Coding on Dual-Parameter Barrier Channels

Yuval Ben-Hur and Yuval Cassuto
The Andrew and Erna Viterbi Department of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 3200003, Israel
Email: {yuvalbh@campus, ycassuto@ee}.technion.ac.il

***Abstract*—This paper studies coding on channels with the barrier property: only errors to and from a special barrier symbol are possible. This model is motivated by information systems that have heterogeneous symbol structure, not admitting the usual multi-bit scaling principle of representation levels. This work is the first that addresses a barrier channel with separate parameters for the transitions into and out of the barrier symbol. Earlier work addressed special-case single-parameter models, and focused primarily on the worst-case coding performance. Our contributions for the general dual-parameter model include derivation of the channel capacity, guaranteed-correction conditions and code-size bounds, a code construction method, and several decoders that correct beyond the guaranteed worst-case capability. A key structural feature of our construction method enables reduction of the decoding problem to a cooperation between two decoders for codes in the Hamming metric.**

I. Introduction

Nowadays, many digital information systems, such as data-storage devices, data processing units and communication systems, represent information using the binary alphabet. Although convenient for hardware implementation, binary representations use only $Q = 2$ levels and thus impose severe limitations on the system efficiency (e.g., information density/rate and power consumption). Even non-binary systems usually employ multi-bit representations, in which an integer power of 2 number of levels are uniformly spaced across the system's dynamic range. However, many next-generation systems are able to span multiple representation levels/states, but without the regularity of the multi-bit scaling principle.

In this paper we study a family of Q-ary channels we call *barrier channels*. Out of the Q symbols of the alphabet, one is designated as the *barrier symbol*, and errors are introduced as transitions only to and from the barrier symbol. Two variants of the barrier channel are studied: the *probabilistic model* $W_Q(p, q)$ where p, q are the transition probabilities to and from the barrier symbol, respectively, and the *combinatorial model* $W_Q(t_d, t_u)$ where t_d, t_u are the numbers of errors in a block to and from the barrier symbol, respectively.

One practical motivation for the ternary ($Q = 3$) barrier channel is a (symbol dependent) AWGN channel as described in Fig. 1. Transitions are dominant to and from the barrier ('0') symbol in the middle, while transitions between the extreme ('1' and '2') symbols occur with negligible probability. Such a channel may apply in memory technologies that scale from binary to ternary cells while maintaining the same dynamic range in the analog domain. The paper studies the general Q-ary barrier channel as an interesting generalization from theoretical perspective, which may also find practical future applications for $Q > 3$. Inspirations for such applications can be found in data storage: magnetic media with multiple stable states [1] representing non-barrier symbols, to which a less-stable barrier state can potentially be added as the barrier symbol, and in communications: by introducing a low-power barrier signal to a constellation of $Q - 1$ orthogonal signals. The particular applicability of the barrier model to ternary ($Q = 3$) alphabets is in line with the recent advent of devices and applications working with ternary alphabets. For example, IoT biosensors generate data streams with 3 possible states [2], novel in-memory memristor-based processors employ ternary logic [3], and next-generation electrically erasable programmable read-only memories (EEPROMs) store ternary symbols [4]. Ternary representations were also considered for traditional CMOS logic [5], wide-band communication systems [6] and even state-of-the-art large language models (LLMs) [7].

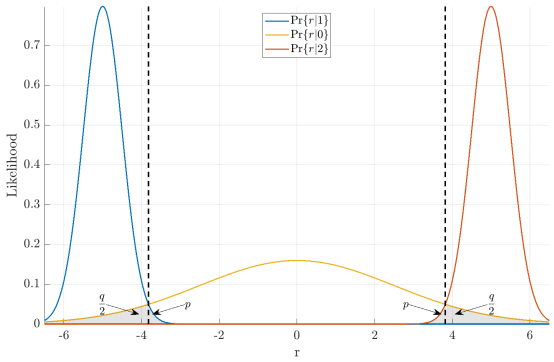


Figure 1: Ternary barrier-channel example: the dominant transitions are to and from the center level. The corresponding barrier parameters are $p = 9 \cdot 10^{-3}$, $q = 1.3 \cdot 10^{-1}$. The transition probability between the extreme symbols is negligible.

Two specific versions of the barrier channel have already been studied before. Codes that correct t barrier errors in only one direction (non-barrier to barrier) are studied in [8]. The results for that version include a necessary and sufficient

Part of the results in this paper was presented at the 2021 IEEE Global Communications Conference (GLOBECOM) and at the 2023 IEEE International Symposium on Information Theory (ISIT).

correction condition, bounds on the size of such codes, and a construction for $t = 1$. A channel that considers barrier errors in both directions is studied in [4] (and its earlier version [9]), but assuming a single parameter that governs both directions of barrier errors. This assumption implies $p = q/2$ in our model, which limits its applicability: in the example of Fig. 1 we observe that $p \ll q$, due to the lower reliability around 0. For the single-parameter version, [4] derives bounds on code sizes, which are further tightened in [10].

The results of [4] include a code-construction method that is useful also in the two-parameter case: obtaining a ternary t-barrier-error-correcting code using binary Hamming-metric constituent codes. Specifically, for t barrier errors of any direction the construction employs: 1) a binary t-error-correcting code in the Hamming metric, and 2) a set of binary t-erasure-correcting codes of *different lengths*. The length of the second code is chosen every encoding instance according to the encoding outcome of the first code. This construction also implies a bounded-distance decoding algorithm for t errors that proceeds by decoding the error-code and erasure-code sequentially.

This paper extends the prior work in several directions. Firstly, it studies barrier channels with two independent parameters: p, q in the probabilistic model, and t_d, t_u in the combinatorial model. This extension holds the promise of more efficient coding tailored for the specific parameters at hand. In the probabilistic model, we derive in Section II the channel capacity of $W_Q(p, q)$ and show significant capacity gaps between channels with different p, q combinations. In the combinatorial model, we prove in Section III conditions for guaranteed correction: one condition that is both necessary and sufficient, and another sufficient condition that is satisfied by the code construction proposed later in the paper. We derive code-size upper and lower bounds based on the two conditions, and show empirically that the derived lower bounds are much tighter than the tightest known prior lower bounds [10] that apply to the dual-parameter barrier channel. Note, however, that these prior bounds apply to a stronger (single-parameter) error model that allows any t barrier errors such that $t = t_d + t_u$ (the d_1 distance of [10] was shown in [4] to capture the capability of correcting t barrier errors.).

Toward practical coding, Section IV presents a code-construction method and Section V adds decoding algorithms that are enabled by that construction method. To obtain a Q-ary (t_d, t_u) barrier-error-correcting code of length n, we use two constituent codes for symmetric errors over smaller alphabets, each of length n. A similar reduction has been proposed before in [4], [8] (for a single parameter t), but requiring to use constituent codes of many different lengths. Using only a single code with a single length in our construction enables decoding algorithms that correct beyond the number of errors guaranteed by the construction.

The baseline decoder for codes constructed with our proposed method is a bounded-distance decoder, which is the first we define in Section V. We then derive a *maximum-likelihood decoder (MLD)* for the construction that employs a reduction to MLD of the constituent codes, which are lower-alphabet codes over the Hamming metric. Based on the MLD, we propose a lower-complexity algorithm we call *cooperative list decoder (CLD)* and its variant *persistent CLD (PCLD)*. As the name implies, CLD/PCLD use a list decoder for decoding the first constituent code, which is a binary code. They use both codes to find the highest likelihood Q-ary codeword that is generated by a word in the list. Finally, we introduce a *deeper cooperation (DC)* decoder that uses cooperation between the constituent codes without resorting to a list decoder. In DC decoder, the constituent decoders pass information on a per-symbol basis, thereby allowing to refine bits of the first code based on a score inferred by the second code. Such cooperation is not possible with the prior construction of [4], [8], since the length of the second code is not known until the first code is correctly decoded. A follow-up DC decoder for graph-based (LDPC) constituent codes has been recently presented in [11]. The performance of all decoders is evaluated and compared by simulations in Section VI.

II. Dual-parameter barrier channel and its capacity

In this section, we specialize the discussion to the probabilistic version of the dual-parameter barrier channel. We define the channel and derive its capacity for a general alphabet size, and in particular to the important ternary case. We designate 0 as the barrier symbol, and the remaining alphabet symbols are $1, \ldots, Q-1$. The probabilistic dual-parameter barrier channel $W_Q(p, q)$ is defined by two parameters: p and q, specifying the transition probabilities of *downward* (to 0) and *upward* (from 0) errors, respectively. A diagram describing $W_3(p, q)$ is given in Fig. 2.

Definition 1: For any input $X \in \mathbb{Z}_Q$ and output $Y \in \mathbb{Z}_Q$ and parameters $0 \le p, q \le 1$, the Q-ary dual-parameter barrier channel $W_Q(p, q)$ has the transition probabilities

$$P(Y=y|X=0) = \begin{cases} 1-q, & y=0, \\ q/(Q-1), & y \neq 0, \end{cases}$$
$$P(Y=y|X=x\neq 0) = \begin{cases} 1-p, & y=x, \\ p, & y=0, \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

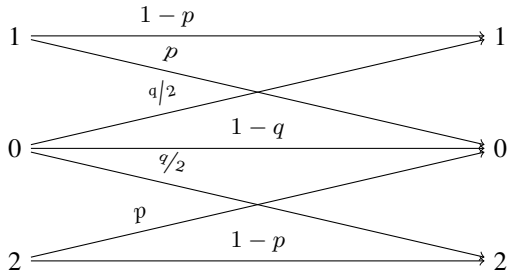


Figure 2: The dual-parameter barrier channel for $Q = 3$.

We now derive the channel capacity of the dual-parameter barrier channel defined in Definition 1. This generalizes the capacity of the single-parameter special case $W_3(q/2, q)$, derived in [4] and the capacity of the ternary dual-parameter barrier channel $W_3(p, q)$, derived in [12]. Unless noted otherwise,

logarithms are base 2 and the capacity is given in units of bits; $h_2(\cdot)$ is the binary entropy function.

Theorem 1: The capacity of $W_Q(p,q)$ for $p+q<1$ is given by

$$\begin{aligned} &h_2\left(\alpha_{p,q}(\varphi^*)\right)+\alpha_{p,q}(\varphi^*)\log(Q-1)\\ &\qquad-\beta_Q(p,q)\varphi^*-\left(h_2(q)+q\log(Q-1)\right), \end{aligned} \tag{2}$$

where

$$\begin{aligned} &\alpha_{p,q}(\varphi)\triangleq(1-p-q)\varphi+q,\\ &\beta_Q(p,q)\triangleq h_2(p)-h_2(q)-q\log(Q-1),\\ &\gamma_Q(p,q)\triangleq 1+Q^{\frac{\beta_Q(p,q)}{1-p-q}-\log(Q-1)},\\ &\text{and }\varphi^*=\min\left\{\frac{\gamma_Q(p,q)^{-1}-q}{1-p-q},1\right\}. \end{aligned} \tag{3}$$

Before we derive the capacity, we show that the capacity-achieving input distribution (CAID) of the channel is equal for all non-zero symbols, i.e., $\Pr\{X=1\}=\cdots=\Pr\{X=Q-1\}$.

Lemma 1: The capacity-achieving input distribution is of the form

$$\Phi^*(x)=\begin{cases}1-\varphi, & x=0\\ \varphi/(Q-1), & \text{otherwise}\end{cases}, \tag{4}$$

where $0\le\varphi\le 1$.

Proof: Define $\varphi\triangleq\sum_{x=1}^{Q-1}\Pr\{X=x\}$ and let $Z\triangleq\mathbb{I}_{[Y\neq 0]}$. Since Z is uniquely determined by Y, we can write $H(Y)=H(Y,Z)=H(Z)+H(Y|Z)$. Clearly, $H(Z)$ is a function of φ alone (due to the symmetry between non-zero symbols in the channel), and $H(Y|Z)$ decomposes to

$$\begin{aligned} &\Pr\{Z=0\}H(Y|Z=0)+\Pr\{Z=1\}H(Y|Z=1)\\ &\qquad=\Pr\{Z=1\}H(Y|Z=1). \end{aligned} \tag{5}$$

Now, $H(Y|Z=1)$ is maximized when $Y|Z=1$ has a uniform distribution, meaning that X should also be uniform over all $X\neq 0$, i.e., $\Pr\{X=x\}=\varphi/(Q-1)$. To complete the proof, we show that $H(Y|X)$ can be optimized by Φ^* as well. Toward this, we formulate $H(Y|X)$ as the following single-variable function of φ,

$$\begin{aligned} H(Y|X)&=\sum_{x=0}^{Q-1}\Pr\{X=x\}H(Y|X=x)\\ &=(1-\varphi)H(Y|X=0)+\sum_{x\neq 0}\Pr\{X=x\}H(Y|X=x)\\ &=(1-\varphi)\left[-(1-q)\log(1-q)-q\log(q/Q-1)\right]\\ &+\varphi\left[-p\log(p)-(1-p)\log(1-p)\right]\\ &=(1-\varphi)\left[h_2(q)+q\log(Q-1)\right]+\varphi h_2(p). \end{aligned} \tag{6}$$

■

We now prove Theorem 1 based on Lemma 1.

Proof: Given the form of Φ^*, we can rewrite $H(Y)$. First, note that

$$\begin{aligned} \Pr\{Y=0\}&=(1-\varphi)(1-q)+\varphi p\\ &=(p+q-1)\varphi+(1-q), \end{aligned} \tag{7}$$

and

$$\begin{aligned} \Pr\{Y=y\neq 0\}&=(1-\varphi)\frac{q}{Q-1}+\varphi\frac{1-p}{Q-1}\\ &=\frac{1-p-q}{Q-1}\varphi+\frac{q}{Q-1}. \end{aligned} \tag{8}$$

Denote $\alpha_{p,q}(\varphi)\triangleq(1-p-q)\varphi+q$ and note that $1-\alpha_{p,q}(\varphi)=(p+q-1)\varphi+(1-q)$. Hence, $\Pr\{Y=y\neq 0\}=\alpha_{p,q}(\varphi)/Q-1$ and $\Pr\{Y=0\}=1-\alpha_{p,q}(\varphi)$. Now,

$$\begin{aligned} H(Y)=&-(1-\alpha_{p,q}(\varphi))\log\left(1-\alpha_{p,q}(\varphi)\right)\\ &-(Q-1)\frac{\alpha_{p,q}(\varphi)}{Q-1}\log\left(\frac{\alpha_{p,q}(\varphi)}{Q-1}\right)\\ =&\ h_2\left(\alpha_{p,q}(\varphi)\right)+\alpha_{p,q}(\varphi)\log(Q-1)\\ =&\ \log_2(Q)\cdot H_Q(\alpha_{p,q}(\varphi)), \end{aligned} \tag{9}$$

where $H_Q(\cdot)$ is the Q-ary entropy function. We can now write the mutual information as

$$\begin{aligned} \mathcal{I}(X,Y)=&h_2\left(\alpha_{p,q}(\varphi)\right)+\alpha_{p,q}(\varphi)\log(Q-1)\\ &-\beta_Q(p,q)\varphi-\left(h_2(q)+q\log(Q-1)\right), \end{aligned} \tag{10}$$

where $\beta_Q(p,q)=h_2(p)-h_2(q)-q\log(Q-1)$. To maximize the latter expression, we take the derivative with respect to φ

$$\mathcal{I}'=(1-p-q)\log\left(\frac{1-\alpha_{p,q}(\varphi)}{\alpha_{p,q}(\varphi)}(Q-1)\right)-\beta_Q(p,q). \tag{11}$$

The second derivative,

$$\mathcal{I}''=\frac{(1-p-q)^2}{(\alpha_{p,q}(\varphi)-1)\alpha_{p,q}(\varphi)}, \tag{12}$$

is always negative since $\alpha_{p,q}(\varphi)$ is between q and $1-p$, both non-negative numbers smaller than 1. Therefore $\mathcal{I}$ is maximized by φ^* for which $\mathcal{I}'=0$. In case φ^* is larger than 1, it is guaranteed that setting it to 1 would maximize $\mathcal{I}$. Equating (11) to 0, we get

$$\log\left(\frac{1-\alpha_{p,q}(\varphi)}{\alpha_{p,q}(\varphi)}\right)=\frac{\beta_Q(p,q)}{1-p-q}-\log(Q-1) \tag{13}$$

which ultimately gives

$$\varphi^*=\min\left\{\frac{\gamma_Q(p,q)^{-1}-q}{1-p-q},1\right\}, \tag{14}$$

where $\gamma_Q(p,q)\triangleq 1+Q^{\frac{\beta_Q(p,q)}{1-p-q}-\log(Q-1)}$. ■

Remark 1: The capacity can also be derived with similar arguments for the (less interesting) case of $p+q>1$, but we omit this derivation. The special case of $p+q=1$ leads to $\varphi^*=1$ for every p, and capacity of $(1-p)\log(Q-1)$, which is equivalent to the capacity of the $(Q-1)$-ary erasure channel with parameter p and the 0 symbol acting as the erasure symbol.

Remark 2: The barrier channel over the binary alphabet ($Q=2$) coincides with the binary asymmetric channel (BAC) [13], and in particular these channels have the same capacity.

Fig. 3 depicts the channel capacity as a function of $p+q$, for several relations between p and q. Note that the channel obtained by not using the barrier symbol is simply the binary erasure channel BEC(p). It can be seen in Fig. 3 that its

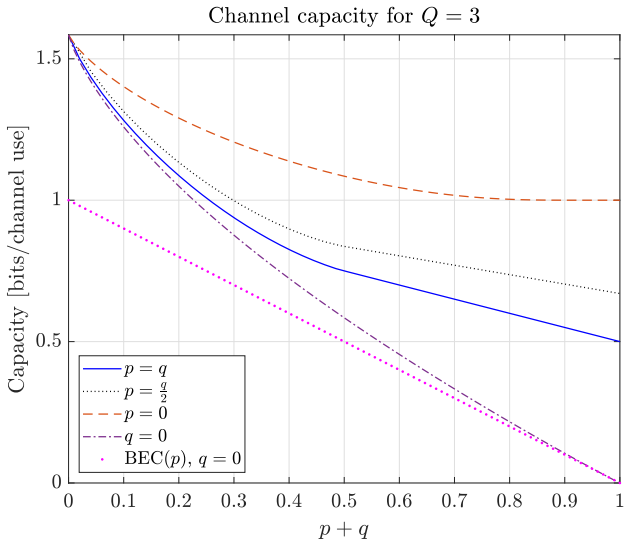


Figure 3: Capacity of the ternary dual-parameter barrier channel $W_3(p, q)$, for several relations between p and q.

capacity is significantly lower than the capacity of ternary barrier channels with various values of p and q. The plot specifically motivates the study of the *dual-parameter* version of the channel, by showing that the known special case $p = q/2$ has large gaps (both upward and downward) to other potentially interesting cases.

III. Dual-parameter barrier error correction

Toward constructing codes with correction guarantees, we move in this section to the *combinatorial* dual-parameter barrier error model. The Q-ary dual-parameter barrier error model $W_Q(t_d, t_u)$ is defined using two (non-negative integer) parameters, t_d and t_u, bounding from above the number of *downward* (non-barrier $\to$ barrier) and *upward* (barrier $\to$ non-barrier) errors, respectively. We denote the set of integers by $\mathbb{Z}$ and the subset of non-negative integers smaller than Q by $\mathbb{Z}_Q$. The notation $-\mathbb{Z}_Q$ denotes the set of corresponding negative integers, including 0. $\mathbb{F}_M$ represents the finite field with M elements. We further denote by $[n]$ the set of integers $\{1, 2, \ldots, n\}$. We now provide a definition of barrier errors over the alphabet $\mathbb{Z}_Q \triangleq \{0, \ldots, Q-1\}$, assuming 0 is the barrier symbol.

Definition 2: Given $n \in \mathbb{N}$ and a codeword $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{Z}_Q^n$, a vector $\boldsymbol{e} = (e_1, \ldots, e_n) \in \{-\mathbb{Z}_Q \cup \mathbb{Z}_Q\}^n$ is called a barrier error vector of $\boldsymbol{c}$ if for every index $i \in [n]$ such that $c_i \neq 0$ either $e_i = -c_i$ (downward error) or $e_i = 0$ (no error). If $c_i = 0$, e_i can be an arbitrary element of $\mathbb{Z}_Q$ (upward error if $e_i \neq 0$).

Given a codeword $\boldsymbol{c} \in \mathbb{Z}_Q^n$ transmitted through $W_Q(t_d, t_u)$, the output $\boldsymbol{r}$ equals $\boldsymbol{c} + \boldsymbol{e}$ (addition over the integers), where $\boldsymbol{e}$ is a (t_d, t_u) barrier error, defined as follows.

Definition 3: Given a codeword $\boldsymbol{c} \in \mathbb{Z}_Q^n$, a barrier error vector $\boldsymbol{e}$ (as defined in Definition 2) is a (t_d, t_u) barrier error if $|\{i \in [n] : e_i < 0\}| \leq t_d$ and $|\{i \in [n] : e_i > 0\}| \leq t_u$.
A code $\mathcal{C} \subseteq \mathbb{Z}_Q^n$ is said to be a (t_d, t_u) barrier-error-correcting code if it corrects any (t_d, t_u) barrier error.

A. Error correction guarantees

Throughout the paper we assume the standard definitions of Hamming weight $w_H(\boldsymbol{x}) \triangleq |\{i \in [n] : x_i \neq 0\}|$ and Hamming distance $d_H(\boldsymbol{x}, \boldsymbol{z}) = |\{i \in [n] : x_i \neq z_i\}|$ for $\boldsymbol{x} = (x_1, \ldots, x_n), \boldsymbol{z} = (z_1, \ldots, z_n) \in \mathbb{Z}_Q^n$. We also use the following indicator mapping function.

Definition 4: Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_Q^n$. The **indicator mapping** of $\boldsymbol{x}$ is defined as $\imath(\boldsymbol{x}) \triangleq (\imath(x_1), \ldots, \imath(x_n))$ where

$$\imath(x_j) = \begin{cases} 1, & x_j \in \mathbb{Z}_Q \setminus \{0\} \\ 0, & x_j = 0 \end{cases} . \tag{15}$$

Based on the indicator mapping we define the indicator distance.

Definition 5: For any $\boldsymbol{x} = (x_1, \ldots, x_n), \boldsymbol{z} = (z_1, \ldots, z_n) \in \mathbb{Z}_Q^n$, define the **indicator distance** as

$$d_\imath(\boldsymbol{x}, \boldsymbol{z}) = d_H(\imath(\boldsymbol{x}), \imath(\boldsymbol{z})). \tag{16}$$

Note that $d_\imath(\boldsymbol{x}, \boldsymbol{z})$ counts coordinates in which $x_i = 0, z_i \neq 0$ or $x_i \neq 0, z_i = 0$. Complementing the indicator distance to the standard Hamming distance is the residual distance, defined next.

Definition 6: For any $\boldsymbol{x} = (x_1, \ldots, x_n), \boldsymbol{z} = (z_1, \ldots, z_n) \in \mathbb{Z}_Q^n$, define the **residual distance** as

$$d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z}) = d_H(\boldsymbol{x}, \boldsymbol{z}) - d_\imath(\boldsymbol{x}, \boldsymbol{z}). \tag{17}$$

Note that $d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z})$ counts coordinates in which $0 \neq x_i \neq z_i \neq 0$.

We first prove a sufficient condition for guaranteed correction of (t_d, t_u) barrier errors. This condition is not tight (i.e., not necessary), but it serves as a basis for a practical construction proposed in Section IV. Later in the section we present a tighter condition that is both sufficient and necessary.

Proposition 1: Let $\mathcal{C} \subseteq \mathbb{Z}_Q^n$ be a code such that for any two codewords $\boldsymbol{x}, \boldsymbol{z} \in \mathcal{C}$, either $d_H(\boldsymbol{x}, \boldsymbol{z}) \geq 2(t_u + t_d) + 1$ or $d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z}) \geq t_d + 1$ (or both). Then, $\mathcal{C}$ is a (t_d, t_u) barrier-error-correcting code.

Proof: Let $\boldsymbol{c}$ be a codeword in a code $\mathcal{C}$ that satisfies the aforementioned condition. Let $\boldsymbol{r} \in \mathbb{Z}_Q^n$ be the output of $W_Q(t_d, t_u)$ for the input codeword $\boldsymbol{c} \in \mathcal{C}$, i.e., there exists a (t_d, t_u) barrier error $\boldsymbol{e}$ such that $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$. Assume there exists another codeword, $\boldsymbol{c}' \in \mathcal{C}$, such that $\boldsymbol{r} = \boldsymbol{c}' + \boldsymbol{e}'$ for some (t_d, t_u) barrier error $\boldsymbol{e}'$. We now consider two cases. If $d_H(\boldsymbol{c}, \boldsymbol{c}') \geq 2(t_u + t_d) + 1$, we immediately get a contradiction, since

$$d_H(\boldsymbol{c}, \boldsymbol{c}') \leq d_H(\boldsymbol{c}, \boldsymbol{r}) + d_H(\boldsymbol{c}', \boldsymbol{r}) \leq 2(t_u + t_d),$$

where the last inequality follows from the bound of $t_u + t_d$ on each of $d_H(\boldsymbol{c}, \boldsymbol{r})$, $d_H(\boldsymbol{c}', \boldsymbol{r})$. Therefore, $d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') \geq t_d + 1$. To show that this also implies a contradiction, define the following index-set sizes

$$\begin{aligned} M_1 &= |\{j \in [n] : 0 \neq c_j \neq c'_j \neq 0\}|, \\ M_2 &= |\{j \in [n] : r_j = 0 \text{ and } 0 \neq c_j \neq c'_j \neq 0\}|. \end{aligned} \tag{18}$$

The condition $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} = \boldsymbol{c}' + \boldsymbol{e}'$ with barrier errors $\boldsymbol{e}, \boldsymbol{e}'$ necessitates that $0 \neq c_j \neq c'_j \neq 0$ implies $r_j = 0$. Thus $M_1 = M_2$. But from the error model we know that $M_2 \leq t_d$

and from the distance assumption $M_1 = d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') \geq t_d + 1$, a contradiction. ∎

Remark 3: A trivial sufficient condition for correcting (t_d, t_u) barrier errors is one that guarantees correction of $t_d + t_u$ symmetric errors: $d_H(\boldsymbol{c}, \boldsymbol{c}') \geq 2(t_u + t_d) + 1$ for any $\boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}$. This condition is improved by Proposition 1 which guarantees correction with a weaker condition, namely allowing $\mathcal{C}$ to contain codewords $\boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}$ with $d_H(\boldsymbol{c}, \boldsymbol{c}') \leq 2(t_u + t_d)$, as long as $d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') \geq t_d + 1$.

To see that the condition in Proposition 1 is not necessary, we give the following example.

Example 1: Let $\mathcal{C} = \{(0,0,0,1), (2,2,2,2)\}$ and let $t_u = t_d = 1$. Obviously, the conditions in Proposition 1 are not satisfied, since

$$\begin{aligned} &d_H((0,0,0,1),(2,2,2,2)) = 4 < 2(t_u + t_d) + 1 = 5, \\ &d_{\bar{\imath}}((0,0,0,1),(2,2,2,2)) = 1 < t_d + 1 = 2. \end{aligned} \tag{19}$$

However, each of the two codewords can be recovered from any error pattern with $t_u = t_d = 1$ using the following decoder: declare the codeword to be $(0,0,0,1)$ if Majority$(r_1, r_2, r_3, r_4) = 0$, and $(2,2,2,2)$ otherwise.

Toward tightening the correction condition, we add the following definition.

Definition 7: For any $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{Z}_Q^n$, define $\delta_{\imath}(\boldsymbol{x}, \boldsymbol{z}) \triangleq |\{i \in [n] : \imath(x_i) = 1, \imath(z_i) = 0\}|$. Then the **asymmetric indicator distance** is defined as

$$d'_{\imath}(\boldsymbol{x}, \boldsymbol{z}) \triangleq \max(\delta_{\imath}(\boldsymbol{x}, \boldsymbol{z}), \delta_{\imath}(\boldsymbol{z}, \boldsymbol{x})). \tag{20}$$

Note that $d'_{\imath}(\boldsymbol{x}, \boldsymbol{z}) = d'_{\imath}(\boldsymbol{z}, \boldsymbol{x})$, and $d'_{\imath}(\boldsymbol{x}, \boldsymbol{z}) \geq d_{\imath}(\boldsymbol{x}, \boldsymbol{z})/2$. Similar definitions and properties appear in [14] studying codes for binary asymmetric channels.

Proposition 2: $\mathcal{C}$ is a (t_d, t_u) barrier-error-correcting code if and only if between any pair of codewords $\boldsymbol{x}, \boldsymbol{z} \in \mathcal{C}$, either $d'_{\imath}(\boldsymbol{x}, \boldsymbol{z}) + d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z}) \geq t_u + t_d + 1$ or $d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z}) \geq t_d + 1$ (or both).

Proof: Let $\boldsymbol{c}$ be a codeword in a code $\mathcal{C}$ that satisfies the aforementioned condition. Let $\boldsymbol{r} \in \mathbb{Z}_Q^n$ be the output of $W_Q(t_d, t_u)$ for the input codeword $\boldsymbol{c} \in \mathcal{C}$, i.e., there exists a (t_d, t_u) barrier error $\boldsymbol{e}$ such that $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$. Assume there exists another codeword, $\boldsymbol{c}' \in \mathcal{C}$, such that $\boldsymbol{r} = \boldsymbol{c}' + \boldsymbol{e}'$ for some (t_d, t_u) barrier error $\boldsymbol{e}'$. The sufficiency of the second condition was already proved in Proposition 1, so we assume $d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') \leq t_d$ and prove that $d'_{\imath}(\boldsymbol{c}, \boldsymbol{c}') + d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') \geq t_u + t_d + 1$ leads to a contradiction. Define M_1 as in (18) (i.e., $M_1 = |\{j \in [n] : 0 \neq c_j \neq c'_j \neq 0\}|$), and also define

$$\begin{aligned} M &= |\{j \in [n] : c_j \neq 0 \text{ and } c'_j = 0\}|, \\ M' &= |\{j \in [n] : c'_j \neq 0 \text{ and } c_j = 0\}|. \end{aligned} \tag{21}$$

Such $\boldsymbol{r}$ exists if and only if the M indices in (21) can be covered by t_u upward errors in $\boldsymbol{c}'$ and $t_d - M_1$ downward errors in $\boldsymbol{c}$, and similarly, the M' indices can be covered by t_u upward errors in $\boldsymbol{c}$ and $t_d - M_1$ downward errors in $\boldsymbol{c}'$. This implies $t_u + t_d - M_1 \geq \max(M, M')$. By definition, $M = \delta_{\imath}(\boldsymbol{c}, \boldsymbol{c}')$, $M' = \delta_{\imath}(\boldsymbol{c}', \boldsymbol{c})$, and $M_1 = d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}')$, and substituting these and (20) to the last inequality gives $t_u + t_d \geq d'_{\imath}(\boldsymbol{c}, \boldsymbol{c}') + d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}')$, a contradiction. For necessity, given $\boldsymbol{c}, \boldsymbol{c}'$ that violate the condition, we can find barrier errors $\boldsymbol{e}, \boldsymbol{e}'$ that cover all the indices counted in M_1, M, M', and thus lead to the same $\boldsymbol{r}$. ∎

B. Bounds on the maximal code size

We now derive upper and lower bounds on the size of the largest (t_d, t_u) barrier-error-correcting code. The analogue of a Hamming ball for the barrier channel is the (t_d, t_u) ball defined next.

Definition 8: Define the (t_d, t_u) ball around a word $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_Q^n$ as $\mathcal{E}_{(t_d,t_u)}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{Z}_Q^n : \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}\}$, where $\boldsymbol{e} = (e_1, \ldots, e_n)$ is a (t_d, t_u)-barrier error.

Note that $\mathcal{E}_{(t_d,t_u)}(\boldsymbol{x})$ defines the set of possible channel outputs when $\boldsymbol{x}$ is transmitted through $W_Q(t_d, t_u)$. The number of words in a (t_d, t_u) ball is a key ingredient in the derivation of code-size bounds.

Proposition 3: Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_Q^n$ and $t_d, t_u \in \mathbb{N}$. The size of the set $\mathcal{E}_{(t_d,t_u)}(\boldsymbol{x})$ is given by

$$|\mathcal{E}_{(t_d,t_u)}(\boldsymbol{x})| = \sum_{\tau_d=0}^{\alpha_{\boldsymbol{x}}} \binom{\omega_{\boldsymbol{x}}}{\tau_d} \cdot \sum_{\tau_u=0}^{\beta_{\boldsymbol{x}}} \binom{n - \omega_{\boldsymbol{x}}}{\tau_u} (Q-1)^{\tau_u}, \tag{22}$$

where $\omega_{\boldsymbol{x}} \triangleq w_H(\boldsymbol{x})$, $\alpha_{\boldsymbol{x}} \triangleq \min\{\omega_{\boldsymbol{x}}, t_d\}$ and $\beta_{\boldsymbol{x}} \triangleq \min\{n - \omega_{\boldsymbol{x}}, t_u\}$.

Proof: Let $\omega_{\boldsymbol{x}} \triangleq w_H(\boldsymbol{x})$ be the number of non-zero elements in $\boldsymbol{x}$. We evaluate the set size by partitioning $\boldsymbol{x}$ to two subsets of indices:

1) $\{i : x_i = 0\}$: Each zero element in $\boldsymbol{x}$ has $Q-1$ possible non-zero assignments, and given $\tau_u \leq \min\{n - \omega_{\boldsymbol{x}}, t_u\}$ such transitions, there are $\binom{n-\omega_{\boldsymbol{x}}}{\tau_u}$ possible index subsets. Summing over all possible words with τ_u transitions yields the second sum in the product in (22).
2) $\{i : x_i \neq 0\}$: Each non-zero element has one possible erroneous assignment (transition to 0), and given $\tau_d \leq \min\{\omega_{\boldsymbol{x}}, t_d\}$ such transitions, there are $\binom{\omega_{\boldsymbol{x}}}{\tau_d}$ possible index subsets. Summing over all possible τ_d assignments yields the first sum in the product in (22).

Since the two subsets of indices are disjoint, the product of sizes yields the size of $\mathcal{E}_{(t_d,t_u)}(\boldsymbol{x})$. ∎

Note that the size of the set $\mathcal{E}_{(t_d,t_u)}(\boldsymbol{x})$ depends on $w_H(\boldsymbol{x})$ rather than $\boldsymbol{x}$. We therefore rename it $|\mathcal{E}_{(t_d,t_u)}(\omega)|$, where $0 \leq \omega \leq n$. An upper bound using these ball volumes will depend on the code's *weight support*, defined next.

Definition 9: Let $\mathcal{C} \subseteq \mathbb{Z}_Q^n$ be a code of length n. Define the weight support of $\mathcal{C}$, $\Omega(\mathcal{C}) \subseteq \{0, 1 \ldots, n\}$, as the set of Hamming weights of codewords in $\mathcal{C}$.

The following proposition provides an upper bound on the size of any (t_d, t_u) barrier-error-correcting code.

Proposition 4: Let $t_d, t_u \in \mathbb{N}$ and let $\mathcal{C} \subseteq \mathbb{Z}_Q^n$ be a (t_d, t_u) barrier-error-correcting code with weight support Ω. Then,

$$|\mathcal{C}| \leq \frac{Q^n}{\min_{\omega \in \Omega} |\mathcal{E}_{(t_d,t_u)}(\omega)|}. \tag{23}$$

Proof: Let $\mathcal{C}$ be a (t_d, t_u) barrier-error-correcting code with given weight support Ω. For every two codewords $\boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}$, the corresponding (t_d, t_u) balls, $\mathcal{E}_{(t_d,t_u)}(\boldsymbol{c})$ and $\mathcal{E}_{(t_d,t_u)}(\boldsymbol{c}')$,

do not intersect (otherwise, there exists a (t_d, t_u) barrier error that cannot be corrected). Consequently,

$$\begin{aligned} Q^n = |\mathbb{Z}_Q^n| &\geq |\cup_{\boldsymbol{c}\in\mathcal{C}} \mathcal{E}_{(t_d,t_u)}(\boldsymbol{c})| = \sum_{\boldsymbol{c}\in\mathcal{C}} |\mathcal{E}_{(t_d,t_u)}(\boldsymbol{c})| \\ &\geq |\mathcal{C}| \min_{\boldsymbol{c}\in\mathcal{C}} |\mathcal{E}_{(t_d,t_u)}(\boldsymbol{c})| = |\mathcal{C}| \min_{\omega\in\Omega} |\mathcal{E}_{(t_d,t_u)}(\omega)|. \end{aligned} \tag{24}$$

■

Toward deriving a *lower bound* on the maximal size of a (t_d, t_u) barrier-error-correcting code, we now define another type of "ball" in $\mathbb{Z}_Q^n$ and calculate its volume. In symmetric error correction, the balls used for lower bounds (Gilbert Varshamov (GV) [15], [16]) are the same as the balls for upper bounds, only with the radius doubled. In the barrier error model there is no such simple doubling that we can use. Instead, we base the lower bounds on balls of words violating the sufficient conditions of Propositions 1 and 2. We start with the ball corresponding to Proposition 1.

Definition 10: For a word $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_Q^n$, define $\mathcal{B}^{(1)}_{(t_d,t_u)}(\boldsymbol{x})$ to be the set consisting of all words $\boldsymbol{y} \in \mathbb{Z}_Q^n$ such that $d_H(\boldsymbol{x}, \boldsymbol{y}) \leq 2(t_u + t_d)$ *and* $d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{y}) \leq t_d$.

Proposition 5: Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_Q^n$, $t_d, t_u \in \mathbb{N}$ and denote $\omega_{\boldsymbol{x}} \triangleq w_H(\boldsymbol{x})$. The size of the set $\mathcal{B}^{(1)}_{(t_d,t_u)}(\boldsymbol{x})$ is

$$\begin{aligned} |\mathcal{B}^{(1)}_{(t_d,t_u)}(\boldsymbol{x})| = \sum_{\tau_u=0}^{\alpha'_{\boldsymbol{x}}} \binom{n-\omega_{\boldsymbol{x}}}{\tau_u} (Q-1)^{\tau_u} \\ \cdot \sum_{\tau_d=0}^{\beta'_{\boldsymbol{x}}} \binom{\omega_{\boldsymbol{x}}}{\tau_d} \cdot \sum_{\tau_r=0}^{\gamma'_{\boldsymbol{x}}} \binom{\omega_{\boldsymbol{x}}-\tau_d}{\tau_r} (Q-2)^{\tau_r}, \end{aligned} \tag{25}$$

where $\alpha'_{\boldsymbol{x}} \triangleq \min\{n - \omega_{\boldsymbol{x}}, 2(t_u + t_d)\}$, $\beta'_{\boldsymbol{x}} \triangleq \min\{\omega_{\boldsymbol{x}}, 2(t_u + t_d) - \tau_u\}$ and $\gamma'_{\boldsymbol{x}} \triangleq \min\{\omega_{\boldsymbol{x}} - \tau_d, t_d, 2(t_d + t_u) - \tau_d - \tau_u\}$.

Proof: To satisfy the first condition, $\tau_u \leq 2(t_u + t_d)$ zero indices change to non-zeros, and $\tau_d \leq 2(t_u + t_d) - \tau_u$ non-zero indices change to zeros. These give the first two sums in the product (25), which also consider the respective bounds $n - \omega_{\boldsymbol{c}}$ and $\omega_{\boldsymbol{c}}$ on τ_u and τ_d. To satisfy the second condition, $\tau_r \leq t_d$ non-zero indices (not selected by the τ_d above) change to a different non-zero symbol. This adds the third sum in (25). ■

Similarly, the balls corresponding to Proposition 2:

Definition 11: For a word $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_Q^n$, define $\mathcal{B}^{(2)}_{(t_d,t_u)}(\boldsymbol{x})$ to be the set consisting of all words $\boldsymbol{y} \in \mathbb{Z}_Q^n$ such that $d'_i(\boldsymbol{x}, \boldsymbol{y}) + d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{y}) \leq t_u + t_d$ *and* $d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{y}) \leq t_d$.

The size of $\mathcal{B}^{(2)}_{(t_d,t_u)}(\boldsymbol{x})$ can be found identically to Proposition 5, but replacing in (25) to the values

$$\begin{aligned} \alpha'_{\boldsymbol{x}} &\triangleq \min\{n - \omega_{\boldsymbol{x}}, t_u + t_d\}, \\ \beta'_{\boldsymbol{x}} &\triangleq \min\{\omega_{\boldsymbol{x}}, t_u + t_d\}, \\ \gamma'_{\boldsymbol{x}} &\triangleq \min\{\omega_{\boldsymbol{x}} - \tau_d, t_d, t_d + t_u - \max\{\tau_d, \tau_u\}\}. \end{aligned} \tag{26}$$

From now on we use $\mathcal{B}_{(t_d,t_u)}$ to represent either $\mathcal{B}^{(1)}_{(t_d,t_u)}$ or $\mathcal{B}^{(2)}_{(t_d,t_u)}$, and the results apply to both. Note that $|\mathcal{B}_{(t_d,t_u)}(\boldsymbol{x})|$ depends only on the weight $\omega_{\boldsymbol{x}}$, hence, we denote it $|\mathcal{B}_{(t_d,t_u)}(\omega_{\boldsymbol{x}})|$. Since $\mathcal{B}_{(t_d,t_u)}(\boldsymbol{c})$ contains all the words violating the sufficient condition of Proposition 1 or 2 with respect to a codeword $\boldsymbol{c}$, greedily selecting $\boldsymbol{c}$ to be in the code, and removing $\mathcal{B}_{(t_d,t_u)}(\boldsymbol{c})$ from the list of candidate codewords, guarantees ending up with a (t_d, t_u) barrier code. This gives a lower bound on the maximal code size, similar to the GV bound.

Proposition 6: Let $t_d, t_u \in \mathbb{N}$ and let $\mathcal{C} \subseteq \mathbb{Z}_Q^n$ be the largest possible (t_d, t_u) barrier-error-correcting code with weight support Ω. Then,

$$|\mathcal{C}| \geq \frac{\sum_{\omega\in\Omega} \binom{n}{\omega}(Q-1)^\omega}{\max_{\omega\in\Omega} |\mathcal{B}_{(t_d,t_u)}(\omega)|}. \tag{27}$$

Proof: To prove the inequality, we use the following chain of equalities and inequalities

$$\begin{aligned} \sum_{\omega\in\Omega} \binom{n}{\omega}(Q-1)^\omega &= |\{\boldsymbol{x} \in \mathbb{Z}_Q^n : w_H(\boldsymbol{x}) \in \Omega(\mathcal{C})| \\ &\leq |\cup_{\boldsymbol{c}\in\mathcal{C}} \mathcal{B}_{(t_d,t_u)}(\boldsymbol{c})| \leq \sum_{\boldsymbol{c}\in\mathcal{C}} |\mathcal{B}_{(t_d,t_u)}(\boldsymbol{c})| \\ &\leq |\mathcal{C}| \max_{\boldsymbol{c}\in\mathcal{C}} |\mathcal{B}_{(t_d,t_u)}(\boldsymbol{c})| = |\mathcal{C}| \max_{\omega\in\Omega} |\mathcal{B}_{(t_d,t_u)}(\omega)|, \end{aligned} \tag{28}$$

where the first equality is the standard counting of weight-ω vectors, the first inequality is from the fact that the greedy codeword selection stops only when all words within the weight support are contained in a codeword ball, the second inequality is because sum of sizes is never smaller than the union size, and the third inequality bounds the sum of sizes by a product with the largest size. ■

Proposition 6 provides a general lower bound on the optimal code size, which can be improved by adding knowledge on the code's weight support. In the case where $\Omega(\mathcal{C})$ is unknown or specifying Ω is undesired, one can simply set $\Omega = \{0, 1, \ldots, n\}$ to obtain general (looser) bounds from (23) and (27). The bound can be further improved using the Generalized Gilbert-Varshamov bound [17]. This amounts to changing the maximal sphere volume in the denominator of (27) to the *average* sphere volume, which gives a strictly better bound.

Proposition 7: Let $t_d, t_u \in \mathbb{N}$ and let $\mathcal{C} \subseteq \mathbb{Z}_Q^n$ be the largest possible (t_d, t_u) barrier-error-correcting code with weight support Ω. Define a_ω to be the fraction of $\mathbb{Z}_Q^n$ words with weight ω out of the words with weights in Ω. Then,

$$|\mathcal{C}| \geq \frac{\sum_{\omega\in\Omega} \binom{n}{\omega}(Q-1)^\omega}{\sum_{\omega\in\Omega} a_\omega |\mathcal{B}_{(t_d,t_u)}(\omega)|}. \tag{29}$$

Proof: This result follows from the proof of Proposition 6 and the generalized Gilbert-Varshamov lower bound in [17]. Specifically, we set $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{Z}_Q^n : w_H(\boldsymbol{x}) \in \Omega\}$ and use the proof of Theorem 4 from [17]. For $\mathcal{B}^{(1)}_{(t_d,t_u)}$, we set $\rho(\boldsymbol{x}, \boldsymbol{z}) = \max\{d_H(\boldsymbol{x}, \boldsymbol{z}), 2t_u + t_d + d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z})\}$ and $d = 2(t_u + t_d) + 1$; while for $\mathcal{B}^{(2)}_{(t_d,t_u)}$, we set $\rho(\boldsymbol{x}, \boldsymbol{z}) = \max\{d'_i(\boldsymbol{x}, \boldsymbol{z}) + d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z}), t_u + d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z})\}$ and $d = t_u + t_d + 1$. ■

To effectively use the bounds stated above, we need to determine Ω wisely. We focus on the lower bounds, because in that case guessing a "good" Ω can prove the existence of a large code with that weight support. Luckily, a simple technique of bounding ω from below gives lower bounds that are better than the known lower bounds, as we show in the following example.

Example 2: For $Q = 3$ we fix t_d to be either 1 or 3 and set $\Omega = \{\omega_{\min}, \ldots, n\}$, where $\omega_{\min}$ is optimized to achieve the largest lower bound for each $t_u \in \{0, \ldots, {}^{n}/_{4}\}$ separately. For Prop. 7, we substitute $a_w = \frac{\binom{n}{\omega}(Q-1)^{\omega}}{\sum_{\omega\in\Omega}\binom{n}{\omega}(Q-1)^{\omega}}$. The results are shown in Fig. 4 for $n = 32$ (upper row) and $n = 128$ (lower row). Each plot compares the values of three lower bounds derived here for (t_d, t_u) barrier errors (solid) with two lower bounds for $t_d + t_u$ barrier errors in any direction from [10] (dotted). In addition, we plot the GV bound for $t_d + t_u$ symmetric errors (dashed). The three new bounds we plot are, in increasing level of tightness: the bound of Prop. 6 based on $\mathcal{B}^{(1)}_{(t_d,t_u)}$, the bound of Prop. 6 based on $\mathcal{B}^{(2)}_{(t_d,t_u)}$, and the bound of Prop. 7 based on $\mathcal{B}^{(2)}_{(t_d,t_u)}$. It can be seen that the new bounds based on $\mathcal{B}^{(2)}$ improve significantly over the previously known bounds. The improvement of Prop. 7 over Prop. 6 is more significant when n is small, probably due to the dominance of large ball volumes in the enumeration of the average ball size. Even the less tight bound based on $\mathcal{B}^{(1)}$ shows improvement over the prior bounds for large t_u values, illustrating the promise of the code construction of Section IV that is driven by the sufficient condition of Prop. 1.

IV. Construction for (t_d, t_u)-barrier errors

In this section, we construct a code that is designated specifically for correcting (t_d, t_u)-barrier errors. A new construction is required since the code constructed in [4] is designed for a weaker error model of t (downward/upward) barrier errors, and setting $t = t_d + t_u$ for correcting (t_d, t_u)-barrier errors is naturally sub-optimal. Additionally, the construction of [4] uses multiple outer erasure codes of *different lengths*, which we also aim to avoid towards allowing simpler encoding and decoding procedures.

A. Code construction

We now construct such a code and prove that it corrects (t_d, t_u)-barrier errors.

Definition 12 (Indicator code): Given $t_d, t_u \in \mathbb{N}$ and $n \in \mathbb{N}$, define the indicator code as a binary code with length n and minimum Hamming distance $2(t_d + t_u) + 1$.

Definition 13 (Base residual code): Given $t_d \in \mathbb{N}$ and $n \in \mathbb{N}$, define the base residual code as a code over $\mathbb{Z}^n_{Q-1}$ with length n and minimum Hamming distance $t_d + 1$.

Construction 1: Let Θ be an indicator code for parameters t_d, t_u and length n; and let Λ be a base residual code for parameter t_d and length n. A word $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathbb{Z}^n_Q$ is a codeword in $\mathcal{C} = \Theta \otimes \Lambda$ if the following conditions hold:

1) There exists $\boldsymbol{\theta} \in \Theta$ such that for every $j = 1, \ldots, n$, $\theta_j = \imath(c_j)$.
2) There exists $\boldsymbol{\lambda} \in \Lambda$ such that for every $j = 1, \ldots, n$, $\lambda_j = c_j - 1$ if $c_j \neq 0$, and $\lambda_j = 0$ if $c_j = 0$.

The supports of the constituent codewords satisfy a property described next.

Definition 14 (Support of $\boldsymbol{x}$): For a word $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}^n_Q$ of length $n \in \mathbb{N}$, we denote the set of indices of non-zero elements as

$$\omega^{(\boldsymbol{x})} \triangleq \{i \in [n] : x_i \neq 0\}. \tag{30}$$

Remark 4: Given codewords $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{\lambda} \in \Lambda$ that constitute a codeword $\boldsymbol{c} \in \mathcal{C}$, note that Properties 1 and 2 of Construction 1 guarantee that the support of $\boldsymbol{\lambda}$ is a subset of the support of $\boldsymbol{\theta}$, i.e., $\omega^{(\boldsymbol{\lambda})} \subseteq \omega^{(\boldsymbol{\theta})}$.

Example 3: Let $Q = 3$, $t_d = t_u = 1$, and consider the indicator code $\Theta = \{(1,1,1,0,0,0,0,0), (1,1,1,1,1,1,1,1)\}$ and base residual code $\Lambda = \{(1,1,0,0,0,0,0,0), (0,0,1,0,0,0,0,0)\}$ (note that both codes satisfy the respective distance requirements from Definitions 12 and 13). Consequently, the resulting barrier code is

$$\mathcal{C} = \left\{ \begin{matrix} (2,2,1,0,0,0,0,0), (1,1,2,0,0,0,0,0), \\ (2,2,1,1,1,1,1,1), (1,1,2,1,1,1,1,1) \end{matrix} \right\}. \tag{31}$$

The codeword $(2,2,1,0,0,0,0,0)$ is a combination of the indicator codeword $(1,1,1,0,0,0,0,0)$ and the residual codeword $(1,1,0,0,0,0,0,0)$. Note that the 0 in the third position in the residual codeword maps to a 1 in the ternary codeword (because $\theta_3 = 1$), while the 0 in the fourth position in the residual codeword maps to a 0 in the ternary codeword (because $\theta_4 = 0$).

To prove the correction capability of the code from Construction 1, we show that the constructed code satisfies the requirements from Proposition 1 for guaranteed correction of any (t_d, t_u)-barrier error.

Theorem 2: Given $t_d, t_u \in \mathbb{N}$ and $n \in \mathbb{N}$, let $\mathcal{C} \subseteq \mathbb{Z}^n_Q$ be a code constructed by Construction 1. Then, $\mathcal{C}$ can correct any (t_d, t_u)-barrier error.

Proof: Let $\boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}$ be a pair of (different) arbitrary codewords ($\boldsymbol{c} \neq \boldsymbol{c}'$). According to Construction 1,

$$d_\imath(\boldsymbol{c}, \boldsymbol{c}') = d_H(\imath(\boldsymbol{c}), \imath(\boldsymbol{c}')) = d_H(\boldsymbol{\theta}, \boldsymbol{\theta}'), \tag{32}$$

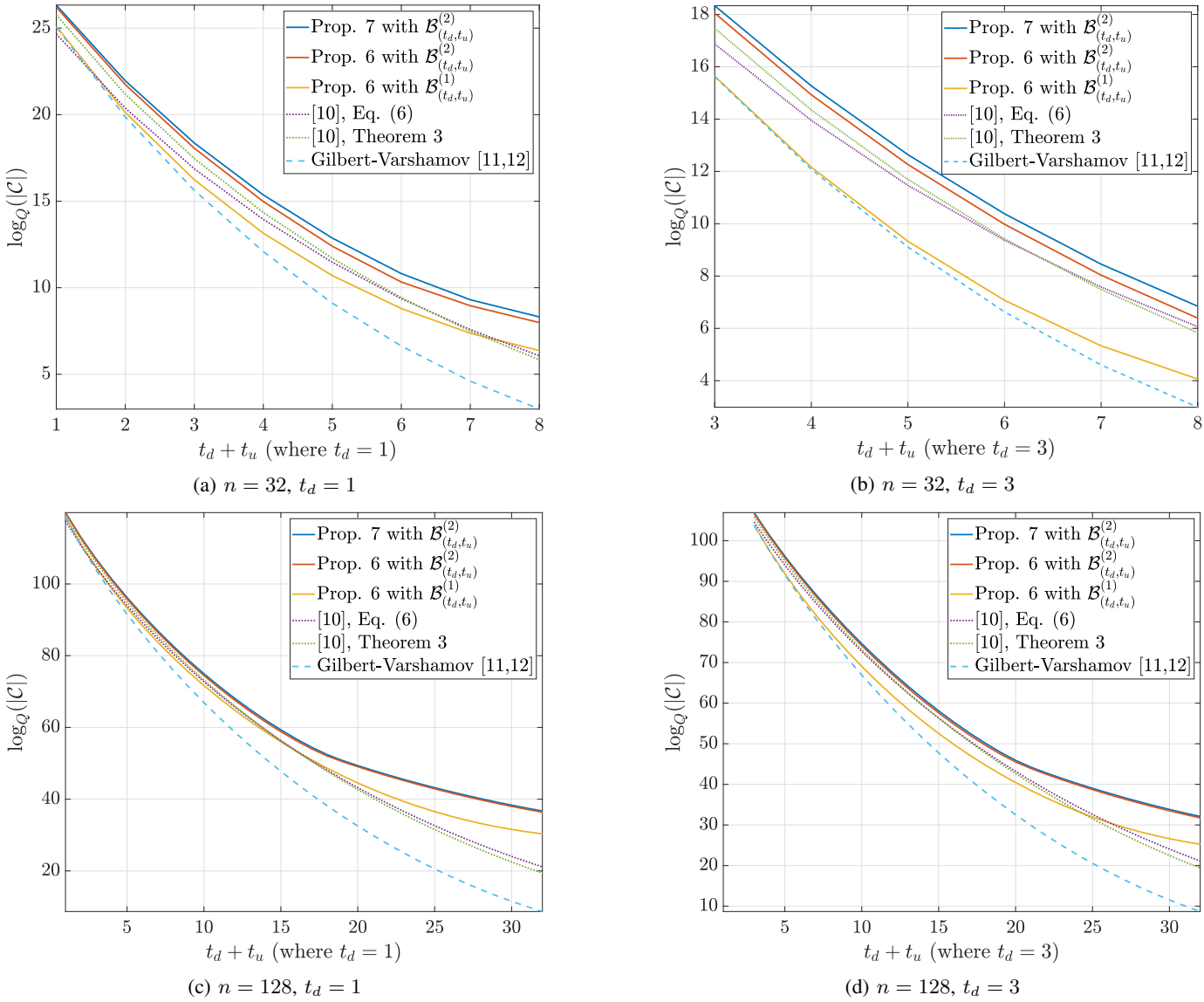
where $\boldsymbol{\theta}, \boldsymbol{\theta}'$ are codewords in Θ. If $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$, then $d_H(\boldsymbol{\theta}, \boldsymbol{\theta}') \geq 2(t_d + t_u) + 1$ (according to the minimum distance of Θ) thereby satisfying the first sufficient condition from Proposition 1. Otherwise $\boldsymbol{\theta} = \boldsymbol{\theta}'$ and it is left to show that $d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') \geq t_d + 1$. In this case, $d_\imath(\boldsymbol{c}, \boldsymbol{c}') = d_H(\imath(\boldsymbol{c}), \imath(\boldsymbol{c}')) = d_H(\boldsymbol{\theta}, \boldsymbol{\theta}') = 0$, hence

$$d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') = d_H(\boldsymbol{c}, \boldsymbol{c}') - d_\imath(\boldsymbol{c}, \boldsymbol{c}') = d_H(\boldsymbol{c}, \boldsymbol{c}'). \tag{33}$$

Now, according to the second condition of Construction 1, $\lambda_j \neq \lambda'_j$ implies $c_j \neq c'_j$, thus $d_H(\boldsymbol{c}, \boldsymbol{c}') \geq d_H(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$, and since $\boldsymbol{\lambda} \neq \boldsymbol{\lambda}'$ (otherwise, $\boldsymbol{c} = \boldsymbol{c}'$), it is guaranteed that $d_H(\boldsymbol{\lambda}, \boldsymbol{\lambda}') \geq t_d + 1$ from the minimum distance of Λ, and $d_{\bar{\imath}}(\boldsymbol{c}, \boldsymbol{c}') \geq t_d + 1$ follows by (33) and a chain of two inequalities. ∎

Our new suggested construction actually satisfies a condition stronger than the sufficient condition in Proposition 1. The constructed code contains only codeword pairs that satisfy either $d_\imath(\boldsymbol{x}, \boldsymbol{z}) \geq 2(t_u + t_d) + 1$ or $d_\imath(\boldsymbol{x}, \boldsymbol{z}) = 0$ and $d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z}) \geq t_d + 1$. It does not contain potentially "legal" codewords for which $d_{\bar{\imath}}(\boldsymbol{x}, \boldsymbol{z}) \geq t_d + 1$ yet $0 < d_\imath(\boldsymbol{x}, \boldsymbol{z}) \leq 2(t_u + t_d)$.

As in [4], our construction also employs a compound structure in which every Q-ary codeword is constructed from codewords originating in two constituent codes: 1) an indicator code, and 2) a residual code. However, [4] uses a set of residual codes *with different lengths* to encode the non-zero symbols



(a) $n = 32$, $t_d = 1$

(b) $n = 32$, $t_d = 3$

(c) $n = 128$, $t_d = 1$

(d) $n = 128$, $t_d = 3$

Figure 4: Comparison of the lower bounds in this paper with bounds from [10] and with the GV lower bound, for $Q = 3$, as a function of the total number of correctable errors $(t_d + t_u)$ where $t_d = 1$ or $t_d = 3$. Each point was obtained by setting the minimum weight in Ω to maximize the value of the bounds.

in each codeword, while we use sub-codes of a single base residual code, thereby allowing simpler encoding/decoding.

B. Structured encoder

To allow efficient encoding of Construction 1, we now formalize the structural relationship between a codeword in $\mathcal{C}$ and the two corresponding constituent codewords in Θ and Λ. To this end, we define the residual sub-code $\Lambda_{\boldsymbol{\theta}}$ of the base code Λ, induced by an arbitrary indicator codeword $\boldsymbol{\theta} \in \Theta$.

Definition 15 ($\boldsymbol{\theta}$-induced sub-code): Let $\Lambda \subseteq \mathbb{Z}_{Q-1}^n$ be a code and let $\boldsymbol{\theta} \in \mathbb{Z}_2^n$. The $\boldsymbol{\theta}$-induced sub-code of Λ is a subset $\Lambda_{\boldsymbol{\theta}} \subseteq \Lambda$ such that for every $\boldsymbol{\lambda} \in \Lambda_{\boldsymbol{\theta}}$

$$\imath(\boldsymbol{\lambda}) \leq \boldsymbol{\theta}, \tag{34}$$

where the inequality is bit-wise. Note that a $\boldsymbol{\lambda}$ from $\Lambda_{\boldsymbol{\theta}}$ satisfies Property 2 in Construction 1 of having zeros in all coordinates where $\boldsymbol{c}$ (and $\boldsymbol{\theta}$) have zeros (and possibly more due to the inequality in (34)).

Now, using a constituent codeword pair: an indicator codeword $\boldsymbol{\theta} \in \Theta$ and a residual codeword $\boldsymbol{\lambda}$ from the $\boldsymbol{\theta}$-induced residual code $\Lambda_{\boldsymbol{\theta}}$, we can encode a codeword in $\mathcal{C} = \Theta \otimes \Lambda$ as

$$\boldsymbol{c} = \Psi(\boldsymbol{\theta}, \boldsymbol{\lambda}) \triangleq \boldsymbol{\theta} \cdot (\mathbf{1} + \boldsymbol{\lambda}), \tag{35}$$

where $\boldsymbol{a} \cdot \boldsymbol{b}$ denotes the element-wise multiplication of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ and $\mathbf{1}$ is a vector of ones. The sub-code structure of Λ allows the simple encoding function of (35) (applying to every $\boldsymbol{\theta}, \boldsymbol{\lambda}$ pair), but its most compelling advantage over [4] lies in *decoding*: one can decode the residual codeword even before the indicator codeword is known. Such decoders are pursued in Section V.

C. Residual encoding/decoding via its parity-check matrix

In general, the indicator and/or base residual codes are not assumed to be linear. Yet, in the remainder of this section we restrict our attention to linear base-residual codes, thanks to their many advantages in encoding and decoding. Even when the base residual code Λ has an efficient encoder, the encoding of $\mathcal{C}$ requires the more refined task of encoding to a specific sub-code $\Lambda_{\boldsymbol{\theta}}$ (the sub-codes in general have different sizes). Toward that task, we now define $\boldsymbol{\theta}$-induced parity-check matrices using the parity-check matrix of the base residual code.

Definition 16: For a matrix $\boldsymbol{X}$ and a binary vector $\boldsymbol{\theta} \in \mathbb{Z}_2^n$, we use the notation $\boldsymbol{X}(\omega^{(\boldsymbol{\theta})})$ to represent the sub-matrix generated by the columns of $\boldsymbol{X}$ whose indices are in $\omega^{(\boldsymbol{\theta})}$.

Definition 17: ($\boldsymbol{\theta}$-induced matrix) Let $\boldsymbol{\theta} \in \mathbb{Z}_2^n$ and let $\boldsymbol{H} \in \mathbb{F}_{Q-1}^{m\times n}$ ($m \leq n$) be a matrix. The $\boldsymbol{\theta}$-induced matrix is defined as

$$\boldsymbol{H}_{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{H} \\ \left(\boldsymbol{I}_n\left([n] \setminus \omega^{(\boldsymbol{\theta})}\right)\right)^\top \end{bmatrix}, \tag{36}$$

where the notation $\boldsymbol{I}_n$ stands for the identity matrix with dimensions $n \times n$.

Proposition 8: (parity-check matrix of $\Lambda_{\boldsymbol{\theta}}$) Let $\boldsymbol{\theta} \in \mathbb{Z}_2^n$ and let Λ be a linear code with a parity-check matrix $\boldsymbol{H}$. Then, the code defined by the parity-check matrix $\boldsymbol{H}_{\boldsymbol{\theta}}$ is the $\boldsymbol{\theta}$-induced sub-code of Λ.

Proof: Let $\boldsymbol{\theta} \in \mathbb{Z}_2^n$ and let $\boldsymbol{H}$ be a parity-check matrix of Λ. Define $Z_{\boldsymbol{\theta}}$ as the code whose parity-check matrix is $\boldsymbol{H}_{\boldsymbol{\theta}}$, defined in Eq. (36). To prove the claim, we show that $Z_{\boldsymbol{\theta}}$ is a subset of Λ and that Eq. (34) holds for its codewords. For any $\boldsymbol{\lambda} \in Z_{\boldsymbol{\theta}}$, we naturally have $\boldsymbol{\lambda} \in \Lambda$ thanks to the upper part of the parity-check matrix in (36), which guarantees $\boldsymbol{H}\boldsymbol{\lambda} = \boldsymbol{0}$.

To prove that (34) holds, recall that $\left(\boldsymbol{I}_n\left([n] \setminus \omega^{(\boldsymbol{\theta})}\right)\right)^\top \boldsymbol{\lambda} = \boldsymbol{0}$ (thanks to the lower part of (36)), which implies $\lambda_j = 0$ for every $j \in [n] \setminus \omega^{(\boldsymbol{\theta})}$. For these indices, we get

$$\lambda_j = \imath\left(\lambda_j\right) = \theta_j = 0, \tag{37}$$

satisfying (34) on these indices with equality, while for the remaining indices the inequality is trivially true since $\theta_j = 1$ and $\imath\left(\lambda_j\right)$ is binary. This proves that $Z_{\boldsymbol{\theta}}$ is the induced sub-code $\Lambda_{\boldsymbol{\theta}}$. ∎

Now, $\Lambda_{\boldsymbol{\theta}}$ can be encoded and/or decoded efficiently for any $\boldsymbol{\theta}$ using the constructive formulation of its parity-check matrix in (36) (alternatively, it can also be decoded as a shortened code). When $\boldsymbol{\theta}$ is not known, we can still decode from the base residual code Λ defined by $\boldsymbol{H}$. Note that $\boldsymbol{H}_{\boldsymbol{\theta}}$ can be further simplified, using its reduced row echelon form, row/column removal and column permutation, to produce a systematic encoder for $\Lambda_{\boldsymbol{\theta}}$.

The rows added to $\boldsymbol{H}$ in the bottom block of $\boldsymbol{H}_{\boldsymbol{\theta}}$ in (36) impose additional constraints on the base residual code Λ, to obtain the sub-code $\Lambda_{\boldsymbol{\theta}}$. We now characterize the added redundancy from these constraints when the base residual code is a random binary code.

Proposition 9: Let Θ be an indicator code and let Λ be a random linear base residual code defined by a $r_\Lambda \times n$ parity-check matrix $\boldsymbol{H}$ whose elements are drawn i.i.d Bernoulli with probability ϵ to draw a 0. For any $\boldsymbol{\theta} \in \Theta$ denote $\tilde{r}_{\boldsymbol{\theta}}$ as the expected redundancy of the $\boldsymbol{\theta}$-induced residual code $\Lambda_{\boldsymbol{\theta}}$. Then,

$$\tilde{r}_{\boldsymbol{\theta}} \leq \min\left\{n - w_H(\boldsymbol{\theta}) + r_\Lambda\left(1 - \epsilon^{w_H(\boldsymbol{\theta})}\right), n\right\}. \tag{38}$$

Proof: The redundancy is the number of independent rows in row-wise concatenation of $\boldsymbol{H}$ and $\left(\boldsymbol{I}_n\left([n] \setminus \omega^{(\boldsymbol{\theta})}\right)\right)^\top$. Thus it is upper bounded[1] by the sum of r_Λ and the number of added rows $n - w_H(\boldsymbol{\theta})$, minus the number of rows in $\boldsymbol{H}$ that are individually dependent on the added rows, that is, are each in the span of $\left(\boldsymbol{I}_n\left([n] \setminus \omega^{(\boldsymbol{\theta})}\right)\right)^\top$. Each row drawn to $\boldsymbol{H}$ is in the span of $\left(\boldsymbol{I}_n\left([n] \setminus \omega^{(\boldsymbol{\theta})}\right)\right)^\top$ if and only if all of its 1s fall in indices in $[n] \setminus \omega^{(\boldsymbol{\theta})}$. The probability of this event is $\epsilon^{w_H(\boldsymbol{\theta})}$. Thus the expected number of rows of $\boldsymbol{H}$ that individually depend on the span of the bottom part is $r_\Lambda \epsilon^{w_H(\boldsymbol{\theta})}$, giving the first argument of the min in (38). Since $\boldsymbol{H}$ and $\left(\boldsymbol{I}_n\left([n] \setminus \omega^{(\boldsymbol{\theta})}\right)\right)^\top$ combined cannot have more than n independent rows, we get the second argument of the min. ∎

The bound in (38) is compared in Fig. 5b to the empirical average rank of $\boldsymbol{H}_{\boldsymbol{\theta}}$ with randomly drawn $\boldsymbol{H}$ matrices. We illustrate the bound using two sets of values for ϵ, n and r_Λ. Setting ϵ to a small value leads to an almost piecewise-linear bound, as in Fig. 5a, because $\epsilon^{w_H(\boldsymbol{\theta})}$ approaches 0 very fast; larger ϵ, on the other hand, allows to better observe the nonlinear transition region, as in Fig. 5b. Proposition 9 is a useful tool to estimate the rate of a barrier-error-correcting code given the weight-distribution of the code Θ ($\{a(w)\}_{w=0}^n$) and the sparsity (ϵ) of the parity-check matrix of Λ. For a ternary code, the rate is:

$$\begin{aligned}&\mathrm{rate}(\{a(w)\}_{w=0}^n, r_\Lambda, \epsilon) \approx \\ &\quad \frac{1}{n}\log_3\left(\sum_{w=0}^{n} a(w)\cdot 2^{\max\{w - r_\Lambda(1-\epsilon^w), 0\}}\right).\end{aligned} \tag{39}$$

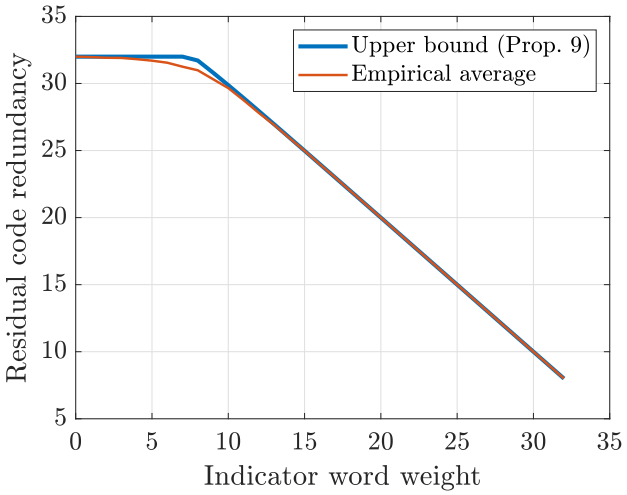
V. Decoding algorithms

Although we commence this section with a description of a bounded-distance decoder and its guarantees, its key content is the formulation of more powerful decoders. Thanks to the property of our code construction that the indicator and residual codes share the same block length, we are able to devise decoding algorithms that decode the constituent codes cooperatively, thereby going beyond the guaranteed correction capability. The first result in the vein is a reduction of *maximum-likelihood (ML)* decoding for $\mathcal{C}$ from Construction 1 (which also applies to the code from [4]) to simpler ML decoding of the constituent codes; then we propose several practical (yet sub-optimal) decoders inspired by this reduction.

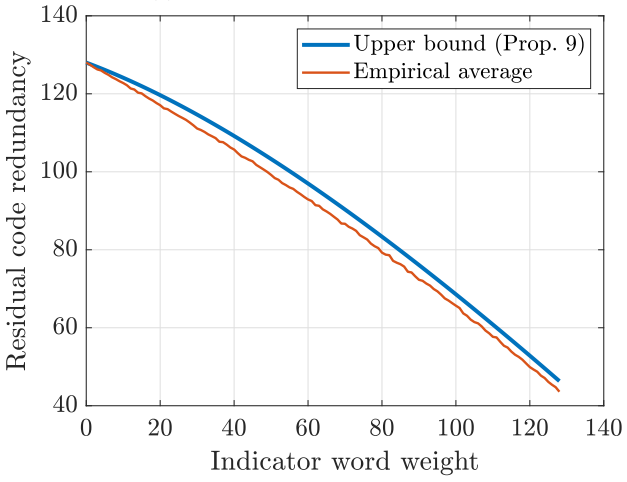
A. Bounded-distance decoding for (t_d, t_u)-barrier errors

The most natural decoding procedure for $\mathcal{C}$ employs a bounded-distance decoder for each of the constituent codes sequentially. The following bounded-distance decoder is a generalization of a similar decoder proposed in [4], and it

[1]The redundancy may be lower if multiple rows of $\boldsymbol{H}$ jointly span a subspace that intersects with the span of $\left(\boldsymbol{I}_n\left([n] \setminus \omega^{(\boldsymbol{\theta})}\right)\right)^\top$.



(a) $n = 32$, $r_\Lambda = 8$ and $\epsilon = 0.66$



(b) $n = 128$, $r_\Lambda = 64$ and $\epsilon = 0.99$

Figure 5: Comparison of the upper bound in Proposition 9 and the empirical average redundancy with randomly drawn base residual codes.

employs two decoding steps: 1) bounded-distance decoding of Θ: find a codeword $\hat{\boldsymbol{\theta}}$ satisfying $d_H(\imath(\boldsymbol{r}), \hat{\boldsymbol{\theta}}) \leq t_d + t_u$; 2) erasure decoding of Λ: find a codeword $\hat{\boldsymbol{\lambda}}$ that agrees with $\hat{\boldsymbol{\theta}}$ where $\hat{\theta}_j = 0$ and with $\boldsymbol{r} - \mathbf{1}$ where $\hat{\theta}_j \neq 0$ and $r_j \neq 0$. Then reconstruct the ternary codeword $\hat{\boldsymbol{c}}$ from $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\lambda}}$ using (35). Declare failure if in any of the Steps 1 and 2 no codeword or more than one codeword were found. The following proposition characterizes the correction capability of the bounded-distance decoder.

Proposition 10: Let $\mathcal{C} = \Theta \otimes \Lambda$ be a code from Construction 1. Given a channel output $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$, the bounded-distance decoder successfully outputs $\boldsymbol{c}$ for any (t_d, t_u)-barrier error $\boldsymbol{e}$.

Proof: Let $\boldsymbol{\theta}$,$\boldsymbol{\lambda}$ be the binary codewords corresponding to $\boldsymbol{c}$ in Construction 1. Since $\boldsymbol{e}$ is a (t_d, t_u)-barrier error, we have $d_H(\imath(\boldsymbol{r}), \boldsymbol{\theta}) \leq t_d + t_u$, so $\boldsymbol{\theta}$ will be found in step 1. $\boldsymbol{\theta}$ is also unique from the properties stated in Definition 12. From item 2 in Construction 1, $\lambda_j = c_j - 1$ if $c_j \neq 0$, and $\lambda_j = 0$ if $c_j = 0$. These are equivalent to $\lambda_j = c_j - 1$ if $\theta_j \neq 0$ and $\lambda_j = 0$ if $\theta_j = 0$, because from item 1 in the construction we have $\theta_j = 0$ if and only if $c_j = 0$. From the properties of barrier errors we have that if $\theta_j \neq 0$ (equiv. $c_j \neq 0$) and $r_j \neq 0$, then $r_j = c_j$, and we proved that $\boldsymbol{\lambda}$ meets the conditions in step 2 of the decoder. For uniqueness, suppose by contradiction that there is another codeword $\boldsymbol{\lambda}'$ satisfying these conditions. Then $\boldsymbol{\lambda}'$ can differ from $\boldsymbol{\lambda}$ only on coordinates where $\theta_j \neq 0$ (equiv. $c_j \neq 0$) and $r_j = 0$. But from the properties of the error there are at most t_d such coordinates, and from the properties stated in Definition 13 this is not possible. ■

B. Decomposed MLD: MLD using simpler constituent codes

In [4], an ML decoder (MLD) – for the special case $p = q/2$, $Q = 3$ – is defined through a distance metric on the ternary alphabet $\mathbb{Z}_3$. For more efficient decoding, we show in this subsection a reduction of ML decoding to simpler ML decoders of the constituent codes Θ and Λ. The advantage of this reduction is that the constituent codes are traditional Hamming-metric codes over lower-order alphabets (Θ is a binary code, and when $Q = 3$ so is Λ).

Let $\boldsymbol{r} = (r_1, \ldots, r_n) \in \mathbb{Z}_Q^n$ be the word output from the channel $W_Q(p, q)$. An MLD for a code $\mathcal{C} = \Theta \otimes \Lambda$ needs to find a pair of codewords $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{\lambda} \in \Lambda_{\boldsymbol{\theta}} \subseteq \Lambda$ that jointly maximize the likelihood of observing $\boldsymbol{r}$. We now decompose this task using individual decoders for Θ, Λ, and a rule for combining the individual decoder outputs.

1) MLD for the code Θ: We define the following decoder for Θ, invoked on the input $\imath(\boldsymbol{r})$ where $\imath(\cdot)$ is the indicator mapping defined in Definition 4, which maps symbols from the channel alphabet to the binary alphabet of Θ.

Definition 18: The **ML indicator decoder** for Θ outputs the codeword

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta} \in \Theta} \{\mu_1 \imath(\boldsymbol{r})\, \boldsymbol{\theta}^T - \mu_2 w_H(\boldsymbol{\theta})\}, \tag{40}$$

where $\mu_1 \triangleq \log\left(\frac{(Q-1)(1-p)(1-q)}{pq}\right)$ and $\mu_2 \triangleq \log\left(\frac{1-q}{p}\right)$. It will be shown later (Proposition 11) that $\hat{\boldsymbol{\theta}}$ in (40) maximizes the indicator vector's likelihood function $\Pr\{\imath(\boldsymbol{r}) | \boldsymbol{\theta}\}$, hence the ML qualification in Definition 18.

2) MLD for the $\Lambda_{\boldsymbol{\theta}}$ code: We first define a mapping from the length-n channel output to the decoder input.

Definition 19: Given $\boldsymbol{\theta} \in \mathbb{Z}_2^n$, the **residual mapping** maps a vector $\boldsymbol{r} \in \mathbb{Z}_Q^n$ to a vector $\boldsymbol{\varrho}^{(\boldsymbol{\theta})}(\boldsymbol{r}) = (\varrho_1, \ldots, \varrho_n)$ such that

$$\varrho_j = \begin{cases} r_j - 1, & \theta_j = 1, r_j \neq 0 \\ ?, & \theta_j = 1, r_j = 0 \\ 0, & \text{otherwise} \end{cases} \tag{41}$$

where ? represents the erasure symbol. Now we define the following decoder for $\Lambda_{\boldsymbol{\theta}}$, invoked on the input $\boldsymbol{\varrho}^{(\boldsymbol{\theta})}(\boldsymbol{r})$.

Definition 20: Given $\boldsymbol{\theta} \in \{0, 1\}^n$, the **residual ML decoder** for $\Lambda_{\boldsymbol{\theta}}$ finds and outputs a codeword $\hat{\boldsymbol{\lambda}}$ such that $\hat{\lambda}_i = \varrho_i^{(\boldsymbol{\theta})}(\boldsymbol{r})$ for every i with $\varrho_i^{(\boldsymbol{\theta})}(\boldsymbol{r}) \neq ?$ (and any value on erased coordinates). The decoder outputs "reject" if no such codeword exists.

3) Combining the individual MLDs: Let $\boldsymbol{r}$ be the channel output. We are now ready to define the combined ML decoder in Algorithm 1.

Algorithm 1 MLD for $\mathcal{C} = \Theta \otimes \Lambda$:

Input: $\boldsymbol{r} \in \mathbb{Z}_Q^n$ - channel output
Output: $\hat{\boldsymbol{c}} \in \mathbb{Z}_Q^n$ - decoded codeword
Initialize: $\Theta' \leftarrow \Theta$
while not returned **do**
    set $\hat{\boldsymbol{\theta}}$ to the output of indicator MLD with input $\imath(\boldsymbol{r})$ and code Θ'
    invoke residual MLD with input $\varrho^{(\hat{\boldsymbol{\theta}})}(\boldsymbol{r})$ and code $\Lambda_{\hat{\boldsymbol{\theta}}}$
    if "reject" **then**
        $\Theta' \leftarrow \Theta' \setminus \hat{\boldsymbol{\theta}}$
    else
        return $\hat{\boldsymbol{c}} = \Psi(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\lambda}})$
    end if
end while

Algorithm 1.1 CLD for $\mathcal{C} = \Theta \otimes \Lambda$:

Input: $\boldsymbol{r} \in \mathbb{Z}_Q^n$ - channel output
Output: $\hat{\boldsymbol{c}} \in \mathbb{Z}_Q^n$ - decoded codeword
Initialize: $\Theta' \leftarrow \{\hat{\boldsymbol{\theta}}_l\}_{l=1}^L$: codewords output by an indicator list-decoder (ordered in descending likelihoods)
for $l = 1, \ldots, L$ **do**
    invoke a residual decoder with input $\varrho^{(\hat{\boldsymbol{\theta}}_l)}(\boldsymbol{r})$ and code $\Lambda_{\hat{\boldsymbol{\theta}}_l}$
    if "reject" **then**
        $\Theta' \leftarrow \Theta' \setminus \hat{\boldsymbol{\theta}}_l$
    else if "fail" **then**
        return *decoding failure*
    else
        return $\hat{\boldsymbol{c}} = \Psi(\hat{\boldsymbol{\theta}}_l, \hat{\boldsymbol{\lambda}})$
    end if
end for
return *decoding failure*

4) Proving the ML property of Algorithm 1: Given a channel output $\boldsymbol{r}$ and a candidate codeword $\boldsymbol{c}$, we can partition the n coordinates to 5 disjoint sets: $\mathcal{S}_0$ where $r_i = c_i = 0$, $\mathcal{S}_1$ where $r_i = c_i \neq 0$, $\mathcal{S}_2$ where $r_i = 0, c_i \neq 0$, $\mathcal{S}_3$ where $r_i \neq 0, c_i = 0$, $\mathcal{S}_4$ where $r_i \neq 0, c_i \neq 0, r_i \neq c_i$. Thus the ML codeword is the one that maximizes $\sum_{s=0}^{3} \ell_s |\mathcal{S}_s|$, subject to $|\mathcal{S}_4| = 0$, where $\ell_0 = \log(1-q)$, $\ell_1 = \log(1-p)$, $\ell_2 = \log(p)$, and $\ell_3 = \log(q/(Q-1))$. This follows from the channel definition and taking the log of the likelihood function $\Pr\{\boldsymbol{r}|\boldsymbol{c}\}$.

Proposition 11: For any $\boldsymbol{r}$, the output of Algorithm 1 is an ML codeword of $\mathcal{C} = \Theta \otimes \Lambda$.

Proof: We first prove that the ML indicator decoder maximizes $\sum_{s=0}^{3} \ell_s |\mathcal{S}_s| + \ell_1 |\mathcal{S}_4|$. This can be seen by substituting in this sum the identities $|\mathcal{S}_1| + |\mathcal{S}_4| = \imath(\boldsymbol{r})\,\boldsymbol{\theta}^T$, $|\mathcal{S}_3| = w_H(\imath(\boldsymbol{r})) - \imath(\boldsymbol{r})\,\boldsymbol{\theta}^T$, $|\mathcal{S}_2| = w_H(\boldsymbol{\theta}) - \imath(\boldsymbol{r})\,\boldsymbol{\theta}^T$, and $|\mathcal{S}_0| = n - w_H(\imath(\boldsymbol{r})) - w_H(\boldsymbol{\theta}) + \imath(\boldsymbol{r})\,\boldsymbol{\theta}^T$. By expressing all set sizes as functions of $w_H(\boldsymbol{\theta})$, $\imath(\boldsymbol{r})\,\boldsymbol{\theta}^T$, and ignoring all terms that do not depend on $\boldsymbol{\theta}$, we get (40).

To show that $\Psi(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\lambda}})$ output by Algorithm 1 is an ML codeword, we first observe that for any $\boldsymbol{c} = \Psi(\boldsymbol{\theta}, \boldsymbol{\lambda})$, $|\mathcal{S}_4| = 0$ if and only if $\boldsymbol{\lambda}$ satisfies the condition that $\lambda_i = \varrho_i^{(\boldsymbol{\theta})}(\boldsymbol{r})$ for every i with $\varrho_i^{(\boldsymbol{\theta})}(\boldsymbol{r}) \neq ?$. This is the condition enforced by the residual MLD in Definition 20, so in particular $\Psi(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\lambda}})$ has $|\mathcal{S}_4| = 0$. That means that maximizing $\sum_{s=0}^{3} \ell_s |\mathcal{S}_s| + \ell_1 |\mathcal{S}_4|$ by the indicator MLD is equivalent to maximizing $\sum_{s=0}^{3} \ell_s |\mathcal{S}_s|$, subject to $|\mathcal{S}_4| = 0$, needed for Q-ary MLD. ∎

The joint ML decoder of Algorithm 1 improves over the bounded-distance decoder of Section V-A that invokes each of the indicator/residual decoder only once. Beyond the unique-decoding capabilities of Θ, the $\boldsymbol{\theta}$ of the ML codeword in $\mathcal{C}$ given $\boldsymbol{r}$ is not necessarily the ML codeword of Θ given $\imath(\boldsymbol{r})$, so having an ML indicator decoder is not sufficient for ML decoding of $\mathcal{C}$. The strength of the joint ML decoder is that it allows the residual decoder to reject wrong Θ codewords even if they have high indicator likelihoods. The MLD algorithm derived here for Construction 1 can be easily adapted to decode also the code construction in [4], as described in Section III.B of [12].

C. Cooperative List Decoding (CLD)

Algorithm 1 simplifies ML decoding compared to the Q-ary ML decoder, by reduction to decoding of binary codes in the Hamming metric and erasure decoding. However, ML decoding of binary codes is still in general a computationally hard problem. To mitigate this hardness, we propose a simplification of Algorithm 1 using *list decoding*, which is a more tractable computational task than ML decoding. We define two variants of this proposition.

1) Cooperative list decoder (CLD): A list decoder for the code Θ outputs a list of likely codewords $\{\hat{\boldsymbol{\theta}}_l\}_{l=1}^L$, where L is the list size. Then the CLD for the code $\mathcal{C} = \Theta \otimes \Lambda$ is obtained by running Algorithm 1.1 which initializes $\Theta' \leftarrow \{\hat{\boldsymbol{\theta}}_l\}_{l=1}^L$ instead of $\Theta' \leftarrow \Theta$ as in Algorithm 1. The residual decoder of CLD is not assumed to be MLD, so a "fail" option is added in case it did not reject but could not find a $\hat{\boldsymbol{\lambda}}$. Other than these changes, the algorithm similarly finds a pair of $\boldsymbol{\theta}, \boldsymbol{\lambda}$ that jointly have the maximum likelihood among all such pairs with codewords in $\{\hat{\boldsymbol{\theta}}_l\}_{l=1}^L$.

2) Persistent cooperative list decoder (PCLD): Identical to CLD, but instead of returning "decoding failure" when the residual decoder fails, it continues to the next codeword in the list (this amounts to merging the 'else if' of Algorithm 1.1 into the 'if' statement, which will now be: if "reject" or "fail").

It is clear that with CLD and/or PCLD, similarly to the MLD of Algorithm 1, a decoding success can occur even if the correct $\boldsymbol{\theta}$ is not the ML codeword given the channel output's indicator word. Thanks to its persistence, PCLD may succeed in finding the correct codeword further down the list, while CLD stops the search upon residual-decoding failure. Note that neither CLD nor PCLD are equivalent to MLD, because the list decoder of Θ in general does not guarantee finding the $\boldsymbol{\theta}$ of the ML codeword. The special case of CLD with $L = 1$ is the bounded-distance decoder described earlier in the section that is based on unique decoding of Θ. Our generalization to $L > 1$ is more powerful not only thanks to more opportunities to find the most likely indicator codeword, but also in its ability to reject wrong indicator codewords using information from the residual decoder. (Indeed in Section VI we show that PCLD outperforms a decoder that uses list decoding for Θ without cooperating with the decoder of Λ.)

3) Success conditions for PCLD: The PCLD succeeds in all instances where the following conditions are met: 1) The correct codeword $\boldsymbol{\theta} \in \Theta$ is in the list $\{\hat{\boldsymbol{\theta}}_l\}_{l=1}^{L}$, 2) all other codewords in the list that have higher or equal likelihoods than $\boldsymbol{\theta}$ given $\imath(\boldsymbol{r})$ reject or fail by the residual decoder, and 3) residual decoding of $\varrho^{(\boldsymbol{\theta})}(\boldsymbol{r})$ does not return decoding failure.

D. Deeper cooperation (DC) joint decoder

To decode codes from Construction 1 more effectively, we now propose an algorithm that decodes the indicator and residual codes jointly with *deeper cooperation* between the decoders of Θ and Λ. In previous decoders, the cooperation was one sided $\Theta \to \Lambda$: the decoder of Θ provided candidate codewords for the decoder of Λ, and the only information passing in the other direction was the "reject"/"fail" indications. In the bounded-distance decoder from Section V-A, a single decoded codeword from Θ is used to decode Λ, with no option to succeed if this codeword is not correct. The PCLD decoder (Section V-C) employs a *list decoder* for Θ, and the decoder then selects from the list the most likely codeword found to be *consistent* with Λ, that is, its residual mapping (41) satisfies all of Λ's parity-check equations. The cost of this one-sided operation in the case of bounded-distance decoding is poor performance beyond Θ's guaranteed-decoding radius, and in the case of PCLD the cost is high computational complexity of list decoding. The main idea of the proposed deeper-cooperation joint decoder is to use information from Λ's decoder to improve the decoding outcome of Θ's decoder. In particular, this information is chosen to be the *inconsistency score* of each index in $[n]$, which is the number of violated parity-check equations of $\boldsymbol{H}$ that the index appears in.

Example 4: We exemplify inconsistency in the residual code induced by an incorrect indicator candidate codeword. Consider a residual code Λ with parity-check equations

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Assume the codeword $\boldsymbol{c} = (0,1,2,0,1,2,0,1,2)$ was transmitted and the channel output is $\boldsymbol{r} = (2,1,2,0,1,0,0,1,2)$ (i.e., upward error in the first symbol and downward error in the sixth symbol). Assuming the (wrong) indicator candidate codeword is $\bar{\boldsymbol{\theta}} = (1,0,1,0,0,1,0,0,1)$, we get $\boldsymbol{\varrho}^{(\bar{\boldsymbol{\theta}})}(\boldsymbol{r}) = (1,0,1,0,0,?,0,0,1)$. The second parity-check equation is violated by $\boldsymbol{\varrho}^{(\bar{\boldsymbol{\theta}})}(\boldsymbol{r})$, hence the first, third and ninth positions have each an inconsistency score of 1, indicating they have lower reliability than other positions. The first parity-check equation has unknown consistency: the erasure in the sixth position implies that the equation may not be violated.

The deeper cooperation (DC) decoding algorithm is given formally in Algorithms 2 and 3 (the latter of which is specified here for $Q = 3$). Given a channel output $\boldsymbol{r}$, we first decode Θ to get an indicator candidate codeword $\bar{\boldsymbol{\theta}}$. We then use Algorithm 3 to order the indices of $[n]$ based on their inconsistency score with respect to $\bar{\boldsymbol{\theta}}$ and Λ's parity-check matrix $\boldsymbol{H}$. For an iteration-count limit of $M = 2^m$, we take a set S of the m indices with the highest inconsistency scores (sorted in non-increasing order), and use this set to generate *bit-flipping subsets*. Each subset is used to obtain a codeword $\hat{\boldsymbol{\theta}}$ by the decoder of Θ (if one found); this codeword is then used to generate $\varrho^{(\hat{\boldsymbol{\theta}})}(\boldsymbol{r})$ which is decoded by Λ, and the combined Q-ary codeword is returned if consistent. This iteration starts with the empty subset (equivalent to decoding Λ with $\bar{\boldsymbol{\theta}}$), and proceeds in increasing subset sizes; within each size, the subsets are ordered by the sum of their ranks in S. This decoding approach mimics the *Chase decoding method [18]*, only that the index ordering is done based on consistency information from Λ and not from the channel reliabilities. Note that the bounded-distance decoder above is a special case of Algorithm 2 with $M = 1$ (and empty set S).

Algorithm 2 Deeper-cooperation (DC) joint decoding of $\mathcal{C} = \Theta \otimes \Lambda$

Input: $\boldsymbol{r} \in \mathbb{Z}_Q^n$: channel output, $M = 2^m$: max # iterations
Output: $\hat{\boldsymbol{c}} \in \mathbb{Z}_Q^n$: decoded codeword
$\bar{\boldsymbol{\theta}} \leftarrow$ decode $\imath(\boldsymbol{r})$ over Θ
$\pi \leftarrow$ Inconsistency-ordering$(\boldsymbol{r}, \bar{\boldsymbol{\theta}}, \boldsymbol{H})$
$S \leftarrow \pi(1:m)$ ▷ highest-inconsistency indices
for $\sigma = \{\}, \ldots, S$ **do** ▷ order subsets $\sigma \subseteq S$ by size, higher inconsistencies first in each size
    set $\boldsymbol{f}$ to be a vector with ones in the indices of σ, and zeros elsewhere
    $\hat{\boldsymbol{\theta}} \leftarrow$ decode $\imath(\boldsymbol{r}) \oplus \boldsymbol{f}$ over Θ
    $\hat{\boldsymbol{\lambda}} \leftarrow$ decode $\varrho^{(\hat{\boldsymbol{\theta}})}(\boldsymbol{r})$ with residual decoder of Λ
    if residual decoder failed or rejected **then**
        continue
    else ▷ $\hat{\boldsymbol{\theta}}$ is consistent with Λ
        return $\hat{\boldsymbol{c}} = \Psi(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\lambda}})$
    end if
end for
return *decoding failure*

Algorithm 3 Inconsistency-ordering of $[n]$

Input: $\boldsymbol{r} \in \mathbb{Z}_3^n$: channel output, $\bar{\boldsymbol{\theta}} \in \mathbb{Z}_2^n$: candidate indicator codeword, $\boldsymbol{H} \in \mathbb{F}_2^{\rho \times n}$: residual-code parity-check matrix
Output: π: sorting of $[n]$ in non-increasing inconsistency
Remove residual parity-check equations with unknown consistency due to erasures

$$\mathcal{E} \leftarrow \{i \in [\rho] : \text{exists } j \in [n], \varrho_j = ? \wedge H_{i,j} = 1\}$$

Extract inconsistent residual parity-check equations

$$\mathcal{S} \leftarrow \left\{ i \in [\rho] \setminus \mathcal{E} : \boldsymbol{H}_{i,:} \boldsymbol{\varrho}^{(\bar{\boldsymbol{\theta}})}(\boldsymbol{r})^\top \neq 0 \right\}$$

Calculate "inconsistency score" for each index $j \in [n]$ as

$$w_j = \sum_{i \in \mathcal{S}} H_{i,j} \text{ where the sum is over } \mathbb{Z} \text{ (not } \mathbb{F}_2)$$

$\pi \leftarrow$ sort $[n]$ in non-increasing w_j
return π

E. Analysis of residual decoder with random linear codes

The fact that the residual decoder of Definition 20 (used in all decoding algorithms) is an erasure decoder allows analyzing its outcomes when the residual code is a random linear code (RLC). In particular, we calculate the probabilities of each of its outcomes "fail" and "reject". These expressions are useful toward simplifying some simulations in the next section, as well as for code design. Throughout this sub-section, we focus on a binary base residual code (corresponding to $Q = 3$), but extension to non-binary codes is possible. The residual decoder is invoked on an input vector $\boldsymbol{\varrho}^{(\boldsymbol{\theta})}(\boldsymbol{r})$, which depends on a candidate indicator codeword $\boldsymbol{\theta}$; the codeword $\boldsymbol{\theta}$ also determines the code $\Lambda_{\boldsymbol{\theta}}$ in which the decoder searches for output codewords.

Definition 21: Define $\Lambda_{\boldsymbol{\theta}}$ as a **random linear code (RLC)** defined by a parity-check matrix $\boldsymbol{H}_{\boldsymbol{\theta}}$ with dimensions $r_{\boldsymbol{\theta}} \times n$, whose each element is an independent and identically distributed (i.i.d.) Bernoulli random variable with parameter $1/2$.

The erasure decoder "fails" when there is more than one codeword that agrees with the non-erased positions of the input. This gives the following well-known result [19].

Proposition 12: When $\boldsymbol{\varrho}^{(\boldsymbol{\theta})}(\boldsymbol{r})$ has $1 \leq e \leq r_{\boldsymbol{\theta}}$ erasures, the residual decoder over RLC returns "fail" with probability

$$\Pr_{\text{fail}}(r_{\boldsymbol{\theta}}, e) = 1 - \prod_{i=0}^{e-1} \left(1 - 2^{i - r_{\boldsymbol{\theta}}}\right). \tag{42}$$

Proof: Immediate from evaluating the probability that the erasure-positions' sub-matrix of $\boldsymbol{H}_{\boldsymbol{\theta}}$ is full rank, and taking the complement. See e.g. Eq. (3.2) in [19]. ∎

When $\boldsymbol{\theta}$ is the correct codeword the residual decoder "fails" with the above probability, and never returns "reject", since by Construction 1 there always exists a $\boldsymbol{\lambda}$ that meets the requirement of Definition 20. When $\boldsymbol{\theta}$ is not the correct codeword, the decoder "fails" with the same $\Pr_{\text{fail}}$ above, but now can also "reject". Since a wrong $\boldsymbol{\theta}$ may induce *errors* in $\boldsymbol{\varrho}^{(\boldsymbol{\theta})}(\boldsymbol{r})$ with respect to Λ (and not just erasures), we assume in this case that each non-erased position in the input is drawn uniformly from $\{0, 1\}$, and get the following:

Proposition 13: When $\boldsymbol{\varrho} \in \{0, 1, ?\}^n$ is a word drawn uniformly from $\{0, 1\}^n$ and then added with $1 \leq e \leq r_{\boldsymbol{\theta}}$ erasures, the residual decoder over RLC returns "reject" with probability

$$\Pr_{\text{reject}}(r_{\boldsymbol{\theta}}, e) = 1 - 2^{-(r_{\boldsymbol{\theta}} - e)}. \tag{43}$$

Proof: For $\boldsymbol{\varrho}$ to *not* be rejected, the product of its non-erased sub-vector with the corresponding sub-matrix of $\boldsymbol{H}_{\boldsymbol{\theta}}$ must be in the span of the remaining e columns of $\boldsymbol{H}_{\boldsymbol{\theta}}$. The probability of this event amounts to the ratio between the size of this span 2^e to the size of the entire space $2^{r_{\boldsymbol{\theta}}}$. $\Pr_{\text{reject}}(\cdot)$ in (43) is the complement of this ratio. ∎

Recall from Section V-C3 that two events related to the residual decoder cause failure of the PCLD: 1) neither rejecting nor failing on a wrong codeword $\hat{\boldsymbol{\theta}}$, and 2) decoding failure of the correct codeword $\boldsymbol{\theta}$. Under the RLC assumption, we can use Proposition 12 to calculate the probability of the second event, and Propositions 12 and 13 to calculate the probability of the first event.

VI. Performance evaluation

We evaluate the presented decoders by simulating their operation on two interesting barrier channel regimes: balanced errors (i.e., $t_u \approx t_d$ or $q \approx 2p$), and extremely unbalanced errors (i.e., $t_u \gg t_d$ or $q \gg p$). Specifically, we examine the decoding performance of different codes and decoders beyond the codes' guaranteed correction capability.

To this end, we simulate the ternary ($Q = 3$) barrier channel and focus on binary Reed–Muller (RM)[2] indicator codes due to their rich structure and extensively studied unique and list decoders. For residual codes, we use widely adopted binary linear codes such as LDPC, and in some experiments random linear codes (RLC), whose analysis in Section V-E allows obtaining approximate results without invoking the erasure decoder (thus speeding up the simulations).

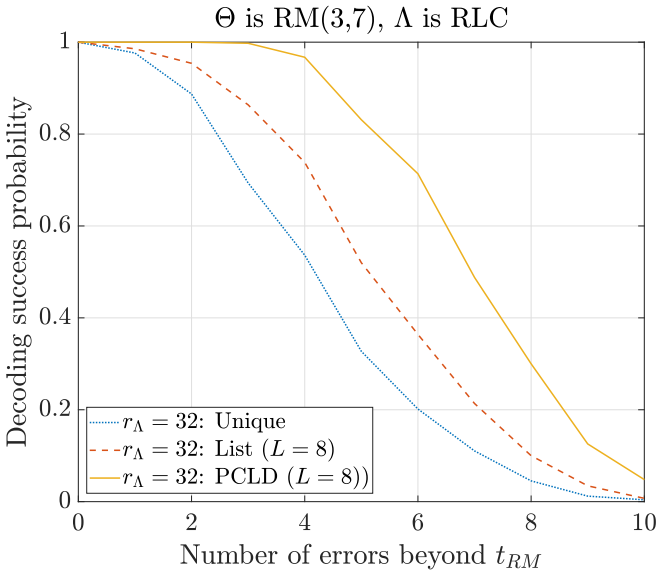
A. Comparing PCLD to unique and list decoders

We first want to compare the performance of *PCLD* (Section V-C2) to prior decoding algorithms. For the comparison we take two algorithms that are improved versions of the bounded-distance decoder from Section V-A. The first we call *unique decoder*, where the indicator decoder may find a (single) codeword beyond the code's guaranteed correction radius. The second we call *list decoder*, where the indicator decoder returns a list of codewords and chooses from it a codeword with the highest likelihood given $\imath(\boldsymbol{r})$. In this sub-section the indicator code is a RM$(3, 7)$ RM code, and the residual code is a RLC. The RLC is not implemented explicitly, but instead its failure/reject probabilities are calculated using Propositions 12, 13 in every decoding instance. The e value substituted in (42),(43) is the number of erasures in $\boldsymbol{\varrho}^{(\hat{\boldsymbol{\theta}})}(\boldsymbol{r})$, where $\hat{\boldsymbol{\theta}}$ is a codeword found by the RM indicator decoder. For $r_{\boldsymbol{\theta}}$ we substitute the bound on $\tilde{r}_{\boldsymbol{\theta}}$ from Proposition 9 with $r_\Lambda = 32$.

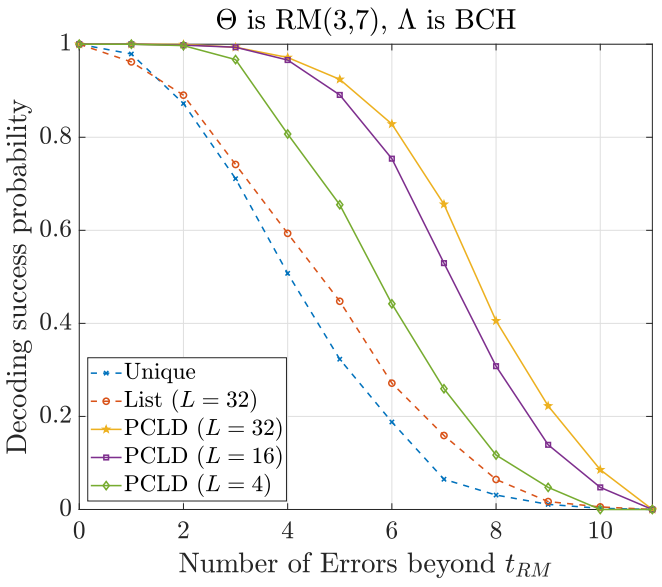
Each simulation instance draws a random codeword $\boldsymbol{c} = \boldsymbol{\theta} \cdot (\mathbf{1} + \boldsymbol{\lambda})$ and passes it through a ternary barrier channel with $q = 2p$ to obtain the received vector $\boldsymbol{r}$. In all three decoders $\imath(\boldsymbol{r})$ is first decoded by a RM decoder: the soft Dumer algorithm [21] in "unique", and the soft Dumer list decoder ($L = 8$) [22] in "list" and in "PCLD". In "unique", the success probability of an instance is $1 - \Pr_{\text{fail}}(r_{\boldsymbol{\theta}}, e)$ if the correct $\boldsymbol{\theta}$ is the codeword returned by the RM decoder, and 0 if not. In "list", the success probability of an instance is $1 - \Pr_{\text{fail}}(r_{\boldsymbol{\theta}}, e)$ if the correct $\boldsymbol{\theta}$ is the highest-likelihood codeword in the list returned by the RM decoder, and 0 if not. In "PCLD", the success probability of an instance is the same as list if the correct $\boldsymbol{\theta}$ is the highest-likelihood codeword in the list, but is in general not 0 even if there are codewords in the list with higher likelihoods. In such cases, $1 - \Pr_{\text{fail}}(r_{\boldsymbol{\theta}}, e)$ is multiplied by a product of probabilities $(\Pr_{\text{reject}} + \Pr_{\text{fail}})$ corresponding to each codeword in the list with higher likelihood than the correct $\boldsymbol{\theta}$ (see Section V-C3). Having run many such instances and averaging the success probabilities, we plot the results in Fig. 6a, where the runs

[2]Implementations of Reed–Muller encoding and decoding procedures used in this section are made available in [20].

are partitioned by the number of barrier errors the instance suffered beyond the guaranteed correction capability of the RM code ($t_{RM} = 7$).



(a) RLC residual code with $r_\Lambda = 32$



(b) BCH residual code for correcting t_{RM} erasures

Figure 6: Decoding-success probability with RM$(3, 7)$ indicator code and different residual codes. The PCLD is compared to unique and list decoders.

It can be seen in Fig. 6a that the PCLD has higher success probabilities than the prior unique and list decoders. In particular, only with PCLD the success probability stays close to 1 even when the number of errors grows beyond the guaranteed correction capability: up to 3 more errors. The gap of PCLD from list decoder attests to the fact that its power stems not from a more powerful indicator decoder, but from the cooperation with the residual decoder.

To complete the performance evaluation of PCLD, we also performed a full simulation: implementing both the indicator and residual decoders. Toward that, we replaced the RLC residual code by a family of BCH codes (to guarantee correcting t_{RM} erasures for every weight of the indicator codeword). It can be seen in Fig. 6b that the PCLD has higher success probabilities compared to the unique decoder and list ($L = 32$) decoder. Moreover, the decoding success probability consistently grows with the indicator-decoder list size, though even a small list of $L = 4$ indicator candidates already provides significant performance gains.

B. Deeper-cooperation joint decoder

In this sub-section we evaluate the performance of the deeper cooperation (DC) joint decoder (from Section V-D), relative to the prior decoders and PCLD considered in the previous sub-section. Here we simulate a full coding scheme implementing both the indicator and residual decoders (and encoders). The performance metric of study is the block-error rate (BLER), i.e., the fraction of code blocks that were output from the decoder with at least one symbol error. The channel is the ternary ($Q = 3$) barrier channel $W_3(p, q)$. We use binary RM$(2, 6)$ and RM$(2, 7)$ codes as indicator codes[3], and decode them using the recursive projection-aggregation (RPA) decoder [23]. The DC decoder uses the non-list version of the RPA algorithm (Line 8 of Alg. 2), while the PCLD and (sequential) List decoders use the list version of RPA. We chose the RPA algorithm thanks to its state-of-the-art performance among unique (non-list) RM decoders. The base-residual code we use in the simulation is a rate $1/2$ binary low-density parity-check (LDPC) code, with length matched to the indicator code (i.e., $n = 64$ in case of RM$(2, 6)$ and $n = 128$ in case of RM$(2, 7)$). The residual codes are decoded by a standard peeling erasure decoder [24], which stops and rejects whenever it encounters a parity check that is not consistent. This configuration resulted in a combined average code rate of ~ 0.4 symbols per channel use for $n = 64$ and ~ 0.3 symbols per channel use for $n = 128$. The channel is realized for asymmetric scenarios, where p is fixed to either 0.001 or 0.05, while q takes values in the range between 0 and 1.

Fig. 7 plots the decoding block-error probabilities (BLER) as a function of the upward channel transition probability q (the downward channel transition probability p in each figure is fixed). It clearly shows the advantage of both of the proposed decoders PCLD and DC over the prior decoders unique and list. The DC decoder almost matches the performance of PCLD for RM$(2, 6)$, even though the latter uses a computationally-intensive list decoder with list-size $L = 32$. Although for the DC decoder we set $M = L = 32$, the multiple serial invocations of the residual and indicator decoders do not induce complexity similar to a list decoder. This translates to significant speed up in implementation compared to PCLD (typically at least $\times 10$ faster in our experiments).

VII. Conclusion

In this work, we study the barrier channel – a non-binary channel in which all errors are either to or from a specific

[3]The implementations of Reed–Muller encoding and decoding procedures used in this section are made available in [20].

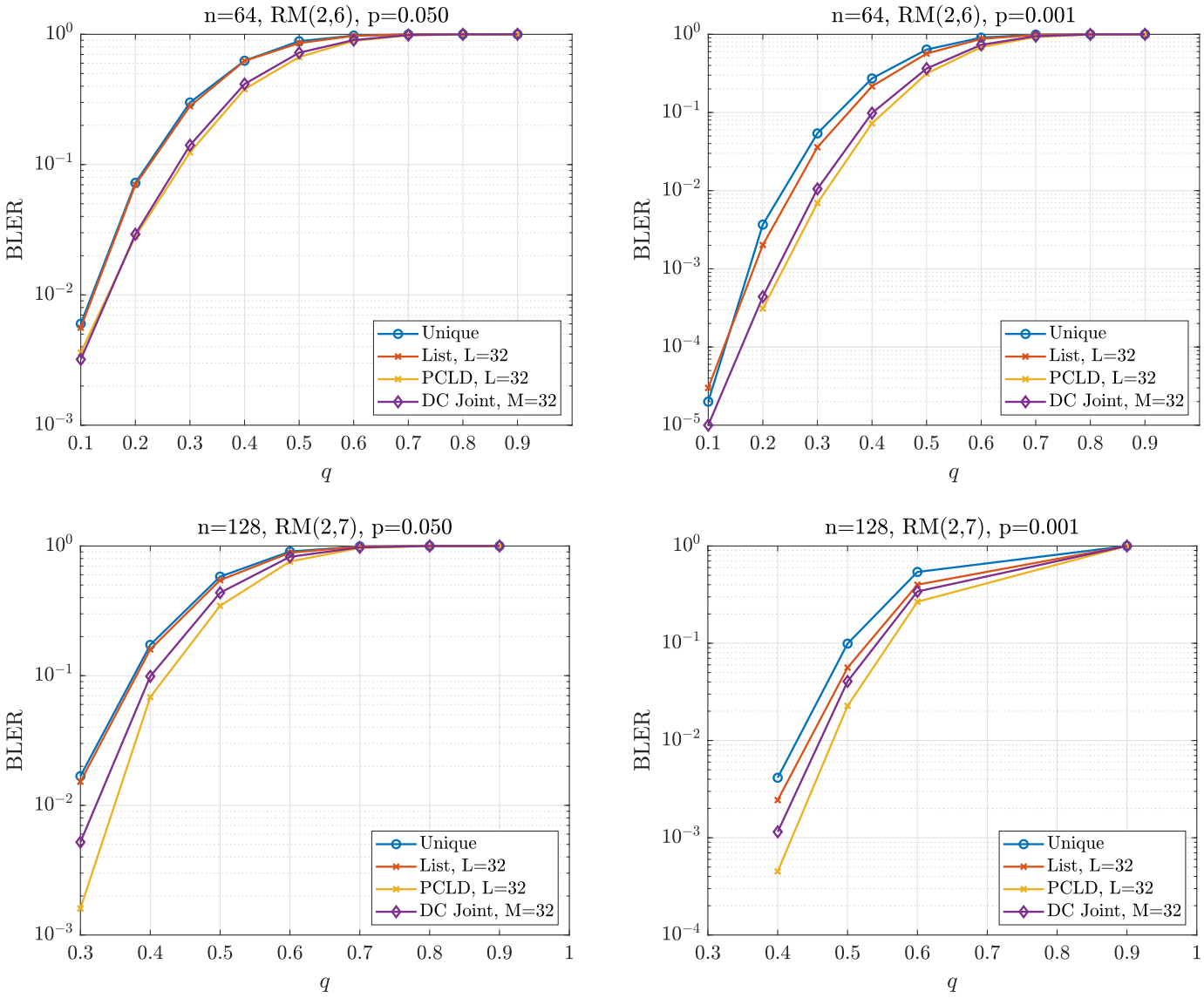


Figure 7: Block error rate (BLER) of RM$(2,6)$ (upper row) and RM$(2,7)$ (lower row) indicator codes and rate $\sim$0.5 irregular LDPC base residual codes simulated over $W_3(p,q)$: A comparison of four decoders.

alphabet symbol called the barrier symbol. Specifically, we focus on a two-parameter (p,q) generalization of this channel, to model different transition probabilities into and out of the barrier symbol. The first part of the paper includes a derivation of the channel capacity, which motivates the two-parameter model by showing large capacity gaps between channels with the same $p+q$.

The second part of the paper studies the corresponding combinatorial error model with the two parameters t_d, t_u. This part includes correction-capability conditions and code-size bounds, and a general code construction building on a pair of constituent codes with the same length.

The last part proposes decoding algorithms that decode the two constituent codes jointly, and as a result can correct beyond the correction guarantees of the codes. Finally, the performance improvements offered by the proposed decoders are demonstrated in simulations.

We point out several interesting directions for further investigation.

- A central question that remains open is the formulation of a code construction induced by the *necessary and sufficient* correction condition of Proposition 2.
- A related problem is to find code constructions that meet the sufficient condition of Proposition 1 more tightly – the proposed Construction 1 satisfies a stronger condition: all pairs of codewords with $d_H(\boldsymbol{c},\boldsymbol{c}') \leq 2(t_u+t_d)$ (and $d_{\bar{i}}(\boldsymbol{c},\boldsymbol{c}') \geq t_d+1$) have $d_i(\boldsymbol{c},\boldsymbol{c}')=0$.
- Another interesting question involves tightening the bounds on maximal code size, using a tighter analysis of ball sizes, or other techniques.
- On the decoding front, it will be interesting to further improve the decoding performance using other forms of cooperation between the two constituent codes.
- Extending the study of barrier channels to the paradigms of zero-error [25] and adversarial channels [26].

VIII. Acknowledgment

This work was supported in part by the US-Israel Binational Science Foundation under grant number 2023627. The authors thank the associate editor and anonymous reviewers for comments and suggestions that improved the paper.

References

[1] Y. Telepinsky, V. Mor, M. Schultz, Y.-M. Hung, A. D. Kent, and L. Klein, "Towards a six-state magnetic memory element," *Applied Physics Letters*, vol. 108, no. 18, p. 182401, 2016.

[2] X. Tang and W. Tang, "A 151nw second-order ternary delta modulator for ECG slope variation measurement with baseline wandering resilience," in *2020 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2020, pp. 1–4.

[3] L. Luo, Z. Dong, X. Hu, L. Wang, and S. Duan, "MTL: Memristor ternary logic design," *International Journal of Bifurcation and Chaos*, vol. 30, no. 15, pp. 205–222, 2020.

[4] N. Bitouzé, A. G. i Amat, and E. Rosnes, "Error correcting coding for a nonsymmetric ternary channel," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5715–5729, 2010.

[5] D. V. Efanov, "Ternary parity codes: Features," in *2019 IEEE East-West Design & Test Symposium (EWDTS)*. IEEE, 2019, pp. 1–5.

[6] S. Vladimirov and O. Kognovitsky, "Wideband data signals with direct ternary maximum length sequence spread spectrum and their characteristics," *Proceedings of Telecommunication Universities*, vol. 3, pp. 28–36, 2017.

[7] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei, "The era of 1-bit LLMs: All large language models are in 1.58 bits," *arXiv preprint arXiv:2402.17764*, 2024.

[8] R. Ahlswede and H. Aydinian, "Error control codes for parallel asymmetric channels," *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 831–836, 2008.

[9] N. Bitouzé and A. G. i Amat, "Coding for a non-symmetric ternary channel," in *2009 Information Theory and Applications Workshop*. IEEE, 2009, pp. 113–118.

[10] L. Tolhuizen, "Bounds for codes for a non-symmetric ternary channel," *arXiv preprint arXiv:1004.1511*, 2010.

[11] Y. Ben-Hur, S. Stern, Y. Cohen, and Y. Cassuto, "Graph codes for dual-parameter barrier channels," in *2024 60th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2024, pp. 1–8.

[12] Y. Ben-Hur and Y. Cassuto, "Coding on dual-parameter barrier channels beyond worst-case correction," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.

[13] S. M. Moser, P.-N. Chen, and H.-Y. Lin, "Error probability analysis of binary asymmetric channels," *Dept. El. & Comp. Eng., Nat. Chiao Tung Univ*, 2009.

[14] S. Al-Bassam and B. Bose, "Asymmetric/unidirectional error correcting and detecting codes," *IEEE Transactions on Computers*, vol. 43, no. 5, pp. 590–597, 1994.

[15] E. N. Gilbert, "A comparison of signalling alphabets," *The Bell System Technical Journal*, vol. 31, no. 3, pp. 504–522, 1952.

[16] R. R. Varshamov, "Estimate of the number of signals in error correcting codes," *Dokl. Acad. Nauk SSSR*, vol. 117, pp. 739–754, 1957.

[17] J. Gu and T. Fuja, "A generalized Gilbert-Varshamov bound derived via analysis of a code-search algorithm," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 1089–1093, 1993.

[18] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, 1972.

[19] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, 2002.

[20] Y. Ben-Hur, "reed-muller-codes-matlab," 2021, available: https://github.com/benhuryuval/reed-muller-codes-matlab.

[21] I. Dumer, "Soft-decision decoding of Reed-Muller codes: a simplified algorithm," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 954–963, 2006.

[22] I. Dumer and K. Shabunov, "Soft-decision decoding of Reed-Muller codes: recursive lists," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1260–1266, 2006.

[23] M. Ye and E. Abbe, "Recursive projection-aggregation decoding of Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 4948–4965, 2020.

[24] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[25] B. Bose, N. Elarief, and L. G. Tallini, "On codes achieving zero error capacities in limited magnitude error channels," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 257–273, 2018.

[26] A. Ravagnani and F. R. Kschischang, "Adversarial network coding," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 198–219, 2019.

[27] J. Iversen, "freezecolors / unfreezecolors," 2022, available: https://github.com/jiversen/freezeColors/releases/tag/v2.5.

Yuval Ben-Hur (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. (cum laude) degrees in Electrical Engineering from the Technion–Israel Institute of Technology, Haifa, Israel, in 2014 and 2018, respectively, where he is currently pursuing the Ph.D. degree in Electrical Engineering.

From 2013 to 2022, he was an Algorithm Engineer and a Team Leader with the Israeli Defense Industry. Since 2022, he has been working on wireless communication systems as an Algorithm Engineer at Qualcomm. His research interests include the application of information and communication theory to emerging data storage and processing devices. He was a recipient of the Best Student Paper Award in data storage from the IEEE Communications Society in 2019 and the Best Student Paper Runner-Up Award from the 2022 IEEE International Workshop on Machine Learning for Signal Processing.

Yuval Cassuto (S'02–M'08–SM'14) is a faculty member at the Viterbi Department of Electrical and Computer Engineering, Technion – Israel Institute of Technology. His research interests lie at the intersection of the theoretical information sciences and the engineering of practical computing and storage systems. He has served on the technical program committees of leading conferences in both theory and systems. During 2010-2011 he has been a Scientist at EPFL, the Swiss Federal Institute of Technology in Lausanne. From 2008 to 2010 he was a Research Staff Member at Hitachi Global Storage Technologies, San Jose Research Center. In 2018-2019 he held a Visiting Professor position at Western Digital Research, and a Visiting Scholar position at UC Berkeley. He received the B.Sc degree in Electrical Engineering, summa cum laude, from the Technion in 2001, and the M.S. and Ph.D. degrees in Electrical Engineering from the California Institute of Technology, in 2004 and 2008, respectively. From 2000 to 2002, he was with Qualcomm, Israel R&D Center, where he worked on modeling, design and analysis in wireless communications. Dr. Cassuto has won the Best Student Paper Award in data storage from the IEEE Communications Society in 2010 as a student, and in 2019 as an adviser. He also won faculty awards from Qualcomm, Intel, and Western Digital. Before that, he won the 2001 Texas Instruments DSP and Analog Challenge $100,000 prize.