Limited-Magnitude Error-Correcting Gray Codes for Rank Modulation

Yonatan Yehezkeally
Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer Sheva 8410501, Israel
yonatany@post.bgu.ac.il

Moshe Schwartz
Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer Sheva 8410501, Israel
schwartz@ee.bgu.ac.il

***Abstract*—We construct Gray codes over permutations for the rank-modulation scheme, which are also capable of correcting errors under the infinity-metric. These errors model limited-magnitude or spike errors, for which only single-error-detecting Gray codes are currently known. Surprisingly, the error-correcting codes we construct achieve better asymptotic rates than that of presently-known constructions not having the Gray property. We also cast the problem of improving upon these results into the context of finding a certain type of auxiliary codes in the symmetric group of even orders.**

I. Introduction

Rank modulation is a method for storing information in non-volatile memories [5], which has been researched in recent years. It calls for storing information in relative values stored in a group of cells rather than the absolute values of single cells. More precisely, it stores information in the permutation suggested by sorting a group of cells by their relative values, e.g., charge levels in flash memory cells. It allows for increased robustness against certain noise mechanisms (e.g., charge leakage in flash memory cells), as well as alleviating some inherent challenges in flash memories (e.g., programming/erasure-asymmetry and programming-overshoot).

Several error models have been studied for rank modulation, including the Kendall τ-metric [1], [6], [8], [16], and the ℓ_∞-metric [7], [10]–[12]. In this paper we focus on the ℓ_∞-metric, which models limited-magnitude or spike noise, i.e., we assume that the rank of any given cell–its position when sorting the group of cells–could not have changed by more than a given amount. [7], [11] have presented constructions for error-correcting codes under this metric, as well as explored some non-constructive lower- and upper-bounds on the parameters of existing codes. [12] has since employed methods of relabeling to optimize the minimal distance of known constructions.

Gray codes were first discussed over the space of binary vectors, where each pair of consecutive vectors differed by a single bit-flip [2]; the concept has since been generalized to include codes over arbitrary alphabets, requiring only that codewords could be ordered in a sequence, where each codeword is derived from the previous by one of a predefined set of functions (see the survey [9]). In particular, in the context of rank modulation, the use of Gray codes has been shown to reduce write-time by eliminating the risk of programming-overshoot and allow integration with other multilevel-cells coding schemes [5].

Gray codes with error-correction capabilities have sometimes been referred to as spread-d circuit codes (see [3] and references therein). Specifically, in the context of rank modulation, such codes were so far only studied for the case of single-error-detection, where they were dubbed snake-in-the-box codes (or, more appropriately, coil-in-the-box codes, when they are cyclic). [13] studied these codes under both the Kendall τ-metric and the ℓ_∞-metric, and more recent papers [4], [15] have nearly categorized and constructed optimally sized coil-in-the-box codes under the former metric.

In this work we focus on the ℓ_∞-metric and present a construction of error-correcting Gray codes capable of correcting an arbitrary number of limited-magnitude errors. The allowed transitions between codewords are the "push-to-the-top" operations, used in all previous works [4], [5], [13], [15]. The resulting codes will be shown to be larger than known constructions in the case of fixed minimal distance, as well as achieve better asymptotic rates than known codes in the case of $d = \Theta(n)$.

The paper is organized as follows. In Section II we present notations and definitions. In Section III we present our construction, and we discuss its performance in comparison with known constructions and bounds in Section IV. We conclude in Section V by reviewing our results and suggest problems for future study. Due to the page limit, proofs are sketched or omitted. The reader is referred to [14] for the full version.

II. Preliminaries

For $n \in \mathbb{N}$, we let S_n be the *Symmetric Group*, the set of all permutations on $[n] = \{1, 2, \ldots, n\}$ (i.e., bijections $\sigma : [n] \xrightarrow[\text{onto}]{1-1} [n]$), with composition as group action, $\sigma\tau(k) = (\sigma \circ \tau)(k) = \sigma(\tau(k))$. Throughout the paper we shall denote the identity permutation $\mathrm{Id} \in S_n$ defined for all $k \in [n]$: $\mathrm{Id}(k) = k$.

We use the *cycle notation* for permutations, i.e., for distinct $\{a_j\}_{j=1}^{k} \subseteq [n]$ we let $\sigma = (a_1, a_2, \ldots, a_k)$ be the permutation such that $\sigma(a_j) = a_{(j \bmod k)+1}$ and $\sigma(b) = b$ for all $b \in [n] \setminus \{a_j\}_{j=1}^{k}$. Trivially, every permutation can be represented as a composition of disjoint cycles. It is also well known that every permutation can be represented as a composition

This work was supported in part by ISF grant no. 130/14.

of *transpositions*, cycles of length 2, and that the parity of the number of transpositions in that representation is unique (although the representation itself is not). We therefore have *even* and *odd* permutations, and the set of even permutations forms a subgroup $A_n \leqslant S_n$ named the *Alternating Group*. We will say that $C \subseteq S_n$ is *parity-preserving* if every two elements $\sigma, \tau \in C$ have the same parity.

We also use the *vector notation* for permutations, $\sigma = [\sigma(1), \sigma(2), \ldots, \sigma(n)]$. This allows us to more easily notate, for $1 \leqslant i < j \leqslant n$, the *"push-to-the-ith-index"* transition $t_{i\uparrow j} : S_n \to S_n$ by

$$t_{i\uparrow j}\left(\left[a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_{j-1}, a_j, a_{j+1}, \ldots, a_n\right]\right) = \left[a_1, a_2, \ldots, a_{i-1}, a_j, a_i, a_{i+1}, \ldots, a_{j-1}, a_{j+1}, \ldots, a_n\right].$$

We follow [4], [5], [13], [15] (among others) in dubbing "push-to-the-1st-index" transitions as *"push-to-the-top"* transitions, and we denote $t_{\uparrow j} = t_{1\uparrow j}$. Finally, we define the *"push-to-the-bottom"* transition on the jth index, $t_{\downarrow j} : S_n \to S_n$,

$$t_{\downarrow j}\left(\left[a_1, a_2, \ldots, a_{j-1}, a_j, a_{j+1}, \ldots, a_n\right]\right) = = \left[a_1, a_2, \ldots, a_{j-1}, a_{j+1}, \ldots, a_n, a_j\right].$$

Given any set S, and a collection of transitions $T \subseteq \{f \mid f : S \to S\}$, we define a *$T$-Gray Code* over S to be a sequence $C = (c_r)_{r=1}^{M} \subseteq S$ such that for all $1 \leqslant r < r' \leqslant M$ we have $c_r \neq c_{q'}$ and such that for all $1 \leqslant r < M$ there exists $t_r \in T$ satisfying $c_{r+1} = t_r(c_r)$ (we say that a sequence C is contained in S, by abuse of notation, if $c_r \in S$ for all r. That is, we may refer to a Gray Code as an unordered set–or simply a code–when desired for simplicity). We call $M = |C|$ the *size* of the code, and $t_1, t_2, \ldots, t_{M-1}$ the transition sequence *generating* C. If there exists $t \in T$ such that $c_1 = t(c_M)$ we say that C is *cyclic*, and include $t_M = t$ in its generating transition sequence. If $C = S$, we say that C is *complete*.

In this paper, we fix $S = S_n$. We say that $C = (c_1, c_2, \ldots, c_M) \subseteq S_n$ is a $G_{i\uparrow}(n, M)$ if it is a cyclic Gray code with transition set $T = \left\{t_{i\uparrow j} \mid i < j \leqslant n\right\}$. When $i = 1$ we refer to C as a "push-to-the-top" code and denote it $G_{\uparrow}(n, M)$, and we likewise denote "push-to-the-bottom" codes $G_{\downarrow}(n, M)$.

It is worthwhile to note that when S is a group, and T consists of the group action of some subset on S, and C is a (complete- and/or cyclic-) Gray code generated by $t_1, t_2, \ldots, t_{M-1}$ $(, t_M)$, then for all $\sigma \in S$ we observe that $(\sigma, t_1(\sigma), t_2(t_1(\sigma)), \ldots)$ is also a (complete- and/or cyclic-respectively) Gray code. In other words, the code is shift invariant. In these cases we might refer to the transition sequence generating the code as the code itself, when desirable for simplicity. It is also of interest to observe that $t_{i\uparrow j}(\sigma) = \sigma \circ (j, j-1, \ldots, i)$, i.e., "push-to-the-$i$th-index" transitions are group actions.

When S is equipped with a metric $d : S \times S \to \mathbb{R}_+$, and $C \subseteq S$ has the property that for all $\sigma, \tau \in C$ either $\sigma = \tau$ or $d(\sigma, \tau) \geqslant d$, for some constant $d > 0$, then C (when considered as an unordered set) is commonly referred to as an *error-correcting code* with *minimal distance* d. If $d(\cdot, \cdot)$ models an error mechanism, such that a single error corresponds to distance 1, and $2p + q < d$, it is well known that C can then correct p errors, and detect q additional errors.

Error-correcting Gray codes have sometimes been referred to as *spread-d circuit codes* (see [3] and references therein), where they were traditionally defined by requiring that for all $c_r, c_{r'} \in C$, $(r - r' \bmod |C|) \geqslant d$ implies $d(c_r, c_{r'}) \geqslant d$. In that way, e.g., spread-1 circuit codes are traditional Gray codes. This eased requirement was made necessary since, working with the Hamming distance d_H in the n-cube, one cannot have codewords at distance less than d in the code sequence attain a distance of at least d. We shall depart from it here to deal with Gray codes which are classic error-correcting codes, but the codes presented in this paper are nevertheless also, in particular, spread-d circuit-codes.

We shall focus on the ℓ_∞-metric defined on S_n by

$$d_\infty(\sigma, \tau) = \max_{j \in [n]} |\sigma(j) - \tau(j)|.$$

That is, it is the metric induced on S_n by the embedding into $\mathbb{Z}^n$ (and, indeed, $\mathbb{R}^n$) implied by the vector notation, and the ℓ_∞-metric in these spaces. [11] studied error-correcting codes in S_n with d_∞, which it dubbed *limited-magnitude Rank-Modulation* codes, and denoted a code C with minimal distance d as an $(n, |C|, d)$-LMRM code. In our case, if a $G_{i\uparrow}(n, M)$ is also an (n, M, d)-LMRM code, we shall denote it a $G_{i\uparrow}(n, M, d)$ (likewise for $G_{\uparrow}$ and $G_{\downarrow}$).

Finally, we organize our notation of codes in Table I.

TABLE I
CODE NOTATIONS FOR $C \subseteq S_n$.

Notation	Definition
$G_{i\uparrow}(n, M)$	$C = (c_r)_{r=1}^{M} \subseteq S_n$ such that for all r: $c_{(r \bmod M)+1} = t_{i\uparrow j_r}(c_r)$.
$G_{\uparrow}(n, M)$	C is a $G_{1\uparrow}(n, M)$.
$G_{\downarrow}(n, M)$	$C = (c_r)_{r=1}^{M} \subseteq S_n$ such that for all r: $c_{(r \bmod M)+1} = t_{\downarrow j_r}(c_r)$.
(n, M, d)-LMRM	$C \subseteq S_n$, $\|C\| = M$ and for all $c_1 \neq c_2 \in C$: $d_\infty(c_1, c_2) \geqslant d$.
$G_{i\uparrow}(n, M, d)$	C is a $G_{i\uparrow}(n, M)$ and an (n, M, d)-LMRM.
$G_{\uparrow}(n, M, d)$	C is a $G_{1\uparrow}(n, M, d)$.
$G^{\text{aux}}_{\uparrow}(k, M)$	C is a $G_{\uparrow}(k, M)$, and for all $q \in [k-1]$: $\sigma \in C \implies (q, k)\sigma \notin C$. (See Section III-A.)

III. CODE CONSTRUCTION

In this section we present the main construction of our paper. In order to do so, we first describe a construction for an auxiliary code which will be a component of the main construction.

A. Auxiliary construction

We define auxiliary codes in S_k in the following way: we say that C is $G^{\text{aux}}_{\uparrow}(k, M)$ if it is $G_{\uparrow}(k, M)$ and for all $q \in [k-1]$ it holds that

$$\sigma \in C \implies (q, k)\sigma \notin C.$$

We study the existence of such codes for use in our construction. Firstly, note that the only existing $G_{\uparrow}^{\text{aux}}(2,M)$ codes are the singletons $\{\mathrm{Id}\},\{(1,2)\}$. However, for $k \geqslant 3$ there do exist $G_{\uparrow}^{\text{aux}}(k,M)$ codes with $M \geqslant 3$, as one such example is $\left\{\mathrm{Id}, t_{\uparrow 3}\,\mathrm{Id}, {t_{\uparrow 3}}^2\,\mathrm{Id}\right\}$. We also note the following:

Lemma 1 If $C \subseteq S_k$ is $G_{\uparrow}^{\text{aux}}(k,M)$, then $M \leqslant \frac{|S_k|}{2}$.

This motivates us to examine another family of codes, namely, parity-preserving codes, due to the following:

Lemma 2 If $C \subseteq S_k$ is parity-preserving then $|C| \leqslant \frac{|S_k|}{2}$.

Lemma 3 If $C \subseteq S_k$ is a parity-preserving $G_{\uparrow}(k,M)$, then C is $G_{\uparrow}^{\text{aux}}(k,M)$.

Parity-preserving $G_{\uparrow}^{\text{aux}}(2m+1,M)$ codes are known to exist, nearly achieving the aforementioned bound:

Lemma 4 [4, Thm. 18] [15] For $k \geqslant 2$, there exist parity-preserving $G_{\uparrow}(2k+1, M_{2k+1})$ codes with

$$M_{2m+1} = |A_{2m+1}| - (2m-1) = \frac{(2m+1)!}{2} - (2m-1).$$

To complete the picture for $k=1$, we recall the $G_{\uparrow}^{\text{aux}}(3,3)$ previously presented, generated by 3 $t_{\uparrow 3}$ transitions, and note its size $3 > \frac{3!}{2} - 1$.

Although not declared, it is tacitly shown in the references that such codes can be assumed to have $t_{\uparrow 2m+1}$ as the first transition in their generating transition sequence.

In comparison, as noted in [13], a parity-preserving $G_{\uparrow}(2m,M)$ must satisfy $M \leqslant \frac{|S_{2m}|}{2m}$, as it must never employ a $t_{\uparrow 2m}$ transition. We therefore examine more general $G_{\uparrow}^{\text{aux}}$ codes. We begin by noting the following lemma:

Lemma 5 [5, Thm. 4,7] For all $n \geqslant 1$ there exist $G_{\uparrow}(n,n!)$ codes, that is, complete and cyclic "push-to-the-top" Gray codes over the symmetric group S_n.

Relying on these codes, we observe:

Theorem 6 For all $m \geqslant 2$ there exists a $G_{\uparrow}^{\text{aux}}(2m, \frac{|S_{2m}|}{2m-1})$, starting at Id and with a generating sequence starting with $t_{\uparrow 2m}$.

Proof sketch: Take a $G_{\uparrow}(2m-2,(2m-2)!)$ C', provided by Lemma 5. We follow the concept of [5, Thm. 7] in extending C' to S_{2m}. Note that $\sigma_0 := t_{\uparrow 2m}\,\mathrm{Id} = [2m,1,\ldots,2m-1]$. If we take $t_{\uparrow i_1}, t_{\uparrow i_2}, \ldots, t_{\uparrow i_{(2m-2)!}}$ to be the transition sequence generating C', then the transition sequence $t_{\downarrow 2m+1-i_1}$, $t_{\downarrow 2m+1-i_2}$, $\ldots$, $t_{\downarrow 2m+1-i_{(2k-2)!}}$ of "push-to-the-bottom" operations, applied iteratively to σ_0, generates $C'' \subseteq S_{2m}$, a $G_{\downarrow}(2m,(2m-2)!)$, all of whose elements' vector notations begin with $[2m,1]$.

We note that $t_{\downarrow 2m+1-j} = {t_{\uparrow 2m}}^{2m-1} t_{\uparrow j}$, so expanding each "push-to-the-bottom" transition of our code in this fashion we get $C \subseteq S_{2m}$, a $G_{\uparrow}(2m,(2m-2)!2m)$, where every block of $2m$ elements is comprised of cyclic shifts of some $\sigma \in C''$. ∎

We remark that, while Theorem 6 does not produce codes much larger than the aforementioned bound, they do at least allow us to permute the last element, and thus combine Lemma 4 and Theorem 6 into the following lemma:

Corollary 7 For all $k \geqslant 3$ there exist a $G_{\uparrow}^{\text{aux}}(k,\tilde{M}_k)$ starting with Id and a $t_{\uparrow k}$ transition, where

$$\tilde{M}_k = \begin{cases} 3 & k = 3; \\ \frac{k!}{2} - (k-2) & 3 < k \equiv 1 \pmod 2; \\ \frac{k!}{(k-1)} & k \equiv 0 \pmod 2. \end{cases}$$

B. Construction

We now present a construction of $G_{\uparrow}(n,\mathcal{M},d)$ codes, for $d \leqslant n$, which we base on Lemma 7 and Lemma 5.

It will simplify the presentation to assume $n = kd$ for some positive k, since in that case every congruence class modulo d of $[n]$ has size k, but the construction is applicable to any $n > d$ with natural amendments. We discuss these changes, focusing on special cases, after presenting the construction.

We construct a $G_{\uparrow}(n,\mathcal{M},d)$ code by an iterative process, starting at the permutation

$$\sigma_0(j) := d\,(j \bmod k) + \left\lceil \frac{j}{k} \right\rceil.$$

Note that for all $0 \leqslant u < d$ and $ku+1 \leqslant i < j \leqslant k(u+1)$ it holds that $\sigma_0(i) \equiv \sigma_0(j) \pmod d$.

We obtain a transition sequence $t_{\uparrow r_1}, t_{\uparrow r_2}, \ldots, t_{\uparrow r_{k!}}$ which generates the $G_{\uparrow}(k,k!)$ provided by Lemma 5. We construct a $G_{k(d-1)+1\uparrow}(n,k!)$ code which we denote C_{d-1} by iteratively applying to σ_0 the transition sequence

$$t_{k(d-1)+1\uparrow k(d-1)+r_1}, t_{k(d-1)+1\uparrow k(d-1)+r_2}, \cdots \\ \cdots, t_{k(d-1)+1\uparrow k(d-1)+r_{k!}}.$$

Note that for every distinct $\sigma,\tau \in C_{d-1}$, there exists j, $m(d-1) < j \leqslant kd = n$, such that $\sigma(j) \neq \tau(j)$. Since by construction $\sigma(j) \equiv \tau(j) \equiv 0 \pmod d$, we observe

$$d_\infty(\sigma,\tau) \geqslant |\sigma(j) - \tau(j)| \geqslant d,$$

implying that C_{d-1} is also a $G_{k(d-1)+1\uparrow}(n,k!,d)$.

Theorem 8 Let $0 < m \leqslant d-1$, and suppose that we have C_m, a $G_{km+1\uparrow}(n,M',d)$ starting at σ_0. Additionally, assume the existence of an auxiliary code $G_{\uparrow}^{\text{aux}}(k+1,\tilde{M}_{k+1})$. Then we can construct a $G_{k(m-1)+1\uparrow}(n,\tilde{M}_{k+1}\cdot M',d)$ starting at σ_0, which we denote C_{m-1}.

Proof sketch: Let $t_{\uparrow k+1}, t_{\uparrow i_2}, \ldots, t_{\uparrow i_M}$ be the transition sequence generating the $G_{\uparrow}^{\text{aux}}(k+1,\tilde{M}_{k+1})$ code provided by Lemma 7. Note that $t_{\uparrow i_M} t_{\uparrow i_{M-1}} \cdots t_{\uparrow i_3} t_{\uparrow i_2} = {t_{\uparrow k+1}}^{-1}$.

We construct the code C_{m-1} as follows: replace each $t_{km+1\uparrow j}$ transition of C_m with $t_{k(m-1)+1\uparrow j}$, followed by $t_{k(m-1)+1\uparrow k(m-1)+i_2}$, $t_{k(m-1)+1\uparrow k(m-1)+i_3}$, and so on until $t_{k(m-1)+1\uparrow k(m-1)+i_M}$. Note that apart from the first transition, every transition acts on an index j' satisfying $k(m-1)+1 < j' \leqslant km$. Thus,

$$t_{km+1\uparrow j} = \left(\prod_{r=2}^{M} t_{k(m-1)+1\uparrow k(m-1)+i_r}\right) t_{k(m-1)+1\uparrow j}$$

(where the product is expanded right-to-left), therefore C_{m-1} expands each "push-to-the-$km+1$st-index" transition of C_m into M "push-to-the-$k(m-1)+1$st-index" transitions. ∎

To conclude this section, note that using Theorem 8 iteratively we obtain a code C_0, which is a $G_\uparrow(n, \tilde{M}_{k+1}{}^{d-1} \cdot k!, d)$.

We now also note that if $n \not\equiv 0 \pmod d$ then $n \bmod d$ congruence classes modulo d of $[n]$ are of size $\left\lceil \frac{n}{d} \right\rceil$, and the rest are of size $\left\lfloor \frac{n}{d} \right\rfloor$. Defining σ_0 by a vector notation consisting of blocks comprised of congruence classes, we can still apply Theorem 6 iteratively if we choose an appropriate $G_\uparrow^{\text{aux}}$ for each step, to achieve C_0, a $G_\uparrow(n, \mathcal{M}, d)$, where

$$\mathcal{M} = \tilde{M}_{\lceil n/d \rceil + 1}{}^{n \bmod d} \cdot \tilde{M}_{\lfloor n/d \rfloor + 1}{}^{d-(n \bmod d)-1} \cdot \left\lfloor \frac{n}{d} \right\rfloor !.$$

A special case is $n < 2d$, where all but $(n \bmod d)$ congruence classes are singletons. We will amend our construction by starting it from

$$C_{m-1} = \left\{ \sigma_0, t_{2m-1 \uparrow 2m+1} \sigma_0, t_{2m-1 \uparrow 2m+1}{}^2 \sigma_0 \right\},$$

where $m = n \bmod d$, effectively only cycling the single member of the congruence class $\equiv m+1 \pmod d$ into previous classes, fixing $\sigma_0(j)$ for $j > 2m+1$.

C. Code size

We note the asymmetry in our construction between congruence classes of odd and even sizes; indeed, a class of size $k \geqslant 2$ (for all classes other than $\equiv 0 \pmod d$), whose contribution is based on the $G_\uparrow(k, k!)$ code provided by Lemma 5) contributes to the code size a factor of

$$\tilde{M}_{k+1} = \begin{cases} 3 & k = 2 \\ \frac{(k+1)!}{2} - (k-1) & 2 < k \equiv 0 \pmod 2 \\ \frac{(k+1)!}{k} & k \equiv 1 \pmod 2 \end{cases}$$

It is therefore important to note that when $\left\lfloor \frac{n}{d} \right\rfloor \equiv 0 \pmod 2$, $[n]$ has $(n \bmod d)$ congruence classes modulo d of odd size $\left\lceil \frac{n}{d} \right\rceil$, and $d - (n \bmod d)$ classes of even size $\left\lfloor \frac{n}{d} \right\rfloor$. We therefore construct a code C_0 of size

$$|C_0| = \left(\frac{(\lceil n/d \rceil + 1)!}{\lceil n/d \rceil} \right)^{n \bmod d} \cdot \left\lfloor \frac{n}{d} \right\rfloor ! \cdot \left(\frac{(\lfloor n/d \rfloor + 1)!}{2} - \left(\left\lfloor \frac{n}{d} \right\rfloor - 1 \right) \right)^{d-(n \bmod d)-1}.$$

It is possible to achieve a slight gain by reordering σ_0 so that the last block consists of a congruence class of odd size, rather than even, where the added complexity of index calculation is inconsequential.

Likewise, when $\left\lfloor \frac{n}{d} \right\rfloor \equiv 1 \pmod 2$, $[n]$ has $(n \bmod d)$ congruence classes modulo d of even size $\left\lceil \frac{n}{d} \right\rceil$, and $d - (n \bmod d)$ classes of odd size $\left\lfloor \frac{n}{d} \right\rfloor$, and we construct a code C_0 of size

$$|C_0| = \left(\frac{(\lceil n/d \rceil + 1)!}{2} - \left(\left\lceil \frac{n}{d} \right\rceil - 1 \right) \right)^{n \bmod d} \cdot \left\lfloor \frac{n}{d} \right\rfloor ! \cdot \left(\frac{(\lfloor n/d \rfloor + 1)!}{\lfloor n/d \rfloor} \right)^{d-(n \bmod d)-1}.$$

We also note the special case $\left\lfloor \frac{n}{d} \right\rfloor = 1$. In this case we only permute $(n \bmod d) = (n-d)$ congruence classes of $[n]$, (and each such class has $2 = \left\lfloor \frac{n}{d} \right\rfloor + 1$ elements). As mentioned, we therefore construct a code of size $|C_0| = 3^{n \bmod d}$.

IV. Comparison to known results

$G_\uparrow(n, M, 2)$ codes were studied in [13, Thm. 24], where it was shown that such codes can be constructed with sizes

$$M = \left\lceil \frac{n}{2} \right\rceil ! \left(\left\lfloor \frac{n}{2} \right\rfloor + \left(\left\lfloor \frac{n}{2} \right\rfloor - 1 \right) ! \right).$$

The construction of Theorem 8 improves this size by a factor of $\approx \left\lfloor \frac{n}{2} \right\rfloor$ for $n \equiv 1, 2 \pmod 4$, and $\approx \frac{1}{2} \left\lfloor \frac{n}{2} \right\rfloor^2$ for $n \equiv 0, 3 \pmod 4$.

Examining error-correcting codes with the ℓ_∞ metric which are not necessarily Gray codes, we observe that the best known general code construction to date, [11, Cst. 1, Thm. 2] and [7, Sec. III-A], presented (n, M, d)-LMRM codes with sizes

$$M = \left\lceil \frac{n}{d} \right\rceil !^{n \bmod d} \left\lfloor \frac{n}{d} \right\rfloor !^{d-(n \bmod d)},$$

which our construction improves upon, more substantially the more even-sized congruence classes modulo d $[n]$ has.

We now go on to examine the case of $d = \Theta(n)$ as n grows to infinity. For an (n, M, d)-LMRM code (and in particular a $G_\uparrow(n, M, d)$), we follow the convention (e.g., [11]) of defining the *Rate* of the code $R = \frac{1}{n} \log_2 M$, and its *normalized distance* $\delta = \frac{d}{n}$.

The following was proven in [11]:

Lemma 9 [11, Thm. 23] For any (n, M, d)-LMRM code

$$R \leqslant 2 - 2\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \left(\delta \left\lfloor \frac{1}{\delta} \right\rfloor - \delta \right) \log_2(\delta) - \left(1 + \delta - \delta \left\lfloor \frac{1}{\delta} \right\rfloor \right) \log_2 \left(1 + \delta - \delta \left\lfloor \frac{1}{\delta} \right\rfloor \right) + o(1).$$

Lemma 10 [11, Thm. 27] For any $0 < \delta \leqslant 1$ the construction of [11, Cst. 1, Thm. 2] and [7, Sec. III-A] yields codes with

$$R = \left(1 - \delta \left\lfloor \frac{1}{\delta} \right\rfloor \right) \log_2 \left(\left\lceil \frac{1}{\delta} \right\rceil ! \right) + \left(\delta + \delta \left\lfloor \frac{1}{\delta} \right\rfloor - 1 \right) \log_2 \left(\left\lfloor \frac{1}{\delta} \right\rfloor ! \right).$$

Lemma 11 [11, Thm. 25] For any $0 < \delta \leqslant 1$ there exist (n, M, d)-LMRM codes satisfying $\frac{d}{n} \geqslant \delta$ with rate $R \geqslant f_{\text{GV}}(\delta) + o(1)$, where

$$f_{\text{GV}} = \begin{cases} \log_2 \frac{1}{\delta} + 2\delta \left(\log_2 e - 1 \right) - 1 & 0 < \delta \leqslant \frac{1}{2} \\ -2\delta \log_2 \frac{1}{\delta} + 2(1-\delta) \log_2 e & \frac{1}{2} \leqslant \delta \leqslant 1 \end{cases}$$

We therefore aim to show that our construction can bridge some of the gap between the given bounds and known constructions.

We find for C_0, by substituting $(n \bmod d) = n - d\left\lfloor \frac{n}{d} \right\rfloor$ and $d = n\delta$, that for $\lfloor 1/\delta \rfloor \equiv 0 \pmod 2$

$$R = \left(1 - \delta\left\lfloor\frac{1}{\delta}\right\rfloor\right)\log_2\left[\left\lceil\frac{1}{\delta}\right\rceil!\left(1+\frac{1}{\lceil 1/\delta\rceil}\right)\right] + \\ + \left(\delta + \delta\left\lfloor\frac{1}{\delta}\right\rfloor - 1\right)\log_2\left(\left(\left\lfloor\frac{1}{\delta}\right\rfloor + 1\right)!\right) \\ - \left(\delta + \delta\left\lfloor\frac{1}{\delta}\right\rfloor - 1\right),$$

except when $\lfloor 1/\delta \rfloor = 2$, where we have

$$R = 3\,(1-2\delta) + (3\delta - 1)\log_2(3) - o(1).$$

Similarly, for $\lfloor 1/\delta \rfloor \equiv 1 \pmod 2$

$$R = \left(1 - \delta\left\lfloor\frac{1}{\delta}\right\rfloor\right)\log_2\left[\left(\left\lceil\frac{1}{\delta}\right\rceil + 1\right)!\left(1 - \frac{2(\lceil 1/\delta\rceil - 1)}{(\lceil 1/\delta\rceil + 1)!}\right)\right] \\ + \left(\delta + \delta\left\lfloor\frac{1}{\delta}\right\rfloor - 1\right)\log_2\left(\left\lfloor\frac{1}{\delta}\right\rfloor!\left(1+\frac{1}{\lfloor 1/\delta\rfloor}\right)\right) \\ + \delta\left\lfloor\frac{1}{\delta}\right\rfloor - 1 - o(1),$$

but, again, for $\lfloor 1/\delta \rfloor = 1$ we must calculate separately

$$R = (1-\delta)\log_2(3).$$

In conclusion, these asymptotic rates and bounds are shown in Figure 1. We note in particular that the rate of codes produced by Theorem 8 is strictly higher than that of previously known constructions (as in Lemma 10). Furthermore, it produces codes with rates higher than those guaranteed by the Gilbert-Varshamov-like Lemma 11 for δ sufficiently close to $\frac{1}{10}$, $\frac{1}{8}$, $\frac{1}{6}$, and all δ greater than ≈ 0.21, whereas known constructions only bypassed these rates for δ greater than ≈ 0.349.

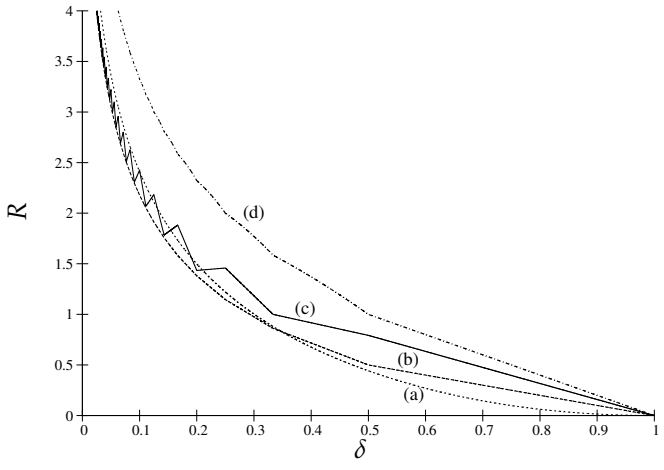

Figure 1. (a) The Gilbert-Varshamov-like lower bound of Lemma 11. (b) The rate of codes from Lemma 10 constructed in [11]. (c) The rate of codes C_0 constructed by Theorem 8. (d) The upper bound of Lemma 9.

V. Conclusion

In this paper we presented the class of $G_{\uparrow}^{\text{aux}}(n, M)$ codes, largely leveraging codes designed for the rank-modulation scheme under the Kendall τ-metric, in order to aid in the construction of error-correcting codes for the ℓ_∞-metric. By doing so, we were able to construct codes that achieve better asymptotic rates than previously-known constructions, while also incorporating the property of being Gray codes.

Furthermore, much as in the case of codes designed for the Kendall τ-metric, our auxiliary construction suffers from asymmetry between the cases of even- and odd-sized congruence classes. This creates an anomaly in the asymptotic rate in certain regions of δ, where a code with a higher normalized distance also has a higher rate. As recent works in that context indicate, this asymmetry is not necessarily inherent to the problem at hand, but rather stems from a limitation in the methods thus far utilized to solve it. We posit that $G_{\uparrow}^{\text{aux}}(2m, M)$ codes with M approaching $\frac{(2m)!}{2}$ do exist, a conjecture that we hope may incite further research.

References

[1] A. Barg and A. Mazumdar, "Codes in permutations and error correction for rank modulation," *IEEE Trans. Inform. Theory*, vol. 56, no. 7, pp. 3158–3165, Jul. 2010.

[2] F. Gray, "Pulse code communication," March 1953, U.S. Patent 2632058.

[3] S. Hood, D. Recoskie, J. Sawada, and D. Wong, "Snakes, coils, and single-track circuit codes with spread k," *J. Comb. Optim.*, vol. 30, no. 1, pp. 42–62, Jul. 2015.

[4] M. Horovitz and T. Etzion, "Constructions of snake-in-the-box codes for rank modulation," *IEEE Trans. Inform. Theory*, vol. 60, no. 11, pp. 7016–7025, Nov. 2014.

[5] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. Inform. Theory*, vol. 55, no. 6, pp. 2659–2673, Jun. 2009.

[6] A. Jiang, M. Schwartz, and J. Bruck, "Correcting charge-constrained errors in the rank-modulation scheme," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2112–2120, May 2010.

[7] T. Kløve, T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Permutation arrays under the Chebyshev distance," *IEEE Trans. Inform. Theory*, vol. 56, no. 6, pp. 2611–2617, Jun. 2010.

[8] A. Mazumdar, A. Barg, and G. Zémor, "Constructions of rank modulation codes," *IEEE Trans. Inform. Theory*, vol. 59, no. 2, pp. 1018–1029, Feb. 2013.

[9] C. D. Savage, "A survey of combinatorial Gray codes," *SIAM Rev.*, vol. 39, no. 4, pp. 605–629, Dec. 1997.

[10] M.-Z. Shieh and S.-C. Tsai, "Decoding frequency permutation arrays under Chebyshev distance," *IEEE Trans. Inform. Theory*, vol. 56, no. 11, pp. 5730–5737, Nov. 2010.

[11] I. Tamo and M. Schwartz, "Correcting limited-magnitude errors in the rank-modulation scheme," *IEEE Trans. Inform. Theory*, vol. 56, no. 6, pp. 2551–2560, Jun. 2010.

[12] ——, "On the labeling problem of permutation group codes for the infinity metric," *IEEE Trans. Inform. Theory*, vol. 58, no. 10, pp. 6595–6604, Oct. 2012.

[13] Y. Yehezkeally and M. Schwartz, "Snake-in-the-box codes for rank modulation," *IEEE Trans. Inform. Theory*, vol. 58, no. 8, pp. 5471–5483, Aug. 2012.

[14] ——, "Limited-magnitude error-correcting Gray codes for rank modulation," *arXiv preprint arXiv:1601.05218*, 2016.

[15] Y. Zhang and G. Ge, "Snake-in-the-box codes for rank modulation under Kendall's τ-metric," *IEEE Trans. Inform. Theory*, vol. 62, no. 1, pp. 151–158, Jan. 2016.

[16] H. Zhou, M. Schwartz, A. Jiang, and J. Bruck, "Systematic error-correcting codes for rank modulation," *IEEE Trans. Inform. Theory*, vol. 61, no. 1, pp. 17–32, Jan. 2015.