \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right] + \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right]$ and $\mathbb{P}\left[C_i^{\overline{l_m}}\right] = \mathbb{P}\left[C_i^{\overline{l_m},0}\right] + \mathbb{P}\left[C_i^{\overline{l_m},1}\right]$, respectively, and rewrite (16) as

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}^{\overline{l_m}}\right] &= \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right] + \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right] \\ &\geq \mathbb{P}\left[C_i^{\overline{l_m}}\right] = \mathbb{P}\left[C_i^{\overline{l_m},0}\right] + \mathbb{P}\left[C_i^{\overline{l_m},1}\right].\end{aligned} \tag{24}$$

The probabilities $\mathbb{P}\left[C_{i_{max}}\right]$ and $\mathbb{P}\left[C_i\right]$ can be expressed as

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}\right] = \mathbb{P}\left[C_{i_{max}}^{l_m,0}\right] + \mathbb{P}\left[C_{i_{max}}^{l_m,1}\right] = \\ p_{l_m}\mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right] + \left(1-p_{l_m}\right)\mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right]\end{aligned} \tag{25}$$

and

$$\begin{aligned}\mathbb{P}\left[C_i\right] = \mathbb{P}\left[C_i^{l_m,0}\right] + \mathbb{P}\left[C_i^{l_m,1}\right] = \\ p_{l_m}\mathbb{P}\left[C_i^{\overline{l_m},1}\right] + \left(1-p_{l_m}\right)\mathbb{P}\left[C_i^{\overline{l_m},0}\right].\end{aligned} \tag{26}$$

By subtracting (26) from (25), we obtain

$$\begin{aligned}\mathbb{P}\left[C_{i_{max}}\right] - \mathbb{P}\left[C_i\right] = \\ \left(1-p_{l_m}\right)\left(\mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right] - \mathbb{P}\left[C_i^{\overline{l_m},0}\right]\right) \\ - p_{l_m}\left(\mathbb{P}\left[C_i^{\overline{l_m},1}\right] - \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right]\right).\end{aligned} \tag{27}$$

Subcase 2.1b: First, suppose that $\boldsymbol{x_{max}}[l_m] = 0$, i.e., $\boldsymbol{x_{max}} \in C_{i_{max}}^{l_m,0}$. This implies $0 \leq p_{l_m} < 0.5$ and $(1-p_{l_m}) > p_{l_m}$. By multiplying both sides of (24) by $(1-p_{l_m})$ and rearranging the terms, we have

$$\begin{aligned}\left(1-p_{l_m}\right)\left(\mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right] - \mathbb{P}\left[C_i^{\overline{l_m},0}\right]\right) \geq \\ \left(1-p_{l_m}\right)\left(\mathbb{P}\left[C_i^{\overline{l_m},1}\right] - \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right]\right) > \\ p_{l_m}\left(\mathbb{P}\left[C_i^{\overline{l_m},1}\right] - \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right]\right),\end{aligned} \tag{28}$$

where the last inequality comes from $(1-p_{l_m}) > p_{l_m}$. Thus, from (28) and (27), we can see that $\mathbb{P}\left[C_{i_{max}}\right] > \mathbb{P}\left[C_i\right]$ holds.

Subcase 2.2b: Finally, suppose that $\boldsymbol{x_{max}}[l_m] = 1$, i.e., $\boldsymbol{x_{max}} \in C_{i_{max}}^{l_m,1}$. This implies $0.5 \leq p_{l_m} \leq 1$ and $p_{l_m} \geq (1-p_{l_m})$. By multiplying both sides of (24) by p_{l_m} and rearranging the terms, we have

$$\begin{aligned}p_{l_m}\left(\mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right] - \mathbb{P}\left[C_i^{\overline{l_m},1}\right]\right) \geq \\ p_{l_m}\left(\mathbb{P}\left[C_i^{\overline{l_m},0}\right] - \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right]\right) \geq \\ \left(1-p_{l_m}\right)\left(\mathbb{P}\left[C_i^{\overline{l_m},0}\right] - \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right]\right),\end{aligned} \tag{29}$$

where the last inequality comes from $p_{l_m} \geq (1 - p_{l_m})$. By again rearranging the terms in the first and the last line of the inequality (29), we get the inequality

$$(1 - p_{l_m}) \left(\mathbb{P}\left[C_{i_{max}}^{\overline{l_m},0}\right] - \mathbb{P}\left[C_i^{\overline{l_m},0}\right] \right) \geq p_{l_m} \left(\mathbb{P}\left[C_i^{\overline{l_m},1}\right] - \mathbb{P}\left[C_{i_{max}}^{\overline{l_m},1}\right] \right). \quad (30)$$

From (27) and (30), it directly follows $\mathbb{P}\left[C_{i_{max}}\right] \geq \mathbb{P}\left[C_i\right]$.

By the principle of double induction, the theorem is true for all binary linear codes of any dimension k, $0 \leq k < n$, and all Hamming weights of their coset leaders $\mathrm{HW}(\boldsymbol{e_l}) \geq 1$.

□

The results of the coset-coset inequality theorem will be helpful in determining the exact output min-entropy of the linear corrector when the distributions of all raw input bits are precisely known. For most real-world TRNGs, these distributions are unknown during the design time and vary, in some range, between TRNG instances and during the operation. Often, the only thing that can be guaranteed and required by the standardization bodies [1]–[3] is the lower bound on entropy. Hence, to practically apply the finding of Theorem 1, that the most probable coset is the one that contains the most probable vector, we will use it in the following lemma to show how this probability can be bounded.

Lemma 2. Let $C_0\left(p_0, \ldots, p_{n-1}\right)$ be the set of codewords of a binary linear code and $C_{i_{max}}\left(p_0, \ldots, p_{n-1}\right)$ be the most probable set as defined in Theorem 1 with corresponding coordinate 1-probabilities given by tuple $(p_0, \ldots, p_{n-1})$, where all p_i might be different. Let $\delta_{max} = \max\left\{|0.5 - p_i|\right\}_{i=0}^{n-1}$ be the maximum coordinate bit bias, and let $(0.5 - \delta_{max}, \ldots, 0.5 - \delta_{max})$ represent a tuple of coordinate 1-probabilities all equal to $0.5 - \delta_{max}$. Then, it holds $\mathbb{P}\left[C_0\left(0.5 - \delta_{max}, \ldots, 0.5 - \delta_{max}\right)\right] \geq \mathbb{P}\left[C_{i_{max}}\left(p_0, \ldots, p_{n-1}\right)\right]$.

Proof. We will decompose the proof into two cases, depending on whether the most probable vector $\boldsymbol{x_{max}}$ is an all-zero vector, and prove both cases by simple induction.

Case 1: Suppose that the most probable vector $\boldsymbol{x_{max}}$ is the all-zero vector, i.e., all coordinate 1-probabilities p_i, $0 \leq i \leq n-1$, are lower than 0.5 and possibly different from each other. According to Theorem 1, the most probable set will be C_0, i.e., $C_{i_{max}} = C_0$. If in some coordinate l_0, we change its probability p_{l_0} to $p_{l_0}^* = 0.5 - \delta_{max}$, the all-zero vector will remain the most probable vector for the tuple of 1-probabilities $\left(p_0, \ldots, p_{l_0}^*, \ldots, p_{n-1}\right)$ and therefore C_0 remains the most probable set. We partition C_0 into two subsets $C_0^{l_0,\,b} = \{\boldsymbol{x} \in C_0 \mid \boldsymbol{x}[l_0] = b\}$, $b \in \{0, 1\}$, according to the value of the element vectors' coordinate in l_0. We now remove the l_0 coordinate of each element in $C_0^{l_0,\,b}$ and obtain subsets $C_0^{\overline{l_0},\,b}$, $b \in \{0, 1\}$. Note that $\mathbb{P}\left[C_0^{\overline{l_0},\,b}\left(p_0, \ldots, p_{l_0}, \ldots, p_{n-1}\right)\right] = \mathbb{P}\left[C_0^{\overline{l_0},\,b}\left(p_0, \ldots, p_{l_0}^*, \ldots, p_{n-1}\right)\right] = \mathbb{P}\left[C_0^{\overline{l_0},\,b}\right]$, since the vectors in $C_0^{\overline{l_0},\,b}$ do not have coordinate l_0 with modified probability. The probability of C_0 before and after the l_0 coordinate probability change will be

$$\mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}, \ldots, p_{n-1}\right)\right] = p_{l_0}\mathbb{P}\left[C_0^{\overline{l_0},1}\right] + (1 - p_{l_0})\,\mathbb{P}\left[C_0^{\overline{l_0},0}\right], \quad (31)$$

and

$$\mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}^* = 0.5 - \delta_{max}, \ldots, p_{n-1}\right)\right] = p_{l_0}^*\mathbb{P}\left[C_0^{\overline{l_0},1}\right] + \left(1 - p_{l_0}^*\right)\mathbb{P}\left[C_0^{\overline{l_0},0}\right] = (0.5 - \delta_{max})\,\mathbb{P}\left[C_0^{\overline{l_0},1}\right] + (0.5 + \delta_{max})\,\mathbb{P}\left[C_0^{\overline{l_0},0}\right], \quad (32)$$

respectively. By subtracting (31) from (32), we obtain

$$\begin{aligned}&\mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}^* = 0.5 - \delta_{max}, \ldots, p_{n-1}\right)\right] \\ &\quad - \mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}, \ldots, p_{n-1}\right)\right] = \\ &(0.5 - \delta_{max} - p_{l_0})\,\mathbb{P}\left[C_0^{\overline{l_0},1}\right] + (p_{l_0} - 0.5 + \delta_{max})\,\mathbb{P}\left[C_0^{\overline{l_0},0}\right] \\ &= (\delta_{max} - (0.5 - p_{l_0}))\left(\mathbb{P}\left[C_0^{\overline{l_0},0}\right] - \mathbb{P}\left[C_0^{\overline{l_0},1}\right]\right). \quad (33)\end{aligned}$$

The first multiplication term in the last line of (33) is non-negative since, by the definition of δ_{max}, it holds $\delta_{max} \geq 0.5 - p_{l_0}$.

Subcase 1.1: If for all $\boldsymbol{x} \in C_0$ it holds $\boldsymbol{x}[l_0] = 0$, then $C_0^{l_0,0} = C_0$ and $C_0^{l_0,1} = C_0^{\overline{l_0},1} = \varnothing$. This implies that the second multiplication term in (33) is also non-negative, since $\mathbb{P}\left[C_0^{\overline{l_0},1}\right] = 0$ and therefore $\mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}^* = 0.5 - \delta_{max}, \ldots, p_{n-1}\right)\right] \geq \mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}, \ldots, p_{n-1}\right)\right]$.

Subcase 1.2: If $\boldsymbol{x}[l_0]$ is not identical for all $\boldsymbol{x} \in C_0$, we have two additional subcases depending on whether C_0 contains the vector element $\mathbf{1_{l_0}}$ – an n-bit vector with Hamming weight 1 that has a 1 in coordinate l_0.

Subsubcase 1.2.1: If $\mathbf{1_{l_0}} \in C_0$, then every element $\boldsymbol{c}$ in $C_0^{l_0,0}$ has exactly one corresponding element $\boldsymbol{c'}$ in $C_0^{l_0,1}$ to which it is related by $\boldsymbol{c'} = \boldsymbol{c} \oplus \mathbf{1_{l_0}}$. Then $C_0^{\overline{l_0},1} = C_0^{\overline{l_0},0}$, and the second multiplication term in (33) is 0. Hence, $\mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}^* = 0.5 - \delta_{max}, \ldots, p_{n-1}\right)\right] = \mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}, \ldots, p_{n-1}\right)\right]$.

Subsubcase 1.2.2: Suppose that $\mathbf{1_{l_0}} \notin C_0$. Then by Theorem 1, $\mathbb{P}\left[C_0^{\overline{l_0},0}\right] \geq \mathbb{P}\left[C_0^{\overline{l_0},1}\right]$, since the set $C_0^{\overline{l_0},0}$ contains the $(n-1)$-bit all-zero vector and $C_0^{\overline{l_0},1}$ is its proper coset. Hence, the second multiplication term in the last line of (33) is also non-negative and $\mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}^* = 0.5 - \delta_{max}, \ldots, p_{n-1}\right)\right] \geq \mathbb{P}\left[C_0\left(p_0, \ldots, p_{l_0}, \ldots, p_{n-1}\right)\right]$.

By a trivial induction over coordinates l_i, $0 \leq i \leq n-1$, and iteratively applying the described coordinate probability substitution, one can easily arrive at the lemma's inequality for Case 1:

$$\mathbb{P}\left[C_0\left(0.5 - \delta_{max}, \ldots, 0.5 - \delta_{max}\right)\right] \geq \mathbb{P}\left[C_0\left(p_0, \ldots, p_{n-1}\right)\right]. \quad (34)$$

Case 2: Suppose that the most probable vector $\boldsymbol{x_{max}}$ is not the all-zero vector, i.e., $\mathrm{HW}(\boldsymbol{x_{max}}) = r \geq 1$ with 1-probabilities not smaller than 0.5 in coordinates $l_0, \ldots, l_{r-1}$.

If we change one of the coordinate 1-probabilities p_{l_0}, that was not smaller than 0.5, to $p^*_{l_0} = 0.5 - \delta_{max}$, the new most probable vector $\boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0}}_{\boldsymbol{max}}$ will be equal to $\boldsymbol{x_{max}}$ in all coordinates except l_0, in which $\boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0}}_{\boldsymbol{max}}$ has a 0.

Subcase 2.1: If $\mathbf{1}_{\boldsymbol{l_0}} \in C_0$, then $\boldsymbol{x_{max}}$ and $\boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0}}_{\boldsymbol{max}}$ are in the same set $C_{i_{max}}$ since $\boldsymbol{x_{max}} = \boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0}}_{\boldsymbol{max}} \oplus \mathbf{1}_{\boldsymbol{l_0}}$. All vectors in $C_{i_{max}}$ can be divided into two subsets $C^{l_0,\,b}_{i_{max}} = \{\boldsymbol{x} \in C_{i_{max}} \mid \boldsymbol{x}[l_0] = b\}$, $b \in \{0,1\}$. We now remove the coordinate l_0 of each element in $C^{l_0,\,b}_{i_{max}}$ to obtain subsets $C^{\overline{l_0},\,b}_{i_{max}}$, $b \in \{0,1\}$. Since every element $\boldsymbol{c}_{\boldsymbol{i_{max}}}$ in $C^{l_0,\,b}_{i_{max}}$ has exactly one corresponding element $\boldsymbol{c'}_{\boldsymbol{i_{max}}}$ in $C^{l_0,1-b}_{i_{max}}$ to which it is related by $\boldsymbol{c'}_{\boldsymbol{i_{max}}} = \boldsymbol{c}_{\boldsymbol{i_{max}}} \oplus \mathbf{1}_{\boldsymbol{l_0}}$, it is clear that $C^{\overline{l_0},\,b}_{i_{max}} = C^{\overline{l_0},1-b}_{i_{max}}$. Then, for the probabilities of $C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})$ and $C_{i_{max}}\left(p_0,\ldots,p^*_{l_0},\ldots,p_{n-1}\right)$, we have

$$\begin{aligned}&\mathbb{P}\left[C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})\right] = p_{l_0}\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right]\\&+(1-p_{l_0})\,\mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right] = \mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right] = \mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right], \quad (35)\end{aligned}$$

and

$$\begin{aligned}&\mathbb{P}\left[C_{i_{max}}\left(p_0,\ldots,p^*_{l_0},\ldots,p_{n-1}\right)\right] = p^*_{l_0}\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right]\\&+\left(1-p^*_{l_0}\right)\mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right] = \mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right] = \mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right], \quad (36)\end{aligned}$$

respectively. Therefore, $\mathbb{P}\left[C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})\right] = \mathbb{P}\left[C_{i_{max}}\left(p_0,\ldots,p^*_{l_0} = 0.5-\delta_{max},\ldots,p_{n-1}\right)\right]$.

Subcase 2.2: If $\mathbf{1}_{\boldsymbol{l_0}} \notin C_0$, then $\boldsymbol{x_{max}}$ and $\boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0}}_{\boldsymbol{max}}$ will be in different sets, which we denote by $C_{i_{max}}$ and C_{i_{max},l_0}. Let $\boldsymbol{e}_{\boldsymbol{x_{max}}}$ be a vector element of the lowest weight in $C_{i_{max}}$ and let $\boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}}}$ be the codeword such that $\boldsymbol{x_{max}} = \boldsymbol{e}_{\boldsymbol{x_{max}}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}}}$. Similarly, let $\boldsymbol{e}_{\boldsymbol{x_{max}},\boldsymbol{l_0}}$ be a vector element of the lowest weight in C_{i_{max},l_0} and let $\boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}},\boldsymbol{l_0}}$ be the codeword such that $\boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0}}_{\boldsymbol{max}} = \boldsymbol{e}_{\boldsymbol{x_{max}},\boldsymbol{l_0}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}},\boldsymbol{l_0}}$. Since $\boldsymbol{x_{max}} = \boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0}}_{\boldsymbol{max}} \oplus \mathbf{1}_{\boldsymbol{l_0}}$, it holds

$$\mathbf{1}_{\boldsymbol{l_0}} = \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}},\boldsymbol{l_0}} \oplus \boldsymbol{e}_{\boldsymbol{x_{max}}} \oplus \boldsymbol{e}_{\boldsymbol{x_{max}},\boldsymbol{l_0}}. \quad (37)$$

Every element $\boldsymbol{e}_{\boldsymbol{x_{max}}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{j}}$ from $C_{i_{max}}$ has one corresponding element in the set C_{i_{max},l_0} from which it differs only in the coordinate l_0:

$$\begin{aligned}&\boldsymbol{e}_{\boldsymbol{x_{max}}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{j}} \oplus \mathbf{1}_{\boldsymbol{l_0}} =\\&\qquad \boldsymbol{e}_{\boldsymbol{x_{max}},\boldsymbol{l_0}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{j}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}}} \oplus \boldsymbol{c}_{\boldsymbol{0},\boldsymbol{x_{max}},\boldsymbol{l_0}}. \quad (38)\end{aligned}$$

We partition $C_{i_{max}}$ into subsets $C^{l_0,\,b}_{i_{max}} = \{\boldsymbol{x} \in C_{i_{max}} \mid \boldsymbol{x}[l_0] = b\}$ and C_{i_{max},l_0} into subsets $C^{l_0,\,b}_{i_{max},l_0} = \{\boldsymbol{x} \in C_{i_{max},l_0} \mid \boldsymbol{x}[l_0] = b\}$, $b \in \{0,1\}$, according to the value of the coordinate l_0. We also remove components in coordinate l_0 of each element in $C^{l_0,\,b}_{i_{max}}$ and $C^{l_0,\,b}_{i_{max},l_0}$, $b \in \{0,1\}$, and obtain $C^{\overline{l_0},\,b}_{i_{max}}$ and $C^{\overline{l_0},\,b}_{i_{max},l_0}$, respectively. Due to (38), the elements in $C_{i_{max}}$ and C_{i_{max},l_0} differ only in the l_0 coordinate, and it follows $C^{\overline{l_0},\,b}_{i_{max}} = C^{\overline{l_0},1-b}_{i_{max},l_0}$. Then for the probabilities of $C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})$ and $C_{i_{max},\,l_0}\left(p_0,\ldots,p^*_{l_0},\ldots,p_{n-1}\right)$, we have

$$\begin{aligned}&\mathbb{P}\left[C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})\right] =\\&\qquad p_{l_0}\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right] + (1-p_{l_0})\,\mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right], \quad (39)\end{aligned}$$

and

$$\begin{aligned}&\mathbb{P}\left[C_{i_{max},\,l_0}\left(p_0,\ldots,p^*_{l_0},\ldots,p_{n-1}\right)\right] =\\&\qquad p^*_{l_0}\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max},l_0}\right] + \left(1-p^*_{l_0}\right)\mathbb{P}\left[C^{\overline{l_0},0}_{i_{max},l_0}\right] =\\&(0.5+\delta_{max})\,\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right] + (0.5-\delta_{max})\,\mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right], \quad (40)\end{aligned}$$

respectively. By subtracting (39) from (40), we obtain

$$\begin{aligned}&\mathbb{P}\left[C_{i_{max},\,l_0}\left(p_0,\ldots,p^*_{l_0} = 0.5-\delta_{max},\ldots,p_{n-1}\right)\right]\\&\qquad -\mathbb{P}\left[C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})\right] =\\&(0.5+\delta_{max}-p_{l_0})\,\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right] + (p_{l_0}-0.5-\delta_{max})\,\mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right]\\&= (\delta_{max}-(p_{l_0}-0.5))\left(\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right] - \mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right]\right). \quad (41)\end{aligned}$$

By the definition of δ_{max}, it follows $\delta_{max} \geq p_{l_0} - 0.5$, and thus, the first multiplication term in the last line of (41) is non-negative. Since in this case $\boldsymbol{x_{max}}$ has a 1 in coordinate l_0, we have two possibilities for the vectors in $C_{i_{max}}$: either $\boldsymbol{x}[l_0] = 1$ holds for all $\boldsymbol{x} \in C_{i_{max}}$ or half of the vectors have a 0 and the other half have a 1 in coordinate l_0.

Subsubcase 2.2.1: If $\boldsymbol{x}[l_0] = 1$ holds for all $\boldsymbol{x} \in C_{i_{max}}$, then $C^{l_0,1}_{i_{max}} = C_{i_{max}}$ and $C^{l_0,0}_{i_{max}} = C^{\overline{l_0},0}_{i_{max}} = \varnothing$. Since $\mathbb{P}\left[C^{l_0,0}_{i_{max}}\right] = 0$, the second multiplication term in the last line of (41) is also non-negative and thus, $\mathbb{P}\left[C_{i_{max},\,l_0}\left(p_0,\ldots,p^*_{l_0} = 0.5-\delta_{max},\ldots,p_{n-1}\right)\right] \geq \mathbb{P}\left[C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})\right]$.

Subsubcase 2.2.2: If not all $\boldsymbol{x} \in C_{i_{max}}$ have identical bit in position l_0, then the most probable $(n-1)$-bit vector obtained from $\boldsymbol{x_{max}}$ by deleting its l_0 coordinate $\boldsymbol{x}^{\overline{\boldsymbol{l_0}}}_{\boldsymbol{max}}$ will be in $C^{\overline{l_0},1}_{i_{max}}$. By Theorem 1, we have $\mathbb{P}\left[C^{\overline{l_0},1}_{i_{max}}\right] \geq \mathbb{P}\left[C^{\overline{l_0},0}_{i_{max}}\right]$. Thus, the second multiplication term in the last line of (41) is also non-negative, and we have $\mathbb{P}\left[C_{i_{max},\,l_0}\left(p_0,\ldots,p^*_{l_0} = 0.5-\delta_{max},\ldots,p_{n-1}\right)\right] \geq \mathbb{P}\left[C_{i_{max}}(p_0,\ldots,p_{l_0},\ldots,p_{n-1})\right]$.

Similarly to Case 1, we can use trivial induction over all coordinates l_i, $0 \leq i \leq r-1$, in which $\boldsymbol{x_{max}}$ has value 1. By iteratively applying the described coordinate probability substitutions, we are saddled with the most probable vector $\boldsymbol{x}^{\boldsymbol{l_0},\boldsymbol{0};\ldots;\boldsymbol{l_{r-1}},\boldsymbol{0}}_{\boldsymbol{max}}$, which is an all-zero vector since all ones are replaced by zeros and therefore $C_{i_{max},\,l_0,\ldots,l_{r-1}} = C_0$. Hence, we arrive at the inequality:

$$\begin{aligned}&\mathbb{P}\left[C_0\left(p_0,\ldots,p^*_{l_0},\ldots,p^*_{l_{r-1}},\ldots,p_{n-1}\right)\right] =\\&\mathbb{P}\left[C_{i_{max},\,l_0,\ldots,l_{r-1}}\left(p_0,\ldots,p^*_{l_0},\ldots,p^*_{l_{r-1}},\ldots,p_{n-1}\right)\right]\\&\qquad \geq \mathbb{P}\left[C_{i_{max}}\left(p_0,\ldots,p_{l_0},\ldots,p_{l_{r-1}},\ldots,p_{n-1}\right)\right], \quad (42)\end{aligned}$$

where $p^*_{l_0} = \cdots = p^*_{l_{r-1}} = 0.5 - \delta_{max}$. We can now apply the inequality (34) from *Case 1* when the all-zero is the most probable vector to obtain the lemma's inequality:

$$\begin{aligned}&\mathbb{P}\left[C_0\left(0.5-\delta_{max},\ldots,0.5-\delta_{max}\right)\right] \geq\\&\qquad \mathbb{P}\left[C_0\left(p_0,\ldots,p^*_{l_0},\ldots,p^*_{l_{r-1}},\ldots,p_{n-1}\right)\right] \geq\\&\qquad \mathbb{P}\left[C_{i_{max}}\left(p_0,\ldots,p_{l_0},\ldots,p_{l_{r-1}},\ldots,p_{n-1}\right)\right]. \quad (43)\end{aligned}$$

Since we have shown that the lemma's inequality is satisfied for all possible Hamming weights of the $\boldsymbol{x_{max}}$, this concludes the proof.

□

We use the previous results from Theorem 1 and Lemma 2 for the main theorem that provides a lower bound on the min-entropy of the output of a linear corrector when only a lower bound on the min-entropy of the noise source of independent bits is known.

Theorem 2. Let $\mathcal{X}$ be a row vector n-bit random variable with independent but not necessarily identically distributed coordinates and let the min-entropy per bit of $\mathcal{X}$ be at least $\mathrm{H}_\infty^{in} > 0$. Let $\boldsymbol{G}$ be a $k \times n$ generator matrix of a binary linear $[n, k]$-code C_0 and let $(\mathfrak{A}_i)_{i=0}^n$ be its weight distribution. Then, the total min-entropy of the output of the linear corrector $\mathcal{Y} = \boldsymbol{G}\mathcal{X}^\intercal$ is lower-bounded by:

$$\mathrm{H}_\infty^{out,\,tot} \;\geq\; -\log_2\left(2^{-k}\sum_{i=0}^{n}\mathfrak{A}_i\left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^i\right). \quad (44)$$

Proof. The proof will be a straightforward application of Theorem 1 and Lemma 2. Consider the min-entropy of the i-th bit of $\mathcal{X}$ with 1-probability $0 < p_i < 1$:

$$\mathrm{H}_\infty^{in,i} = -\log_2\left(\max\left\{p_i,\,1-p_i\right\}\right) = -\log_2\left(0.5+\left|0.5-p_i\right|\right). \quad (45)$$

If we also denote the maximal bit bias with $\delta_{max} = \max\left\{\left|0.5-p_i\right|\right\}_{i=0}^{n-1}$, then the lower bound on the min-entropy per bit of $\mathcal{X}$ is simply given by

$$\mathrm{H}_\infty^{in} = \min\left\{\mathrm{H}_\infty^{in,i}\right\}_{i=0}^{n-1} = -\log_2\left(0.5+\max\left\{\left|0.5-p_i\right|\right\}_{i=0}^{n-1}\right) = -\log_2\left(0.5+\delta_{max}\right). \quad (46)$$

By definition of the linear corrector $\mathcal{Y} = \boldsymbol{G}\mathcal{X}^\intercal$ and the fact that the generator matrix $\boldsymbol{G}$ of the C_0 code is equivalent to the parity-check matrix of its dual code $C_0^\perp$, every k-bit output $\boldsymbol{y}$ of the linear corrector will be a syndrome for the dual code $C_0^\perp$ of an n-bit vector $\boldsymbol{x}$ which is a realization of $\mathcal{X}$. Since all n-bit vectors belonging to the same coset of $C_0^\perp$ have the same syndrome, determining the probability of each output is equivalent to determining the probability of the corresponding coset of $C_0^\perp$. By Theorem 1, the most probable output will correspond to the syndrome of the most probable input vector. From (46) and the Lemma 2, it holds

$$\max_{y\in\mathcal{Y}}\mathbb{P}\left[\mathcal{Y}=y\right] = \mathbb{P}\left[C_{i_{max}}^\perp\left(p_0,\ldots,p_{n-1}\right)\right] \leq \mathbb{P}\left[C_0^\perp\left(p=1-2^{-\mathrm{H}_\infty^{in}},\ldots,p=1-2^{-\mathrm{H}_\infty^{in}}\right)\right], \quad (47)$$

where $C_{i_{max}}^\perp$ is a coset of $C_0^\perp$ that contains the most probable vector $\boldsymbol{x}$. Since the number of vectors of $C_0^\perp$ with Hamming weight i is given by its weight distribution $\left(\mathfrak{A}_i^\perp\right)_{i=0}^n$, we can determine the lower bound of the total output min-entropy as

$$\begin{aligned}\mathrm{H}_\infty^{out,\,tot} &= -\log_2\left(\max_{y\in\mathcal{Y}}\mathbb{P}\left[\mathcal{Y}=y\right]\right) \geq \\ &\quad -\log_2\left(\sum_{i=0}^{n}\mathfrak{A}_i^\perp\,p^i\left(1-p\right)^{n-i}\right) \\ &= -\log_2\left(2^{-n\mathrm{H}_\infty^{in}}\sum_{i=0}^{n}\mathfrak{A}_i^\perp\left(2^{\mathrm{H}_\infty^{in}}-1\right)^i\right). \quad (48)\end{aligned}$$

By substituting $2^{1-\mathrm{H}_\infty^{in}}-1$ for z in the MacWilliams identity (7), we have

$$\sum_{i=0}^{n}\mathfrak{A}_i^\perp\left(2^{\mathrm{H}_\infty^{in}}-1\right)^i = 2^{n\mathrm{H}_\infty^{in}}2^{-k}\sum_{i=0}^{n}\mathfrak{A}_i\left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^i, \quad (49)$$

and thus the theorem follows.

□

According to Lemma 2, our new bound (44) is tight when independent input bits are not identically distributed and only the lower bound on the input min-entropy is known, and it is met with equality when independent input bits are identically distributed with $p < 0.5$. In addition, thanks to Theorem 1, it is possible to determine the value and probability of the linear corrector's most probable output when the distributions of the input bits are precisely known. Since $\sum_{i=d}^{n}\mathfrak{A}_i = 2^k - 1$ and $\left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^i \leq \left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^d$ for $i \geq d$, it is straightforward to show that the lower bound from Theorem 2 is always tighter than the overly conservative state-of-the-art bound given by (3):

$$\begin{aligned}&-\log_2\left(2^{-k}\sum_{i=0}^{n}\mathfrak{A}_i\left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^i\right) = \\ &\quad -\log_2\left(2^{-k}+2^{-k}\sum_{i=d}^{n}\mathfrak{A}_i\left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^i\right) \geq \\ &\quad -\log_2\left(2^{-k}+2^{-k}\left(2^k-1\right)\left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^d\right) > \\ &\quad -\log_2\left(2^{-k}+\left(2^{1-\mathrm{H}_\infty^{in}}-1\right)^d\right). \quad (50)\end{aligned}$$

Finally, it is worth mentioning, as pointed out by one of the reviewers, that the results presented in this section can alternatively be obtained using established Fourier techniques outlined in the works of Redinbo [36] and Meneghetti [37].

V. Selection of the Linear Correctors

Improvement of the new bound over the old one given by (3) varies depending on the corrector's underlying code for which the bounds are calculated. From (3), it can be observed that for identical H_∞^{in} and fixed corrector length n and dimension k, the total output entropy is largest for the corrector based on a code with the greatest possible minimum distance d. Linear codes that achieve the greatest minimum distance among all known $[n, k]$-codes are called the *best known linear codes* (BKLCs) [38], [39]. On the other hand, it is clear from

(44) that the relationship between H_∞^{in} and $\mathrm{H}_\infty^{out,\,tot}$ is more complex and the codes' complete weight distribution should be considered. However, computing the weight distribution of a general binary linear code is an NP-hard problem [40] and requires a significant computing effort for codes with high dimensions and high differences between the length and dimension. In this section, we first calculate the new bound for the correctors based on the codes from the set of linear codes whose weight distributions can be conveniently determined or already available in the literature. We then outline the process of selecting the optimal corrector for a given min-entropy rate of raw bits that maximizes the throughput of post-processed bits while maintaining the desired security level. To demonstrate the practical advantages of our new bound, we compare the efficiencies and output min-entropies of correctors selected using the new bound against those selected using the old one.

A. Optimal Extracting Linear Correctors

Both large output min-entropy and low throughput reduction are desirable corrector's properties. Most security applications and standards [1]–[3] specify the output entropy requirements in terms of the min-entropy per bit $\mathrm{H}_\infty^{out,\,1}$. To conservatively guarantee the entropy rate $\mathrm{H}_\infty^{out,\,1}$ for every output bit, we require the total output min-entropy to be at least $\mathrm{H}_\infty^{out,\,tot} = k-1+\mathrm{H}_\infty^{out,\,1}$. This requirement is more strict than $\mathrm{H}_\infty^{out,\,tot} = k\mathrm{H}_\infty^{out,\,1}$, which would only guarantee the average min-entropy rate $\mathrm{H}_\infty^{out,\,1}$ across all output bits, while the min-entropy of individual bits might be lower. Since the throughput reduction is equal to the inverse of the underlying code's rate, a corrector based on a linear code is *optimal extracting* if there are no codes in the considered set with simultaneously higher code rate k/n and a lower or equal required H_∞^{in} to achieve $\mathrm{H}_\infty^{out,\,1}$. We denote this value of H_∞^{in} as $\mathrm{H}_\infty^{in,\,req}$. Post-processing of the raw bits with some specific (targeted) min-entropy rate is performed by selecting an optimal extracting corrector whose $\mathrm{H}_\infty^{in,\,req}$ is closest to the targeted min-entropy from below. By doing so, $\mathrm{H}_\infty^{out,\,1}$ can be obtained at the corrector's output with the lowest possible throughput reduction.

B. Construction of Corrector Sets

We first construct two sets from which the optimal extracting correctors will be determined: the set of correctors with output min-entropy determined according to the old bound (OBC) and the set of correctors with output min-entropy calculated by the new bound from Theorem 2 (NBC). The OBC is a set of $32,741$ elements and consists of the correctors based on the non-trivial ($n \neq k$) BKLCs from [39], BCH codes up to length 511 from [28] and binary linear codes available at [41]–[48]. On the other hand, the NBC set has a total of $16,613$ elements. It comprises correctors that are derived from binary linear codes with known weight distributions. These weight distributions are obtained from various sources, namely [41]–[48]. Additionally, the NBC set includes all non-trivial BKLCs and BCH codes found in the OBC. The length of these codes is restricted to $n < 81$, except for those with $n \geq 81$ that satisfy the condition $\min(k,\, n-k) \leq 38$. Computing the weight distributions using MAGMA [38] of BKLCs and BCH codes under these restrictions requires at most 60s per code of the real CPU time on Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz with 24 cores and 48 threads. To handle the BKLCs with generator matrices that contain one or more all-zero columns, we used codes with equivalent minimum distances but modified generator matrices to ensure that each column had at least one non-zero entry.

For hardware implementations of the correctors, opting for

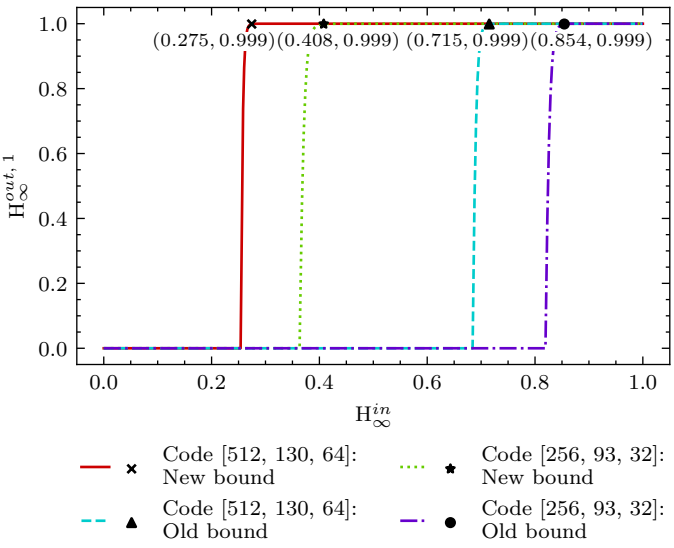

Fig. 1. Relation between input and output min-entropy rate according to both old and new bounds for Reed-Muller $[512, 130, 64]$ and $[256, 93, 32]$ code-based correctors. The output min-entropy rate is computed as $\mathrm{H}_\infty^{out,\,1} = \max\left(\mathrm{H}_\infty^{out,\,tot} - k + 1,\, 0\right)$, where $\mathrm{H}_\infty^{out,\,tot} = f\left(\mathrm{H}_\infty^{in}\right)$ is determined for both the old and the new bound. All min-entropy values are rounded to three decimals.

TABLE I
OPTIMAL LINEAR CORRECTORS AND PERFORMANCES FOR $\mathrm{H}_\infty^{out,\,1} \geq 0.999$

Target H_∞^{in}	Corrector construction		Extraction efficiency (η)	
	OBC	NBC	Old bound	New bound
0.1	$[511, 31, 219]^{a}$	$[511, 31, 219]^{a}$	0.60665234	0.60665360 (+0.0002%)
0.2	$[254, 31, 96]^{*}$	$[243, 38, 83]^{*}$	0.61022	0.78187 (+28.13%)
0.3	$[255, 47, 85]^{a,b}$	$[512, 130, 64]^{c}$	0.61437	0.84635 (+37.76%)
0.4	$[126, 29, 42]^{*}$	$[122, 38, 31]^{*}$	0.57538	0.77867 (+35.33%)
0.5	$[127, 35, 36]^{*}$	$[127, 50, 27]^{a,b}$	0.55117	0.78740 (+42.86%)
0.6	$[87, 29, 24]^{*}$	$[127, 64, 19]^{d}$	0.55554	0.83989 (+51.19%)
0.7	$[59, 23, 16]^{*}$	$[256, 163, 16]^{c}$	0.55688	0.90960 (+63.34%)
0.8	$[46, 22, 12]^{*}$	$[512, 382, 16]^{c}$	0.59781	0.93262 (+56.01%)
0.9	$[63, 35, 12]^{*}$	$[255, 219, 10]^{*}$	0.61727	0.95424 (+54.59%)

[*] BKLC code from [39] [a] BCH code from [28] [b] BCH code from [41] [c] Reed-Muller code from [41], [42] [d] Quadratic residue code from [43]

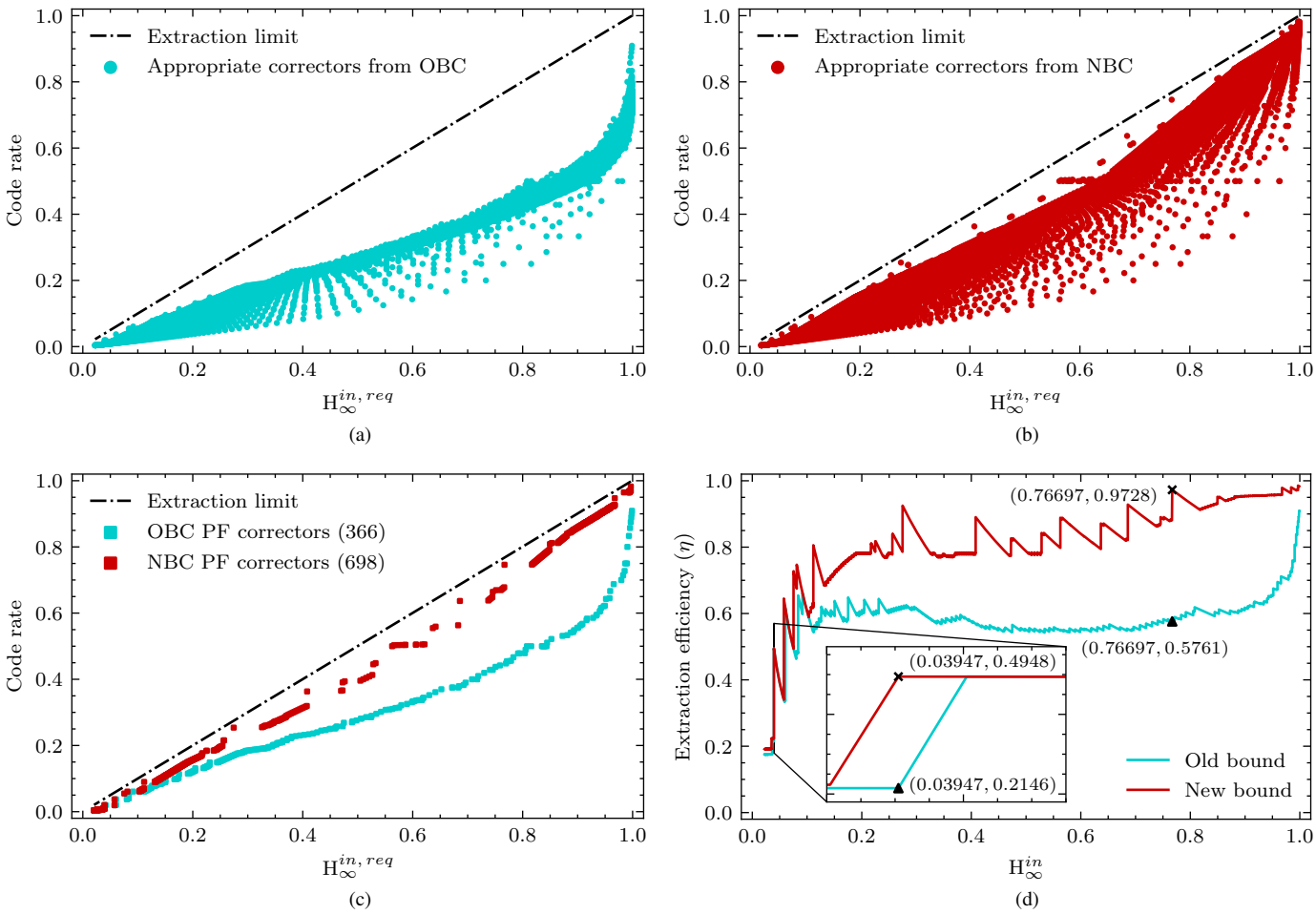

Fig. 2. Performances of linear correctors from OBC and NBC for $H_\infty^{out,\,1} \geq 0.999$ and extraction efficiency according to the old and the new bound.

cyclic codes generally results in smaller area requirements. This is because they can be implemented with only several registers and XOR gates, utilizing the well-known generator or parity-check polynomial constructions [14], [25]. To also provide optimal extracting correctors based only on cyclic codes, we form two new sets out of OBC and NBC, consisting only of cyclic constructions – OBCCYC and NBCCYC. The OBCCYC and NBCCYC sets consist out of 803 and 637 correctors, respectively. Comprehensive lists of elements in all four sets, accompanied by corresponding weight distributions for the NBC and NBCCYC sets, are publicly available via our Github repository [26].

Once the design parameter $H_\infty^{out,\,1}$ has been set, we calculate the code rate k/n and $H_\infty^{in,\,req}$ according to (3) for each corrector in the OBC and OBCCYC sets such that $H_\infty^{out,\,tot} = k - 1 + H_\infty^{out,\,1}$ is reached. Likewise, by numerically solving (44) via bisection for the same $H_\infty^{out,\,tot}$, we obtain $H_\infty^{in,\,req}$ and the code rate for every corrector in the NBC and NBCCYC sets. If $H_\infty^{in,\,req}$ is smaller than $H_\infty^{out,\,1}$, the corrector can be used for increasing the min-entropy rate and is referred to as an *appropriate corrector*. We form the subsets of appropriate correctors from each of the four corrector sets. Finally, we construct sets of *optimal extracting* correctors from sets of appropriate correctors, which we also call Pareto frontier (PF) correctors. It is important to note that, due to the disparity between the new and old bound, the optimal extracting correctors within the NBC/NBCCYC sets generally do not correspond to the optimal extracting correctors within the OBC/OBCYC sets.

C. Practical Corrector Selection and Efficiency Comparisons

In this work, we use $H_\infty^{out,\,1} = 0.999$, as it is the maximum between the requirement of the latest version of AIS-31 [3] (0.98) and NIST SP 800-90B [1] upper bound for the min-entropy rate after non-cryptographic post-processing (0.999). With this setting, we identified $24,221$ appropriate correctors from the OBC set, $15,873$ from the NBC set, 522 from the OBCCYC set and 435 from the NBCCYC set.

We evaluated the improvement in lowering $H_\infty^{in,\,req}$ offered by the new bound by calculating the difference between the $H_\infty^{in,\,req}$ values for $H_\infty^{out,\,1} = 0.999$ according to the new and the old bound for $9,908$ appropriate correctors common to both the OBC and NBC sets. Our analysis revealed that the new bound yields a considerable relative improvement in $H_\infty^{in,\,req}$ surpassing $15\,\%$ for most constructions. We found that the greatest absolute improvement is achieved for the Reed-Muller $[256, 93, 32]$ code-based corrector, for which the new bound lowers $H_\infty^{in,\,req}$ from 0.854296 to 0.407964, while the

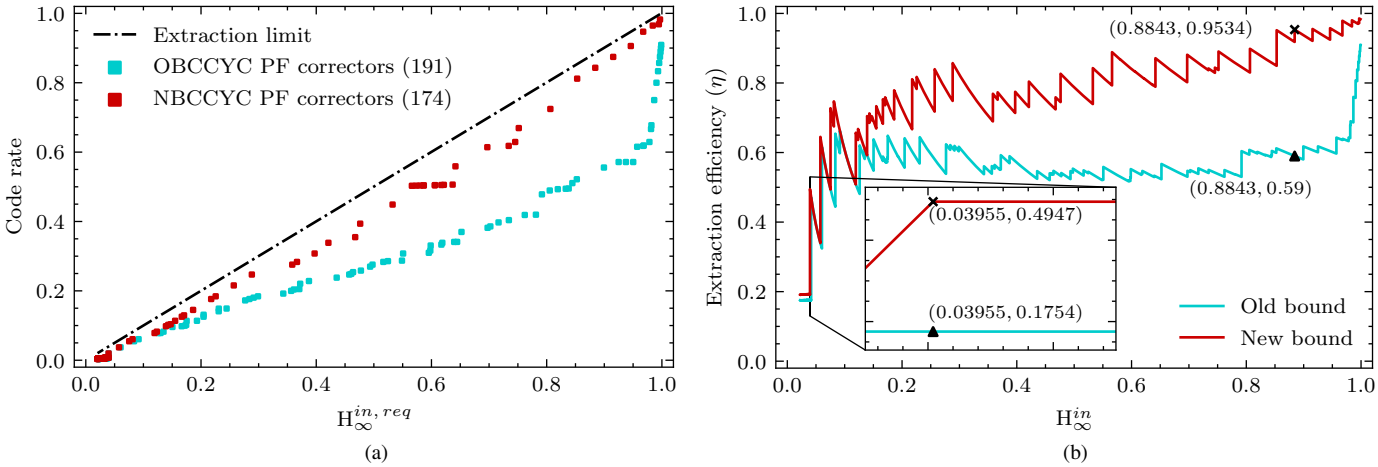

Fig. 3. Performances of optimal linear correctors from OBCCYC and NBCCYC for $H_\infty^{out,1} \geq 0.999$ and extraction efficiency according to the old and the new bound.

largest relative improvement of $61.62\,\%$ is obtained for the Reed-Muller $[512, 130, 64]$ corrector, as indicated in Fig. 1. It is worthwhile to note that the old bound fails to guarantee that every output bit will have at least some entropy for $H_\infty^{in} = 0.274447$ in the case of $[512, 130, 64]$ corrector and $H_\infty^{in} = 0.407964$ in the case of $[256, 93, 32]$ corrector, by taking a conservative approach to calculating the output min-entropy rate $H_\infty^{out,1} = \max\left(H_\infty^{out,tot} - k + 1,\, 0\right)$. Even for correctors based on codes with very large minimum distances, such as the $[512, 10, 256]$ corrector, our bound still offers a discernible improvement of $0.01\,\%$. This indicates that the state-of-the-art min-entropy bound for these correctors is already quite close to the new bound, underscoring that further improvements for the same $H_\infty^{out,1}$ are not feasible.

Appropriate correctors from OBC and NBC sets in a *code rate - required input min-entropy* plane are shown in Fig. 2a – 2c. The dash-dotted lines show the theoretical extraction limit for $H_\infty^{out,1} = 0.999$, i.e., the highest possible code rate of $H_\infty^{in}/H_\infty^{out,1}$ for $H_\infty^{in} < H_\infty^{out,1}$. Fig. 2c displays optimal extracting (PF) correctors from both sets to examine the benefits of the new bound. Although the NBC set of appropriate correctors is much smaller than its OBC counterpart, the optimal extracting solutions obtained by our bound always dominate over the solutions with the old bound. Further, the new bound provides more optimal extracting correctors than the old one, though the correctors from the OBC set are more evenly spread. Our analysis also revealed that the OBC set's optimal extracting correctors have the smallest $H_\infty^{in,req}$ value of 0.022275, whereas the NBC set's optimal extracting correctors have the smallest $H_\infty^{in,req}$ value of 0.020351. These results suggest that the new bound permits a marginally broader range of admissible raw bit min-entropies.

Fig. 2d shows the extraction efficiency for targeted H_∞^{in} in the common range for both bounds – $(0.022275, 0.999)$, by using the optimal extracting correctors selected according to the state-of-the-art and the new bound from the OBC and NBC sets, respectively. The extraction efficiency is calculated using (6). For the old bound, we obtained $H_\infty^{out,tot}$ as described in (3), while for the new bound, we utilized (44). As indicated by peaks in the graph, extraction efficiency reaches local maxima for targeted min-entropies that coincide with $H_\infty^{in,req}$ of the optimal extracting correctors. Here, we observe that the extraction efficiencies for both bounds are consistently greater than 0.5 starting from $H_\infty^{in} = 0.08374$ and that the new bound extraction efficiency outperforms the old bound one for the entire input min-entropy range. The largest absolute efficiency difference of 0.39668 is reached for $H_\infty^{in} = 0.76697$, while the highest relative efficiency increase of $130.56\,\%$ is achieved for $H_\infty^{in} = 0.03947049$. Additionally, we computed the average relative efficiency increase resulting from the new bound to be $41.2\,\%$, while starting from $H_\infty^{in} = 0.1777221$ this relative increase consistently exceeds $20\,\%$. The performances of optimal correctors from both sets for nine targeted input min-entropies are summarized in Table I, together with constructions of corresponding correctors.

The code rates of the optimal extracting correctors based only on cyclic codes from OBCCYC and NBCCYC sets versus their $H_\infty^{in,req}$ is depicted in Fig. 3a. In this case, there are fewer optimal correctors from the NBCCYC set, but we found that the new bound still provides a narrowly larger range of admissible input min-entropies, as the values of the smallest $H_\infty^{in,req}$ for correctors in OBCCYC and NBCCYC sets are identical to the ones in OBC and NBC sets, respectively. Based on the results shown in the plot of Fig. 3b, which displays the relationship between the extraction efficiency and the targeted H_∞^{in}, it is evident that the extraction efficiency achieved with the new bound-selected cyclic correctors always surpasses that of the old bound-selected cyclic correctors for all targeted H_∞^{in}. Notably, the maximum relative efficiency increase of $182.04\,\%$ achieved for $H_\infty^{in} = 0.03955041$ in this case is higher than the increase observed without imposing the cyclicity restriction. Table II summarizes the performances and constructions of optimal correctors based on cyclic codes for nine targeted input min-entropies. Optimal extracting corrector constructions from all sets and their $H_\infty^{in,req}$ are available in our online repository [26].

TABLE II
OPTIMAL LINEAR CORRECTORS BASED ON CYCLIC CODES AND PERFORMANCES FOR $H_\infty^{out,1} \geq 0.999$

Target H_∞^{in}	Corrector construction		Extraction efficiency (η)	
	OBCCYC	NBCCYC	Old bound	New bound
0.1	$[511, 31, 219]^{a}$	$[511, 31, 219]^{a}$	0.60665234	0.60665360 (+0.0002%)
0.2	$[255, 29, 95]^{a,b}$	$[255, 37, 91]^{a,b}$	0.56862	0.72549 (+27.59%)
0.3	$[255, 47, 85]^{a,b}$	$[255, 63, 63]^{a,b}$	0.61437	0.82353 (+34.04%)
0.4	$[127, 29, 43]^{*}$	$[117, 36, 32]^{*,c}$	0.57086	0.76921 (+34.75%)
0.5	$[127, 35, 36]^{*}$	$[127, 50, 27]^{a,b}$	0.55117	0.78740 (+42.86%)
0.6	$[127, 42, 32]^{*}$	$[127, 64, 19]^{d}$	0.55117	0.83989 (+52.38%)
0.7	$[55, 21, 15]^{*}$	$[127, 78, 15]^{a,b}$	0.54543	0.87738 (+60.86%)
0.8	$[23, 11, 8]^{*}$	$[127, 85, 13]^{a,b}$	0.59779	0.83661 (+39.95%)
0.9	$[63, 35, 12]^{*}$	$[255, 215, 11]^{a,b}$	0.61727	0.93682 (+51.77%)

* BKLC code from [39] a BCH code from [28] b BCH code from [41]
c Code from [46] d Quadratic residue code from [43]

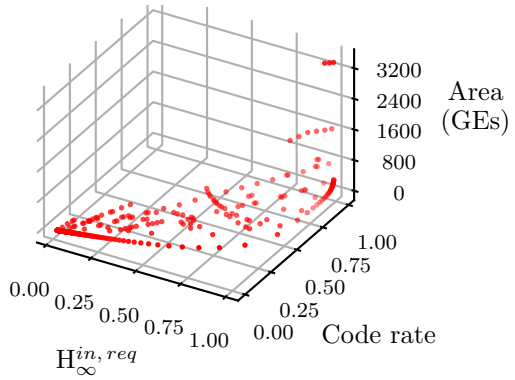

Fig. 4. Optimal area-efficient cyclic code-based correctors.

D. Implementation Cost Criterion

As a final selection criterion, we take an estimation of the implementation cost (chip area) of the correctors based on cyclic codes. Cyclic codes possess a distinct structure that results in a simplified implementation compared to general codes. Our objective is to find a balance between the code rate, required input min-entropy, and the area the corrector based on cyclic code would occupy. In doing so, we ensure that the chosen correctors not only provide a small reduction in throughput and high extraction efficiency but are also practical for real-world applications.

To evaluate the implementation cost of each corrector in the NBCCYC set, without including a controller counter, we estimate the number of gate equivalents (GEs). The area of each corrector is assessed based on two distinct implementation methods, utilizing the generator and parity-check polynomials of the corresponding code, as delineated in [25]. We employ XOR2_1 and DFFR_X1 gates from the NanGate 45 nm open standard-cell library [49]. Each XOR2_1 gate consumes 2 GEs, while the DFFR_X1 gate utilizes 6.67 GEs. Here, one GE corresponds to the size of a NAND2_X1 gate. We first calculate the area of each corrector using both implementation flavors. Subsequently, for each individual corrector, we select the implementation yielding the smaller area. We then conduct a three-dimensional optimization to derive a set of optimal area efficiency correctors. A corrector based on cyclic code is *optimal area-efficient* if there are no other codes in NBCCYC that concurrently exhibit a higher code rate, equal or lower $H_\infty^{in,req}$, and a smaller area.

TABLE III
OPTIMAL AREA-EFFICIENT LINEAR CORRECTORS BASED ON CYCLIC CODES AND PERFORMANCES FOR $H_\infty^{out,1} \geq 0.999$ (NEW BOUND)

Target H_∞^{in}	Corrector construction	Extraction efficiency (η)	Area (NanGate 45 nm)
0.1	$[51, 1, 51]$	0.1959	8.67 GEs
0.2	$[127, 15, 55]$	0.5905	122.05 GEs
0.3	$[63, 9, 28]$	0.4762	68.03 GEs
0.4	$[11, 1, 11]$	0.2270	8.67 GEs
0.5	$[87, 31, 22]$	0.7126	244.77 GEs
0.6	$[127, 64, 21]$	0.8399	484.88 GEs
0.7	$[15, 5, 7]$	0.4761	39.35 GEs
0.8	$[23, 12, 7]$	0.6521	94.04 GEs
0.9	$[31, 21, 5]$	0.7527	162.07 GEs

TABLE IV
IMPLEMENTATION COST COMPARISONS FOR DIFFERENT POST-PROCESSING ALGORITHMS

Post-processing	Reference	Technology	Area
Keccak-f [1600]	$[50]^{a}$	NanGate 45 nm	31,361 GEs
Keccak-f [1600]	$[51]^{a}$	NanGate 45 nm	28,100 GEs
SHA-256	[52]	NanGate 45 nm	15,000 GEs
Keccak-f [1600]	$[51]^{b}$	NanGate 45 nm	12,800 GEs
SHA-256	[53]	STD110 0.25 μm	8,588 GEs
Keccak-f [1600]	$[54]^{b}$	UMC 0.13 μm	5,522 GEs
Linear corrector $[511, 484, 7]$	This work[c]	NanGate 45 nm	3443.04 GEs

a round-based b serial (slice-based)
c largest optimal area-efficient NBCCYC corrector

The 434 optimal area-efficient correctors that we found are displayed in Fig. 4. Table III provides an overview of the constructions and performances of these correctors for nine targeted input min-entropies. Comparing these correctors to the correctors found with the new bound listed in Table II, it is immediately evident that the correctors in Table III exhibit significantly lower extraction efficiency, particularly for $H_\infty^{in,req} = 0.1$ and $H_\infty^{in,req} = 0.4$. However, these constructions require only 8.67 GEs, whereas correctors based on $[511, 31, 219]$ and $[117, 36, 32]$ codes require 224.77 GEs and 272.12 GEs, respectively. On the other hand, for $H_\infty^{in,req} = 0.5$, the efficiency of the $[87, 31, 22]$ corrector differs from that of the $[127, 50, 27]$ corrector by only 0.0748, while consuming much less area: 375.50 GEs vs 244.77 GEs. The estimated implementation costs for all correctors from the NBCCYC set are also available in [26].

Table IV shows the area usage (in GEs) for the largest optimal area-efficient linear corrector $[511, 484, 7]$ and several implementations of two NIST-approved cryptographic hash functions that can be used for post-processing (conditioning) [1] – SHA-3 (based on Keccak-f [1600]) and SHA-256. It can be observed that the areas of various implementations of Keccak-f [1600] and SHA-256 vary significantly due to the technology and architectural choices. However, even the implementation of the largest linear corrector $[511, 484, 7]$ from our work demonstrates a remarkable reduction in the area footprint, consuming only 3443.04 GEs. This represents a considerable saving in comparison to the cryptographic post-processing algorithms. Further, when considering only implementations using identical technology – Nangate 45 nm, the $[511, 484, 7]$ corrector is more than three times smaller than the most area-efficient implementation of Keccak-f [1600].

VI. Conclusion

In this paper, we have presented a novel tight bound on the output min-entropy of linear correctors based on the weight distribution of the corresponding binary linear code. Our proposed bound, which relies on the code's weight distribution, enables more efficient use of linear correctors than the old bound, which only requires knowledge of the code's minimum distance. We have demonstrated how the new bound can be used to select an optimal extracting corrector that meets output min-entropy rate requirements and maximizes throughput. Moreover, we have made publicly available optimal constructions for general correctors and correctors based on cyclic codes for $\mathrm{H}_{\infty}^{out,\,1} = 0.999$, allowing for easy implementation and integration into existing TRNG designs. Our findings indicate a potential for advancements in optimal extracting solutions through further research in characterizing binary linear codes' weight distributions. Future work will concentrate on constructing tight output min-entropy bounds for a wider spectrum of non-IID noise sources and, potentially, non-linear correctors.

Acknowledgment

The authors would like to thank the anonymous reviewers for their useful feedback and highlighting the connection between our findings and those presented in the work by Redinbo [36] using Fourier methods.

References

[1] M. S. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, "NIST special publication 800-90B: Recommendation for the entropy sources used for random bit generation," Tech. Rep., Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Jan. 2018.

[2] W. Killmann and W. Schindler, "A proposal for: Functionality classes for random number generators," *ser. BDI, Bonn*, 2011.

[3] M. Peter and W. Schindler, "A proposal for functionality classes for random number generators, version 2.35 – draft," *ser. BDI, Bonn*, 2022.

[4] J. Balasch, F. Bernard, V. Fischer, M. Grujić, M. Laban, O. Petura, V. Rožić, G. van Battum, I. Verbauwhede, M. Wakker, and B. Yang, "Design and testing methodologies for true random number generators towards industry certification," in *Proc. 2018 IEEE 23rd Eur. Test Symp. (ETS)*, 2018, pp. 1–10.

[5] B. Yang, V. Rožić, M. Grujić, N. Mentens, and I. Verbauwhede, "ES-TRNG: A High-throughput, Low-area True Random Number Generator based on Edge Sampling," *IACR Trans. on Cryptograph. Hardw. Embed. Syst.*, vol. 2018, no. 3, pp. 267–292, Aug. 2018.

[6] O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet, "A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices," in *Proc. 26th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2016, pp. 1–10.

[7] Y. Ma, T. Chen, J. Lin, J. Yang, and J. Jing, "Entropy estimation for adc sampling-based true random number generators," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 11, pp. 2887–2900, 2019.

[8] D. Johnston, *Random Number Generators—Principles and Practices, A Guide for Engineers and Programmers*. Berlin, Boston: De Gruyter, Sep. 2018.

[9] J. Von Neumann, "Various techniques used in connection with random digits," *Appl. Math. Ser.*, vol. 12, pp. 36–38, 1951.

[10] Y. Peres, "Iterating von Neumann's procedure for extracting random bits," *Ann. Statist.*, pp. 590–597, 1992.

[11] P. Elias, "The efficient construction of an unbiased random sequence," *Ann. Math. Statist.*, pp. 865–870, 1972.

[12] R. B. Davies, "Exclusive or (xor) and hardware random number generators," Author-hosted manuscript at http://www.robertnz.net/pdf/xor2.pdf, 2002.

[13] M. Dichtl, "Bad and good ways of post-processing biased physical random numbers," in *Proc. Int. Workshop Fast Softw. Encryption*, 2007, pp. 137–152.

[14] P. Lacharme, "Post-processing functions for a biased physical random number generator," in *Proc. Int. Workshop Fast Softw. Encryption*, 2008, pp. 334–342.

[15] ——, "Analysis and construction of correctors," *IEEE Trans. Inf. Theory*, vol. 55, no. 10, pp. 4742–4748, 2009.

[16] A. Tomasi, A. Meneghetti, and M. Sala, "Code generator matrices as RNG conditioners," *Finite Fields Appl.*, vol. 47, pp. 46–63, Sep. 2017.

[17] M. Grujić and I. Verbauwhede, "TROT: A three-edge ring oscillator based true random number generator with time-to-digital conversion," *IEEE Trans. Circuits Syst. I*, vol. 69, no. 6, pp. 2435–2448, 2022.

[18] A. Zeh, A. Glew, B. Spinney, B. Marshall, D. Page, D. Atkins, K. Dockser, M.-J. O. Saarinen, N. Menhorn, and R. Newell, "RISC-V cryptographic extension proposals," Online available at: https://github.com/riscv/riscv-crypto, 2021.

[19] M.-J. O. Saarinen, G. R. Newell, and B. Marshall, "Development of the RISC-V entropy source interface," *J. Cryptograph. Eng.*, vol. 12, no. 4, pp. 371–386, Jan. 2022.

[20] K. Ugajin, Y. Terashima, K. Iwakawa, A. Uchida, T. Harayama, K. Yoshimura, and M. Inubushi, "Real-time fast physical random number generator with a photonic integrated circuit," *Opt. Express*, vol. 25, no. 6, pp. 6511–6523, Mar 2017.

[21] R. Ali, Y. Wang, Z. Hou, H. Ma, Y. Zhang, and W. Zhao, "Process variation-resilient STT-MTJ based TRNG using linear correcting codes," in *Proc. 2019 IEEE/ACM Int. Symp. Nanoscale Architectures (NANOARCH)*, 2019, pp. 1–6.

[22] J. Park, S. Cho, T. Lim, and M. Tehranipoor, "QEC: A quantum entropy chip and its applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst*, vol. 28, no. 6, pp. 1471–1484, 2020.

[23] T. Lyp, N. Karimian, and F. Tehranipoor, "LISH: A new random number generator using ECG noises," in *Proc. 2021 IEEE Int. Conf. Consum. Electron. (ICCE)*, 2021, pp. 1–6.

[24] N. Massari, A. Tontini, L. Parmesan, M. Perenzoni, M. Gruijć, I. Verbauwhede, T. Strohm, D. Oshinubi, I. Herrmann, and A. Brenneis, "A monolithic SPAD-based random number generator for cryptographic application," in *Proc. IEEE 48th Eur. Solid State Circuits Conf. (ESSCIRC 2022)*, 2022, pp. 73–76.

[25] S.-H. Kwok, Y.-L. Ee, G. Chew, K. Zheng, K. Khoo, and C.-H. Tan, "A comparison of post-processing techniques for biased random number generators," in *Proc. IFIP Int. Workshop Inf. Security Theory Practices*. Springer, 2011, pp. 175–190.

[26] M. Grujić and I. Verbauwhede, "Optimal linear correctors - repository," https://github.com/KULeuven-COSIC/Optimizing-Linear-Correctors/, 2023.

[27] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes*. Elsevier, 1977, vol. 16.

[28] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Pearson-Prentice Hall, 2004.

[29] J. MacWilliams, "A theorem on the distribution of weights in a systematic code," *Bell Syst. Tech. J.*, vol. 42, no. 1, pp. 79–94, 1963.

[30] H. Zhou and J. Bruck, "Linear extractors for extracting randomness from noisy sources," in *Proc. 2011 IEEE Int. Symp. Inf. Theory*, Jul. 2011, pp. 1738–1742.

[31] ——, "Linear transformations for randomness extraction," *arXiv preprint arXiv:1209.0732*, 2012.

[32] A. Meneghetti, M. Sala, and A. Tomasi, "A weight-distribution bound for entropy extractors using linear binary codes," *arXiv preprint arXiv:1405.2820*, 2014.

[33] I. Sason, "Entropy Bounds for Discrete Random Variables via Maximal Coupling," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7118–7131, Nov. 2013.

[34] D. Sullivan, "A fundamental inequality between the probabilities of binary subgroups and cosets," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 91–94, Jan. 1967.

[35] M. Živković, "On two probabilistic decoding algorithms for binary linear codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 6, pp. 1707–1716, Nov. 1991.

[36] G. Redinbo, "Inequalities between the probability of a subspace and the probabilities of its cosets," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 533–536, Jul. 1973.

[37] A. Meneghetti, "Optimal Codes and Entropy Extractors," Ph.D. dissertation, Università degli studi di Trento, 2017.

[38] W. Bosma, J. Cannon, and C. Playoust, "The Magma algebra system. I. The user language," *J. Symbolic Comput.*, vol. 24, no. 3-4, pp. 235–265, 1997.

[39] M. Grassl, "Bounds on the minimum distance of linear codes and quantum codes," Online available at: http://www.codetables.de, 2007.

[40] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[41] M. Terada, J. Asatani, and T. Koumoto, "Weight Distribution," Online available at: https://isec.ec.okayama-u.ac.jp/home/kusaka/wd/.

[42] T. Sugita, T. Kasami, and T. Fujiwara, "The weight distribution of the third-order Reed-Muller code of length 512," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1622–1625, Sep. 1996.

[43] N. J. Sloane, "List of weight distributions in the on-line encyclopedia of integer sequences," Online available at: https://oeis.org/wiki/List_of_weight_distributions.

[44] T.-K. Truong, Y. Chang, and C.-D. Lee, "The weight distributions of some binary quadratic residue codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1776–1782, 2005.

[45] M. Tomlinson, C. J. Tjhai, M. A. Ambroze, M. Ahmed, and M. Jibril, *Error-Correction Coding and Decoding: Bounds, Codes, Decoders, Analysis and Applications*. Springer Nature, 2017.

[46] D. Schomaker and M. Wirtz, "On binary cyclic codes of odd lengths from 101 to 127," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 516–518, 1992.

[47] Y. Desaki, T. Fujiwara, and T. Kasami, "The weight distributions of extended binary primitive BCH codes of length 128," *IEEE Trans. Inf. Theory*, vol. 43, no. 4, pp. 1364–1371, 1997.

[48] T. Fujiwara and T. Kasami, "The weight distribution of (256, k) extended binary primitive bch code with k<= 63, k>= 207," IEICE, IT97, Tech. Rep., 1993.

[49] Silvaco, "Nangate 45 nm open cell library." [Online]. Available: https://si2.org/open-cell-library/

[50] D. Knichel and A. Moradi, "Composable gadgets with reused fresh masks: First-order probing-secure hardware circuits with only 6 fresh masks," *IACR Trans. Cryptograph. Hardw. Embedded Syst.*, pp. 114–140, Jun. 2022.

[51] B. Bilgin, J. Daemen, V. Nikov, S. Nikova, V. Rijmen, and G. Van Assche, "Efficient and first-order dpa resistant implementations of Keccak," in *Proc. 12th Int. Conf. Smart Card Res. Adv. Appl. (CARDIS)*, 2014, pp. 187–199.

[52] L. Baldanzi, L. Crocetti, F. Falaschi, M. Bertolucci, J. Belli, L. Fanucci, and S. Saponara, "Cryptographically secure pseudo-random number generator IP-core based on SHA2 algorithm," *Sensors*, vol. 20, no. 7, p. 1869, 2020.

[53] M. Kim, J. Ryou, and S. Jun, "Efficient hardware architecture of sha-256 algorithm for trusted mobile computing," in *Proc. Inf. Security Cryptol.*, 2009, pp. 240–252.

[54] P. Pessl and M. Hutter, "Pushing the limits of SHA-3 hardware implementations to fit on RFID," in *Cryptograph. Hardw. Embed. Syst. – CHES 2013*, 2013, pp. 126–141.