Codes Correcting Erasures and Deletions for Rank Modulation

Ryan Gabrys, *Member, IEEE*, Eitan Yaakobi, *Member, IEEE*, Farzad Farnoud, *Member, IEEE*, Frederic Sala, *Student Member, IEEE*, Jehoshua Bruck, *Fellow, IEEE*, and Lara Dolecek, *Senior Member, IEEE*

***Abstract*— Error-correcting codes for permutations have received considerable attention in the past few years, especially in applications of the rank modulation scheme for flash memories. While codes over several metrics have been studied, such as the Kendall τ, Ulam, and Hamming distances, no recent research has been carried out for erasures and deletions over permutations. In rank modulation, flash memory cells represent a permutation, which is induced by their relative charge levels. We explore problems that arise when some of the cells are either erased or deleted. In each case, we study how these erasures and deletions affect the information carried by the remaining cells. In particular, we study models that are symbol-invariant, where unaffected elements do not change their corresponding values from those in the original permutation, or permutation-invariant, where the remaining symbols are modified to form a new permutation with fewer elements. Our main approach in tackling these problems is to build upon the existing works of error-correcting codes and leverage them in order to construct codes in each model of deletions and erasures. The codes we develop are in certain cases asymptotically optimal, while in other cases, such as for codes in the Ulam distance, improve upon the state of the art results.**

***Index Terms*— Coding theory, flash memories, rank modulation codes, permutation codes, deletion codes.**

I. Introduction

FLASH memory is the storage medium of choice in portable consumer electronic applications, and high performance solid-state drives (SSDs) are being introduced into mobile computing, enterprise storage, data warehousing, and data-intensive computing systems. The rapid increase in the capacity of flash memories makes them attractive in these applications. However, the capacity increase of these technologies presents major challenges in the areas of device reliability and endurance. These challenges can be overcome through innovative coding and data handling techniques.

Flash memories are comprised of blocks of cells. Each cell can store one or more bits of information. For example, a typical block in current flash memories contains about one million cells. The number of levels per cell is a fixed value that varies between 2 and 16 in current technologies. One of the main challenges in flash memories is to exactly program each cell to its target level. In order to overcome this difficulty, ***rank modulation codes*** were proposed and studied in [11]. In this setup, the information is carried by the relative values between the cells rather than by their absolute levels. Thus, every group of cells induces a permutation, which is derived by the ranking of the level of each cell in the group. Shortly after the work in [11], several papers explored codes which correct errors in permutations specifically for the rank modulation scheme; see e.g., [1], [4], [8], [12], [26]. These works include different metrics such as Kendall τ, Ulam, and Hamming distances. However, none of these papers studied the setup where cells in the permutations are either deleted or erased. The goal of the present work is to establish the foundations and present results for these faulty mechanisms.

The paradigms we explore are derived from a hardware implementation of the modulation process in rank modulation codes [14], [15], [20]. In particular, while reading and comparing between the levels of the cells, it may happen that some cells are corrupted and thus cannot be read correctly. This leads to erasures in the case where the locations of the corrupted cells are known, and deletions otherwise. Furthermore, the missing information about the corrupted cells may or may not affect the values of the other cells. We provide more detail below.

Assume that a permutation π is stored in the flash memory cells. Depending on the read mechanism, while reading π, some of the cells could be interpreted as erasures, or, alternatively, as deletions. In traditional models, each symbol is affected by such errors independently; however, when using permutations, erasures and deletions can potentially affect the other symbols as well. To simplify our discussion, we assume for now that only one symbol is either erased or deleted. We consider four different models, which correspond to two choices of characteristics: first, ***erasure*** or ***deletion***:

Manuscript received June 2, 2015; accepted September 21, 2015. Date of publication October 26, 2015; date of current version December 18, 2015. This work was supported in part by the U.S.–Israel Binational Science Foundation, Jerusalem, Israel, under Grant 2010075, in part by the Smart Scholarship, in part by NSF under Grant CCF-1029030, Grant CCF-1150212, and Grant CIF-1218005, in part by the NISE Program at SSC Pacific, in part by NSF GRFP, and in part by Intellectual Ventures. This paper was presented at the IEEE International Symposium on Information Theory in 2014 [6] and [7].

R. Gabrys is with Spawar Systems Center, San Diego, CA 92152 USA (e-mail: ryan.gabrys@navy.mil).

E. Yaakobi is with the Department of Computer Science, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: yaakobi@cs.technion.ac.il).

F. Farnoud and J. Bruck are with the Electrical Engineering Department, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: farnoud@caltech.edu; bruck@caltech.edu).

F. Sala and L. Dolecek are with the Electrical Engineering Department, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: fredsala@ucla.edu; doleck@ee.ucla.edu).

Communicated by M. Schwartz, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2015.2493147

whether the location of the lost cell is known or not known, and, second, ***symbol invariance*** or ***permutation invariance***: whether the other symbols cannot be changed or may be changed as a result of the erasure/deletion. We enumerate these models as follows and provide an example of each. Suppose that the stored permutation is $(5, 3, 2, 4, 1)$ and the third symbol (that is, 2) is affected by one of the four types of errors. For each of the four error models, we provide the resulting output.

1) Symbol-invariant erasure (SIE): $(5, 3, ?, 4, 1)$,
2) Permutation-invariant erasure (PIE): $(4, 2, ?, 3, 1)$,
3) Symbol-invariant deletion (SID): $(5, 3, 4, 1)$,
4) Permutation-invariant deletion (PID): $(4, 2, 3, 1)$.

Note the reasoning underlying the "symbol-invariant" and "permutation-invariant" terminology: In the case of symbol invariance, only the affected symbol is changed, but, as a result, the remaining symbols no longer form a permutation over $\{1, 2, \ldots, n'\}$, where n' is the number of available elements. Meanwhile, in the case of permutation invariance, the other symbols are affected so that the (non-erased) symbols continue to form a permutation.

Our main contribution in the first three models of erasures and deletions is to first identify an existing distance metric for permutations that will provide codes in the corresponding model. We then introduce suitable codes in this model. In particular, we show that codes based upon the Hamming distance can be used to correct symbol-invariant erasures (SIEs). We also show that the models of permutation-invariant erasures (PIEs) and symbol-invariant deletions (SIDs) are equivalent and that codes designed for the Ulam distance can be used in these setups. For the fourth model, we introduce an asymptotically optimal single PID-correcting code construction.

To the best of our knowledge, the research on codes combating errors defined using the proposed models is very limited. We could only specify the paper by Levenshtein [19] which falls under the SID model and the follow up works in [24] and [25]. These works contained explicit constructions only for the case of a single SID. In [17], deletion-correcting codes capable of correcting a large number of deletions were studied. In particular, the work in [17] considers codes of length n capable of correcting t deletions where the quantity $n - t$ is a constant. In this work, we focus on SID correcting codes capable of correcting a small number of deletions (greater than one).

The rest of the paper is organized as follows. In Section II, we formally define the erasure and deletion models studied in the paper and review the Hamming and Ulam distances for permutations. In Section III, we show how to use existing codes in these three distance metrics in order to construct codes for the proposed erasure and deletion models. In Section IV, we give a construction of codes in the Ulam distance, corresponding to the second and third models, which improves upon the best known codes. Afterwards, we focus on the fourth and most challenging model, dealing with permutation-invariant deletions. Section V considers some simple properties of PIDs. In Section VI, we derive an upper bound on the maximum cardinality of a code which can correct a single PID. In Section VII we give a construction for such a code which is asymptotically optimal.

II. Definitions and Preliminaries

Let $\mathbb{S}_n$ denote the set of all $n!$ permutations of n elements, chosen to be $\{1, 2, \ldots, n\}$. We use the vector notation to denote a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$. Given some permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n) \in \mathbb{S}_n$, its inverse permutation is $\pi^{-1} = (\pi_1^{-1}, \pi_2^{-1}, \ldots, \pi_n^{-1})$, where π_i^{-1} is the location of the element i in π. For example, for $\pi = (6, 1, 3, 2, 5, 4)$ we have $\pi^{-1} = (2, 4, 3, 6, 5, 1)$. The set $\{1, \ldots, n\}$ is denoted by $[n]$, and for two positive integers $a < b$, the set $\{a, \ldots, b\}$ is denoted by $[a, b]$. We denote the concatenation of two sequences $\boldsymbol{x}, \boldsymbol{y}$ by $(\boldsymbol{x}, \boldsymbol{y})$.

We now formally define the four models of symbol-invariant/permutation-invariant erasures and deletions. For a permutation $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$, and a set of positions $I \subseteq [n]$, $\pi(I)$ is the set $\pi(I) = \{\pi_i : i \in I\}$. For an integer $a \in [n]$ and a subset $J \subseteq [n]$, the integer $a(J) \in [n]$ is defined as $a(J) = a - |\{i \in J : i < a\}|$.

As an analogy, if a is the finishing position of a runner in a race, and those runners finishing in positions given by set J are disqualified, $a(J)$ is the new finishing position of the original runner. For example, assume $\pi = (6, 1, 3, 2, 5, 4)$ and $I = \{1, 4, 5\}$, so that $\pi(I) = \{6, 2, 5\}$. We take $a = \pi_3$ and $J = \pi(I)$ in the above definition yielding $\pi_3(\pi(I)) = 3(\pi(\{1, 4, 5\})) = 3(\{6, 2, 5\}) = 3 - 1 = 2$.

Definition 1: Assume that $\pi = (\pi_1, \ldots, \pi_n)$ is a permutation in $\mathbb{S}_n$ and $I \subseteq \{1, \ldots, n\}$ is a positions set of size t ($0 \leqslant t \leqslant n$). We consider the following four models of erasures and deletions:

1) ***Symbol-Invariant Erasure (SIE)****: The permutation π suffered t symbol-invariant erasures in the positions set I, resulting in the vector $\pi' = (\pi'_1, \ldots, \pi'_n)$, if*
   a) *for all $i \in I$, $\pi'_i = ?$, and*
   b) *for all $i \in [n] \setminus I$, $\pi'_i = \pi_i$.*
2) ***Permutation-Invariant Erasure (PIE)****: The permutation π suffered t permutation-invariant erasures in the positions set I, resulting in the vector $\pi' = (\pi'_1, \ldots, \pi'_n)$, if*
   a) *for all $i \in I$, $\pi'_i = ?$, and*
   b) *for all $i \in [n] \setminus I$, $\pi'_i = \pi_i(\pi(I))$.*
3) ***Symbol-Invariant Deletion (SID)****: The permutation π suffered t symbol-invariant deletions in the positions set I, resulting in the vector $\pi' = (\pi'_1, \ldots, \pi'_{n-t})$, if*[1] *for all $k \in [n] \setminus I$ and $i = k(I)$, $\pi'_i = \pi_k$.*
4) ***Permutation-Invariant Deletion (PID)****: The permutation π suffered t permutation-invariant deletions in the positions set I, resulting in the permutation $\pi' = (\pi'_1, \ldots, \pi'_{n-t}) \in \mathbb{S}_{n-t}$, if for all $k \in [n] \setminus I$ and $i = k(I)$, $\pi'_i = \pi_k(\pi(I))$.*

A code $\mathcal{C} \subseteq \mathbb{S}_n$ is called a ***t-SIE/PIE/SID/PID-correcting code*** *if it can correct up to t SIEs/PIEs/SIDs/PIDs, respectively.*

The next example illustrates these four models.

[1]Note that for $k_1 \neq k_2$, we have $k_1(I) \neq k_2(I)$.

Example 2: Let $\pi = (6, 1, 3, 2, 5, 4) \in \mathbb{S}_6$ and $I = \{1, 4, 5\}$, so that there are three errors. Then, the following vectors are the received ones for each model:

1) Symbol-Invariant Erasures: $\pi' = (?, 1, 3, ?, ?, 4)$.
2) Permutation-Invariant Erasures: $\pi' = (?, 1, 2, ?, ?, 3)$.
3) Symbol-Invariant Deletions: $\pi' = (1, 3, 4)$.
4) Permutation-Invariant Deletions: $\pi' = (1, 2, 3)$.

Note that, of the four models, the errors in the first model in Definition 1 are the easiest to handle and those in the last model are the hardest with respect to the amount of information that is lost. We continue by reviewing the Hamming and Ulam distances.

The ***Hamming distance*** between two permutations $\pi, \sigma \in \mathbb{S}_n$, denoted by $d_H(\pi, \sigma)$, is defined as the number of positions for which π and σ differ. For two permutations $\pi, \sigma \in \mathbb{S}_n$, let $\ell(\pi, \sigma)$ be the length of a longest common subsequence of π and σ. The ***Ulam distance*** between π and σ is defined as $d_U(\pi, \sigma) = n - \ell(\pi, \sigma)$, see also [2], [5], [8].

Example 3: Let $\pi = (4, 3, 1, 2, 5)$ and $\sigma = (4, 3, 5, 1, 2)$. Then $d_H(\pi, \sigma) = 3$ and $d_U(\pi, \sigma) = 1$.

The minimum distance of a code, according to any metric, is the minimum distance between any two distinct codewords in the code. For a code $\mathcal{C} \subseteq \mathbb{S}_n$, its minimum Hamming and Ulam distances are denoted by $d_H(\mathcal{C})$ and $d_U(\mathcal{C})$, respectively.

III. Codes From Existing Codes in Other Metrics

In this section we present codes for the erasure and deletion models defined in Section II. In particular, we show how existing codes in the Hamming and Ulam distances can be used for these models.

Let us start with symbol-invariant erasures. First notice that if only a single SIE occurred then it is immediate to complete the missing symbol since its position is marked by '?' and its value is the missing element in the permutation. Thus, the code $\mathcal{C} = \mathbb{S}_n$ is a single SIE-correcting code. For more than a single SIE, we show in the next theorem that codes in the Hamming distance are necessary and sufficient.

Theorem 4: A code $\mathcal{C} \subseteq \mathbb{S}_n$ is a t-SIE-correcting code if and only if $d_H(\mathcal{C}) \geqslant t + 1$.

Proof: We show that if $d_H(\mathcal{C}) \geqslant t + 1$ then $\mathcal{C}$ is a t-SIE-correcting code by proving the contrapositive. Assume that $\mathcal{C}$ has minimum distance $d_H(\mathcal{C}) \geqslant t + 1$ but is not a t-SIE-correcting code. Thus, there are two permutations $\pi, \sigma \in \mathcal{C}$ and two positions sets $I_1, I_2 \subseteq [n]$, each of size at most t, such that if π' is the result of SIEs in the positions set I_1 in π and σ' is the result of SIEs in the positions set I_2 in σ, then $\pi' = \sigma'$. First notice that $I_1 = I_2$. Otherwise, assume without loss of generality that there exists $i \in I_1 \backslash I_2$, so we get $\pi'_i = ?$ and $\sigma'_i \neq ?$, which is a contradiction. Hence, we can denote $I_1 = I_2 = I$ and $|I| \leqslant t$. Then we get that for every $i \in [n] \setminus I$, $\pi_i = \pi'_i = \sigma'_i = \sigma_i$, and thus $d_H(\pi, \sigma) \leqslant |I| \leqslant t$. However, we assumed that $d_H(\mathcal{C}) \geqslant t + 1$, a contradiction.

Now, for the second direction, assume that $\mathcal{C}$ is a t-SIE-correcting code. We view $\mathcal{C}$ as a code in $[n]^n$ so that its minimum Hamming distance has to be at least $t + 1$, as desired. ■

We next move to the models of PIEs and SIDs. Note that with PIEs the locations are known but the values are not known, while with SIDs the values are known while the locations are unknown.

The next lemma establishes the surprising fact that the two models are equivalent. Let $\pi = (\pi_1, \ldots, \pi_n)$ be a permutation in $\mathbb{S}_n$. Suppose π suffered t PIEs in the positions set I. We denote the result by the length-n vector $\pi' = (\pi'_1, \ldots, \pi'_n)$. With a slight abuse of notation, we denote by $\pi'^{-1}_{PIE} = (\pi'^{-1}_1, \ldots, \pi'^{-1}_{n-t})$ the length-$(n - t)$ vector which specifies the locations of the $(n - t)$ non-erased symbols in π'. Similarly, if π suffered t SIDs in the positions set I, let the length-$(n - t)$ vector $\pi' = (\pi'_1, \ldots, \pi'_{n-t})$ be the resulting sequence. We denote by $\pi'^{-1}_{SID} = (\pi'^{-1}_1, \ldots, \pi'^{-1}_n)$ the length-n vector which specifies the locations of the symbols $[n]$ in π', where a '?' indicates that the symbol does not appear in π'.

The preceding definitions allow us to use the concept of permutation inverse for sequences that may not be permutations in $\mathbb{S}_n$. For example, if $\pi = (4, 3, 5, 1, 2)$ and two PIEs occurred in the second and fourth positions, then $\pi' = (2, ?, 3, ?, 1)$ and $\pi'^{-1}_{PIE} = (5, 1, 3)$. If two SIDs occurred in the second and fourth positions of π, then $\pi' = (4, 5, 2)$, and $\pi'^{-1}_{SID} = (?, 3, ?, 1, 2)$.

Lemma 5: If π' is the result of t PIEs over a permutation $\pi \in \mathbb{S}_n$, then π'^{-1}_{PIE} is the result of t SIDs over the permutation π^{-1}. If π' is the result of t SIDs over a permutation $\pi \in \mathbb{S}_n$, then π'^{-1}_{SID} is the result of t PIEs over the permutation π^{-1}.

Proof: For the first claim, let π' be the result of t PIEs in π given by the positions set I. We note that for every $k \in [n]\backslash I$, $\pi'_k = \pi_k(\pi(I))$ and hence the location of $i \in [n - t]$ in π' is the location of the symbol k_i in π such that $i = k_i(\pi(I))$. Therefore, for $\pi'^{-1}_{PIE} = (\pi'^{-1}_1, \ldots, \pi'^{-1}_{n-t})$, we have that for $i \in [n - t]$, $\pi'^{-1}_i = \pi^{-1}_{k_i}$, where $i = k_i(\pi(I))$. Therefore, π'^{-1}_{PIE} is the vector resulting from π^{-1} suffering t SIDs in the positions set $\pi(I)$.

To prove the second claim, assume that π suffered t SIDs in the positions set I, resulting in the vector $\pi' = (\pi'_1, \ldots, \pi'_{n-t})$, such that for $i \in [n - t]$, $\pi'_i = \pi_{k_i}$, where $i = k_i(I)$. Let $\pi'^{-1}_{SID} = (\pi'^{-1}_1, \ldots, \pi'^{-1}_n)$ be as defined previously. That is, for $i \in \pi(I)$ have $\pi'^{-1}_i = ?$ and for $i \notin \pi(I)$, $\pi'^{-1}_i = \pi^{-1}_i(I)$, or, since $\pi^{-1}(\pi(I)) = I$, $\pi'^{-1}_i = \pi^{-1}_i(\pi^{-1}(\pi(I)))$. Hence, the vector π'^{-1}_{SID} equals the vector resulting from π^{-1} suffering t PIEs in the positions set $\pi(I)$. ■

As a result of Lemma 5 we conclude the following corollary.

Corollary 6: There exists a t-PIE-correcting code of cardinality M if and only if there exists a t-SID-correcting code of cardinality M.

To complete this discussion we only need to consider codes in one of these two models. We will show how codes in the Ulam distance are necessary and sufficient for codes in the SID model. The following property was stated by Levenshtein [19] but we prove it here for the completeness of the results in the paper.

Theorem 7: A code $\mathcal{C} \subseteq \mathbb{S}_n$ is a t-SID-correcting code if and only if $d_U(\mathcal{C}) \geqslant t + 1$.

Proof: We first show that if $\mathcal{C}$ is a t-SID-correcting code then $d_U(\mathcal{C}) \geqslant t + 1$. Assume on the contrary that $\mathcal{C}$ is t-SID-correcting but that $d_U(\mathcal{C}) \leqslant t$. Let $\pi, \sigma \in \mathcal{C}$ be such that $d_U(\pi,\sigma) = t' \leqslant t$. Hence, π and σ have a common subsequence of length $\ell(\pi,\sigma) = n - d_U(\pi,\sigma) = n - t' \geqslant n - t$. Let S be the set of symbols in this common subsequence. Let I_π be the positions set of the symbols $[n]\backslash S$ in π and similarly let I_σ be the positions set of the symbols $[n] \setminus S$ in σ. Then, if π' is the result of t' SIDs in π in the positions set I_π and σ' is the result of t' SIDs in σ in the positions set I_σ, with $t' \leqslant t$, we get that $\pi' = \sigma'$. Therefore, the code $\mathcal{C}$ is not a t-SID-correcting code, which is a contradiction.

Next we prove the other direction. Assume that $\mathcal{C}$ is not a t-SID-correcting code but that $d_U(\mathcal{C}) \geqslant t+1$. Thus, there exist two distinct permutations $\pi, \sigma \in \mathcal{C}$ and two positions sets $I_1, I_2 \subseteq [n]$, each of size at most t, such that if π' is the result of SIDs in the positions set I_1 in π and σ' is the result of SIDs in the positions set I_2 in σ, then $\pi' = \sigma'$. First we have that $|I_1| = |I_2|$ and since $\pi' = \sigma'$, π and σ have a common subsequence of length $n - |I_1| \geqslant n - t$. Therefore, $d_U(\pi,\sigma) \leqslant n - (n-t) = t$, a contradiction. ∎

In the next section, we introduce novel constructions of codes correcting PIEs and SIDs.

IV. New Codes Correcting PIEs/SIDs

The purpose of this section is to present novel codes correcting permutation-invariant erasures or symbol-invariant deletions. Recall that according to Lemma 5 and Theorem 7, correcting either class of errors is equivalent to correcting errors in the Ulam metric. For this reason, our proposed codes will target this metric. We begin by introducing some notation, tools, and codes in subsection IV-A. We build on these results to introduce the main construction in subsection IV-B. In subsection IV-C, we comment on how the proposed construction improves upon the state of the art codes for the Ulam metric given in [8] in the case where the Ulam distance is at most 6. For ease of presentation, in the remainder in this section we use the terms deletion and PID (erasure and SIE, respectively) interchangeably.

A. Notation and Location-Correcting Auxiliary Code

For a sequence $\boldsymbol{x}$ with elements from the set $[n] \cup \{?\}$ and for $s \in [n]$, let $D(\boldsymbol{x}, s)$ be the result of removing all occurrences[2] of the symbol s in $\boldsymbol{x}$. If s is not contained in $\boldsymbol{x}$, then $D(\boldsymbol{x}, s) = \boldsymbol{x}$. More generally, for $\mathcal{I} \subset [n]$, $D(\boldsymbol{x}, \mathcal{I})$ is the result of removing all symbols from $\mathcal{I}$ in $\boldsymbol{x}$. (Note that we use cursive for set $\mathcal{I}$ containing the symbols to be deleted and regular case for the set I containing the locations of symbols to be deleted.) For $\mathcal{I}' \subset [n]$, let $\sigma = Er(\boldsymbol{x}, \mathcal{I}')$ be the result of substituting each of the elements of $\mathcal{I}'$ in $\boldsymbol{x}$ with the symbol '?'.

Next, we introduce the first of several auxiliary codes needed for our new construction. In order to distinguish between the different auxiliary codes, we label them with a superscript based on a code attribute. As the following code can determine the locations of deletions (given that these locations satisfy a certain constraint to be explained later), we refer to it by $\mathcal{C}^L$. For the remainder of this section, we assume that n, ℓ are positive integers where $n > \ell$ and $\ell | n$. We define

$$\mathcal{C}^L_{\ell,n} = \{\pi \in \mathbb{S}_n : i \in [n], \pi_i \equiv i - 1 (\text{mod} \ell)\}.$$

For example, $\pi = (3,4,2,6,1,5) \in \mathcal{C}^L_{3,6}$. For $\sigma = (\sigma_1, \dots, \sigma_n) \in \mathbb{S}_n$, the vector $\sigma(\text{mod}\ell) = (\sigma'_1, \dots, \sigma'_n)$ is defined by $\sigma'_i = \sigma_i (\text{mod} \ell)$ for $i \in [n]$. Thus, for any codeword $\pi \in \mathcal{C}^L_{\ell,n}$, the vector $\pi(\text{mod}\ell)$ is a periodic sequence of length n and period ℓ where each period is $(0, \dots, \ell-1)$.

We now describe a decoding map $\mathcal{D}^L_{\ell,n}$ and an associated algorithm for the code $\mathcal{C}^L_{\ell,n}$. This algorithm inserts '?' symbols into σ so that in the resulting vector, for each i, if the element at position i is not a '?', then it equals $i-1$ modulo ℓ. Suppose that $\pi \in \mathcal{C}^L_{\ell,n}$ is the stored codeword and σ is the retrieved word, where $\sigma = D(\pi, \mathcal{I})$ with $\mathcal{I} \subset [n]$, $|\mathcal{I}| = t < n$. The input to $\mathcal{D}^L_{\ell,n}$ is σ.

1) Initialize $j = 0$ and let $\sigma^{(1)} = (\sigma, 0)$.
2) Let $j = j + 1$.
3) Let i be the smallest element in $[n - t + j]$ such that $\sigma^{(j)}_i \not\equiv i - 1 (\text{mod}\ell)$ and $\sigma^{(j)}_i \neq ?$. If no such symbol exists, go to step 5).
4) Let $\sigma^{(j+1)}$ be the result of inserting the symbol '?' into $\sigma^{(j)}$ at position i_j. Go to step 2).
5) Define $\hat{\pi} = (\sigma^{(j)}_1, \dots, \sigma^{(j)}_{n-t+j-1})$.

We illustrate the decoding map $\mathcal{D}^L_{\ell,n}$ with the following example.

Example 8: Suppose $n = 12$ and $\ell = 3$ and let

$$\pi = (3,7,5,9,1,8,6,4,2,12,10,11) \in \mathcal{C}^L_{3,12},$$

and let $\mathcal{I} = \{1,3,8,11\}$. Then,

$$\sigma = D(\pi, \mathcal{I}) = (7,5,9,6,4,2,12,10).$$

Then, according to step 1), we have

$$\sigma^{(1)} = (7,5,9,6,4,2,12,10,0).$$

Next, continuing the process, we have that

$$\begin{aligned}
\sigma^{(2)} &= (?,7,5,9,6,4,2,12,10,0),\\
\sigma^{(3)} &= (?,7,5,9,?,6,4,2,12,10,0),\\
\sigma^{(4)} &= (?,7,5,9,?,?,6,4,2,12,10,0),\\
\sigma^{(5)} &= (?,7,5,9,?,?,6,4,2,12,10,?,0),
\end{aligned}$$

and, finally,

$$\hat{\pi} = (?,7,5,9,?,?,6,4,2,12,10,?).$$

Let $Consec(\pi, \mathcal{I}, \ell)$ be equal to 1 if there exists a subset of ℓ symbols from $\mathcal{I}$ that appear consecutively in π. Otherwise, $Consec(\pi, \mathcal{I}, \ell) = 0$. If $Consec(\pi, \mathcal{I}, \ell) = 1$, then let $L_\ell(\pi, \mathcal{I})$ be the set of symbols that constitute the first occurrence of a subset of ℓ symbols from $\mathcal{I}$ that appear consecutively in π. If $Consec(\pi, \mathcal{I}, \ell) = 0$, then let $L_\ell(\pi, \mathcal{I}) = \emptyset$. For example, let π be as in Example 8, then, $Consec(\pi, \{3,7,5,8,6,4\}, 3) = 1$ and $L_3(\pi, \{3,7,5,8,6,4\}) = \{3,5,7\}$.

Let $\boldsymbol{x}$ be a sequence with symbols from $[n] \cup \{?\}$ and let $|\boldsymbol{x}|$ denote the length of $\boldsymbol{x}$. The following claim follows directly

[2]Note that here, $\boldsymbol{x}$ is not a permutation over S_n but rather a sequence over $[n] \cup \{?\}$, so that there may be several occurrences of a particular symbol.

from the definition of the map $\mathcal{D}^L_{\ell,n}$. Recall that the operation $Er(\pi, \mathcal{I})$ substitutes each of the elements of $\mathcal{I}$ present in π with the symbol '?'.

Claim 9: For positive integers n, ℓ, suppose $\pi \in \mathcal{C}^L_{\ell,n}$, $\mathcal{I} \subset [n]$, and $\sigma = D(\pi, \mathcal{I})$. If $Consec(\pi, \mathcal{I}, \ell) = 0$, then $\mathcal{D}^L_{\ell,n}(\sigma) = Er(\pi, \mathcal{I})$. Otherwise, $|\mathcal{D}^L_{\ell,n}(\sigma)| \leqslant n-\ell < |\pi| = n$.

Claim 9, in other words, states that if a certain constraint (that is, longest maximal substring deleted from π has length less than ℓ) is met, then the output of $\mathcal{D}^L_{\ell,n}$ is π with symbols of $\mathcal{I}$ replaced by '?'s. Otherwise, the algorithm returns a sequence with length shorter than the length of π.

The decoding map can only fail to place elements in their proper positions if consecutive elements corresponding to entire periods of length ℓ are deleted. As a result, we note that the output sequence must have length $n - k\ell$ for some $k \geqslant 0$. In particular, if the number of deletions is smaller than 2ℓ, the output sequence must have length n if $Consec(\pi, \mathcal{I}, \ell) = 0$ and length $n - \ell$ if $Consec(\pi, \mathcal{I}, \ell) = 1$.

This concludes the analysis of the location-correcting code $\mathcal{C}^L_{\ell,n}$. We will rely on $\mathcal{C}^L_{\ell,n}$ as a building block for our novel code construction in the following.

B. Code Construction

We begin by giving two more auxiliary codes. The first code, which we call $\mathcal{C}^E_{n'} \subseteq \mathbb{S}_{n'}$, can correct a single SID. The superscript E in $\mathcal{C}^E_{n'}$ refers to the fact that $\mathcal{C}^E_{n'}$ has a prescribed edit or deletion distance. In particular, $\mathcal{C}^E_{n'}$ can correct a single insertion or deletion. A construction of such a code and an associated decoder $\mathcal{D}^E_{n'}$ were given in [19].

The decoder $\mathcal{D}^E_{n'}$ operates as follows. Recall that the operation $D(\pi, s)$ removes the element s from π. Suppose $\sigma = D(\pi, s)$, where $\pi \in \mathcal{C}^E_{n'}$. The output of $\mathcal{D}^E_{n'}(\sigma)$ is the ordered triplet (s, s', s'') where s is the symbol deleted from π to obtain σ, s' is the symbol immediately before s in π, and s'' is the symbol immediately after s in π. If s is the final symbol in π, then $s'' = 0$ and similarly if s is the first symbol in π, then $s' = 0$.

The second code $\mathcal{C}^H_{d,n} \subseteq \mathbb{S}_n$ has minimum Hamming distance d. The structure of the code $\mathcal{C}^H_{d,n} \subseteq \mathbb{S}_n$ will be described in more detail in Section IV-C. For the next subsection, we only require that $\mathcal{C}^H_{d,n}$ has minimum Hamming distance d.

Using the three auxiliary codes, we now present a code capable of correcting $2\ell - 1$ SIDs. The idea is to constrain our codewords (or certain functions of our codewords) to simultaneously be members of each of the codes $\mathcal{C}^L_{\ell,n}$, $\mathcal{C}^E_{n/\ell}$, and $\mathcal{C}^H_{3\ell-2,n}$. Below, we explain the intuition behind this idea, and follow up by providing more detail later on in this section.

Our goal is to correct up to $2\ell - 1$ SIDs. We first identify the locations of the deletions. If fewer than ℓ of the deletions are consecutive, we can correctly identify their locations with $\mathcal{D}^L_{\ell,n}$. If ℓ or more of the deletions are consecutive, we look to identify the location of the first element of the consecutive deletions with $\mathcal{D}^E_{n/\ell}$. Doing so allows us to fix all the remaining deletion locations. Once these locations have been determined, we can treat the deletions as erasures, which can be corrected by the decoder of the third (Hamming metric) code.

For a permutation $\pi \in \mathbb{S}_n$ and an integer $0 \leqslant i \leqslant \ell - 1$, let $R_\ell(\pi, i)$ be the subsequence of π that only contains the symbols from the set $\{s \in [n] : s \equiv i (\text{mod} \ell)\}$. (The notation R refers to the fact that we restrict π to elements congruent to i modulo ℓ.) For shorthand, denote $\{1, 2, \ldots, n\}$ as $[n]$. For positive integers m, n, k, and $\boldsymbol{x} \in [n]^m$, let $(\boldsymbol{x} - k)/\ell = (y_1, \ldots, y_m)$ be such that for $i \in [m]$, $y_i = \lfloor (x_i - k)/\ell \rfloor$. We introduce the code $\mathcal{C}^U_{2\ell,n}$, where U refers to the Ulam metric.

Construction 10: For positive integers n, ℓ where $n > \ell$ and $\ell | n$, let $\mathcal{C}^U_{2\ell,n} \subseteq \mathbb{S}_n$ be the code consisting of all permutations $\pi \in \mathbb{S}_n$ that satisfy the following conditions:

1) $\pi \in \mathcal{C}^L_{\ell,n}$,
2) $(R_\ell(\pi, i) - i)/\ell \in \mathcal{C}^E_{n/\ell}$ for $0 \leqslant i \leqslant \ell - 1$, and
3) $\pi \in \mathcal{C}^H_{3\ell-2,n}$.

We show that the code $\mathcal{C}^U_{2\ell,n}$ can recover from up to any $m = 2\ell - 1$ SIDs (or, equivalently, that it has Ulam distance at least 2ℓ.) Suppose $\sigma = D(\pi, \mathcal{I}) \in [n]^{n-m}$ where $\mathcal{I} \subseteq [n]$, $|\mathcal{I}| = m$, and $\pi \in \mathcal{C}^U_{2\ell,n}$.

We detail the decoding procedure. We begin by computing $Consec(\pi, \mathcal{I}, \ell)$. According to Claim 9, $\mathcal{D}^L_{\ell,n}$ can determine the locations of all the deletions unless $Consec(\pi, \mathcal{I}, \ell) = 1$. Recall that if $Consec(\pi, \mathcal{I}, \ell) = 1$, then a substring of length at least ℓ is deleted from π and $\mathcal{D}^L_{\ell,n}$ returns a vector of length less than n. In this case, the decoder $\mathcal{D}^E_{n/\ell}$ is used to determine the location where the substring (of length at least ℓ) was deleted from π. The locations of any remaining deletions are discovered with a further application of $\mathcal{D}^L_{\ell,n}$. Now, in either case, the deletion locations are known, so that we may treat them as erasures. Using $\mathcal{D}^H_{3\ell-2,n}$, the values of the deleted symbols are recovered.

Next, we formally define the decoding procedure. We refer to the decoding map for $\mathcal{C}^U_{2\ell,n}$ as $\mathcal{D}^U_{2\ell,n} : [n]^{n-m} \to \mathbb{S}_n$. The input to the map is σ and the output is an estimate $\hat{\pi}$ of the codeword $\pi \in \mathcal{C}^U_{2\ell,n}$. The decoder is presented in Algorithm 1, where we use the following notation. Let $s', s'' \in [0, n]$ and $\sigma \in [n]^{n-m}$ be a vector. We define $\sigma[s', s'']$ as follows. If $s' = 0$, then $\sigma[s', s'']$ is the set of symbols that appear before the symbol s'' in σ. Similarly, if $s'' = 0$, then $\sigma[s', s'']$ is the set of symbols that appear after the symbol s' in σ. Lastly, if $s', s'' > 0$, $\sigma[s', s'']$ refers to the set of symbols in σ that are between the symbols s', s'', exclusive.

Theorem 11: The code $\mathcal{C}^U_{2\ell,n}$ has Ulam distance at least 2ℓ.

Proof: Let π be a permutation in $\mathcal{C}^U_{2\ell,n}$ affected by deletions of elements given by the set $\mathcal{I}$. The result is proven by showing that if $\mathcal{I}$ has size no more than $2\ell - 1$, the output $\hat{\pi}$ of Algorithm 1 equals π. In this proof, let $B = L_\ell(\pi, \mathcal{I})$. In step 1, we compute $\mathcal{D}^L_{\ell,n}(\sigma)$. The output of step 2 has two possibilities.

In the first case, suppose that at step 2 we have $|\sigma^{(1)}| = n$. Then, from Claim 9, we have $\sigma^{(1)} = Er(\pi, \mathcal{I})$ and $B = \emptyset$. In step 10, since π belongs to a code with minimum Hamming distance $3\ell - 2$, and $|\mathcal{I}| = 2\ell - 1 \leqslant 3\ell - 2$, the values of the deleted symbols can be recovered. Thus, when $|\sigma^{(1)}| = n$, we have that $\hat{\pi} = \pi$.

Algorithm 1 $\mathcal{D}^U_{2\ell,n} : [n]^{n-m} \to \mathbb{S}_n$

```
input : the retrieved permutation $\sigma = D(\pi, \mathcal{I})$
output: estimate $\hat{\pi}$ of $\pi$
1  $\sigma^{(1)} \longleftarrow \mathcal{D}^L_{\ell,n}(\sigma)$;
2  if $|\sigma^{(1)}| = n$ then
3      $\sigma^{(4)} \longleftarrow \sigma^{(1)}$;
4  else
5      $k \longleftarrow \min\{x \in \mathbb{Z}_\ell : |R_\ell(\sigma, x)| = \frac{n}{\ell} - 1\}$;
       $(s, s', s'') \longleftarrow \mathcal{D}^E_{n/\ell}((R_\ell(\sigma, k) - k)/\ell)$;
       $\mathcal{S} = \sigma[\ell \cdot s' + k, \ell \cdot s'' + k]$;
6      $\sigma^{(2)} \longleftarrow D(\sigma, \mathcal{S})$;
7      $\sigma^{(3)} \longleftarrow$ result of inserting $\ell \cdot s + k$ between $\ell \cdot s' + k$
       and $\ell \cdot s'' + k$ in $\sigma^{(2)}$;
8      $\sigma^{(4)} \longleftarrow \mathcal{D}^L_{\ell,n}(\sigma^{(3)})$;
9  end
10 $\hat{\pi} \longleftarrow \mathcal{D}^H_{3\ell-2,n}(\sigma^{(4)})$;
```

Next we have the case that $|\sigma^{(1)}| = n - \ell$, which, as seen from our previous discussion, is the only possibility if $|\sigma^{(1)}| \neq n$ and there are fewer than 2ℓ deletions. Since out of the (at most) $2\ell - 1$ deleted symbols, the ℓ symbols of B are consecutive in π, each element of B belongs to a distinct equivalence class modulo ℓ. There are up to $\ell - 1$ other deleted symbols so that, since there are ℓ equivalence classes modulo ℓ, there exists an equivalence class modulo ℓ with precisely one deleted symbol. Hence, there exists k such that $|R_\ell(\sigma, k)| = \frac{n}{\ell} - 1$ (the algorithm arbitrarily picks the smallest possible value for k in step 5). The deleted symbol from $R_\ell(\pi, k)$, that is the only deleted symbol that is equal to $k(\text{mod}\ell)$, is $\ell \cdot s + k$, where $(s, s', s'') = \mathcal{D}^E_{n/\ell}(\frac{R_\ell(\sigma,k)-k}{\ell})$. For simpliciy of presentation, in the algorithm and in this discussion we ignore the possibility that $s' = 0$ or $s'' = 0$; these cases can be handled similarly.

Since the ℓ symbols of B are consecutive in π, there is one element of B that is equal to $k(\text{mod}\ell)$. But since $\ell \cdot s + k$ is the only deleted element from π that is equal to k modulo ℓ, we have $\ell \cdot s + k \in B$. This in turn implies that the symbols of B are located between the symbol $\ell \cdot s' + k$ and the symbol $\ell \cdot s'' + k$ in π. So the size of the set $\mathcal{S}$ in step 5 is at most $(2\ell - 1) - \ell = \ell - 1$, where $2\ell - 1$ is the number of symbols between $\ell \cdot s' + k$ and $\ell \cdot s'' + k$ in π and ℓ is the number of symbols in B. Hence, $|\sigma^{(2)}| = n - |\mathcal{I}| - |\mathcal{S}| \leqslant n - (3\ell - 2)$.

In step 7 of the algorithm, $\ell \cdot s + k$ is inserted in its correct position. So now there are at most $3\ell - 3$ elements missing from $\sigma^{(3)}$ compared to π. In other words, there is a set $\mathcal{I}'$ such that $\sigma^{(3)} = D(\pi, \mathcal{I}')$, where $|\mathcal{I}'| \leqslant 3\ell - 3$. Since $\ell \cdot s + k \notin \mathcal{I}'$, we have $Consec(\pi, \mathcal{I}', \ell) = 0$. Hence, by Claim 9, $\sigma^{(4)} = Er(\pi, \mathcal{I}')$ at step 10. The decoder $\mathcal{D}^H_{3\ell-2,n}$ can recover π from $\sigma^{(4)}$ in step 12 since $|\mathcal{I}'| \leqslant 3\ell - 3$ and the minimum Hamming distance of the code $\mathcal{C}^U_{2\ell,n}$ is $3\ell - 2$. ■

We illustrate Algorithm 1 with the following example.

Example 12: Take $\pi = (9, 7, \mathbf{8}, \mathbf{6}, \mathbf{1}, 11, 3, \mathbf{10}, 2, 12, \mathbf{4}, 5) \in \mathcal{C}^U_{6,12}$ and $\ell = 3$. Suppose $\sigma = D(\pi, \mathcal{I}) = (9, 7, 11, 3, 2, 12, 5)$ (the elements that were deleted to produce $D(\pi, \mathcal{I})$ are labeled in π in bold) where $\mathcal{I} = \{1, 4, 6, 8, 10\}$. We will show that, if Algorithm 1 is invoked with σ as input, then $\hat{\pi} = \pi$.

At step 1, we have $\sigma^{(1)} = (9, 7, 11, 3, ?, 2, 12, ?, 5)$. Note that one modulo 3 period has not been recovered. Then, since $|\sigma^{(1)}| = 9 < 12$, we proceed to step 5. At step 5, $k = 0$ since $|R_3(\sigma, 0)| = |(9, 3, 12)| = 4 - 1$. Then at step 6, the input to the decoder $\mathcal{D}^E_4$ is $(3, 1, 4)$ and the output is $(s, s', s'') = (2, 3, 1)$. At step 7, the set $\mathcal{S} = \sigma[9, 3] = (7, 11)$. Notice that $|\mathcal{S}| = 2$ in this case. After the elements from the set $\mathcal{S}$ are removed from σ, at step 8 we have $\sigma^{(2)} = (9, 3, 2, 12, 5)$. At step 9, we insert the symbol 6 into $\sigma^{(2)}$ giving that $\sigma^{(3)} = (9, 6, 3, 2, 12, 5)$. At step 10, $\sigma^{(4)} = (9, ?, ?, 6, ?, ?, 3, ?, 2, 12, ?, 5)$. Since there are 6 total erasures and $\mathcal{D}^H_{7,12}$ is the decoder for a code with Hamming distance 7, the output of $\mathcal{D}^H_{7,12}$ at step 12 will be $\hat{\pi} = (9, 7, 8, 6, 1, 11, 3, 10, 2, 12, 4, 5) = \pi$ as desired.

In the following subsection, we consider the size of a code created according to Construction 10 and compare the cardinalities of our codes to those of the codes from [8].

C. The Cardinality of $\mathcal{C}^U_{2\ell,n}$

In this subsection, we consider the size of a code $\mathcal{C}^U_{2\ell,n}$. In order to do so, we consider in more details the constraints a permutation π must satisfy to be a codeword in $\mathcal{C}^U_{2\ell,n}$.

We first revisit a mapping, referred to as the ***signature***, which was used in [24] to construct 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 $1 \leqslant i \leqslant n - 1$,

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

The following is well-known.

Theorem 13 (See [19], [24]): For $a \in \mathbb{Z}_n$, the code

$$\mathcal{C}^a_n = \{\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n : \sum_{i=1}^{n-1} i\alpha(\pi)_i \equiv a \mod n\}$$

can correct a single SID.

In the following we assume ℓ is an integer and $n + 1$ an odd prime, where $\ell | n$. Recall that $\mathcal{C}^U_{2\ell,n} \subseteq \mathcal{C}^L_{\ell,n}$. We will introduce a set of constraints that will partition the set $\mathcal{C}^L_{\ell,n}$ into a number of subsets, each of which will be shown to satisfy the second and third properties required by the construction of $\mathcal{C}^U_{2\ell,n}$. Then, we will apply the pigeonhole principle to lower bound the maximal cardinality of $\mathcal{C}^U_{2\ell,n}$.

The constraints we introduce will partition the codewords $\pi = (\pi_1, \ldots, \pi_n) \in \mathcal{C}^L_{\ell,n}$ into $(n+1)^{3\ell-3} \left(\frac{n}{\ell}\right)^\ell$ sets. The constraints will be specified by the vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_{3\ell-3}, a_{3\ell-2}, \ldots, a_{4\ell-3})$ where for $1 \leqslant i \leqslant 3\ell - 3$, $a_i \in GF(n+1)$ and for $3\ell - 2 \leqslant i \leqslant 4\ell - 3$, $a_i \in \mathbb{Z}_{n/\ell}$. Specifically, the constraints we introduce are

$$a_1 = \sum_{i=1}^{n} \pi_i,$$
$$a_2 = \sum_{i=1}^{n} i \cdot \pi_i,$$
$$\vdots$$
$$a_{3\ell-3} = \sum_{i=1}^{n} i^{3\ell-4} \cdot \pi_i,$$
$$a_{3\ell-2} = \sum_{i=1}^{n/\ell-1} i \cdot \alpha\left(R_\ell(\pi, i)/\ell\right)_i,$$
$$\vdots$$
$$a_{4\ell-3} = \sum_{i=1}^{n/\ell-1} i \cdot \alpha\left((R_\ell(\pi, i) - (\ell-1))/\ell\right)_i,$$

where the operations in the first $3\ell - 3$ equations take place over the field $GF(n+1)$ and the operations in the last ℓ equations are over $\mathbb{Z}_{n/\ell}$. We refer to the set of sequences satisfying the constraints corresponding to vector $\boldsymbol{a}$ by $\mathcal{C}_n^{\boldsymbol{a}}$.

For any choice of $\boldsymbol{a}$, notice that the first $3\ell - 3$ equations written above generate a Vandermonde matrix which is full rank over $GF(n+1)$. Therefore, we have that each of the sets $\mathcal{C}_n^{\boldsymbol{a}}$ has Hamming distance at least $3\ell-2$. Furthermore, the final ℓ equations ensure that the sets $\mathcal{C}_n^{\boldsymbol{a}}$ can be decomposed into ℓ single deletion codes as required by the second condition in Construction 10. Therefore, each of the sets $\mathcal{C}_n^{\boldsymbol{a}}$ forms a code $\mathcal{C}_{2\ell,n}^{U}$.

Notice that we can actually remove the first constraint since if $\pi \in \mathbb{S}_n$, then $\sum_{i=1}^{n} \pi_i = 0$ in $GF(n+1)$. Now, observe that the cardinality of $\mathcal{C}_{\ell,n}^{L}$ is given by $((n/\ell)!)^\ell$. This follows from a simple counting argument: we have n/ℓ choices of elements for each of the terms in the first period, $n/\ell - 1$ choices for the elements in the second period, and so on. Next, we have a total of $(n+1)^{3\ell-4}(n/\ell)^\ell$ choices for vector $\boldsymbol{a}$. Then, applying the pigeonhole principle, there must exist some $\boldsymbol{a}$ such that the corresponding code $\mathcal{C}_n^{\boldsymbol{a}} = \mathcal{C}_{2\ell,n}^{U}$ has cardinality satisfying

$$|\mathcal{C}_{2\ell,n}^{U}| \geqslant \frac{((\frac{n}{\ell})!)^\ell}{(n+1)^{3\ell-4}\left(\frac{n}{\ell}\right)^\ell}.$$

We now compare the cardinality of our codes to those of the state of the art codes from [4] and [8]. Let $A_H(n,t)$ be the size of a code of length n over $\mathbb{S}_n$ with Hamming distance at least t. In [8], a construction was provided that produced codes of Ulam distance t, length n, and cardinality at least $A_{FSM}(n,t)$ where

$$A_{FSM}(n,t) = \prod_{i=1}^{k} A_H\left(\left\lceil \frac{n}{2^i} \right\rceil - 1, \frac{3t}{2}\right), \tag{2}$$

and $k = \lfloor \log_2 \frac{n}{3t/2+1} \rfloor$. In [4], the codes from (2) were improved upon and new codes of Ulam distance t, length n, and cardinality at least $A_{CV}(n,t)$ were derived such that

$$A_{CV}(n,t) = \sum_{j=0}^{s} \prod_{i=1}^{u_j} A_H\left(\left\lceil \frac{n - j(3t+1)}{2^i} \right\rceil - 1, \frac{3t}{2}\right), \tag{3}$$

where $s = \left\lfloor \frac{n}{3t+1} \right\rfloor$ and $u_j = \left\lfloor \log_2 \frac{n-j(3t+1)}{3t/2+1} \right\rfloor$ for $j \in [0, s]$.

Let $A_{new}(n,t) = \frac{(\frac{2n}{t}!)^{t/2}}{(n+1)^{3(t/2)-4}\left(\frac{2n}{t}\right)^{t/2}}$. In this section, we have provided a construction for a code with Ulam distance t and length n and showed that it must have at least $A_{new}(n,t)$ codewords. In the following lemma, we show that $A_{new}(n,t) > A_{FSM}(n,t)$ and $A_{new}(n,t) > A_{CV}(n,t)$ when n is large and $t \leqslant 6$. The proof can be found in Appendix A.

Lemma 14: For sufficiently large n such that $n+1$ is a prime, $t|2n$, and $t \leqslant 6$, $A_{New}(n,t) > A_{FSM}(n,t)$ and $A_{New}(n,t) > A_{CV}(n,t)$.

V. Basic Properties of PIDs

In this section, we consider some properties of permutation-invariant deletions. We first revisit the previously introduced model for PIDs. We also introduce the notion of permutation-invariant insertions (PIIs), which will be necessary in the sequel.

Recall from Definition 1 that a permutation π experiences t PIDs in the positions set $I \subseteq [n]$, resulting in the permutation $(\pi'_1, \ldots, \pi'_{n-t}) \in \mathbb{S}_{n-t}$, if for all $k \in [n] \setminus I$ and $i = k(I)$, $\pi'_i = \pi_k(\pi(I))$, where for $a \in [n]$ and $I \subseteq [n]$, $a(I) \in [n]$ is the integer $a(I) = a - |\{i \in I : i < a\}|$. For convenience, throughout the remainder of this paper, we will write π' as $\pi_{\downarrow,\mathcal{I}}$, where the $\downarrow$ indications deletions and $\mathcal{I}$ is the set of elements (not locations) to be deleted.

In particular, a permutation $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$ experiences a single PID of the symbol π_j at position $j \in [n]$, resulting in the permutation $\pi_{\downarrow,\{\pi_j\}}$ if

$$\pi_{\downarrow,\{\pi_j\}} = (\pi'_1, \ldots, \pi'_{j-1}, \pi'_{j+1}, \ldots, \pi'_n) \in \mathbb{S}_{n-1}, \tag{4}$$

where for $1 \leqslant i \leqslant n$ $(i \neq j)$, we have

$$\pi'_i = \begin{cases} \pi_i, & \text{if } \pi_i < \pi_j \\ \pi_i - 1, & \text{otherwise.} \end{cases} \tag{5}$$

For shorthand, we will write $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\{\pi_j\}}$.

For a permutation $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$, it is straightforward to see that if $(\pi_{i_1}, \ldots, \pi_{i_t})$ is a decreasing vector (i.e., $\pi_{i_1} > \pi_{i_2} > \cdots > \pi_{i_t}$) and $\mathcal{I} = \{\pi_{i_1}, \ldots, \pi_{i_t}\}$, then from (4),

$$(((\pi_{\downarrow,\pi_{i_1}})_{\downarrow,\pi_{i_2}})\ldots)_{\downarrow,\pi_{i_t}} = \pi_{\downarrow,\mathcal{I}}. \tag{6}$$

Let $\mathcal{B}_{D,t}(\pi)$ be the set of permutations resulting from t PIDs in π, i.e., $\mathcal{B}_{D,t}(\pi) = \{\pi_{\downarrow,\mathcal{I}} : \mathcal{I} \subset [n], |\mathcal{I}| = t\}$, where $1 \leqslant t \leqslant n$. For shorthand, we will write $\mathcal{B}_{D,1}(\pi) = \mathcal{B}_D(\pi)$. Therefore, a code $\mathcal{C}$ is a ***t-PID-correcting code*** if for any distinct $\pi, \sigma \in \mathcal{C}$, $\mathcal{B}_{D,t}(\pi) \cap \mathcal{B}_{D,t}(\sigma) = \emptyset$.

Let us define the model of permutation-invariant insertions (PIIs). We say that a permutation $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$ experiences a single PII of the symbol $s \in [n+1]$ at position $j \in [n+1]$, resulting in the permutation $\pi^{\uparrow,s,j}$ if

$$\pi^{\uparrow,s,j} = (\pi'_1, \ldots, \pi'_{j-1}, s, \pi'_j, \ldots, \pi'_n) \in \mathbb{S}_{n+1}, \tag{7}$$

where for $1 \leqslant i \leqslant n$, we have

$$\pi'_i = \begin{cases} \pi_i, & \text{if } \pi_i < s \\ \pi_i + 1, & \text{otherwise.} \end{cases}$$

For example, if $\pi = (3,1,4,2)$, then $\pi^{\uparrow,3,2} = (4,3,1,5,2)$.

Let $\mathcal{B}_I(\pi)$ be the set of all permutations obtained by applying a single PII to π, that is,

$$\mathcal{B}_I(\pi) = \{\pi^{\uparrow,s,j} : s, j \in [n+1]\}.$$

More generally, let $\mathcal{B}_{I,t}(\pi) = \{\sigma \in \mathbb{S}_{n+t} : \exists(\sigma^{(0)} = \pi, \sigma^{(1)}, \sigma^{(2)}, \ldots, \sigma^{(t-1)}, \sigma^{(t)} = \sigma), \sigma^{(i)} \in \mathcal{B}_I(\sigma^{(i-1)}), 1 \leqslant i \leqslant t\}$, so that $\mathcal{B}_{I,t}(\pi)$ is the set of permutations possible given that t successive single insertions occur to π. (Here, $\sigma^{(i)}$ is formed by a single insertion into $\sigma^{(i-1)}$ for $1 \leqslant i \leqslant t$.) We say that a code $\mathcal{C}$ is a ***t-PII-correcting code*** if for any $\pi, \sigma \in \mathcal{C}$, $\mathcal{B}_{I,t}(\pi) \cap \mathcal{B}_{I,t}(\sigma) = \emptyset$.

The following claims will be useful in subsequent derivations. Claim 15 is straightforward to verify. The proofs of Claims 16 and 17 are included in Appendix B.

Claim 15: Let $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$. 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 16: Let $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$, $s, t \in [n]$ and suppose that $s < t$. Then $(\pi_{\downarrow,s})_{\downarrow,t-1} = (\pi_{\downarrow,t})_{\downarrow,s}$.

Claim 17: Let $\pi \in \mathbb{S}_n$ and $j, k, s, t \in [n+1]$ such that $s \leqslant t$. Then,

- *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 [18], there is also a duality between insertions and deletions in permutations.

Lemma 18: 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 15, $\pi = \tau^{\uparrow,\pi_j,j}$ 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 17, 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 $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$.

Now, suppose $j > k$, where, as before, $\pi_j \leqslant \sigma_k$. From Claim 17, we have

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

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 $\pi_j < \sigma_k$ and $j = k$. Then from Claim 17, we have $(\tau^{\uparrow,\pi_j,j})^{\uparrow,\sigma_k+1,j} = (\tau^{\uparrow,\sigma_k,j})^{\uparrow,\pi_j,j+1}$ and again $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$. Therefore, $\mathcal{B}_I(\pi) \cap \mathcal{B}_I(\sigma) \neq \emptyset$ in all cases, as desired.

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

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

Recall that from Claim 15, we have $\pi = \theta_{\downarrow,s}$ and $\sigma = \theta_{\downarrow,t}$. First, if $s = t$, then $j = k$. Furthermore, if $j = k$ and $s = t$ then $\pi = \sigma$ and the result is straightforward.

Next, suppose, without loss of generality, that $s < t$. From Claim 16,

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

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 next theorem generalizes the previous result.

Theorem 19: For any $\pi, \sigma \in \mathbb{S}_n$, $\mathcal{B}_{D,t}(\pi) \cap \mathcal{B}_{D,t}(\sigma) \neq \emptyset$ if and only if $\mathcal{B}_{I,t}(\pi) \cap \mathcal{B}_{I,t}(\sigma) \neq \emptyset$.

Proof: We first show by induction on t that for all n if $\mathcal{B}_{D,t}(\pi) \cap \mathcal{B}_{D,t}(\sigma) \neq \emptyset$ for $\pi, \sigma \in \mathbb{S}_n$, then there exist $t-1$ permutations $\tau'^{(1)}, \ldots, \tau^{(t-1)} \in \mathbb{S}_n$, such that

$$\mathcal{B}_D(\tau^{(i)}) \cap \mathcal{B}_D(\tau^{(i+1)}) \neq \emptyset,$$

for $0 \leqslant i \leqslant t-1$, while $\tau^{(0)} = \pi$ and $\tau^{(t)} = \sigma$.

It is clear to see that the claim holds for all n if $t = 1$. Let us assume that the induction assertion holds for some $t \geqslant 1$ and we will prove its correctness for $t+1$. Let $\pi, \sigma \in \mathbb{S}_n$ be such that $\mathcal{B}_{D,t+1}(\pi) \cap \mathcal{B}_{D,t+1}(\sigma) \neq \emptyset$. According to (6), there exist $\pi', \sigma' \in \mathbb{S}_{n-1}$ such that $\pi' \in \mathcal{B}_D(\pi)$, $\sigma' \in \mathcal{B}_D(\sigma)$ and

$$\mathcal{B}_{D,t}(\pi') \cap \mathcal{B}_{D,t}(\sigma') \neq \emptyset.$$

According to the induction assumption there exist $t-1$ permutations $\tau'^{(1)}, \ldots, \tau'^{(t-1)} \in \mathbb{S}_{n-1}$, such that

$$\mathcal{B}_D(\tau'^{(i)}) \cap \mathcal{B}_D(\tau'^{(i+1)}) \neq \emptyset,$$

for $0 \leqslant i \leqslant t-1$, while $\tau'^{(0)} = \pi'$ and $\tau'^{(t)} = \sigma'$. According to Lemma 18, for $1 \leqslant i \leqslant t$, there exists $\tau'^{(i)} \in \mathbb{S}_n$ such that

$$\tau_i \in \mathcal{B}_I(\tau'^{(i-1)}) \cap \mathcal{B}_I(\tau'^{(i)}).$$

Therefore, we have t permutations $\tau^{(1)}, \ldots, \tau^{(t)}$, where according to Claim 15, $\tau'^{(i-1)}, \tau'^{(i)} \in \mathcal{B}_D(\tau^{(i)})$ for $1 \leqslant i \leqslant t$. In particular, for $1 \leqslant i \leqslant t-1$,

$$\tau'^{(i)} \in \mathcal{B}_D(\tau^{(i)} \cap \mathcal{B}_D(\tau^{(i+1)}),$$

and thus $\mathcal{B}_D(\tau^{(i)}) \cap \mathcal{B}_D(\tau^{(i+1)}) \neq \emptyset$. Note also that $\pi' = \tau'^{(0)} \in \mathcal{B}_D(\pi)$ and $\sigma' = \tau'^{(t)} \in \mathcal{B}_D(\sigma)$, and thus, if we let $\tau^{(0)} = \pi$ and $\tau^{(t+1)} = \sigma$, then $\mathcal{B}_D(\tau^{(i)}) \cap \mathcal{B}_D(\tau^{(i+1)}) \neq \emptyset$ for $0 \leqslant i \leqslant t$.

By a similar induction it is possible to show the opposite property. Namely, if there exist $t-1$ permutations $\tau^{(1)}, \ldots, \tau^{(t-1)} \in \mathbb{S}_n$, such that $\mathcal{B}_D(\tau^{(i)}) \cap \mathcal{B}_D(\tau^{(i+1)}) \neq \emptyset$, for $0 \leqslant i \leqslant t-1$, while $\tau^{(0)} = \pi$, then $\mathcal{B}_{D,t}(\pi) \cap \mathcal{B}_{D,t}(\sigma) \neq \emptyset$.

The same assertion holds for insertions. That is, for all t and n $\mathcal{B}_{I,t}(\pi) \cap \mathcal{B}_{I,t}(\sigma) \neq \emptyset$ for $\pi, \sigma \in \mathbb{S}_n$ if and only if there exist $t-1$ permutations $\tau^{(1)}, \ldots, \tau^{(t-1)} \in \mathbb{S}_n$, such that

$$\mathcal{B}_I(\tau^{(i)}) \cap \mathcal{B}_I(\tau^{(i+1)}) \neq \emptyset,$$

for $0 \leqslant i \leqslant t-1$, while $\tau^{(0)} = \pi$ and $\tau^{(t)} = \sigma$.

Lastly, since single insertions and deletions are equivalent (Lemma 18), we deduce that $\mathcal{B}_{D,t}(\pi) \cap \mathcal{B}_{D,t}(\sigma) \neq \emptyset$ if and only if $\mathcal{B}_{I,t}(\pi) \cap \mathcal{B}_{I,t}(\sigma) \neq \emptyset$. ■

The next corollary follows directly from Theorem 19.

Corollary 20: For all $t \geqslant 1$, $\mathcal{C}$ is a t-PID-correcting code if and only if it is a t-PII-correcting code.

VI. Bounds on PII/PID Balls

In this section, we establish a relationship between our notion of PIIs and PIDs and the theory of permutation patterns. We quote certain results from the existing literature on permutation patterns; these results compute the sizes of PII and PID balls. Based on these results, we then produce an asymptotic upper bound on the size of a code that is capable of correcting a single PID.

There is a wealth of literature on the subject of permutation patterns, e.g., [16], [21]. Most works are concerned with determining which permutations, of any length, avoid certain small permutations. Here, by avoid, we mean that a permutation must not have any positions whose ordering induces the smaller permutation. A related problem was recently studied by Homberger [10]. This work introduces a series of results regarding the number of permutations patterns contained in larger permutations of fixed size. These results are quite useful for the study of PIIs and PIDs. We cite several theorems from this work. First, we define the concept of consecutive runs:

*Definition 21: 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 $(5, 4, 3, 2)$.

In general, we note that the size of the PID ball $\mathcal{B}_D(\pi)$ depends on the permutation π. However, this size can be characterized solely as a function of the number of consecutive runs in π. Following [10], we also introduce the notion of permutation ***bonds***. We say that a pair of consecutive elements (v_i, v_{i+1}) in a vector $\boldsymbol{v}$ $(1 \leqslant i \leqslant n-1)$ is a bond if $|v_i - v_{i+1}| = 1$. Observe that a consecutive run of length k in a permutation is made up of $k-1$ bonds.

Note that if π_j and π_k are part of the same consecutive run in π, deleting either π_j or π_k will yield the same permutation. That is $\pi_{\downarrow,\pi_j} = \pi_{\downarrow,\pi_k}$. For a permutation $\pi \in \mathbb{S}_n$, we denote by $R(\pi)$ the number of consecutive runs in π and by $C(\pi)$ the number of bonds in π. Using these observations, the following result immediately follows from [10, Th. 6].

Theorem 22: For all $\pi \in \mathbb{S}_n$, $|\mathcal{B}_D(\pi)| = R(\pi) = n - C(\pi)$.

Before we proceed to derive our upper bound on the maximum PID-correcting code cardinality $A_{PID}(n)$, we briefly discuss the subject of insertions. Recall that when working with sequences, the size of insertion balls does not depend on the sequence $\boldsymbol{x}$, as shown by Levenshtein in [18]. Surprisingly, the analogous result does not hold for permutations. That is, in general, the size of the t-PII ball $|\mathcal{B}_{I,t}(\pi)|$ is a function of π. However, in the single PII case $t = 1$, the PII ball size $|\mathcal{B}_I(\pi)|$, does not depend on the permutation π. We have the following simple formula, given in [10]: For all $\pi \in \mathbb{S}_n$, $|\mathcal{B}_I(\pi)| = n^2 + 1$.

The case $t = 1$ is the sole case where the PII balls are of equal size. We have that, for example, $|\mathcal{B}_{I,2}(1, 2, 3, 4)| = 207$ while $|\mathcal{B}_{I,2}(1, 3, 4, 2)| = 208$. The following table illustrates how different permutations of length 4 may have different size PII balls for $t > 1$ [3]:

t \ π	(1,2,3,4)	(1,3,2,4)	(1,3,4,2)
1	17	17	17
2	207	207	208
3	2279	2278	2300

We are ready to proceed with our computation of the upper bound on $A_{PID}(n)$. Ideally, we would first like to compute the number of permutations $\pi \in S_n$ with r consecutive runs, so that $R(\pi) = r$. However, it turns out that this is a challenging quantity to compute. Using an approach similar to the Goulden-Jackson method [22], Homberger derives a generating function for the number of permutations with k bonds in [10]. Let $a_{n,k}$ be the number of permutations $\pi \in S_n$ which contain exactly k bonds. We set $a_{0,0} = 1$. Then,

$$\sum_{n\geqslant 0}\sum_{k\geqslant 0} a_{n,k} z^n u^k = \sum_{m\geqslant 0} m! \left(z + \frac{2z^2(u-1)}{1 - z(u-1)}\right)^m.$$

Since $R(\pi) = n - C(\pi)$, in principle, we could obtain our desired values from the generating function above. However, working with this function is quite difficult. Instead, we will rely on an upper bound. We introduce some useful terminology. For positive integers n, r where $r < n$, define

$$F(n,r) = \binom{n-1}{r-1} \cdot 2^{\min\{r, n-r\}} \cdot r!. \tag{8}$$

Lemma 23: The number of permutations in $\mathbb{S}_n$ with r $(1 \leqslant r \leqslant n)$ consecutive runs is at most $F(n, r)$.

Proof: Consider the set of permutations in $\mathbb{S}_n$ 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. ■

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 assume that all log functions are base 2.

We also need the following claim and lemma.

Claim 24: For $2 \leqslant r \leqslant n - \log(n)$, $F(n, r-1) \leqslant F(n, r)$.

Proof: It is easy to see that $F(n, r-1) \leqslant F(n,r)$ is equivalent to the statement

$$\frac{r-1}{n-r+1} \cdot 2^{\min\{r-1, n-r+1\}} \leqslant 2^{\min\{r, n-r\}} \cdot r. \tag{9}$$

We proceed by casework. First, if $r \leqslant n-r$, then $r-1 \leqslant n-r-1 < n-r+1$, so (9) reduces to $\frac{r-1}{n-r+1} \leqslant 2r$. The left-hand side is smaller than 1, so this inequality clearly holds for $r \geqslant 2$.

The next case is $r = n-r+1$. Then, $r > n-r$ while $r-1 < n-r+1$. (9) now reduces to $\frac{r-1}{n-r+1} \leqslant 2^{n-2r+1} \cdot r$. Since $n-2r+1 = 0$ and $n-r+1 = r$, (9) is equivalent to $\frac{r-1}{r^2} \leqslant 1$, which is true when $r \geqslant 2$.

Finally we have the case $r \geqslant n-r+2$. Since $n-r < r$ and $n-r+1 \leqslant r-1$, we must show that $\frac{r-1}{n-r+1} \cdot 2 \leqslant r$, or, equivalently, $2(1-\frac{1}{r}) \leqslant (n-r+1)$. Now, $(n-r+1) \geqslant \log(n)+1 \geqslant 2$ for $n \geqslant 2$, so, since $r \leqslant n - \log n$, the result holds. ■

Lemma 25: For $n \geqslant 4$, 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 24, the maximum number of permutations in $\mathbb{S}_n$ with no more 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 (8) 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. ■

Now we are ready to apply the preceding results to state the main theorem in this section. Using a similar approach as in [18], we provide an upper bound for the maximum size of a single-deletion-correcting code $A_{PID}(n)$.

Theorem 26: For any $0 < \epsilon < 1$ there exists an N_ϵ such that for all $n \geqslant N_\epsilon$, with n a power of two, $A_{PID}(n) \leqslant \frac{n!}{n(n-\log(n))}(1+\epsilon)$.

Proof: Suppose $\mathcal{C}$ is a single PID-correcting code over $\mathbb{S}_n$. Let $S_1 = \{\pi \in \mathbb{S}_n : |\mathcal{B}_D(\pi)| > n - \log(n)\}$. We first consider an upper bound on $|\mathcal{C} \cap S_1|$. Since for all $\pi \in \mathcal{C} \cap S_1$ the sets $\mathcal{B}_D(\pi) \subseteq \mathbb{S}_{n-1}$ are disjoint, and since $|\mathcal{B}_D(\pi)| > n - \log(n)$, we get,

$$|\mathcal{C} \cap S_1| \leqslant \frac{(n-1)!}{n-\log(n)}.$$

Let $S_2 = \{\pi \in \mathbb{S}_n : |\mathcal{B}_D(\pi)| \leqslant n - \log(n)\}$. Clearly, $|\mathcal{C} \cap S_2| \leqslant |S_2|$ and from Lemma 25, $|S_2| \leqslant \frac{n!(n-\log(n))^2}{(\log(n))!}$. Thus, we have

$$|\mathcal{C}| = |\mathcal{C} \cap S_1| + |\mathcal{C} \cap S_2| \leqslant \frac{(n-1)!}{n-\log(n)} + \frac{n!(n-\log(n))^2}{(\log(n))!}$$
$$= \frac{n!}{n(n-\log(n))}\left(1 + \frac{n(n-\log(n))^3}{(\log(n))!}\right).$$

We conclude that $A_{PID}(n) \leqslant \frac{n!}{n(n-\log(n))}\left(1+\frac{n(n-\log(n))^3}{(\log(n))!}\right)$. 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 $A_{PID}(n) \leqslant \frac{n!}{n(n-\log(n))}(1+\epsilon)$, as desired. ■

VII. Code Construction

In this section, we introduce an asymptotically optimal construction of single-PID-correcting codes. 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 a 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 27: 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 $\pi_{\downarrow,\pi_j} = f^{-1}(M')$.

Recall the mapping known as the ***signature***, which was previously introduced in Section IV-C. For convenience, we restate the definition. For $\boldsymbol{y} \in [m]^n$, the binary length-$(n-1)$ signature $\alpha(\boldsymbol{y}) = (\alpha(\boldsymbol{y})_1, \ldots, \alpha(\boldsymbol{y})_{n-1})$ is defined as follows. For $1 \leqslant i \leqslant n-1$,

$$\alpha(\boldsymbol{y})_i = \begin{cases} 1, & \text{if } y_{i+1} \geqslant y_i. \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Recall that for a permutation $\pi = (\pi_1, \ldots, \pi_n) \in \mathbb{S}_n$, 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 28: 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 29: For $\pi \in \mathbb{S}_n$, we have $(\pi_{\downarrow,\pi_j})^{-1} = (\pi^{-1})_{\downarrow,j}$.

Proof: From Claim 27, 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 27, 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 $\boldsymbol{x} \in GF(2)^n$ is a binary vector, we denote by $\mathcal{B}_D(\boldsymbol{x})$ the set of all possible vectors obtainable by deleting one bit from $\boldsymbol{x}$.[3]

[3]That is, this is a deletion ball resulting from a single deletion to a binary sequence.

If a symbol from a permutation π is deleted, then a bit 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 29 is the following.

Corollary 30: 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 is called a ***binary single-deletion-correcting code*** if it can correct any single-bit deletion. As a result of Corollary 30, 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-PID-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 31: For a binary sequence s, a **run** is a maximal-length 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 32: Continuing with the setup from Example 28, 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. Similarly, the final three symbols in π comprise a descending run and so the final two symbols in $\alpha(\pi)$ are zeros.

The following claim is straightforward to verify.

Claim 33: Consider a permutation $\pi \in \mathbb{S}_n$ where $n \geqslant 2$ 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 $\alpha(\sigma)$ can be obtained by deleting a 1 from the run in $\alpha(\pi)$ which begins in position i and ends in position j of $\alpha(\pi)$, then we can conclude that there exists a permutation $\sigma = \pi_{\downarrow,x}$ where x is at position k in π and $i \leqslant k \leqslant j+1$. A similar statement holds for descending runs and a deletion of 0.

B. Code Construction

Let us first review the binary single-deletion-correcting code we will use in our construction. Namely, the code from [18], initially introduced in [25], known as a ***Varshamov-Tennegolts code*** (VT code), is defined as follows. For a positive integer $n \geqslant 2$ and $a \in \mathbb{Z}_{n+1}$, $\mathcal{C}_n^a$ is the code $\mathcal{C}_n^a = \{\boldsymbol{x} \in GF(2)^n : \sum_{i=1}^n i x_i \equiv a \mod n+1\}$.

Lemma 34 (See [18], [25]): For any integer $n \geqslant 2$ and $a \in \mathbb{Z}_{n+1}$, the code $\mathcal{C}_n^a$ is a binary single-deletion-correcting code.

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

Construction 35: Given an integer $n > 2$ and $a_1, a_2 \in \mathbb{Z}_n$, let

$$\mathcal{C}_n^{a_1,a_2} = \left\{\pi \in \mathbb{S}_n : \alpha(\pi) \in \mathcal{C}_{n-1}^{a_1}, \alpha(\pi^{-1}) \in \mathcal{C}_{n-1}^{a_2}\right\}. \quad (11)$$

We first comment on the cardinality of the codes $\mathcal{C}_n^{a_1,a_2}$. Recall from the previous section that $A_{PID}(n)$ represents the maximum cardinality of a single-PID-correcting code. Note that the codes $\mathcal{C}_n^{a_1,a_2}$ 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 $R(n)$ for $n > 2$ the maximum cardinality of a code according to Construction 35, that is, $R(n) = \max_{a_1,a_2 \in \mathbb{Z}_n}\{|\mathcal{C}_n^{a_1,a_2}|\}$, then applying the pigeonhole principle gives the following.

$$R(n) \geqslant \frac{n!}{n^2}.$$

Then, applying our bound on $A_{PID}(n)$ from Theorem 26 yields the result,

Corollary 36: Construction 35 is asymptotically optimal, that is,

$$\lim_{n\to\infty} \frac{R(n)}{A_{PID}(n)} = 1.$$

We continue by proving the correctness of Construction 35. The decoding algorithm for the construction is presented as Algorithm 2. We show that Algorithm 2 can correct a single PID. For simplicity, for a permutation σ, we use $x \prec_\sigma y$ if and only if $\sigma^{-1}(x) < \sigma^{-1}(y)$.

Theorem 37: For $n > 2$ and $a_1, a_2 \in \mathbb{Z}_n$, the code $\mathcal{C}_n^{a_1,a_2}$ is a single-PID-correcting permutation code.

Proof: Suppose that $\pi \in \mathcal{C}_n^{a_1,a_2}$ 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

Algorithm 2 DECODE

```
input : the retrieved permutation $\sigma$
output: the deleted symbol $x$
1 compute $\alpha(\sigma)$ and $\alpha(\sigma^{-1})$ from $\sigma$;
2 $\alpha(\pi) \longleftarrow$ VTDEC($\alpha(\sigma)$);
3 $d \longleftarrow$ deleted element of $\alpha(\pi)$;
4 $i \longleftarrow$ start of run of deleted element in $\alpha(\pi)$;
5 $j \longleftarrow$ end of run of deleted element in $\alpha(\pi)$;
6 $\alpha(\pi^{-1}) \longleftarrow$ VTDEC($\alpha(\sigma^{-1})$);
7 $e \longleftarrow$ deleted element of $\alpha(\pi^{-1})$;
8 $p \longleftarrow$ start of run of deleted element in $\alpha(\pi^{-1})$;
9 $q \longleftarrow$ end of run of deleted element in $\alpha(\pi^{-1})$;
10 if $d = 1$ and $e = 1$ then
11     $S \longleftarrow \{\sigma_i, \ldots, \sigma_j\} \cap \{p, \ldots, q\}$;
12     if $S = \emptyset$ then
13         if $\sigma^{-1}(q) < i$ then $x \longleftarrow q+1$;
14         else if $\sigma^{-1}(p) > j$ then $x \longleftarrow p$;
15         else Find $x \in \{p+1, \ldots, q\}$ such that
               $\sigma^{-1}(x-1) < i$ and $\sigma^{-1}(x) > j$;
16     else $x \longleftarrow \min S$;
17 else if $d = 1$ and $e = 0$ then
18     if $\sigma^{-1}(p) < i$ then $x \longleftarrow p$;
19     else if $\sigma^{-1}(q) > j$ then $x \longleftarrow q+1$;
20     else Find $x \in \{p+1, \ldots, q\}$ such that $\sigma^{-1}(x) < j$
           and $\sigma^{-1}(x-1) > j$;
21 else
22     ··· /* Essentially similar to the case
           of $d = 1$                          */
23 end
```

elements were deleted and show that there is a 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 permutation σ, we compute $\alpha(\sigma)$. Corollary 30 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 run endpoints i and j of Claim 33. 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 33, $\pi = \sigma^{\uparrow,x,k}$ such that

$$i \leqslant k \leqslant j+1, \tag{12}$$

$$\sigma_i < \sigma_{i+1} < \cdots < \sigma_j, \tag{13}$$

$$x \leqslant \sigma_m \text{ iff } k \leqslant m \text{ for } m \in \{i, \ldots, j\}. \tag{14}$$

By Lemma 29, we have $\sigma^{-1} = (\pi^{-1})_{\downarrow,k}$ and $\pi^{-1} = \left(\sigma^{-1}\right)^{\uparrow,k,x}$. We thus may apply Claim 33 to 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 $\pi_p^{-1}, \pi_{p+1}^{-1}, \ldots, k, \ldots, \pi_q^{-1}, \pi_{q+1}^{-1}$ is a run in 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 33 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 $\alpha\left(\sigma^{-1}\right)$.

Now we have determined the ranges of possible values for x and k. We use the combination of these ranges to specify x and k.

There are two different cases depending on the deleted element of $\alpha\left(\pi^{-1}\right)$ being a 1 or a 0. We only consider the former case; the latter follows using the same idea. Suppose that a 1 in a run of 1s in $\alpha\left(\pi^{-1}\right)$ is deleted. We have $\pi_p^{-1} < \pi_{p+1}^{-1} < \cdots < k < \cdots < \pi_q^{-1} < \pi_{q+1}^{-1}$ and thus

$$x \in \{p, p+1, \ldots, q+1\}, \tag{15}$$

$$p \prec_\sigma p+1 \prec_\sigma \cdots \prec_\sigma q, \tag{16}$$

$$k \leqslant \sigma^{-1}(y) \text{ iff } x \leqslant y \text{ for } y \in \{p, \ldots, q\}, \tag{17}$$

where $p \prec_\sigma p+1$ denotes that the symbol p appears before the symbol $p+1$ in the permutation σ. Let $S = \left\{\sigma_i, \ldots, \sigma_j\right\} \cap \{p, \ldots, q\}$. Suppose S is empty. Because $\sigma_i, \ldots, \sigma_j$ is an increasing run in σ and $p, p+1, \ldots, q$ is an increasing subsequence in σ, we have that one of the following cases holds: 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: First, in case a), from (12), we have $\sigma^{-1}(q) < k$, which using (17) implies that $x > q$. From (15), we find $x = q+1$. Next, in case b), similarly to case a), from (12), (17), and (15), we have $x = p$. Lastly, in case c), from (12) it follows that $\sigma^{-1}(z-1) < k \leqslant \sigma^{-1}(z)$. Using (17), we find that $x = z$. Hence, we can identify x if S is empty. Having identified x, we can find the unique position k in $\{i, \ldots, j+1\}$ that satisfies condition (14).

Now suppose S is nonempty and take u to be the smallest element in S and v to be the largest element in S. From (13) and (16), it is not difficult to show that every integer between u and v is also in S, i.e.,

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

and that the elements of S form a consecutive run in σ. Based on (12)–(17), 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 S is increased by 1. Thus π is determined uniquely. ■

We illustrate the preceding proof by the following example.

Example 38: Consider the code $\mathcal{C}_8^{a_1,a_2}$, where $a_1 = 0$ and $a_2 = 2$. Suppose the stored codeword is $\pi = (7,4,5,6,8,2,1,3) \in \mathcal{C}_8^{0,2}$, 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

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

and, using the decoders $\mathcal{D}_{a_1}^n$ and $\mathcal{D}_{a_2}^n$, computes

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

We thus have $i = 2$, $j = 4$, $p = 4$, and $q = 5$. Furthermore, $S = \{4,5,7\} \cap \{4,5\} = \{4,5\}$, 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.

VIII. Conclusion

In this work, we explored the problem of dealing with erasures and deletions in rank modulation systems. Specifically, we gave four error models: symbol-invariant erasures (SIEs), permutation-invariant erasures (PIEs), symbol-invariant deletions (SIDs), and permutation-invariant deletions (PIDs). These models select between erasures and deletions and between symbol-invariance, where unaffected symbols do not change value, and permutation-invariance, where these symbols change value in such a way that the final result remains a permutation. We showed that for the SIE model, codes over the Hamming metric are sufficient. Next, we showed that the SID and PIE models are equivalent and related by permutation inverses. In addition, we proved that the Ulam distance is the appropriate metric for correcting SIDs or PIEs. Furthermore, we gave a new construction of codes in the Ulam metric that, in certain cases, has larger cardinality than the best-known codes.

We also studied the more difficult model of PID errors. We gave the basic properties of such errors and showed that certain aspects of PIDs are equivalent to problems in permutation patterns. We computed an upper bound on the size of the largest code capable of correcting a single PID. We then introduced an asymptotically optimal single PID-correcting code construction along with an associated decoding algorithm.

Lastly, we note that most of our results are asymptotic and the study of these codes for smaller length is left for future research.

Appendix A Proof of Lemma 14

We first revisit some notation from Section IV-C before proceeding to the proof of the lemma. Let $A_H(n,t)$ be the maximum size of a code of length n over $\mathbb{S}_n$ with Hamming distance at least t. Recall from Section IV-C,

$$A_{FSM}(n,t) = \prod_{i=1}^{k} A_H\left(\left\lceil \frac{n}{2^i}\right\rceil - 1, \frac{3t}{2}\right), \tag{18}$$

where $k = \lfloor \log_2 \frac{n}{3t/2+1} \rfloor$, and

$$A_{CV}(n,t) = \sum_{j=0}^{s}\prod_{i=1}^{u_j} A_H\left(\left\lceil \frac{n-j(3t+1)}{2^i}\right\rceil - 1, \frac{3t}{2}\right), \tag{19}$$

where $s = \left\lfloor \frac{n}{3t+1} \right\rfloor$ and $u_j = \left\lfloor \log_2 \frac{n-j(3t+1)}{3t/2+1} \right\rfloor$ for $j \in [0,s]$, and

$$A_{New}(n,t) = \frac{((\frac{2n}{t})!)^{t/2}}{(n+1)^{\frac{3t}{2}-4}(2n/t)^{t/2}}.$$

Lemma 14: For sufficiently large n where $n+1$ is a prime, $t|2n$, and $t \leqslant 6$, $A_{New}(n,t) > A_{FSM}(n,t)$ and $A_{New}(n,t) > A_{CV}(n,t)$.

Proof: Clearly $A_{CV}(n,t) \geqslant A_{FSM}(n,t)$ and so we only need to prove that $A_{New}(n,t) > A_{CV}(n,t)$ for appropriate values of n and t.

Since $\log m! \geqslant m \log(m/e)$, we find

$$\begin{aligned}\log A_{New}(n,t) &\geqslant \frac{t}{2}\left(\frac{2n}{t}\right)\log\frac{2n}{te} \\ &\quad - \left(\frac{3t}{2}-4\right)(\log(n+1)) - \frac{t}{2}\log\frac{2n}{t} \\ &= n\log n - n\log\frac{te}{2} + O(\log n). \end{aligned} \tag{20}$$

From [9, Th. 4], we have

$$A_H(m,u) \leqslant \frac{m!}{(u-1)!}.$$

Using this upper bound on $A_H(m,u)$, we find

$$A_{CV}(n,t) \leqslant n \prod_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} \frac{\lceil \frac{n}{2^i} - 1\rceil!}{(3t/2-1)!}.$$

Thus,

$$\begin{aligned}\log A_{CV}(n,t) &\leqslant \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} \log\left(\left(\frac{n}{2^i}\right)!\right) + O(\log n) \\ &\leqslant \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} \log\left(\sqrt{2\pi e n}(n/2^i)^{n/2^i}e^{-n/2^i}\right) \\ &\quad + O(\log n),\end{aligned} \tag{21}$$

where we have used $m! \leqslant \sqrt{2\pi e m}\,(m/e)^m$ (for large enough m) we drop floors and ceilings for convenience. We use (21) to write

$$\begin{aligned}\log A_{CV}(n,t) &\leqslant \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} \frac{n}{2^i}\log\frac{n}{2^i} - \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} \frac{n}{2^i}\log e \\ &\quad + O\left((\log n)^2\right) \\ &\leqslant \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} \frac{n}{2^i}(\log n - i) \\ &\quad - n\log e \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} 2^{-i} + O\left((\log n)^2\right) \\ &= n\log n \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} 2^{-i} - n \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} i2^{-i} \\ &\quad - n\log e \sum_{i=1}^{\lfloor \log \frac{n}{3t/2+1}\rfloor} 2^{-i} + O\left((\log n)^2\right).\end{aligned}$$

Since $\log\frac{n}{3t/2+1} - 1 \leqslant \lfloor \log\frac{n}{3t/2+1}\rfloor \leqslant \log\frac{n}{3t/2+1}$, $\sum_{i=1}^{x} 2^{-i} = 1 - 2^{-x}$, and $\sum_{i=1}^{x} i2^{-i} = 2 - (2+x)2^{-x}$, where the latter two expressions hold for any positive integer x, we have

$$\begin{aligned}\log A_{CV}(n,t) &\leqslant n\log n\left(1 - \frac{3t+2}{2n}\right) \\ &\quad - \left((2(n-2) - (3t+2)\log\frac{n}{3t+2} - 6t\right) \\ &\quad - n\log e + 3t\log e + O\left((\log n)^2\right) \\ &= n\log n - (2+\log e)\,n + O\left((\log n)^2\right).\end{aligned}$$

From this and (20), we bound the difference between code sizes as

$$\log A_{new}(n,t) - \log A_{CV}(n,t) \geqslant n\log\frac{8}{t} + O\left((\log n)^2\right).$$

Recall that by construction, (19) holds when t is even. Hence, $\log A_{new}(n,t) - \log A_{CV}(n,t) > 0$ for n sufficiently large when $t \leqslant 6$. ■

Appendix B Proofs of Claims and Lemmas From Section V

Claim 16: Let $\pi = (\pi_1,\ldots,\pi_n) \in \mathbb{S}_n$, $s,t \in [n]$ and suppose that $s < t$. Then $(\pi_{\downarrow,s})_{\downarrow,t-1} = (\pi_{\downarrow,t})_{\downarrow,s}$.

Proof: Suppose $s = \pi_j$ and $t = \pi_k$ where $j,k \in [n]$ and for now we assume that $j < k$. From (4), we can write

$\sigma = \pi_{\downarrow,\pi_j}$ where $\sigma = (\sigma_1, \ldots, \sigma_{j-1}, \sigma_{j+1}, \ldots, \sigma_n) \in \mathbb{S}_{n-1}$, and the elements σ_i for $1 \leqslant i \leqslant n$ $(i \neq j)$ are such that $\sigma_i = \pi_i$ if $\pi_i < s$ and $\sigma_i = \pi_i - 1$ otherwise. Since $t > s$, the symbol $(t-1)$ appears in position $(k-1)$ in σ. Thus, we can write $\sigma^{(2)} = (\pi_{\downarrow,s})_{\downarrow,t-1} = \sigma_{\downarrow,\pi_k-1} = \sigma_{\downarrow,\sigma_k}$ where

$$\sigma^{(2)} = (\sigma_1^{(2)}, \sigma_{j-1}^{(2)}, \sigma_{j+1}^{(2)}, \ldots, \sigma_{k-1}^{(2)}, \sigma_{k+1}^{(2)}, \ldots, \sigma_n^{(2)}) \in \mathbb{S}_{n-2},$$

and for $1 \leqslant i \leqslant n$,

$$\sigma_i^{(2)} = \begin{cases} \pi_i, & \text{if } \pi_i < s \\ \pi_i - 1, & \text{if } s < \pi_i < t \\ \pi_i - 2, & \text{otherwise.} \end{cases}$$

Let $\zeta = \pi_{\downarrow,t} = (\zeta_1, \ldots, \zeta_{k-1}, \zeta_{k+1}, \ldots, \zeta_n) \in \mathbb{S}_{n-1}$, and the elements ζ_i for $1 \leqslant i \leqslant n$ are such that $\zeta_i = \pi_i$ if $\pi_i < t$ and $\zeta_i = \pi_i - 1$ otherwise. Since $t > s$ and $j < k$, $\zeta_j = s$. Under this setup, it follows that $\zeta_{\downarrow,s} = (\pi_{\downarrow,t})_{\downarrow,s} = \sigma^{(2)} = (\pi_{\downarrow,s})_{\downarrow,t-1}$.

The proof for the case where $j > k$ is identical and so the details are omitted. ■

Claim 17: Let $\pi \in \mathbb{S}_n$ and $j, k, s, t \in [n+1]$ such that $s \leqslant t$. Then,

- 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 $(\pi^{\uparrow,s,j})^{\uparrow,t+1,j} = (\pi^{\uparrow,t,j})^{\uparrow,s,j+1}$.

Proof: We first show that for the case where $j < k$, $(\pi^{\uparrow,s,j})^{\uparrow,t+1,k+1} = (\pi^{\uparrow,t,k})^{\uparrow,s,j}$. From (7), we can write $\sigma = \pi^{\uparrow,s,j} = (\sigma_1, \ldots, \sigma_{j-1}, s, \sigma_{j+1}, \ldots, \sigma_n) \in \mathbb{S}_{n+1}$ where for $1 \leqslant i \leqslant n$, we have $\sigma_i = \pi_i$ if $\pi_i < s$ and $\sigma_i = \pi_i + 1$ otherwise. Thus, since $j < k$ and $s \leqslant t$, we can write $\sigma^{(2)} = (\pi^{\uparrow,s,j})^{\uparrow,t+1,k+1} = \sigma^{\uparrow,t+1,k+1}$ where

$$\sigma^{(2)} = (\sigma_1^{(2)}, \ldots, \sigma_{j-1}^{(2)}, s, \sigma_j^{(2)}, \ldots, \sigma_{k-1}^{(2)}, t+1, \sigma_k^{(2)}, \ldots, \sigma_n^{(2)}) \in \mathbb{S}_{n+2},$$

and for $1 \leqslant i \leqslant n$,

$$\sigma_i^{(2)} = \begin{cases} \pi_i, & \text{if } \pi_i < s \\ \pi_i + 1, & \text{if } s \leqslant \pi_i < t \\ \pi_i + 2, & \text{otherwise.} \end{cases}$$

Repeating similar steps, $\sigma^{(2)} = (\pi^{\uparrow,t,k})^{\uparrow,s,j} = \pi^{\uparrow,s,j})^{\uparrow,t+1,k+1}$.

We now consider the case where $j > k$. In this case let $j' = k$ and $k' = j$. Then we need to show that if $j' < k'$, then $(\pi^{\uparrow,s,k'})^{\uparrow,t+1,j'} = (\pi^{\uparrow,t,j'})^{\uparrow,s,k'+1}$. Using similar ideas as before we can write $\zeta = \pi^{\uparrow,s,k'} = (\zeta_1, \ldots, \zeta_{k'-1}, s, \zeta_{k'}, \ldots, \zeta_n) \in \mathbb{S}_{n+1}$ where for $1 \leqslant i \leqslant n$, we have $\zeta_i = \pi_i$ if $\zeta_i < s$ and $\zeta_i = \pi_i + 1$ otherwise. Then, since $j' < k'$ and $s \leqslant t$, we have $\zeta^{(2)} = \zeta^{\uparrow,t+1,j'} = (\pi^{\uparrow,s,k'})^{\uparrow,t+1,j'}$ where

$$\zeta^{(2)} = (\zeta_1^{(2)}, \ldots, \zeta_{j'-1}^{(2)}, t+1, \zeta_{j'}^{(2)}, \ldots, \zeta_{k'-1}^{(2)}, s, \zeta_{k'}^{(2)}, \ldots, \zeta_n^{(2)}) \in \mathbb{S}_{n+2},$$

and for $1 \leqslant i \leqslant n$,

$$\zeta_i^{(2)} = \begin{cases} \pi_i, & \text{if } \pi_i < s \\ \pi_i + 1, & \text{if } s \leqslant \pi_i < t \\ \pi_i + 2, & \text{otherwise.} \end{cases}$$

Repeating similar steps, we have $\zeta^{(2)} = (\pi^{\uparrow,t,j'})^{\uparrow,s,k'+1} = (\pi^{\uparrow,s,k'})^{\uparrow,t+1,j'}$.

We now consider the case where $j = k$ and $s < t$. Then we can write $\tau = (\pi^{\uparrow,s,j})^{\uparrow,t+1,j}$ where

$$\tau = (\tau_1, \ldots, \tau_{j-1}, t+1, s, \tau_j, \ldots, \tau_n) \in \mathbb{S}_{n+2},$$

and for $1 \leqslant i \leqslant n$,

$$\tau_i = \begin{cases} \pi_i, & \text{if } \pi_i < s \\ \pi_i + 1, & \text{if } s \leqslant \pi_i < t \\ \pi_i + 2, & \text{otherwise.} \end{cases}$$

Since $\tau = (\pi^{\uparrow,s,j})^{\uparrow,t+1,j} = (\pi^{\uparrow,t,j})^{\uparrow,s,j+1}$, the statement in the claim holds. ■

ACKNOWLEDGEMENT

The authors thank two anonymous reviewers and the Associate Editor Prof. Moshe Schwartz for their valuable comments and suggestions.

REFERENCES

[1] A. Barg and A. Mazumdar, "Codes in permutations and error correction for rank modulation," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3158–3165, Jul. 2010.
[2] W. A. Beyer, M. L. Stein, and S. M. Ulam, "Metric in biology, an introduction," Univ. California, Los Alamos, NM, USA, Tech. Rep. LA-4973, 1972.
[3] M. Bóna, "Exact enumeration of 1342-avoiding permutations: A close link with labeled trees and planar maps," *J. Combinat. Theory*, vol. 80, no. 2, pp. 257–272, Nov. 1997.
[4] Y. M. Chee and V. K. Vu, "Breakpoint analysis and permutation codes in generalized Kendall tau and Cayley metrics," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 2959–2963.
[5] M. Deza and T. Huang, "Metrics on permutations: A survey," *J. Combinat. Inf. Syst. Sci.*, vol. 23, pp. 173–185, 1998.
[6] R. Gabrys, E. Yaakobi, F. Farnoud, and J. Bruck, "Codes correcting erasures and deletions for rank modulation," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 2759–2763.
[7] R. Gabrys, E. Yaakobi, F. Farnoud, F. Sala, J. Bruck, and L. Dolecek, "Single-deletion-correcting codes over permutations," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun./Jul. 2014, pp. 2764–2768.
[8] F. Farnoud, V. Skachek, and O. Milenkovic, "Error-correction in flash memories via codes in the Ulam metric," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 3003–3020, May 2013.
[9] P. Frankl and M. Deza, "On the maximum number of permutations with given maximal or minimal distance," *J. Combinat. Theory, A*, vol. 22, no. 3, pp. 352–360, 1977.
[10] C. Homberger, "Counting fixed-length permutation patterns," *Online J. Anal. Combinat.*, vol. 7, Nov. 2012.
[11] A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, "Rank modulation for flash memories," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2659–2673, Jun. 2009.
[12] A. Jiang, M. Schwartz, and J. Bruck, "Correcting charge-constrained errors in the rank-modulation scheme," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2112–2120, May 2010.
[13] M. Kendall and J. D. Gibbons, *Rank Correlation Methods*. New York, NY, USA: Oxford Univ. Press, 1990.
[14] M. Kim, J. K. Park, and C. M. Twigg, "Rank modulation hardware for flash memories," in *Proc. IEEE 55th Int. Midwest Symp. Circuits Syst.*, Aug. 2012, pp. 294–297.
[15] M. Kim, M. Shaterian, and C. M. Twigg, "Rank determination algorithm by current comparing for rank modulation flash memories," in *Proc. IEEE 56th Int. Midwest Symp. Circuits Syst.*, Aug. 2013, pp. 1354–1357.
[16] S. Kitaev, *Patterns in Permutations and Words*. New York, NY, USA: Springer-Verlag, 2011.
[17] A. Klein, "On perfect deletion-correcting codes," *J. Combinat. Designs*, vol. 12, no. 1, pp. 72–77, Nov. 2003.
[18] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Phys. Doklady*, vol. 10, pp. 707–710, Feb. 1966.

[19] V. I. Levenshtein, “On perfect codes in deletion and insertion metric,” *Discrete Math. Appl.*, vol. 3, no. 1, pp. 3–20, 1991.

[20] Y. Li, Y. Ma, E. E. Gad, M. Kim, A. Jiang, and J. Bruck, “Implementing rank modulation,” in *Proc. Non-Volatile Memory Workshop (NVMW)*, Mar. 2015, pp. 1–2.

[21] S. Linton, N. Ruškuc, and V. Vatter, *Permutation Patterns*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[22] J. Noonan and D. Zeilberger, “The Goulden–Jackson cluster method: Extensions, applications and implementations,” *J. Difference Equ. Appl.*, vol. 5, nos. 4–5, pp. 355–377, 1999.

[23] K. H. Rosen, *Discrete Mathematics and Its Applications*. London, U.K.: Chapman & Hall, 2004.

[24] G. M. Tenengolts, “Nonbinary codes, correcting single deletion or insertion (Corresp.),” *IEEE Trans. Inf. Theory*, vol. 30, no. 5, pp. 766–769, Sep. 1984.

[25] R. R. Varshamov and G. M. Tenengolts, “Codes which correct single asymmetric errors,” *Avtomatika Telemekhanika*, vol. 6, no. 2, pp. 288–292, 1965.

[26] H. Zhou, A. Jiang, and J. Bruck, “Systematic error-correcting codes for rank modulation,” in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 2978–2982.

Ryan Gabrys received the B.S. degree in mathematics and computer science from the University of Illinois at Champaign-Urbana in 2005. He holds M.Eng. and Ph.D. degrees in electrical engineering from the University of California at San Diego and the University of California at Los Angeles, respectively. Since 2005, he has worked at Space and Naval Warfare Systems Center San Diego developing future naval system capabilities. His research interests include coding theory and its applications to storage and communications.

Eitan Yaakobi (S'07–M'12) is an Assistant Professor at the Computer Science Department at the Technion Israel Institute of Technology. He received the B.A. degrees in computer science and mathematics, and the M.Sc. degree in computer science from the Technion — Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. Between 2011-2013, he was a postdoctoral researcher in the department of Electrical Engineering at the California Institute of Technology. His research interests include information and coding theory with applications to non-volatile memories, associative memories, data storage and retrieval, and voting theory. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010-2011.

Farzad Farnoud is a postdoctoral scholar at the California Institute of Technology. He received his MS degree in Electrical and Computer Engineering from the University of Toronto in 2008. From the University of Illinois at Urbana-Champaign, he received his MS degree in mathematics and his PhD in Electrical and Computer Engineering in 2012 and 2013, respectively. His research interests include the informationtheoretic and algorithmic analysis of genomic evolutionary processes, ranking-based information processing, and coding for flash memory. He is a recipient of the Robert T. Chien Memorial Award for demonstrating excellence in research in electrical engineering from the University of Illinois at Urbana-Champaign.

Frederic Sala received the B.S.E. degree in Electrical Engineering from the University of Michigan, Ann Arbor, in 2010 and the M.S. degree in Electrical Engineering from the University of California, Los Angeles (UCLA) in 2013. He is currently pursuing the Ph.D. degree in Electrical Engineering at UCLA, where he is associated with the LORIS and CoDESS labs.

His research interests include information theory and coding with a focus on error-correction codes, including applications to synchronization and data storage in non-volatile memories. He is a recipient of the NSF Graduate Research Fellowship and the UCLA Edward K. Rice Outstanding Masters Student Award.

Jehoshua Bruck (S'86–M'89–SM'93–F'01) is the Gordon and Betty Moore Professor of computation and neural systems and electrical engineering at the California Institute of Technology (Caltech). His current research interests include information theory and systems and the theory of computation in nature.

Dr. Bruck received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University, in 1989. His industrial and entrepreneurial experiences include working with IBM Research where he participated in the design and implementation of the first IBM parallel computer; cofounding and serving as Chairman of Rainfinity (acquired in 2005 by EMC), a spin-off company from Caltech that created the first virtualization solution for Network Attached Storage; as well as cofounding and serving as Chairman of XtremIO (acquired in 2012 by EMC), a start-up company that created the first scalable all-flash enterprise storage system.

Dr. Bruck is a recipient of the Feynman Prize for Excellence in Teaching, the Sloan Research Fellowship, the National Science Foundation Young Investigator Award, the IBM Outstanding Innovation Award and the IBM Outstanding Technical Achievement Award.

Lara Dolecek (S'05–M'10–SM'12) is an Associate Professor with the Electrical Engineering Department at the University of California, Los Angeles (UCLA). She holds a B.S. (with honors), M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences, as well as an M.A. degree in Statistics, all from the University of California, Berkeley. She received the 2007 David J. Sakrison Memorial Prize for the most outstanding doctoral research in the Department of Electrical Engineering and Computer Sciences at UC Berkeley. Prior to joining UCLA, she was a postdoctoral researcher with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. She received IBM Faculty Award (2014), Northrop Grumman Excellence in Teaching Award (2013), Intel Early Career Faculty Award (2013), University of California Faculty Development Award (2013), Okawa Research Grant (2013), NSF CAREER Award (2012), and Hellman Fellowship Award (2011). She is an Associate Editor for Coding Theory for IEEE TRANSACTIONS ON COMMUNICATIONS and served as an associate editor for IEEE COMMUNICATION LETTERS and as the lead guest editor for 2014 IEEE JSAC special issue on emerging data storage systems. Her research interests span coding and information theory, graphical models, statistical algorithms, and computational methods, with applications to emerging systems for data storage, processing, and communication. She is a Senior Member of IEEE.