Single-Deletion-Correcting Codes over Permutations

Ryan Gabrys*, **Eitan Yaakobi**‡, **Farzad Farnoud**†, **Frederic Sala***, **Jehoshua Bruck***, and **Lara Dolecek**†
*Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA 90095, USA
†Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125, USA
‡Computer Science Department, Technion - Israel Institute of Technology, Haifa 32000, Israel
{rgabrys, fredsala, dolecek}@ee.ucla.edu, {farnoud, bruck}@caltech.edu, yaakobi@cs.technion.ac.il

***Abstract*—Motivated by the rank modulation scheme for flash memories, we consider an information representation system with relative values (permutations) and study codes for correcting deletions. In contrast to the case of a deletion in a regular (with absolute values) representation system, a deletion in this new paradigm results in a new permutation over the remaining symbols. For example, the deletion of 3 (or 2) from $(1,3,2,4)$ yields $(1,2,3)$; while the deletion of 1 yields $(2,1,3)$. Codes for correcting deletions in permutations were studied by Levenshtein under a different model, however, he considered absolute values where the deletions are missing symbols.**

We study the single deletion relative-values model and prove that a code can correct a single deletion if and only if it can correct a single insertion. Using the concept of a signature of a permutation, we construct single-deletion correcting codes and prove that they are asymptotically optimal with respect to an upper bound that we derive. Finally, we describe an efficient decoding algorithm.

I. Introduction

Channel codes play an important role in the rapid growth of Flash memories. In order to overcome the complicated task of exactly programming each flash memory cell to its designated level, the novel framework of ***rank modulation codes*** was proposed in [3]. Under this scheme, the cells form a permutation which is derived from the relative amount of charge stored in the cells. Most of the error-correcting codes designed for rank modulation schemes have focused on constructing codes under different error metrics, such as the Kendall's τ distance, [6], [11] and the Ulam distance [1].

In this paper, we consider a related problem, and focus on the setup where cells can be deleted. Deletions in permutations can be modeled in two ways. Suppose the permutation $(1,5,4,3,2)$ is stored and the symbol 3 is deleted. For the first model, we assume the other symbols are not affected, and the received vector, which is no longer a permutation, is $(1,5,4,2)$. For the second model, all symbols greater than 3 are decreased by one, so the received vector is the permutation $(1,4,3,2)$. One important difference between these two models is the resulting size of the deletion ball. In the first model, every deleted symbol results in a different vector so that the size of the deletion ball is the same, regardless of the stored permutation. In contrast, the deletion ball size in the second model depends on the permutation; for example, the outcome of every deleted symbol in the permutation $(1,2,3,4,5)$ is the permutation $(1,2,3,4)$, so the deletion ball consists of only this single permutation. The first model was first studied by Levenshtein [5].

To the best of our knowledge, no prior work has been done using the second model. In this paper, we design codes for the second model where both the stored vector and the received vector are permutations. The first model of insertions and deletions is studied in our parallel work in [2].

The deletion model studied here has an analogue for insertions. Suppose that the symbol 3 is inserted into the fourth position of the permutation $(1,5,4,3,2)$. Then, the resulting permutation is $(1,6,5,3,4,2)$ where all the symbols in the original permutation of value greater than or equal to 3 were incremented by one. Although the purpose of this work is to study deletions, we show that there exists a duality between insertions and deletions.

The remainder of the paper is organized as follows. In Section II, we formally define the single deletion and insertion models studied in the paper. We prove some basic properties when deletions and insertions occur. These properties will be useful in proving that a code over permutations can correct a single deletion if and only if it can correct a single insertion. We proceed in Section III to give an asymptotic upper bound on the maximal size of a code over permutations that can correct a single deletion. This will be done in a similar fashion to the equivalent upper bound given by Levenshtein for deletions in binary vectors [4]. Our main contribution in the paper is given in Section IV. We present a construction of codes over permutations capable of correcting a single deletion. The correctness of this construction will be verified by providing a decoder for our code and proving its correctness. We lastly show in this section that our construction is asymptotically optimal. Due to the lack of space, some of the proofs in the paper are omitted.

II. Definitions and Preliminaries

In this section, we establish the notation that will be used throughout the paper. Afterwards, we consider some properties of codes that correct a single deletion.

The set of permutations of length n will be referred to as $\mathbb{S}_n$ and we refer to the set $\{1,2,\ldots,n\}$ as $[n]$. Let us first formally define the models for a single deletion and a single insertion for permutations.

Definition 1. *We say that a permutation $\pi = (\pi_1,\ldots,\pi_n) \in \mathbb{S}_n$ experiences a **single deletion** of the symbol π_j at position $j \in [n]$, resulting in the permutation $\pi_{\downarrow,\pi_j} = (\pi'_1,\ldots,\pi'_{n-1}) \in \mathbb{S}_{n-1}$ if the following holds for all $i \in [n-1]$:*

1) *For every $i < j$, if $\pi_i < \pi_j$ then $\pi'_i = \pi_i$.*
2) *For every $i < j$, if $\pi_i > \pi_j$ then $\pi'_i = \pi_i - 1$.*
3) *For every $i \geqslant j$, if $\pi_{i+1} < \pi_j$ then $\pi'_i = \pi_{i+1}$.*
4) *For every $i \geqslant j$, if $\pi_{i+1} > \pi_j$ then $\pi'_i = \pi_{i+1} - 1$.*

Definition 2. *We say that a permutation $\pi = (\pi_1,\ldots,\pi_n)$ experiences a **single insertion** of the symbol $s \in [n+1]$ at position $j \in [n+1]$, resulting in the permutation $\pi^{\uparrow,s,j} = (\pi'_1,\ldots,\pi'_{n+1}) \in \mathbb{S}_{n+1}$ if the following holds for all $i \in [n+1]$:*

1) *For every $i < j$, if $\pi_i < s$ then $\pi'_i = \pi_i$.*

2) *For every $i < j$, if $\pi_i \geqslant s$ then $\pi'_i = \pi_i + 1$.*
3) *If $i = j$, $\pi'_i = s$.*
4) *For every $i > j$, if $\pi_i < s$ then $\pi'_i = \pi_{i-1}$.*
5) *For every $i > j$, if $\pi_i \geqslant s$ then $\pi'_i = \pi_{i-1} + 1$.*

For a permutation $\pi \in \mathbb{S}_n$, let $\mathcal{B}_D(\pi)$ be the set of all permutations given that a single deletion occurred to π, i.e., $\mathcal{B}_D(\pi) = \{\pi_{\downarrow,s} \ : \ s \in [n]\}$. We say that a code $\mathcal{C}$ is a ***single-deletion-correcting code*** if for any $\pi, \sigma \in \mathcal{C}$, $\mathcal{B}_D(\pi) \cap \mathcal{B}_D(\sigma) = \emptyset$. Similarly, $\mathcal{B}_I(\pi)$ is the set of all permutations given that a single insertion occurred to π, that is, $\mathcal{B}_I(\pi) = \{\pi^{\uparrow,s,j} \ : \ s, j \in [n+1]\}$. We say that a code $\mathcal{C}$ is a ***single-insertion-correcting code*** if for any $\pi, \sigma \in \mathcal{C}$, $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) = \emptyset$.

We first begin with a few basic properties that are consequences of Definitions 1 and 2.

Claim 1. *Let $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$. Then, for any $j \in [n]$, $\pi = (\pi_{\downarrow,\pi_j})^{\uparrow,\pi_j,j}$. Similarly for any $s, j \in [n+1]$ $\pi = (\pi^{\uparrow,s,j})_{\downarrow,s}$.*

Claim 2. *Let $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$, $s, t \in [n]$ and suppose that $s < t$. Then $(\pi_{\downarrow,s})_{\downarrow,t-1} = (\pi_{\downarrow,t})_{\downarrow,s}$.*

Claim 3. *Let $\pi \in \mathbb{S}_n$ and $j, k, s, t \in [n+1]$ such that $s \leqslant t$. If $j < k$, then $(\pi^{\uparrow,s,j})^{\uparrow,t+1,k+1} = (\pi^{\uparrow,t,k})^{\uparrow,s,j}$. If $j > k$, then $(\pi^{\uparrow,s,j})^{\uparrow,t+1,k} = (\pi^{\uparrow,t,k})^{\uparrow,s,j+1}$. If $j = k$ and $s < t$, then $(\pi^{\uparrow,s,j})^{\uparrow,t+1,j} = (\pi^{\uparrow,t,j})^{\uparrow,s,j+1}$.*

The last three claims will be useful in showing that, similarly to the traditional setup of deletions and insertions in vectors [4], there is also a duality between insertions and deletions in permutations.

Lemma 1. *For any $\pi, \sigma \in \mathbb{S}_n$, $\mathcal{B}_D(\pi) \cap \mathcal{B}_D(\sigma) \neq \emptyset$ if and only if $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$.*

Proof: We first prove that if $\mathcal{B}_D(\pi) \cap \mathcal{B}_D(\sigma) \neq \emptyset$ then $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$. Let $\tau \in \mathcal{B}_D(\pi) \cap \mathcal{B}_D(\sigma)$ so there exist $j, k \in [n]$ such that $\pi_{\downarrow,\pi_j} = \sigma_{\downarrow,\sigma_k} = \tau$. Notice that from Claim 1, $\pi = \tau^{\uparrow,\pi_j,j}$ and $\sigma = \tau^{\uparrow,\sigma_k,k}$. Assume without loss of generality that $\pi_j \leqslant \sigma_k$. We first consider the case where $j < k$. From Claim 3, we conclude that

$$\pi^{\uparrow,\sigma_k+1,k+1} = (\tau^{\uparrow,\pi_j,j})^{\uparrow,\sigma_k+1,k+1} = (\tau^{\uparrow,\sigma_k,k})^{\uparrow,\pi_j,j} = \sigma^{\uparrow,\pi_j,j},$$

and thus $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$. Suppose $j > k$ where, as before, $\pi_j \leqslant \sigma_k$. From Claim 3, we have, $(\tau^{\uparrow,\pi_j,j})^{\uparrow,\sigma_k+1,k} = (\tau^{\uparrow,\sigma_k,k})^{\uparrow,\pi_j,j+1}$ and so $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$ in this case as well. Now suppose $j = k$. First notice that if $j = k$ and $\pi_j = \sigma_k$, then $\pi = \sigma$ and so the result trivially holds. Therefore we assume $\pi_j < \sigma_k$. Then from Claim 3, we have $(\tau^{\uparrow,\pi_j,j})^{\uparrow,\sigma_k+1,j} = (\tau^{\uparrow,\sigma_k,j})^{\uparrow,\pi_j,j+1}$ and again $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$.

Now we prove that if $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$ then $\mathcal{B}_D(\pi) \cap \mathcal{B}_D(\sigma) \neq \emptyset$. Let $\theta \in \mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma)$ and thus there exists $j, k, s, t \in [n+1]$ such that

$$\pi^{\uparrow,s,j} = \sigma^{\uparrow,t,k} = \theta = (\theta_1, \dots, \theta_{n+1}).$$

Recall that from Claim 1, we have $\pi = \theta_{\downarrow,s}$ and $\sigma = \theta_{\downarrow,t}$. Notice that if $s = t$, then $j = k$ by Definition 2. Furthermore, if $j = k$ and $s = t$ then $\pi = \sigma$ and the result is straightforward. Suppose, without loss of generality, that $s < t$. From Claim 2,

$$\pi_{\downarrow,t-1} = (\theta_{\downarrow,s})_{\downarrow,t-1} = (\theta_{\downarrow,t})_{\downarrow,s} = \sigma_{\downarrow,s}.$$

By the assumption $s < t$, we have $s, (t-1) \in [n]$, and thus we showed that $\mathcal{B}_D(\pi) \cap \mathcal{B}_D(\sigma) \neq \emptyset$, as required. ■

The following corollary follows directly from Lemma 1.

Corollary 1. *A code $\mathcal{C} \subseteq \mathbb{S}_n$ is a single-deletion-correcting code if and only if it is a single-insertion-correcting code.*

The following definition will be useful for characterizing permutations in the next section.

Definition 3. *For a permutation $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$, a **consecutive run** is a substring of maximal length in π that contains consecutively valued symbols, increasing or decreasing.*

For example, if $\pi = (1, 5, 4, 3, 2) \in \mathbb{S}_5$, then π has 2 consecutive runs: (1) and $(5, 4, 3, 2)$.

III. An Upper Bound on the Cardinality of Single-Deletion-Correcting Codes

The goal of this section is to derive an upper bound on the maximum size of a single-deletion-correcting code. We refer to the maximum size of such codes over $\mathbb{S}_n$ by $A(n)$ and our upper bound on $A(n)$ will be given in Theorem 1.

We first note that the size of the deletion ball $\mathcal{B}_D(\pi)$ depends on the permutation π. However, as will be shown in the following lemma, the size of the deletion ball for a permutation π can be solely characterized as a function of the number of consecutive runs in π. For a vector, $\boldsymbol{v} = (v_1, \dots, v_n)$ and two integers i_1, i_2, we denote by $\boldsymbol{v}_{[i_1,i_2]}$ the vector $\boldsymbol{v}_{[i_1,i_2]} = (v_{i_1}, \dots, v_{i_2})$. As a formality, if $i_2 < i_1$, then $\boldsymbol{v}_{[i_1,i_2]}$ is the empty string.

Lemma 2. *Let $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$ and suppose the symbols π_j and π_k ($j, k \in [n]$) belong to the same consecutive run in π. Then, $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_k}$.*

Proof: We denote $\sigma = \pi_{\downarrow,\pi_j}$ and $\theta = \pi_{\downarrow,\pi_k}$ and assume without loss of generality that $j < k$. Let $(\pi_j, \pi_{j+1}, \dots, \pi_k)$ be a substring of the consecutive run shared by the symbols π_j, π_k. Assume that $\pi_k > \pi_j$, while the opposite case is proven similarly.

Since π_j and π_k are in the same consecutive run in π, then from Definition 1, $\sigma_{[1,j-1]} = \theta_{[1,j-1]}$ and $\sigma_{[k,n-1]} = \theta_{[k,n-1]}$. Since $(\pi_j, \dots, \pi_k)$ is a substring of an increasing consecutive run, then $\pi_{j+1} = \pi_j + 1, \pi_{j+2} = \pi_{j+1} + 1, \dots, \pi_k = \pi_{k-1} + 1$. If π_k is deleted from $(\pi_j, \dots, \pi_k)$ then the resulting substring is $(\pi_j, \dots, \pi_{k-1})$. If π_j is deleted from $(\pi_j, \pi_{j+1}, \dots, \pi_k)$, then the resulting substring is $(\pi_{j+1} - 1, \pi_{j+2} - 1, \dots, \pi_k - 1) = (\pi_j, \dots, \pi_{k-1})$, and therefore $\sigma_{[j,k-1]} = \theta_{[j,k-1]}$. Thus, we conclude that $\sigma = \theta$. ■

The converse of Lemma 2 is proven next.

Lemma 3. *Let $\pi = (\pi_1, \dots, \pi_n) \in \mathbb{S}_n$, $j, k \in [n]$. If $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_k}$ then π_j and π_k belong to the same consecutive run.*

Proof: Assume without loss of generality, that $j < k$. Let us denote

$$\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_k} = \pi' = (\pi'_1, \dots, \pi'_{n-1}). \tag{1}$$

The proof will be by induction on $(k - j)$. For the base case, we prove that the result holds when $(k - j) = 1$. Since $\pi' = \pi_{\downarrow,\pi_j}$, we have that $\pi'_j = \pi_{j+1} - 1$ if $\pi_j < \pi_{j+1}$ or $\pi'_j =$

π_{j+1} if $\pi_j > \pi_{j+1}$. Similarly, since $\pi' = \pi_{\downarrow,\pi_k}$, we have that $\pi'_j = \pi_j$ if $\pi_j < \pi_k$ or $\pi'_j = \pi_j - 1$ if $\pi_j > \pi_k$. Thus, since $\pi \in \mathbb{S}_n$, we have that either $\pi'_j = \pi_j = \pi_{j+1} - 1$ if $\pi_j < \pi_k$ or $\pi'_j = \pi_j - 1 = \pi_{j+1}$ otherwise. In either case, (π_j, π_{j+1}) is a substring of a consecutive run.

Suppose that for all $(k-j) < m$, if $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_k}$ then π_j and π_k belong to the same consecutive run and consider the case where $(k-j) = m$ and $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_k}$. First, we show that if $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_k}$, then $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_{j+1}}$. Since assumption (1) holds, then as before either 1) $\pi'_j = \pi_{j+1}$, or 2) $\pi'_j = \pi_{j+1} - 1$. Using the same logic, we can conclude that (π_j, π_{j+1}) is a substring of a consecutive run and from Lemma 2, it follows that $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_{j+1}} = \pi_{\downarrow,\pi_k}$.

Let $\ell = j+1$. Since $(k-\ell) < m$, we can use the inductive hypothesis to conclude that since $\pi_{\downarrow,\pi_k} = \pi_{\downarrow,\pi_\ell}$, then π_k, π_ℓ are in the same consecutive run. Since π_j, π_ℓ are in the same consecutive run and π_ℓ, π_k are in the same consecutive run, it follows that π_j, π_k are in the same consecutive run and the proof is complete. ∎

For a permutation $\pi \in \mathbb{S}_n$, we denote by $R(\pi)$ the number of consecutive runs in π. The following is a corollary of Lemmas 2 and 3.

Corollary 2. *For all $\pi \in \mathbb{S}_n$, $|\mathcal{B}_D(\pi)| = R(\pi)$.*

We introduce some terminology that will be used for the derivation of the upper bound on $A(n)$. For positive integers n, r where $r < n$, we define

$$F(n,r) = \binom{n-1}{r-1} \cdot 2^{\min\{r,n-r\}} \cdot r!. \qquad (2)$$

To simplify the notation, we assume that n is a power of two so that the floors and ceilings can be dropped for convenience. We also assume that all log functions are base 2.

Lemma 4. *The number of permutations from $\mathbb{S}_n$ with r ($1 \leqslant r \leqslant n$) consecutive runs is at most $F(n,r)$.*

Proof: Consider the set of permutations from $\mathbb{S}_n$ that contain r consecutive runs. We proceed by over-counting this quantity. We first partition the elements from $[n]$ into r consecutive runs. This is equivalent to computing the number of solutions to the problem $\sum_{j=1}^{r} t_j = n$, where $t_j \geqslant 1$ and each t_j is an integer. There are $\binom{n-1}{r-1}$ such solutions.

If $r \leqslant \frac{n}{2}$ then there can be at most r consecutive runs of length greater than one. Each consecutive run can be either increasing or decreasing and so there are at most 2^r ways to re-arrange the numbers within each consecutive run. If $r > \frac{n}{2}$ then there are at most $n-r$ consecutive runs of length greater than one. In this case there are 2^{n-r} ways to re-arrange the numbers within each consecutive run. Then, if we permute each (block of symbols that constitute each) consecutive run we have at most

$$\binom{n-1}{r-1} \cdot 2^{\min\{r,n-r\}} \cdot r!$$

permutations in $\mathbb{S}_n$ with r consecutive runs. ∎

We need the following claim and lemma in order to prove Theorem 1.

Claim 4. *For $2 \leqslant r \leqslant n - \log(n)$, $F(n, r-1) \leqslant F(n,r)$.*

Lemma 5. *The number of permutations in $\mathbb{S}_n$ with at most $n - \log(n)$ consecutive runs is at most $\frac{n!(n-\log(n))^2}{(\log(n))!}$.*

Proof: From Claim 4, the maximum number of permutations in $\mathbb{S}_n$ with fewer than $n - \log(n)$ consecutive runs is at most $\sum_{r=1}^{n-\log(n)} F(n,r) \leqslant \sum_{r=1}^{n-\log(n)} F(n, n-\log(n))$. Substituting the expression for $F(n, n-\log(n))$ from (2) gives that there are at most

$$(n-\log(n)) \cdot \binom{n-1}{n-\log(n)-1} \cdot 2^{\log(n)} \cdot (n-\log(n))!$$
$$= \frac{n!(n-\log(n))^2}{(\log(n))!}$$

permutations in $\mathbb{S}_n$ with at most $n - \log(n)$ consecutive runs. ∎

Using a similar approach as in [4], we provide an upper bound for the maximum size of a single-deletion-correcting code.

Theorem 1. *For any $0 < \epsilon < 1$ there exists an N_ϵ such that for all $n \geqslant N_\epsilon$, $A(n) \leqslant \frac{n!}{n(n-\log(n))}(1+\epsilon)$.*

Proof: Suppose $\mathcal{C}$ is a single-deletion-correcting code over $\mathbb{S}_n$. Let $\mathbb{S}'_n = \{\pi' \in \mathbb{S}_n : |\mathcal{B}_D(\pi')| > n - \log(n)\}$. We first consider an upper bound on $|\mathcal{C} \cap \mathbb{S}'_n|$. Since the sets $\mathcal{B}_D(\pi') \subseteq \mathbb{S}_{n-1}$, for all $\pi' \in \mathcal{C} \cap \mathbb{S}'_n$ are disjoint and $|\mathcal{B}_D(\pi')| > n - \log(n)$ we get,

$$|\mathcal{C} \cap \mathbb{S}'_n| \leqslant \frac{(n-1)!}{n-\log(n)}.$$

Let $\mathbb{S}''_n = \{\pi'' \in \mathbb{S}_n : |\mathcal{B}_D(\pi'')| \leqslant n - \log(n)\}$. Clearly, $|\mathcal{C} \cap \mathbb{S}''_n| \leqslant |\mathbb{S}''_n|$ and from Lemma 5, $|\mathbb{S}''_n| \leqslant \frac{n!(n-\log(n))^2}{(\log(n))!}$. Thus, we have

$$|\mathcal{C}| = |\mathcal{C} \cap \mathbb{S}'_n| + |\mathcal{C} \cap \mathbb{S}''_n| \leqslant \frac{(n-1)!}{n-\log(n)} + \frac{n!(n-\log(n))^2}{(\log(n))!}$$
$$= \frac{n!}{n(n-\log(n))}\left(1 + \frac{n(n-\log(n))^3}{(\log(n))!}\right).$$

Since this upper bound holds for any single-deletion-correcting code $\mathcal{C}$, we conclude that $A(n) \leqslant \frac{n!}{n(n-\log(n))}\left(1 + \frac{n(n-\log(n))^3}{(\log(n))!}\right)$. Lastly, since $\lim_{n\to\infty} \frac{n(n-\log(n))^3}{(\log(n))!} = 0$, there exists an N_ϵ such that for all $n \geqslant N_\epsilon$, we have $A(n) \leqslant \frac{n!}{n(n-\log(n))}(1+\epsilon)$. ∎

IV. Code Construction

In this section, we first consider some properties regarding deletions in permutations. Afterwards, an asymptotically optimal construction of single-deletion-correcting codes is provided. Lastly, we prove the correctness of the construction and discuss a decoding algorithm.

A. Permutation Matrices, Signatures, and Runs

A ***permutation matrix*** is a square binary matrix such that in each row and each column there is precisely one 1. For a permutation $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$, the ***permutation matrix of*** π is a permutation matrix $M = f(\pi)$ such that $M_{ij} = 1$ if $j = \pi_i$. Notice that if M is an $n \times n$ permutation matrix, then there exists an unique permutation π such that $M = f(\pi)$. Hence, the mapping f is invertible. We denote its inverse by f^{-1} and write $f^{-1}(f(\pi)) = \pi$. The next claim follows from Definition 1.

Claim 5. *For $\pi \in \mathbb{S}_n$, the matrix $f(\pi_{\downarrow,\pi_j})$ is the result of removing row j and column π_j from $f(\pi)$. Furthermore, if row j and column π_j are removed from $f(\pi)$ thus resulting in M', then $\pi_{\downarrow,\pi_j} = f^{-1}(M')$.*

We next use a mapping, referred to as the ***signature***, that will be useful in our code construction. This mapping was also used in [9] in the context of single-deletion-correcting codes over non-binary vectors. Let $m > 2$ be an integer. For $\boldsymbol{y} \in [m]^n$, define the binary length-$(n-1)$ signature $\alpha(\boldsymbol{y}) = (\alpha(\boldsymbol{y})_1, \ldots, \alpha(\boldsymbol{y})_{n-1})$ as follows. For $1 \leqslant i \leqslant n-1$,

$$\alpha(\boldsymbol{y})_i = \begin{cases} 1, & \text{if } y_{i+1} \geqslant y_i. \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

For a permutation $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$, recall from [7] that the inverse permutation $\pi^{-1} = (\pi_1^{-1}, \ldots, \pi_n^{-1}) \in \mathbb{S}_n$ is such that for $i \in [n]$, π_i^{-1} is the location of the element i in π. It is straightforward to verify that the permutation of the transpose of $f(\pi)$ corresponds to the inverse permutation for π. In other words, we have $f(\pi^{-1}) = (f(\pi))^T$. For shorthand, we will refer to the signature of the inverse permutation as the ***inverse signature***. The next example illustrates a signature and an inverse signature.

Example 1. Suppose $\pi = (1,3,5,4,2) \in \mathbb{S}_5$. Then $\alpha(\pi) = (1,1,0,0)$. Furthermore, $\pi^{-1} = (1,5,2,4,3) \in \mathbb{S}_5$ and $\alpha(\pi^{-1}) = (1,0,1,0)$.

We are now ready to prove the following lemma.

Lemma 6. *For $\pi \in \mathbb{S}_n$, we have $(\pi_{\downarrow,\pi_j})^{-1} = (\pi^{-1})_{\downarrow,j}$.*

Proof: From Claim 5, if the symbol π_j (where $j \in [n]$) is deleted from π, then the permutation matrix $f(\pi_{\downarrow,\pi_j})$ is the result of removing row j and column π_j from $f(\pi)$. Alternatively, we can obtain $(f(\pi_{\downarrow,\pi_j}))^T$ by removing row π_j and column j from $(f(\pi))^T$. From Claim 5, removing row π_j and column j from $(f(\pi))^T$ corresponds to the deletion of symbol j from π^{-1} since $f^{-1}((f(\pi))^T) = \pi^{-1}$. ∎

In the following, if the input to $\mathcal{B}_D$ is a binary vector $\boldsymbol{x} \in GF(2)^n$, then the output of $\mathcal{B}_D(\boldsymbol{x})$ is the set of all possible vectors obtainable by deleting one bit from $\boldsymbol{x}$.

If a symbol from a permutation π is deleted, then a symbol from its signature, $\alpha(\pi)$, is deleted as well. That is, $\alpha(\pi_{\downarrow,x}) \in \mathcal{B}_D(\alpha(\pi))$. Hence, an immediate consequence of Lemma 6 is the following.

Corollary 3. *Let $\pi \in \mathbb{S}_n$ and $j \in [n]$. Then $\alpha(\pi_{\downarrow,\pi_j}) \in \mathcal{B}_D(\alpha(\pi))$ and $\alpha((\pi_{\downarrow,\pi_j})^{-1}) \in \mathcal{B}_D(\alpha(\pi^{-1}))$.*

A binary code will be called a ***binary single-deletion-correcting code*** if it can correct any single-bit deletion. As a result of Corollary 3, in the next subsection we will leverage binary single-deletion-correcting codes which will be invoked over the signature and inverse signature of permutations in order to construct single-deletion-correcting codes for permutations.

In the rest of this subsection, we define runs in permutations and binary sequences and present a claim that will be useful in the next subsection.

Definition 4. *For a binary sequence s, a **run** is a maximal substring of s that is all-zero or all-one. For a permutation π, an **ascending run** is a maximal substring of π whose values are increasing and a **descending run** is a maximal substring of π whose values are decreasing. A substring of π is a **run** if it is an ascending or a descending run.*

Example 2. Continuing with the setup from Example 1, let $\pi = (1,3,5,4,2) \in \mathbb{S}_5$ and $\alpha(\pi) = (1,1,0,0)$. Notice that the signature of a permutation reflects the structure of the runs in the permutation. For example, the first three symbols in π comprise an ascending run and so the first two symbols of $\alpha(\pi)$ are ones.

The following claim is straightforward to verify.

Claim 6. *Consider a permutation $\pi \in \mathbb{S}_n$ and its signature $\alpha(\pi)$. There is an ascending run starting at position i and ending at position $j+1$ in π if and only if there is a run of ones starting at position i and ending at position j in $\alpha(\pi)$. Furthermore, if $\sigma = \pi_{\downarrow,x}$, where x is at position k in π with $i \leqslant k \leqslant j+1$, then $\alpha(\pi)$ can be converted to $\alpha(\sigma)$ by deleting a 1 from this run. A similar statement holds for descending runs and deletion of 0s.*

B. Code construction

Let us first review the binary single-deletion-correcting code we will use in our construction. Namely, the code from [10], known as a ***Varshamov-Tennegolts code*** (VT code), is defined as follows. For a positive integer $n > 2$ and $a \in \mathbb{Z}_{n+1}$, $\mathcal{C}_a^n$ is the code $\mathcal{C}_a^n = \{\boldsymbol{x} \in GF(2)^n : \sum_{i=1}^n ix_i \equiv a \mod n+1\}$.

Lemma 7. *(cf. [4]) For any integer $n > 2$ and $a \in \mathbb{Z}_{n+1}$, the code $\mathcal{C}_a^n$ is a binary single-deletion-correcting code.*

We are now ready to present our code construction of single-deletion-correcting codes over permutations.

Construction 1. Given an integer $n > 2$ and $a_1, a_2 \in \mathbb{Z}_n$, let

$$\mathcal{C}_{a_1,a_2}^n = \{\pi \in \mathbb{S}_n : \alpha(\pi) \in \mathcal{C}_{a_1}^{n-1}, \alpha(\pi^{-1}) \in \mathcal{C}_{a_2}^{n-1}\}. \tag{4}$$

We first comment on the cardinality of the codes $\mathcal{C}_{a_1,a_2}^n$. Recall from the previous section that $A(n)$ represents the maximum cardinality of a single-deletion-correcting code. Note that the codes $\mathcal{C}_{a_1,a_2}^n$ for $a_1, a_2 \in \mathbb{Z}_n$ partition the space $\mathbb{S}_n$ into n^2 mutually disjoint codes. Hence, if we denote by $F(n)$ for $n > 2$ the maximum cardinality of a code according to Construction 1, that is, $F(n) = \max_{a_1,a_2 \in \mathbb{Z}_n}\{|\mathcal{C}_{a_1,a_2}^n|\}$, then applying the pigeonhole principle gives the following result.

Corollary 4. *Construction 1 is asymptotically optimal, that is,*

$$\lim_{n\to\infty} \frac{F(n)}{A(n)} = 1.$$

C. Proof of Correctness of Construction 1

In this subsection, we prove the correctness of Construction 1. From our proof, a decoding algorithm for the codes can be easily derived. Due to lack of space, however, we omit an explicit presentation of the decoding algorithm.

For simplicity, for a permutation σ, we use $x \prec_\sigma y$ if and only if $\sigma^{-1}(x) < \sigma^{-1}(y)$.

Theorem 2. *For $n > 2$ and $a_1, a_2 \in \mathbb{Z}_n$, the code $\mathcal{C}_{a_1,a_2}^n$ is a single-deletion-correcting permutation code.*

Proof: Suppose that $\pi \in \mathcal{C}_{a_1,a_2}^n$ and that $\sigma = \pi_{\downarrow,x}$ for some $x \in [n]$. We show that π is uniquely identifiable from

σ. To do this, we first identify the runs in σ and σ^{-1} from which x was deleted and show that there is an unique way to increase the length of these runs by an insertion in a consistent way.

Let k denote the position of x in π, that is, we have $\pi = \sigma^{\uparrow,x,k}$. From the received word σ, we compute $\alpha(\sigma)$. Corollary 3 implies that $\alpha(\sigma) \in \mathcal{B}_D(\alpha(\pi))$. Using a decoder for a VT code, we find $\alpha(\pi)$ from $\alpha(\sigma)$ since there is a deletion that converts $\alpha(\pi)$ to $\alpha(\sigma)$. By comparing $\alpha(\pi)$ and $\alpha(\sigma)$, we find the i and j of Claim 6. Hence, $\pi_i, \pi_{i+1}, \ldots, \pi_{j+1}$ form a run in π. Without loss of generality, assume that this run is an ascending run, or equivalently, the deleted element in $\alpha(\pi)$ is a 1. Thus, by Claim 6, $\pi = \sigma^{\uparrow,x,k}$ such that

$$i \leqslant k \leqslant j+1, \tag{5}$$

$$\sigma_i < \sigma_{i+1} < \cdots < \sigma_j, \tag{6}$$

$$x \leqslant \sigma_m \text{ iff } k \leqslant m \text{ for } m \in \{i, \ldots, j\}. \tag{7}$$

By Lemma 6, we have $\sigma^{-1} = (\pi^{-1})_{\downarrow,k}$ and $\pi^{-1} = (\sigma^{-1})^{\uparrow,k,x}$. We thus may apply Claim 6 by substituting π with π^{-1}. Let the corresponding values of i and j of the claim be denoted by p and q, respectively, for this case. The claim implies that the substring $\pi^{-1}_p, \pi^{-1}_{p+1}, \ldots, k, \ldots, \pi^{-1}_q, \pi^{-1}_{q+1}$ is a run of π^{-1} and that $p \leqslant x \leqslant q+1$.

The values of p and q can be determined from $\alpha\left(\sigma^{-1}\right)$ as follows. By Claim 6 and using a decoder for a VT code, we find $\alpha\left(\pi^{-1}\right)$ from $\alpha\left(\sigma^{-1}\right)$ since there is a deletion that converts $\alpha\left(\pi^{-1}\right)$ to $\alpha\left(\sigma^{-1}\right)$. We then find p and q by comparing $\alpha\left(\pi^{-1}\right)$ and $\alpha\left(\sigma^{-1}\right)$.

There are two different cases depending on the deleted element of $\alpha\left(\pi^{-1}\right)$ being a 1 or a 0. Due to lack of space, here, we only consider the former case. Suppose that a 1 in a run of 1s in $\alpha\left(\pi^{-1}\right)$ is deleted. We have $\pi^{-1}_p < \pi^{-1}_{p+1} < \cdots < k < \cdots < \pi^{-1}_q < \pi^{-1}_{q+1}$ and thus

$$x \in \{p, p+1, \ldots, q+1\}, \tag{8}$$

$$p \prec_\sigma p+1 \prec_\sigma \cdots \prec_\sigma q, \tag{9}$$

$$k \leqslant \sigma^{-1}(y) \text{ iff } x \leqslant y \text{ for } y \in \{p, \ldots, q\}, \tag{10}$$

where $p \prec_\sigma p+1$ denotes that the symbol p appears before the symbol $p+1$ in the permutation σ. Let $A = \{\sigma_i, \ldots, \sigma_j\} \cap \{p, \ldots, q\}$. Suppose A is empty. Because $\sigma_i, \ldots, \sigma_j$ is an increasing run of σ and $p, p+1, \ldots, q$ in an increasing subsequence of σ, we have one of the following cases: a) $\sigma^{-1}(q) < i$; b) $\sigma^{-1}(p) > j$; or c) $\sigma^{-1}(z-1) < i$ and $\sigma^{-1}(z) > j$ for some $z \in \{p+1, \ldots, q\}$. Note that if cases a) and b) do not hold, then $q > p$ and so the set $\{p+1, \ldots, q\}$ used in case c) is nonempty.

We consider each case separately: a) From (5), we have $\sigma^{-1}(q) < k$, which using (10) implies that $x > q$. From (8), we find $x = q+1$. b) Similar to case a), from (5), (10), and (8), we have $x = p$. c) From (5) it follows that $\sigma^{-1}(z-1) < k \leqslant \sigma^{-1}(z)$. Using (10), we find that $x = z$. Hence, we can identify x if A is empty. Having identified x, we can find the unique position k in $\{i, \ldots, j+1\}$ that satisfies condition (7).

Now suppose A is nonempty and so $u = \min A$ and $v = \max A$ are well defined. From (6) and (9), it is not difficult to show that every integer between u and v is also in A, i.e.,

$$A = \{u, u+1, \ldots, v\},$$

and that the elements of A form a consecutive run in σ. Based on (5)–(10), it is straightforward (but tedious) to see that the set of possible values for x is exactly $\{u, u+1, \ldots, v, v+1\}$ and that $k = \sigma^{-1}(x)$ if $u \leqslant x \leqslant v$ and $k = \sigma^{-1}(v)+1$ if $x = v+1$. Furthermore, with the aforementioned values for x and k, the resulting permutation $\sigma^{\uparrow,x,k}$ is the same; it is a permutation in which the length of the consecutive run formed by the element of A is increased by 1. Thus π is determined uniquely. ∎

We illustrate the preceding proof by the following example.

Example 3. Consider the code $\mathcal{C}^8_{a_1,a_2}$, where $a_1 = 7$ and $a_2 = 0$. Suppose the stored codeword is $\pi = (7,4,5,6,8,2,1,3) \in \mathcal{C}^8_{a_1,a_2}$, and the retrieved permutation is $\sigma = \pi_{\downarrow,5} = (6,4,5,7,2,1,3)$. The decoder is given σ, a_1, and a_2, and from these it computes

$$\alpha(\sigma) = (0,1,1,0,0,1),$$
$$\alpha(\pi) = (0,1,1,1,0,0,1),$$
$$\alpha(\sigma^{-1}) = (0,1,0,1,0,1),$$
$$\alpha(\pi^{-1}) = (0,1,0,1,1,0,1).$$

We thus have $i = 2$, $j = 4$, $p = 4$, and $q = 5$. Furthermore, $A = \{4,5,7\} \cap \{4,5\} = \{4,5\}$, implying that $x \in \{4,5,6\}$. Hence, the possible pairs of values for (x,k) are $(4,2)$, $(5,3)$, and $(6,4)$. Note that $\pi = \sigma^{\uparrow,4,2} = \sigma^{\uparrow,5,3} = \sigma^{\uparrow,6,4}$, and so the decoding is successful.

ACKNOWLEDGMENT

The work of E.Y., F.F. , and J.B. was supported in part by Intellectual Ventures, an NSF grant CIF-1218005, and the U.S.-Israel Binational Science Foundation, Jerusalem, Israel, under Grant No. 2010075. The work of R.G., F.S., and L.D. was supported in part by the SMART scholarship, NSF GRFP, and NSF grants CCF-1029030 and CCF-1150212.

REFERENCES

[1] F. Farnoud, V. Skachek, and O. Milenkovic, "Error-correction in flash memories via codes in the Ulam metric," *IEEE Trans. Information Theory*, vol. 59, no. 5, pp. 3003–3020, May 2013.

[2] R. Gabrys, E. Yaakobi, F. Farnoud, and J. Bruck, "Codes correcting erasures and deletions for rank modulation," *Proc. IEEE International Symposium on Information Theory*, Honolulu, HI, June-July 2014.

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

[4] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet physics doklady*, vol. 10, p. 707–710, 1966.

[5] V. I. Levenshtein, "On perfect codes in deletion and insertion metric," *Discretnaya Mathematika*, vol. 3, no. 1, pp. 3–20, 1991.

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

[7] K.H. Rosen, *Discrete Mathematics and its Applications*, Chapman and Hall, 2004.

[8] F. Sala, R. Gabrys, and L. Dolecek, "Deletions in Multipermutations," *IEEE International Symposium on Information Theory*, Honolulu, HI, June-July 2014.

[9] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion (Corresp.)," *IEEE Trans. Information Theory*, vol. 30, no. 5, pp. 766–769, Sep. 1984.

[10] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors (in Russian)," *Avtomatika i Telemekhanika*, vol. 6, no. 2, 1965 (trans: *Automation and Remote Contr.*, vol. 26, pp. 286–290).

[11] H. Zhou, A. Jiang, and J. Bruck, "Systematic error-correcting codes for rank modulation," *Proc. IEEE International Symposium on Information Theory*, pp. 2978–2982, Cambridge, MA, July 2012.