# Insertion and Deletion Correction in Polymer-based Data Storage

Anisha Banerjee, Antonia Wachter-Zeh and Eitan Yaakobi

*Abstract*—Synthetic polymer-based storage seems to be a particularly promising candidate that could help to cope with the ever-increasing demand for archival storage requirements. It involves designing molecules of distinct masses to represent the respective bits $\{0, 1\}$, followed by the synthesis of a polymer of molecular units that reflects the order of bits in the information string. Reading out the stored data requires the use of a tandem mass spectrometer, that fragments the polymer into shorter substrings and provides their corresponding masses, from which the *composition*, i.e. the number of $1$s and $0$s in the concerned substring can be inferred. Prior works have dealt with the problem of unique string reconstruction from the set of all possible compositions, called *composition multiset*. This was accomplished either by determining which string lengths always allow unique reconstruction, or by formulating coding constraints to facilitate the same for all string lengths. Additionally, error-correcting schemes to deal with substitution errors caused by imprecise fragmentation during the readout process, have also been suggested. This work builds on this research by generalizing previously considered error models, mainly confined to substitution of compositions. To this end, we define new error models that consider insertions of spurious compositions and deletions of existing ones, thereby corrupting the composition multiset. We analyze if the reconstruction codebook proposed by Pattabiraman *et al.* is indeed robust to such errors, and if not, propose new coding constraints to remedy this.

*Index Terms*—Polymer-based data storage, string reconstruction, Composition errors, insertions, deletions

## I. INTRODUCTION

As we progress through this digital age, our rate of data generation continues to rise unhindered, and with it, so do our storage requirements. Since current data storage media are not particularly advantageous in regard to longevity or density, several molecular storage techniques [1]–[9] have been proposed. The work in [1] involving synthetic polymer-based storage systems appears to be especially favorable, given its promise of efficient synthesis, low read latency and cost. Under this paradigm, a string of information bits is encoded into a chain of molecules linked by means of phosphate bonds, such that the component molecules may only assume one of two significantly differing masses, which represent bits $0$ and $1$ respectively. The stored data can be read out by employing a tandem mass (MS/MS) spectrometer, which

essentially splits the synthesized polymer at the phosphate linkages and outputs the masses of the resulting fragments. In this manner, the user is given access to the masses of all substrings in the encoded string.

The previous work [10] dealt with the problem of reconstructing a binary string from such an MS/MS readout, under the following modeling assumptions:

*Assumption 1.* Masses of the component molecules are chosen such that one can always uniquely infer the *composition*, i.e., the number of $0$s and $1$s forming a certain fragment, from its mass.

*Assumption 2.* While fragmenting a polymer for the purpose of mass spectrometry analysis, the masses of all constituent substrings are observed with identical frequency.

This proposed setting simplifies the recovery of the original information string into the problem of binary string reconstruction from its composition multiset. More specifically, the reconstruction process now involves determining the binary string from a set of compositions of all of its substrings of each possible length. It is worth noting that this setup does not allow for differentiation between a string and its reversal, since their sets of substring compositions would be identical.

While the authors of [10] primarily focused on string lengths that ensured unique reconstruction from a composition multiset, subsequent works [11]–[13] extended this research by building a code that allows for unique reconstruction of each member codeword from its composition multiset alone, regardless of the string length. It was found that a redundancy proportional to the logarithm of the information length is sufficient to guarantee unique reconstruction. Similar coding constraints were also proposed to also cope with possible errors in the composition multiset. The work in [14] takes a step further by dealing with the recovery of multiple strings from the mass spectrometry readout of a mixture of synthesized polymers.

Since the errors introduced during an MS/MS readout are often context-dependent, we devote this work to the generalization of the error model considered in [11], [12]. Specifically, we investigate the impact of inserting and deleting one or more compositions on the reconstructability of the encoded strings. In addition to this, new coding constraints are proposed to enable the correction of such errors. We also consider a special kind of substitution error, namely a *skewed substitution error*. This category of errors is motivated by imperfect fragmentations of a given polymer during the MS/MS readout process, as a result of which the observed molecular mass of a shorter monomer chain is lower than what

the true mass of its perfectly fragmented version would have been. In this scenario, errors occur only in one direction, i.e., the the measured mass can only be lower than the true mass, not higher. An error-correcting scheme is also suggested for this setting.

The organization of this work is as follows. Section II introduces relevant terminology, notations and some preliminary results to be exploited subsequently. Section III discusses coding constructions proposed in earlier works [11]–[13], while Section IV describes the error models pertaining to insertions, deletions and skewed substitutions of one or multiple compositions and also briefly summarizes error-correcting codes to deal with the same. We demonstrate the equivalence between codes correcting deletions and insertions of multisets in Section V. Sections VI and VII delve deeper into the constructions capable of correcting deletions of multiple multisets. We also talk about skewed substitution errors and related coding constructions in Section VIII. Finally, we conclude with Section IX, where a few open problems are discussed.

## II. PRELIMINARIES

Let $s = s_1 s_2 \ldots s_n$ denote a binary string of $n$ bits. Any substring $s_i \ldots s_j$ where $i \leq j$, may be indicated by $s_i^j$. The *composition* of this substring, denoted by $c(s_i^j)$, is said to be $0^z 1^w$, where $z$ and $w$ refer to the number of 0s and 1s in $s_i^j$ respectively, such that $z + w = j - i + 1$. We also define $C_k(s)$ as the set of compositions of all length-$k$ substrings in $s$. Evidently, $C_k(s)$ should contain $n - k + 1$ compositions.

**Example 1.** *Consider $s = 001010111$. Then, the multiset of compositions for substrings of length 7 is given by: $C_7(s) = \{0^4 1^3, 0^3 1^4, 0^2 1^5\}$.*

Upon combining the multisets for all $1 \leq k \leq n$, we obtain the *composition multiset* of $s$:

$$C(s) = \bigcup_{k \in [n]} C_k(s).$$

where $[n] = \{1, \ldots, n\}$. As stated earlier, [10] determined string lengths for which unique reconstruction (up to reversal) from such sets is possible. For the remaining string lengths, the authors exploited a bivariate generating polynomial representation, to find strings that are equicomposable with a given string. Here, two distinct strings $s, t \in \{0,1\}^n$ are said to be *equicomposable* if a common composition multiset is shared, i.e., $C(s) = C(t)$.

A code $\mathcal{C}$ is called a *composition-reconstructable code* if for all $s, t \in \mathcal{C}$, it holds that $C(s) \neq C(t)$. For all $n$, denote by $A(n)$ the size of the largest composition reconstructable code. Since composition multisets are identical for a binary string and its reversal, it holds that

$$A(n) \leq 2^{\lceil \frac{n}{2} \rceil} + \frac{1}{2}(2^n - 2^{\lceil \frac{n}{2} \rceil}) = 2^{n-1} + 2^{\lceil \frac{n}{2} \rceil - 1},$$

where the term $2^{\lceil \frac{n}{2} \rceil}$ describes the number of palindromic strings of length $n$, and [10] determined string lengths $n$ where it is possible to achieve this bound with equality. Specifically,

it was shown that binary strings of length $\leq 7$, one less than a prime, or one less than twice a prime, are uniquely reconstructable up to reversal.

### A. Unique Reconstruction Codes

For values of $n$ where it is not possible to achieve the aforementioned bound, it is necessary to formulate a code, as done in [11], [12].

The first major coding-theoretic problem concerning polymer-based storage involved designing constraints in order to guarantee unique reconstruction for codewords of a fixed length, i.e., to formulate a composition-reconstructable code. To this end, [12] introduced the following composition-reconstructable code for even codeword lengths.

*Construction 1 [12]:*

$$\mathcal{S}_R(n) = \{s \in \{0,1\}^n, s_1 = 0, s_n = 1, \text{ and}$$
$$\exists I \subset \{2, \ldots, n-1\} \text{ such that}$$
$$\text{for all } i \in I, s_i \neq s_{n+1-i}, \quad (1)$$
$$\text{for all } i \notin I, s_i = s_{n+1-i},$$
$$s_{[n/2] \cap I} \text{ is a Catalan-Bertrand string.}\}$$

In this context, a Catalan-Bertrand string refers to any binary vector wherein each prefix contains strictly more 0s than 1s. When $n$ is odd, the codebook $\mathcal{S}_R(n)$ is defined as:

$$\mathcal{S}_R(n) = \bigcup_{s \in \mathcal{S}_R(n-1)} \{s_1^{(n-1)/2} 0 s_{(n+1)/2}\}^n, s_1^{(n-1)/2} 1 s_{(n+1)/2}^n\}. \quad (2)$$

The number of redundant bits can thus be upper-bounded in terms of $n$ as $1/2 \log(n) + 5$ [11]. Alternatively, we obtain the following statement from [12].

**Theorem 1.** *[12, pg. 3] There exist efficiently encodable and decodable reconstruction codes with $k$ information bits and redundancy at most $\frac{1}{2} \log(k) + 6$.*

From the definition of $A(n)$, we can also deduce that,

$$|\mathcal{S}_R(n)| \leq A(n).$$

This construction sets $s_1 = 0$ and $s_n = 1$ to avoid confusion among reversals, while the remaining bits are chosen such that the weight of a prefix and a suffix of equal length are unequal if the said prefix includes a Catalan-Bertrand string, i.e.,

$$\text{wt}(s_2^i) \begin{cases} = \text{wt}(s_{n-i+1}^{n-1}), & \text{if } [i] \cap I = \emptyset, \\ < \text{wt}(s_{n-i+1}^{n-1}), & \text{otherwise,} \end{cases} \quad (3)$$

where $i < \lceil \frac{n}{2} \rceil$ and $\text{wt}(\cdot)$ denotes the Hamming weight of the argument. The latter inequality stems from the fact that if $s_{[i] \cap I}$ has strictly more 0s than 1s, then $s_{\{n-i+1, \ldots, n-1\} \cap I}$ contains strictly more 1s than 0s, thus causing a weight mismatch. Here, we note that the embedded Catalan-Bertrand string may begin from index 2 at the earliest.

## B. Reconstruction from Error-Free Composition Multisets

The decoder of the composition-reconstructable code $\mathcal{S}_R(n)$ recovers a string from its composition multiset by employing the approach outlined in [10], [11]. Since the underlying principles of this process help us in formulating coding constructions for the more general error models involving insertions and deletions, we briefly discuss it in this subsection. For further details, the reader is referred to [10], [11].

The algorithm begins by deducing the following sequence that characterizes the string to be recovered, say $s \in \mathcal{S}_R(n)$,

$$\boldsymbol{\sigma}_s = (\sigma_1, \ldots, \sigma_{\lceil n/2 \rceil}),$$

where $\sigma_i = \text{wt}(s_i s_{n-i+1})$ for $i \in \{1, \ldots, \lfloor n/2 \rfloor\}$. When $n$ is odd, we set $\sigma_{\lceil \frac{n}{2} \rceil} = \text{wt}(s_{\lceil \frac{n}{2} \rceil})$, i.e., the weight of the central element.

**Example 2.** For $s = 001010111$. the sequence of $\sigma_i$'s is $\boldsymbol{\sigma}_s = (1, 1, 2, 0, 1)$.

These values can be computed by exploiting some inherent properties of composition multisets. In particular, we make use of *cumulative weights*, which are defined for each multiset $C_k(s)$ as:

$$w_k(s) = \sum_{0^z 1^w \in C_k(s)} w.$$

**Example 3.** For instance, the multiset $C_7(s) = \{0^4 1^3, 0^3 1^4, 0^2 1^5\}$ has a cumulative weight $w_7(s) = 12$.

It is easy to see that for all $k \leq \lceil \frac{n}{2} \rceil$, these weights obey the following relations:

$$w_1(s) = \sum_{i=1}^{\lceil \frac{n}{2} \rceil} \sigma_i, \tag{4}$$

$$w_k(s) = \sum_{i=1}^{k} i \sigma_i + k \sum_{i=k+1}^{\lceil n/2 \rceil} \sigma_i \tag{5}$$

$$= k w_1(s) - \sum_{i=1}^{k-1} i \sigma_{k-i}. \tag{6}$$

We also observe a symmetry relation for any given set of cumulative weights:

$$w_k(s) = w_{n-k+1}(s), \quad \forall\, k \in [n]. \tag{7}$$

In light of this, the multisets $C_i$ and $C_{n-i+1}$ are henceforth said to be *symmetric*. For notational convenience, we also define:

$$\widetilde{C}_i(s) = C_i(s) \cup C_{n-i+1}(s)$$

Now to demonstrate the functioning of the reconstruction algorithm, we consider the following example.

**Example 4.** *In this example, we reconstruct the string* $s = 001010111$ *from its composition multiset* $C(s)$, *which is stated below:*

$$
\begin{aligned}
C(s) = \{ & 0, 0, 1, 0, 1, 0, 1, 1, 1, 0^2, 0^1 1^1, 0^1 1^1, 0^1 1^1, 0^1 1^1, \\
& 0^1 1^1, 1^2, 1^2, 0^2 1^1, 0^2 1^1, 0^1 1^2, 0^2 1^1, 0^1 1^2, 0^1 1^2, \\
& 1^3, 0^3 1^1, 0^2 1^2, 0^2 1^2, 0^2 1^2, 0^1 1^3, 0^1 1^3, 0^3 1^2, \\
& 0^3 1^2, 0^2 1^3, 0^2 1^3, 0^1 1^4, 0^4 1^2, 0^3 1^3, 0^2 1^4, 0^2 1^4, \\
& 0^4 1^3, 0^3 1^4, 0^2 1^5, 0^4 1^4, 0^3 1^5, 0^4 1^5 \}.
\end{aligned} \tag{8}
$$

*The reconstruction process involves the following steps:*

1) *Firstly, we deduce its* $\boldsymbol{\sigma}_s$ *sequence from (4) and (6):*

$$\boldsymbol{\sigma}_s = (1, 1, 2, 0, 1).$$

2) *We create a multiset* $\mathcal{T}$ *to include all compositions that can be determined from* $\boldsymbol{\sigma}_s$. *More explicitly, one can infer the compositions* $c(s_5), c(s_4^6), \ldots, c(s_1^9)$ *by noting that for any* $i < \lceil n/2 \rceil$,

$$
c(s_i s_{n-i+1}) = \begin{cases} 0^2, & \text{if } \sigma_i = 0. \\ 0^1 1^1, & \text{if } \sigma_i = 1. \\ 1^2, & \text{if } \sigma_i = 2. \end{cases}
$$

$$\mathcal{T} = \{1, 0^2 1, 0^2 1^3, 0^3 1^4, 0^4 1^5\}.$$

3) *The process now assigns the bits of* $s$ *pairwise, in an inward manner, starting with bit pair* $(s_1, s_9)$. *Since* $\sigma_1 = 1$, *we could set* $s_1 = 0$ *and* $s_9 = 1$ *or vice-versa. Due to (1), we opt for the former, i.e.* $(s_1, s_9) = (0, 1)$.

4) *Using the reconstructed prefix and suffix, we update* $\mathcal{T}$:

$$\mathcal{T} = \{0, 1, 1, 0^2 1, 0^2 1^3, 0^3 1^4, 0^4 1^5, 0^3 1^5, 0^4 1^4\}.$$

5) *The two longest compositions in the multiset* $C(s) \backslash \mathcal{T}$ *are* $\{0^4 1^3, 0^2 1^5\}$. *These denote the compositions of substrings* $s_1^7$ *and* $s_3^9$. *Conversely, their complements* $\{1^2, 0^2\}$ *correspond to substrings* $s_1^2$ *and* $s_8^9$. *Combining this with the knowledge of bits* $s_1$ *and* $s_9$, *we reconstruct* $s$ *up to its prefix-suffix pair of length 2, i.e.* $(s_1^2, s_8^9) = (00, 11)$.

6) *To recover the remaining bits, we simply repeat steps 4 and 5.*

## III. SUBSTITUTION-CORRECTING CONSTRUCTIONS

We now turn our attention to the problem of reconstruction from erroneous composition multisets. Substitution errors were considered in [11] under the asymmetric and symmetric setting. In this error model, some compositions in $C(s)$ are arbitrarily altered. If the errors occur such that each multiset $\widetilde{C}_i$ includes at most one substituted composition, then they are said to be *asymmetric*. On the contrary, a pair of *symmetric* substitution errors would occur in the multisets $C_i$ and $C_{n-i+1}$, for any $i \in [n]$.

**Definition 1.** *A composition multiset* $C(s)$ *of the string* $s \in \{0, 1, \}^n$ *is said to have suffered an **asymmetric substitution error**, if for some* $i \in [n]$, *a single composition of the multiset*

$C_i(s)$ is modified, but its symmetric counterpart $C_{n-i+1}(s)$ remains unaffected.

**Definition 2.** *If a composition multiset $C(s)$ is corrupted by having one composition substituted in each of the multisets $C_i(s)$ and $C_{n-i+1}(s)$, then **two symmetric substitution errors** are said to have occurred.*

To exemplify this, we consider the following.

**Example 5.** *Let $s = 001010111$. The symmetric multiset pair $C_3(s)$ and $C_7(s)$ is given by*

$$C_3(s) = \{0^2 1, 0^2 1, 01^2, 0^2 1, 01^2, 01^2, 1^3\},$$
$$C_7(s) = \{0^4 1^3, 0^3 1^4, 0^2 1^5\}.$$

*For instance, an asymmetric substitution error is said to have occurred if $C_7(s)$ is corrupted to*

$$C_7'(s) = \{0^4 1^3, 0^3 1^4, 0^3 1^4\}.$$

*On the contrary, if $C_3(s)$ is also corrupted in addition to $C_7(s)$ as follows,*

$$C_3'(s) = \{1^3, 0^2 1, 01^2, 0^2 1, 01^2, 01^2, 1^3\},$$

*then two symmetric substitution errors are said to have occurred.*

We recall an important construction from [11] that corrects such composition substitution errors. In the following, we designate a code $\mathcal{S}_{CA}^{(t)}$ as a *t-asymmetric composition code*, if for all $s, v \in \mathcal{S}_{CA}^{(t)}$, there exists no $\mathcal{I} \subseteq [[\frac{n}{2}]]$ with $|\mathcal{I}| \leq t$ such that

$$|\widetilde{C}_i(s) \setminus \widetilde{C}_i(v)| = 1 \quad \forall\, i \in \mathcal{I},$$
$$\widetilde{C}_i(s) = \widetilde{C}_i(v) \quad \forall\, i \in \left[\left[\frac{n}{2}\right]\right] \setminus \mathcal{I}.$$

*Construction 2 [11], [12]:* A single (asymmetric or symmetric) composition code for odd values of $n$ is stated below.

$$\mathcal{S}_{CA}^{(1)}(n) = \{s = s_1 s_1^* s_2 \ldots s_{\lceil \frac{n-2}{2} \rceil} \ldots s_{n-3} s_n^* s_{n-2} \in \{0,1\}^n :$$
$$s_1 \ldots s_{n-2} \in \mathcal{S}_R(n-2), \operatorname{wt}(s) \bmod 2 = 0,$$
$$\sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(s) = 0 \bmod 3, \text{ where } s_1^* \leq s_n^*\}.$$

A similar construction exists for even $n$. The size of this code equals $\frac{|\mathcal{S}_R(n-2)|}{2}$. However, subsequently in Section VII we conclude by means of Lemma 7, that the code $\mathcal{S}_R(n)$ is also capable of correcting a single composition error.

*Construction 3 [11]:* A codebook $\mathcal{S}_{CA}^{(t)}(n)$ that is capable of rectifying $t$-asymmetric substitution errors is proposed in [11], and for the sake of brevity, we henceforth call it a $t$-asymmetric composition code. $\mathcal{S}_{CA}^{(t)}(n)$ constitutes all codewords $s = (\tilde{s}_1^{m/2} b_1^{n-m} \tilde{s}_{m/2+1}^m)$, such that the components $\tilde{s}_1^m$ and $b_1^{n-m}$ are constructed as follows:

- We choose $\tilde{s} = (\tilde{s}_1^{m/2} \tilde{s}_{m/2+1}^m) \in \mathcal{S}_R^{(t)}(m)$, described by the sequence $\sigma_{\tilde{s}}$.

$$\mathcal{S}_R^{(t)}(m) = \{s \in \{0,1\}^m, s_1^t = \mathbf{0}, s_{m-t+1}^m = \mathbf{1}, \text{ and}$$
$$\exists I \subset \{t+1, \ldots, m-t\} \text{ such that}$$
$$\text{for all } i \in I, s_i \neq s_{m+1-i}, \quad (9)$$
$$\text{for all } i \notin I, s_i = s_{m+1-i},$$
$$s_{[m/2] \cap I} \text{ is a Catalan-Bertrand string.}\}$$

- A systematic Reed-Solomon code over the alphabet $\{0, 1, 2\}$ is used to map $\sigma_{\tilde{s}}$ to a sequence $\sigma_s$ by appending the values $(\sigma_{m/2+1}, \ldots, \sigma_{n/2})$, which help to construct $b = b_1^{n-m}$ as follows:

$$b_k b_{n-k+1} = \begin{cases} 00, & \text{if } \sigma_{m/2+k} = 0. \\ 01, & \text{if } \sigma_{m/2+k} = 1. \\ 11, & \text{if } \sigma_{m/2+k} = 2. \end{cases}$$

where $k \in [(n-m)/2]$.

The upcoming construction, designed to correct substitution errors in symmetric multiset pairs, exploits a bivariate generating polynomial representation $P_s(x, y)$ of string $s$, that works as follows. Let the first term always be $\left(P_s(x, y)\right)_0 = 1$. Now by representing bits $0$ and $1$ as $y$ and $x$ respectively, we define the subsequent terms as:

$$\left(P_s(x, y)\right)_i = \begin{cases} y\left(P_s(x, y)\right)_{i-1}, & \text{if } s_i = 0 \\ x\left(P_s(x, y)\right)_{i-1}, & \text{if } s_i = 1. \end{cases}$$

**Example 6.** *For $s = 001010111$, the bivariate generating polynomial is given by $P_s(x, y) = 1 + y + y^2 + xy^2 + xy^3 + x^2 y^3 + x^2 y^4 + x^3 y^4 + x^4 y^4 + x^5 y^4$.*

The corresponding construction can be defined more explicitly as follows. A code $\mathcal{S}_{CS}^{(t)}$ is called a *t-symmetric composition code*, if for all $s, v \in \mathcal{S}_{CS}^{(t)}$, there exists no $\mathcal{I} \subseteq [[\frac{n}{2}]]$ with $|\mathcal{I}| \leq t$ such that

$$\left| \bigcup_{i \in \mathcal{I}} (\widetilde{C}_i(s) \setminus \widetilde{C}_i(v)) \right| \leq t,$$
$$\widetilde{C}_i(s) = \widetilde{C}_i(v) \quad \forall\, i \in \left[\left[\frac{n}{2}\right]\right] \setminus \mathcal{I}.$$

*Construction 4 [11]:* The authors of [11] also suggest a construction that corrects any $t$ symmetric composition substitutions in an entire composition multiset as follows.

$$\mathcal{S}_{CS}^{(t)}(n) = \{s \in \{0,1\}^n, \text{ s.t. } P_s(\alpha^{\ell_1}, \alpha^{\ell_2}) = a_{\ell_1, \ell_2},$$
$$\operatorname{wt}(s) \equiv a \mod (2t+1)\} \quad (10)$$

for all $\ell_1, \ell_2 \in \{0, 1, \ldots, 4t\}$, $a \in \{0, 1, \ldots, 2t\}$ and where $(a_{\ell_1, \ell_2})_{\ell_1, \ell_2 = 0}^{4t}$ denotes a random vector from $\mathbb{F}_q^{(4t+1)^2}$.

## IV. New Error Models

The subsequent sections explore error models that involve corrupting a valid composition multiset via the insertion or deletion of one or more multisets.

**Definition 3.** *An **asymmetric multiset deletion** is said to have occurred in the composition multiset $C(s)$ of a string $s \in$*

$\{0,1\}^n$, if for some $i \in [n]$, the multiset $C_i(s)$ is entirely missing, while $C_{n-i+1}(s)$ is uncorrupted.

**Definition 4.** *A **pair of symmetric multiset deletions** is said to have occurred if the composition multiset $C(s)$ of a string $s \in \{0,1\}^n$, if for some $i \in [n]$ such that $i \neq n-i+1$, the multisets $C_i(s)$ and $C_{n-i+1}(s)$ are entirely eliminated.*

**Example 7.** *Let $s = 001010111$. If the composition multiset $C(s)$ is corrupted to*

$$
\begin{aligned}
C'(s) = &\bigcup_{i \in [n] \setminus \{3\}} C_i(s), \\
= &\{0, 0, 1, 0, 1, 0, 1, 1, 1, 0^2, 0^1 1^1, 0^1 1^1, 0^1 1^1, 0^1 1^1, \\
&0^1 1^1, 1^2, 1^2, 0^3 1^1, 0^2 1^2, 0^2 1^2, 0^2 1^2, 0^1 1^3, 0^1 1^3, \\
&0^3 1^2, 0^3 1^2, 0^2 1^3, 0^2 1^3, 0^1 1^4, 0^4 1^2, 0^3 1^3, 0^2 1^4, \\
&0^2 1^4, 0^4 1^3, 0^3 1^4, 0^2 1^5, 0^4 1^4, 0^3 1^5, 0^4 1^5 \}.
\end{aligned}
$$

*then an asymmetric multiset deletion is said to have occurred. More specifically, the multiset $C_3(s) = \{0^2 1^1, 0^2 1^1, 0^1 1^2, 0^2 1^1, 0^1 1^2, 0^1 1^2, 1^3\}$ has been deleted. On the other hand, if*

$$
\begin{aligned}
C'(s) = &\bigcup_{i \in [n] \setminus \{3,7\}} C_i(s), \\
= &\{0, 0, 1, 0, 1, 0, 1, 1, 1, 0^2, 0^1 1^1, 0^1 1^1, 0^1 1^1, 0^1 1^1, \\
&0^1 1^1, 1^2, 1^2, 0^3 1^1, 0^2 1^2, 0^2 1^2, 0^2 1^2, 0^1 1^3, 0^1 1^3, \\
&0^3 1^2, 0^3 1^2, 0^2 1^3, 0^2 1^3, 0^1 1^4, 0^4 1^2, 0^3 1^3, 0^2 1^4, \\
&0^2 1^4, 0^4 1^4, 0^3 1^5, 0^4 1^5 \}.
\end{aligned}
$$

*we say that a pair of symmetric multiset deletions has occurred. Here compared to $C(s)$, we are missing the multisets $C_3(s) = \{0^2 1^1, 0^2 1^1, 0^1 1^2, 0^2 1^1, 0^1 1^2, 0^1 1^2, 1^3\}$ and $C_7(s) = \{0^4 1^3, 0^3 1^4, 0^2 1^5\}$.*

**Definition 5.** *A composition multiset $C(s)$ of a string $s \in \{0,1\}^n$ is said to have suffered a **composition insertion error**, if for some $i \in [n]$ the multiset $C_i(s)$ contains $n-i+2$ compositions, i.e. an unknown and invalid composition has been registered.*

**Example 8.** *Once again, let $s = 001010111$. If $C_7(s)$ has been altered as follows,*

$$
C_7'(s) = \{0^4 1^3, 0^3 1^4, 0^2 1^5, 0^1 1^6\}.
$$

*we say that a composition insertion error has taken place.*

The main contribution of this work consists of studying the aforementioned error models and proposing new coding constraints to combat the same. We also establish an equivalence between codes that correct composition insertions and composition deletions. Consequently, we restrict our attention to the latter for the remainder of this paper.

To this end, we first propose the following composition reconstruction code that allows for the correction of $t$ asymmetric multiset deletions. Specifically, a code $\mathcal{S}_{DA}^{(t)}$ is termed as a *t-asymmetric multiset deletion composition code*, if for

all $s, v \in \mathcal{S}_{DA}^{(t)}$, there exists no $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| \leq t$ such that for all $i \in \mathcal{I}$,

$$
\begin{aligned}
C_i(s) &\neq C_i(v), \\
C_{n-i+1}(s) &= C_{n-i+1}(v), \\
C_j(s) &= C_j(v) \quad \forall j \in [n] \setminus \mathcal{I}.
\end{aligned}
$$

*Construction 5:*

$$
\begin{aligned}
\mathcal{S}_{DA}^{(t)}(n) = &\{s \in \{0,1\}^n, s_1 = 0, s_n = 1, \text{ and} \\
&\exists I \subset \{2, \ldots, \frac{n}{2}\}, |I| \geq t, \text{ such that} \\
&\quad \forall\, i \in I, s_i \neq s_{n+1-i}, \\
&\quad \text{and } \forall i \notin I, s_i = s_{n+1-i}, \\
&s_{[n/2] \cap I} \text{ is a string wherein each} \\
&\text{prefix has at least } t \text{ more 0s than 1s.}\}
\end{aligned}
\tag{11}
$$

The corresponding proof follows behind Theorem 2. Evidently, this construction is inspired from (9), in that it requires at least $t$ 0s in $s_1^{n/2}$ and at least $t$ 1s in $s_{n/2+1}^n$, however their locations are not necessarily restricted as in (9). The extension to odd codeword lengths is similar to (2).

Following this, we investigate the case of symmetric multiset deletions, and discover that when two or more symmetric multiset pairs are missing, additional constraints are needed to bolster the code $S_R(n)$ so as to guarantee unique reconstructability. In this context, a code $\mathcal{S}_{DS}^{(t)}$ is termed as a *t-symmetric multiset deletion composition code*, if for all $s$, $v \in \mathcal{S}_{DS}^{(t)}$, there exists no $\mathcal{I} \subseteq \left[\left[\frac{n}{2}\right]\right]$ with $|\mathcal{I}| \leq t$ such that

$$
\begin{aligned}
\widetilde{C}_i(s) &\neq \widetilde{C}_i(v), \forall\, i \in \mathcal{I} \\
C_i(s) &= C_i(v) \quad \forall\, i \in \left[\left[\frac{n}{2}\right]\right] \setminus \mathcal{I}.
\end{aligned}
$$

For the elementary case of two deleted symmetric multiset pairs, we propose the following code.

*Construction 6:*

$$
\begin{aligned}
\mathcal{S}_{DS}^{(2)}(n) = &\{s \in \mathcal{S}_R(n), \\
&\sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(s) \bmod 7 = a, \ 0 \leq a \leq 6\}.
\end{aligned}
\tag{12}
$$

Theorem 8 proves that this code can indeed correct the deletion of two symmetric multiset pairs. We also generalize this construction to accommodate for the deletion of any $t$ consecutive symmetric multiset pairs, where $t \geq 2$. More explicitly, a code $\mathcal{S}_{DS}'^{(t)}$ is termed as a *t-symmetric consecutive multiset deletion composition code*, if for all $s, v \in \mathcal{S}_{DS}'^{(t)}$, there exists no $\mathcal{I} = \{i, i+1, \ldots i+p-1\} \subseteq \left[\left[\frac{n}{2}\right]\right]$ with $p \leq t$ such that

$$
\begin{aligned}
\widetilde{C}_j(s) &\neq \widetilde{C}_j(v), \forall\, j \in \mathcal{I} \\
C_j(s) &= C_j(v) \quad \forall\, j \in \left[\left[\frac{n}{2}\right]\right] \setminus \mathcal{I}.
\end{aligned}
$$

*Construction 7:*

$$\mathcal{S}_{DS}^{\prime(t)}(n) = \{\boldsymbol{s} \in \mathcal{S}_R(n), \sum_{i=1}^{\frac{m}{2}} w_i(\boldsymbol{s}) \bmod A = a, \qquad (13)$$
$$0 \le a \le A - 1\}$$

where $t \ge 2$ and

$$A = \left\lceil \frac{4t^3}{3} + \frac{2t}{3} - \frac{31}{4} \right\rceil.$$

Theorem 11 proves that $\mathcal{S}_{DS}^{\prime(t)}(n)$ is capable of correcting the deletion of $t$ consecutive symmetric multiset pairs.

**Definition 6.** *A composition multiset $C(\boldsymbol{s})$ of the string $\boldsymbol{s} \in \{0, 1,\}^n$ is said to have suffered an **asymmetric skewed substitution error**, if for some $i \in [n]$, a single composition of multiset $C_i(\boldsymbol{s})$ is replaced with one of a lower Hamming weight, such that the symmetric counterpart $C_{n-i+1}(\boldsymbol{s})$ remains unaffected.*

**Example 9.** *For instance, if an erroneous measurement corrupts the composition $0^2 1^4$, the measured compositions could be $0^3 1^3$ or $0^4 1^2$, but not $0^1 1^5$.*

Formally, a code $\mathcal{C}^{\prime(t)}$ is referred to as a *t-asymmetric skewed composition code*, if for all $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{C}^{\prime(t)}$, there exists no $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| \le t$ such that for all $i \in \mathcal{I}$,

$$C_i(\boldsymbol{s}) \ne C_i(\boldsymbol{v}),$$
$$C_{n-i+1}(\boldsymbol{s}) = C_{n-i+1}(\boldsymbol{v}),$$
$$C_j(\boldsymbol{s}) = C_j(\boldsymbol{v}) \quad \forall j \in [n] \setminus \mathcal{I}$$

We subsequently prove in Lemma 7 of Section VIII that the code $\mathcal{S}_{DA}^{(t)}(n)$ (Construction 5) is sufficiently robust to allow the correction of $t$ skewed asymmetric substitution errors in its composition set.

These results, along with the earlier constructions proposed in [11]–[13], have been summarized in Table I.

## V. CODE EQUIVALENCE: INSERTION AND DELETION OF MULTISETS

In this section, we demonstrate how codes which can correct the deletion of a group of $t$ multisets, can also correct the occurrence of insertion errors in those $t$ multisets.

**Lemma 1.** *A code can correct the deletion of $t$ composition multisets, if and only if it can correct any number of composition insertion errors in those $t$ multisets.*

*Proof.* We prove this by contradiction. Let there be two binary strings $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R(n)$, such that:

$$D_t(\boldsymbol{s}) \cap D_t(\boldsymbol{v}) \ne \emptyset. \qquad (14)$$

where $D_t(\boldsymbol{s})$ constitutes all codewords in $\mathcal{S}_R(n)$ that $\boldsymbol{s}$ becomes equicomposable with upon the deletion of at most $t$ multisets, i.e.,

$$D_t(\boldsymbol{s}) = \{\boldsymbol{u} \in \mathcal{S}_R(n) \text{ such that } \exists \mathcal{I} \subseteq [n], |\mathcal{I}| \le t,$$
$$\bigcup_{i \in [n] \setminus \mathcal{I}} C_i(\boldsymbol{s}) = \bigcup_{i \in [n] \setminus \mathcal{I}} C_i(\boldsymbol{u})\}.$$

Equation (14) implies that at least $n - t$ composition multisets of $\boldsymbol{s}$ and $\boldsymbol{v}$ are identical. In other words, when a specific group of $t$ multisets disappears from the multiset information of $\boldsymbol{s}$ and $\boldsymbol{v}$, they become indistinguishable. Let these differing multisets correspond to substring lengths $i_1, i_2, \ldots i_t$. This allows us to write that:

$$\bigcup_{j \in [n] \setminus \{i_1, \ldots i_t\}} C_j(\boldsymbol{s}) = \bigcup_{j \in [n] \setminus \{i_1, \ldots i_t\}} C_j(\boldsymbol{v}).$$

If we perform a set union operation on both sides of the previous equation with $\bigcup_{i \in \{i_1, \ldots i_t\}} C_i(\boldsymbol{s}) \cup C_i(\boldsymbol{v})$, then we get:

$$\bigcup_{i \in \{i_1, \ldots i_t\}} (C_j(\boldsymbol{v}) \backslash C_j(\boldsymbol{s})) \cup \bigcup_{j \in [n]} C_j(\boldsymbol{s})$$
$$= \bigcup_{i \in \{i_1, \ldots i_t\}} (C_j(\boldsymbol{s}) \backslash C_j(\boldsymbol{v})) \cup \bigcup_{j \in [n]} C_j(\boldsymbol{v}).$$

This effectively means that if the multisets $C_{i_1}(\boldsymbol{s}), \ldots C_{i_t}(\boldsymbol{s})$ are corrupted by the insertion of some specific erroneous compositions, then the multiset information may correspond to both $\boldsymbol{s}$ and $\boldsymbol{v}$, and vice-versa. This lets us write the following:

$$I_t(\boldsymbol{s}) \cap I_t(\boldsymbol{v}) \ne \emptyset. \qquad (15)$$

where $I_t(\boldsymbol{s})$ denotes the set of all codewords $\boldsymbol{u} \in \mathcal{S}_R(n)$ whose composition multisets, upon suffering any number of insertion errors in at most $t$ distinct multisets, resemble $C(\boldsymbol{s})$ after corruption by certain composition insertions in those affected multisets. In other words, at least $n - t$ distinct multisets of $\boldsymbol{s}$ and $\boldsymbol{u}$ are identical. Consequently, we can write

$$I_t(\boldsymbol{s}) = D_t(\boldsymbol{s})$$
$$= \{\boldsymbol{u} \in \mathcal{S}_R(n) \text{ such that } \exists \mathcal{I} \subseteq [n], |\mathcal{I}| \le t,$$
$$\forall i \in [n] \setminus \mathcal{I}, \ C_i(\boldsymbol{s}) = C_i(\boldsymbol{u})\}$$

$\square$

Owing to this result, we deem it sufficient to focus on multiset deletion-correcting codes. The subsequent sections examine how multiset deletions affect the reconstructability of an encoded string drawn from $\mathcal{S}_R(n)$. Similar to [11], we categorize such deletion errors into two major settings.

## VI. ASYMMETRIC MULTISET DELETION-CORRECTING COMPOSITION-RECONSTRUCTION CODES

We begin by considering an error model where a complete multiset $C_k(\boldsymbol{s})$ can be deleted from the composition multiset $C(\boldsymbol{s})$. This is formally referred to as a single asymmetric multiset deletion [see Definition 3]. We investigate whether the reconstruction codebook [see Construction 1] guarantees unique recoverability under this model. To proceed in this direction, we first take note of the following lemma, which results from a specific case of [11, Lemma 4].

**Lemma 2.** *Let $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R(m)$ share the same $\boldsymbol{\sigma}$ sequence and satisfy $|C_j(\boldsymbol{s}) \backslash C_j(\boldsymbol{v})| \le 2$ for all $j \in [m]$. If the longest prefix-suffix pair shared by $\boldsymbol{s}$ and $\boldsymbol{v}$ is of length $i$, then their*

| Code | Symbol | Upper bound on redundancy | Proof |
|---|---|---|---|
| Composition-reconstructable code | $\mathcal{S}_R(n)$ | $\frac{1}{2}\log_2 n + 5$ | [11], [12] |
| Single composition error-correcting code | $\mathcal{S}_{CA}^{(1)}(n)$ | $\frac{1}{2}\log_2(n-2) + 8$ | [11], [12] |
| $t$-asymmetric composition code | $\mathcal{S}_{CA}^{(t)}(n)$ | $\left(\frac{1}{2} + 3t\right)\log_2 n + 2t + 5$ | [11] |
| $t$-composition code | $\mathcal{S}_{CS}^{(t)}(n)$ | $156t^2 \log_2 n$ | [11], [13] |
| $t$-asymmetric multiset deletion composition code | $\mathcal{S}_{DA}^{(t)}(n)$ | $\frac{1}{2}\log_2(n-2t) + 2t + 3$ | Th. 2 |
| 2-symmetric multiset deletion composition code | $\mathcal{S}_{DS}^{(2)}(n)$ | $\frac{1}{2}\log_2(n-2) + 8$ | Th. 8 |
| $t$-symmetric consecutive multiset deletion composition code | $\mathcal{S}_{DS}'^{(t)}(n)$ | $\frac{1}{2}\log_2(n-2)$ $+ \log_2\left\lceil \frac{4t^3}{3} + \frac{2t}{3} - \frac{31}{4}\right\rceil + 5$ | Th. 11 |

Table I: Summary of constructions

*corresponding composition multisets $C_{m-i-1}$ and $C_{m-i-2}$ each differ in at least 2 compositions.*

To shortly highlight the implications of this lemma, we consider the strings $s = 001011101$ and $v = 001110101$. Clearly, they are both specified by $\sigma = (1,0,2,1,1)$. Since the longest prefix-suffix pair shared by them is $(001, 101)$, i.e., of length 3, their respective multisets $C_4$ and $C_5$ differ by at least 2 compositions.

**Lemma 3.** *Consider a string $s \in \mathcal{S}_R(n)$. Given $C'(s) = \bigcup_{i \in [n] \backslash \{k\}} C_i(s)$ for any $k \in [n]$, $s$ can be fully recovered.*

*Proof.* **Case 1.** $n$ is even

From the steps of the reconstruction algorithm as described in Section II-B, it is evident that we only require the composition multisets $C_n(s), \ldots, C_{\frac{n}{2}}(s)$. Hence, if $k < \frac{n}{2}$, the reconstruction of $s$ is straightforward. On the contrary, if $k \geq \frac{n}{2}$, one can still infer the cumulative weight of the missing multiset $C_k(s)$ from (7). Consequently, $\sigma_s$ can be obtained accurately.

In the absence of $C_k(s)$, the prefix and suffix can be constructed upto $s_1^{n-k-1}$ and $s_{k+2}^n$. When $\sigma_k = \sigma_{n-k+1} \in \{0,2\}$, there remains no ambiguity concerning the bits $s_{n-k}$ and $s_{k+1}$. However, when $\sigma_k = 1$, one can either have $(s_{n-k}, s_{k+1}) = (0,1)$ or $(s_{n-k}, s_{k+1}) = (1,0)$ if both of these possibilities guarantee weight mismatch between $s_1^{n-k}$ and $s_{k+1}^n$. Now since $s \in \mathcal{S}_R(n)$, Lemma 2 tells us that choosing the bits $s_{n-k}$ and $s_{k+1}$ incorrectly, will lead to an incompatibility with the multiset $C_{k-1}(s)$. Thus there exists only one valid choice for these bits, implying that $s$ is uniquely recoverable.

**Case 2.** $n$ is odd

Similar to the previous case, it can be argued that for any missing composition multiset $C_k(s)$, where $k \neq \lceil \frac{n}{2} \rceil$, $s$ can be easily and uniquely determined. The more interesting case occurs when $k = \lceil \frac{n}{2} \rceil$, since the absence of $C_{\lceil \frac{n}{2} \rceil}(s)$, and thus $w_{\lceil \frac{n}{2} \rceil}(s)$, prevents us from computing $\sigma_{\lceil \frac{n}{2} \rceil - 1}$ and $\sigma_{\lceil \frac{n}{2} \rceil}$. However, their sum is known from (4), i.e.

$$\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} = w_1(s) - \sum_{i=1}^{\lceil \frac{n}{2} \rceil - 2} \sigma_i. \qquad (16)$$

Since $\sigma_{\lceil \frac{n}{2} \rceil - 1} = \mathrm{wt}(s_{\lceil \frac{n}{2} \rceil - 1} s_{\lceil \frac{n}{2} \rceil + 1}) \in \{0,1,2\}$ and $\sigma_{\lceil \frac{n}{2} \rceil} = \mathrm{wt}(s_{\lceil \frac{n}{2} \rceil}) \in \{0,1\}$, these values can be inferred directly when $\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} \in \{0,3\}$. However, an ambiguity arises when $\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} \in \{1,2\}$.

Let $v \in \mathcal{S}_R(n)$ be a string with which $s$ becomes equicomposable when the multiset $C_{\lceil n/2 \rceil}$ is deleted, i.e.,

$$\bigcup_{i \in [n] \backslash \{\lceil \frac{n}{2} \rceil\}} C_i(s) = \bigcup_{i \in [n] \backslash \{\lceil \frac{n}{2} \rceil\}} C_i(v). \qquad (17)$$

Also, let $v$ be specified by $\sigma_v = (\sigma_1', \ldots, \sigma_{\lceil n/2 \rceil}')$. As a consequence of (17), we can write:

$$\sigma_i = \sigma_i', \quad \forall\ 1 \leq i \leq \left\lceil \frac{n}{2} \right\rceil - 2$$
$$\sigma_{\lceil \frac{n}{2} \rceil - 1} + \sigma_{\lceil \frac{n}{2} \rceil} = \sigma_{\lceil \frac{n}{2} \rceil - 1}' + \sigma_{\lceil \frac{n}{2} \rceil}'. \qquad (18)$$

To verify whether the reconstructability of $s$ is affected, we simply check if there exists a suitable $v$ that satisfies (17) and (18). We also note that (17) directly implies the equality of the prefix-suffix pairs $(s_1^{\lceil \frac{n}{2} \rceil - 2}, s_{\lceil \frac{n}{2} \rceil + 2}^n) = (v_1^{\lceil \frac{n}{2} \rceil - 2}, v_{\lceil \frac{n}{2} \rceil + 2}^n)$.

Figure 1: Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\lceil\frac{n}{2}\rceil-2}, \boldsymbol{s}_{\lceil\frac{n}{2}\rceil+2}^n) = (\boldsymbol{v}_1^{\lceil\frac{n}{2}\rceil-2}, \boldsymbol{v}_{\lceil\frac{n}{2}\rceil+2}^n)$, where $v_+ = 1 - v_-$.

We jointly depict the specific subcases in Fig. 1, wherein we allow for $\sigma_{\lceil\frac{n}{2}\rceil-1} + \sigma_{\lceil\frac{n}{2}\rceil} \in \{1,2\}$ since for both $\boldsymbol{s}$ and $\boldsymbol{v}$, we have:

$$\sigma_{\lceil\frac{n}{2}\rceil-1} + \sigma_{\lceil\frac{n}{2}\rceil} = 2 - b.$$

where $b \in \mathbb{F}_2$. To proceed with the proof, we try to determine the conditions under which $C_{\lceil\frac{n}{2}\rceil-1}(\boldsymbol{s}) = C_{\lceil\frac{n}{2}\rceil-1}(\boldsymbol{v})$ holds. This would require the following set equality:

$$\left\{\begin{array}{l} \{c(\boldsymbol{s}_1^{\lceil\frac{n}{2}\rceil-2}), 1-b\} \\ \{c(\boldsymbol{s}_2^{\lceil\frac{n}{2}\rceil-2}), b, 1-b\} \\ \{c(\boldsymbol{s}_{\lceil\frac{n}{2}\rceil+2}^n), 1-b\} \\ \{c(\boldsymbol{s}_{\lceil\frac{n}{2}\rceil+2}^{n-1}), b, 1-b\} \end{array}\right\} = \left\{\begin{array}{l} \{c(\boldsymbol{v}_1^{\lceil\frac{n}{2}\rceil-2}), v_+\} \\ \{c(\boldsymbol{v}_2^{\lceil\frac{n}{2}\rceil-2}), v_+, 1-b\} \\ \{c(\boldsymbol{v}_{\lceil\frac{n}{2}\rceil+2}^n), 1-v_+\} \\ \{c(\boldsymbol{v}_{\lceil\frac{n}{2}\rceil+2}^{n-1}), 1-v_+, 1-b\}. \end{array}\right\}.$$

By checking the above relation exhaustively for all possibilities of $(b, v_+) \in \{0,1\}^2$, we conclude that the multisets $C_{\lceil n/2\rceil-1}(\boldsymbol{s})$ and $C_{\lceil n/2\rceil-1}(\boldsymbol{s})$ can never match. Therefore, $\boldsymbol{v}$ does not exist and $\boldsymbol{s}$ retains its unique reconstructability. $\qquad\square$

It follows directly from the preceding lemma that

**Lemma 4.** *The code $\mathcal{S}_R(n)$ is a single asymmetric multiset deletion composition code.*

As a second step, $\mathcal{S}_R(n)$ is now generalized to $\mathcal{S}_{DA}^{(t)}(n)$ [see Construction 5] to allow correcting the deletion of $t$ asymmetric multisets. To prove why this construction works, we first consider the following lemma.

**Lemma 5.** *Let $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_{DA}^{(t)}(n)$ be specified by an identical $\boldsymbol{\sigma}$ sequence, such that the longest prefix-suffix pair shared by them is of length $k$. Then their corresponding multisets $C_{n-i-1}, \ldots, C_{n-i-t-1}$ differ by at least two compositions.*

*Proof.* Since $\boldsymbol{s}$ and $\boldsymbol{v}$ bear the same $\boldsymbol{\sigma}$ sequence and their prefix-suffix pair of length $k+1$ do not match, we conclude that $\sigma_{k+1} = 1$ and $s_{k+1} \neq v_{k+1}$. Without loss of generality, we assume $s_{k+1} = 0$ and it becomes obvious that $|C_{n-k-1}(\boldsymbol{s}) \backslash C_{n-k-1}(\boldsymbol{v})| = 2$.



Figure 2: Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are related such that $(\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) = (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n)$ and $c(\boldsymbol{s}_{k+2}^{n-k-1}) = c(\boldsymbol{v}_{k+2}^{n-k-1})$

As for the remaining multisets, we undertake the approach used in [11, Lemma 4], i.e., we design a set of strings $\mathcal{V}_{\boldsymbol{s}}$,

such that for each $\boldsymbol{v} \in \mathcal{V}_{\boldsymbol{s}}$, $\boldsymbol{s}$ and $\boldsymbol{v}$ are specified by the same $\boldsymbol{\sigma}$ sequence, and satisfy:

$$\begin{aligned} (\boldsymbol{s}_1^k, \boldsymbol{s}_{n-k+1}^n) &= (\boldsymbol{v}_1^k, \boldsymbol{v}_{n-k+1}^n), \\ c(\boldsymbol{s}_{t+2}^{n-t-1}) &= c(\boldsymbol{v}_{t+2}^{n-t-1}), \qquad (19) \\ |C_{n-k-j}(\boldsymbol{s}) \backslash C_{n-k-j}(\boldsymbol{v})| &\leq 2, \qquad \forall\, j \in [t+1]. \end{aligned}$$

Equation (19) follows directly from the premise of a common $\boldsymbol{\sigma}$ sequence. Similar to [11, Lemma 4], we note that $|C_{n-k-2}(\boldsymbol{s}) \backslash C_{n-k-2}(\boldsymbol{v})|$ is minimized when $\sigma_{k+2} = 1$ and $(s_+, v_+) = (1,0)$, thereby leading to $|C_{n-k-2}(\boldsymbol{s}) \backslash C_{n-k-2}(\boldsymbol{v})| = 2$. Now, if an additional condition is upheld:

$$(\boldsymbol{s}_{k+3}^{t+1}, \boldsymbol{s}_{n-t}^{n-k-2}) = (\boldsymbol{v}_{k+3}^{t+1}, \boldsymbol{v}_{n-t}^{n-k-2}). \qquad (20)$$

we can show that $|C_{n-k-j}(\boldsymbol{s}) \backslash C_{n-k-j}(\boldsymbol{v})| = 2$ for any $j \in [t+1]$, by examining the following set equality:

$$\left\{\begin{array}{l} \{c(\boldsymbol{s}_1^k), 01, c(\boldsymbol{s}_{k+3}^{n-k-j})\} \\ \{c(\boldsymbol{s}_2^k), 01, c(\boldsymbol{s}_{k+3}^{n-k-j+1})\} \\ \vdots \\ \{c(\boldsymbol{s}_{j-1}^k), 01, c(\boldsymbol{s}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{s}_j^k), 0^21, c(\boldsymbol{s}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{s}_{n-k+1}^n), 01, c(\boldsymbol{s}_{k+j+1}^{n-k-2})\} \\ \{c(\boldsymbol{s}_{n-k+1}^{n-1}), 01, c(\boldsymbol{s}_{k+j}^{n-k-2})\} \\ \vdots \\ \{c(\boldsymbol{s}_{n-k+1}^{n-j+2}), 01, c(\boldsymbol{v}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{s}_{n-k+1}^{n-j+1}), 01^2, c(\boldsymbol{v}_{k+3}^{n-k-2})\} \end{array}\right\}$$
$$= \left\{\begin{array}{l} \{c(\boldsymbol{v}_1^k), 01, c(\boldsymbol{v}_{k+3}^{n-k-j})\} \\ \{c(\boldsymbol{v}_2^k), 01, c(\boldsymbol{v}_{k+3}^{n-k-j+1})\} \\ \vdots \\ \{c(\boldsymbol{v}_{j-1}^k), 01, c(\boldsymbol{v}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{v}_j^k), 01^2, c(\boldsymbol{v}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{v}_{n-k+1}^n), 01, c(\boldsymbol{v}_{k+j+1}^{n-k-2})\} \\ \{c(\boldsymbol{v}_{n-k+1}^{n-1}), 01, c(\boldsymbol{v}_{k+j}^{n-k-2})\} \\ \vdots \\ \{c(\boldsymbol{v}_{n-k+1}^{n-j+2}), 01, c(\boldsymbol{v}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{v}_{n-k+1}^{n-j+1}), 0^21, c(\boldsymbol{v}_{k+3}^{n-k-2})\} \end{array}\right\}. \qquad (21)$$

By exploiting (19) and (20), one can simplify this to:

$$\left\{\begin{array}{l} \{c(\boldsymbol{s}_j^k), 0^21, c(\boldsymbol{s}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{s}_{n-k+1}^{n-j+1}), 01^2, c(\boldsymbol{s}_{k+3}^{n-k-2})\} \end{array}\right\} = \left\{\begin{array}{l} \{c(\boldsymbol{s}_j^k), 01^2, c(\boldsymbol{s}_{k+3}^{n-k-2})\} \\ \{c(\boldsymbol{s}_{n-k+1}^{n-j+1}), 0^21, c(\boldsymbol{s}_{k+3}^{n-k-2})\} \end{array}\right\}.$$

Upon combining (19) and (20), further reduction is possible:

$$\left\{\begin{array}{l} \{c(\boldsymbol{s}_j^k), 0^21\} \\ \{c(\boldsymbol{s}_{n-k+1}^{n-j+1}), 01^2\} \end{array}\right\} = \left\{\begin{array}{l} \{c(\boldsymbol{s}_j^k), 01^2\} \\ \{c(\boldsymbol{s}_{n-k+1}^{n-j+1}), 0^21\} \end{array}\right\}. \qquad (22)$$

We note that the preceding equality only holds if:

$$c(\boldsymbol{s}_j^k) = c(\boldsymbol{s}_{n-k+1}^{n-j+1})$$

However from Fig. 2 and the definition of $\mathcal{S}_R^{*(t)}(n)$ in (11), we observe that:

$$\mathrm{wt}(\boldsymbol{v}_2^{k+1}) + t \le \mathrm{wt}(\boldsymbol{v}_{n-k+1}^{n-1})$$
$$\implies \mathrm{wt}(\boldsymbol{v}_1^k) + t + 2 \le \mathrm{wt}(\boldsymbol{v}_{n-k+1}^n).$$

This inequality allows us to conclude that (22) never holds for any $j \in [t+1]$, consequently proving the statement of this lemma. $\square$

The preceding lemma now helps us establish that the code $\mathcal{S}_{DA}^{(t)}(n)$ is robust to the deletion of any $t$ asymmetric multisets.

**Theorem 2.** *Given the composition multisets $C_i(\boldsymbol{s})$ for $i \in [n] \setminus \{i_1, \dots i_t\}$, where $\boldsymbol{s} \in \mathcal{S}_{DA}^{(t)}(n)$ [see Construction 5], such that no two of the deleted multisets are mutually symmetric, $\boldsymbol{s}$ can be uniquely recovered.*

*Proof.* **Case 1.** The deleted multisets are consecutive. This case is directly implied by Lemma 5.
**Case 2.** All of the deleted multisets are not consecutive.
Since the reconstruction algorithm functions in an outside-in manner, the missing multiset encountered first, corresponds to that of highest substring length. In the following analysis, we assume that $i_t > i_{t-1} > \dots > i_1$.

If $i_t = n$, we can directly infer $C_n(\boldsymbol{s})$ from the cumulative weight of $C_1(\boldsymbol{s})$. Alternatively when $i_t < n$ and additionally $i_t, \dots, i_{t-j+1}$ are consecutive, the prefix-suffix pair $(\boldsymbol{s}_1^{n-i_t-1}, \boldsymbol{s}_{i_t+2}^n)$ an incorrect assignment of the bit pair $(s_{n-i_t}, s_{i_t+1})$ will certainly cause an incompatibility with the multiset $C_{i_{t-j+1}-1}(\boldsymbol{s}) = C_{i_{t-j}}(\boldsymbol{s})$, as Lemma 5 suggests. Thus, the backtracking algorithm can detect the mistake and accurately reconstruct the string upto $(\boldsymbol{s}_1^{n-i_t+j}, \boldsymbol{s}_{i_t-j+1}^n)$. Absence of the other missing multisets $C_{i_{t-j}}, \dots, C_{i_1}$ can be dealt with similarly. $\square$

The previous theorem implies the following.

**Theorem 3.** $\mathcal{S}_{DA}^{(t)}(n)$ *is a $t$-asymmetric multiset deletion composition code.*

We also bound the number of redundant bits required by $\mathcal{S}_{DA}^{(t)}$ as follows.

**Lemma 6.** *The code $\mathcal{S}_{DA}^{(t)}$ requires at most $\frac{1}{2}\log(n-2t) + 2t + 3$ bits of redundancy.*

*Proof.* We refer to (11) and additionally recount from [11] that $\frac{1}{2}\binom{2h}{h}$ indicates the number of all strings of length $2h$ wherein every prefix of which contains strictly more 0s than 1s. For odd lengths $2h+1$, this term serves as a lower bound. Similarly, to count all strings $\boldsymbol{s} \in \{0,1\}^p$ wherein each prefix (of length exceeding $t$) contains at least $t$ more 0s than 1s, we simply note that such strings satisfy $\boldsymbol{s}_1^{t-1} = \boldsymbol{0}$ and $\boldsymbol{s}_t^p$ should

be a standard Catalan-Bertrand string. By virtue of this, we derive a lower bound on dimension of the codebook:

$$|\mathcal{S}_{DA}^{(t)}(n)| \ge \sum_{i=t}^{n/2-1} 2^{n/2-2-i} \binom{n/2-1}{i} \binom{i-t+1}{\lfloor (i-t+1)/2 \rfloor}.$$

After some algebraic manipulation of this expression, we conclude that the maximum number of redundant bits necessary is $\frac{1}{2}\log(n-2t) + 2t + 3$. $\square$

## VII. SYMMETRIC MULTISET DELETION-CORRECTING COMPOSITION-RECONSTRUCTION CODES

As mentioned in Section IV, errors under this category occur in such a way that the affected multisets occur in pairs. We begin directly with the case when two symmetric multisets are inaccessible.

**Lemma 7.** *Consider a string $\boldsymbol{s} \in \mathcal{S}_R(n)$. Assume that for any $1 \le k \le \lceil \frac{n-1}{2} \rceil$, one is given $C'(\boldsymbol{s}) = \bigcup_{i \in [n] \setminus \{k, n-k+1\}} C_i(\boldsymbol{s})$. Then, $\boldsymbol{s}$ can be fully recovered.*

*Proof.* **Case 1.** $n$ is odd.

Since the deleted multisets $C_k(\boldsymbol{s})$ and $C_{n-k+1}(\boldsymbol{s})$ can never be consecutive when $n$ is odd, we can infer from [11, Lemma 4] that any attempt to substitute $C_{n-k+1}(\boldsymbol{s})$ with another multiset, say $C'_{n-k+1}$, that may or may not preserve the value of $\sigma_{k-1}(\boldsymbol{s})$, will surely cause a disagreement with $C_{n-k}(\boldsymbol{s})$. Hence, there exists no valid alternative choices for the multiset pair $\{C_k(\boldsymbol{s}), C_{n-k+1}(\boldsymbol{s})\}$, thus implying that $\boldsymbol{s}$ is uniquely reconstructable.

**Case 2.** $n$ is even.

As in the previous case, we can argue that for any $k \ne \{\frac{n}{2}, \frac{n}{2}+1\}$, i.e., when the missing multisets are non-consecutive, $\boldsymbol{s}$ remains unique reconstructable by virtue of [11, Lemma 4]. The only case left to be analyzed is when the deleted multisets are adjacent, i.e $C_{\frac{n}{2}}(\boldsymbol{s})$ and $C_{\frac{n}{2}+1}(\boldsymbol{s})$. More specifically, we examine the existence of any $\boldsymbol{v} \in \mathcal{S}_R(n)$, such that

$$\bigcup_{i \in [n] \setminus \{\frac{n}{2}, \frac{n}{2}+1\}} C_i(\boldsymbol{v}) = \bigcup_{i \in [n] \setminus \{\frac{n}{2}, \frac{n}{2}+1\}} C_i(\boldsymbol{s}).$$

This directly leads to the following relations:

$$(\boldsymbol{s}_1^{n/2-2}, \boldsymbol{s}_{n/2+3}^n) = (\boldsymbol{v}_1^{n/2-2}, \boldsymbol{v}_{n/2+3}^n),$$
$$\sigma_i = \sigma'_i, \quad \forall\ 1 \le i \le \frac{n}{2} - 2$$
$$\sigma_{\frac{n}{2}-1} + \sigma_{\frac{n}{2}} = \sigma'_{\frac{n}{2}-1} + \sigma'_{\frac{n}{2}}.$$

where the sequence $\boldsymbol{\sigma}_{\boldsymbol{v}} = (\sigma'_1, \dots, \sigma'_{n/2})$ describes $\boldsymbol{v}$.
*Subcase (i): $\boldsymbol{\sigma}_{\boldsymbol{s}} = \boldsymbol{\sigma}_{\boldsymbol{v}}$*
We only study this subcase for when $\sigma_{\frac{n}{2}-1} = \sigma'_{\frac{n}{2}-1} = 1$ and $(s_{\frac{n}{2}-1}, s_{\frac{n}{2}+2}) \ne (v_{\frac{n}{2}-1}, v_{\frac{n}{2}+2})$, since the alternative involves $C_{n/2+1}(\boldsymbol{s}) = C_{n/2+1}(\boldsymbol{v})$ and as a result of this, Lemma 3 precludes the existence of $\boldsymbol{v}$, since $C(\boldsymbol{s})$ and $C(\boldsymbol{v})$ cannot differ by a single multiset alone. This situation is illustrated in Fig. 3.

We now proceed to ascertain if there exists some $\boldsymbol{v}$ for which $C_{n/2-1}(\boldsymbol{s}) = C_{n/2-1}(\boldsymbol{v})$ holds. Alternatively, we need the following set equality relation to hold:

$$
\begin{Bmatrix}
\{c(\boldsymbol{s}_1^{\frac{n}{2}-2}),0\} \\
\{c(\boldsymbol{s}_2^{\frac{n}{2}-2}),0,s_+\} \\
\{c(\boldsymbol{s}_3^{\frac{n}{2}-2}),0,s_+,s_-\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+3}^n),1\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-1}),1,s_-\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-2}),1,s_+,s_-\}
\end{Bmatrix}
=
\begin{Bmatrix}
\{c(\boldsymbol{v}_1^{\frac{n}{2}-2}),1\} \\
\{c(\boldsymbol{v}_2^{\frac{n}{2}-2}),1,v_+\} \\
\{c(\boldsymbol{v}_3^{\frac{n}{2}-2}),1,v_+,v_-\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+3}^n),0\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-1}),0,v_-\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-2}),0,v_+,v_-\}
\end{Bmatrix}.
\tag{23}
$$

Due to the weight mismatch property between prefix and suffix of equal lengths, we note from Fig. 3 that if $\boldsymbol{v}$ must uphold:

$$
\begin{aligned}
\mathrm{wt}(\boldsymbol{s}_2^{n/2-2}) + 1 &< \mathrm{wt}(\boldsymbol{s}_{n/2+3}^{n-1}) \\
\implies \mathrm{wt}(\boldsymbol{s}_1^{n/2-2}) + 3 &\leq \mathrm{wt}(\boldsymbol{s}_{n/2+3}^n).
\end{aligned}
\tag{24}
$$

Now to prove that (23) never holds, it suffices to show that the composition $\{c(\boldsymbol{s}_{\frac{n}{2}+3}^n),1\}$ can never be matched to any two elements on the RHS in (23), even when (24) holds with equality. It is easy to see this when $v_+ + v_- < 2$. On the contrary when $v_+ + v_- = 2$, the compositions $\{c(\boldsymbol{v}_1^{\frac{n}{2}-2}),1\}$ and $\{c(\boldsymbol{v}_2^{\frac{n}{2}-2}),1,v_+\}$ become identical, and cannot be matched simultaneously to the components of RHS in (23). Therefore, $\boldsymbol{v}$ does not exist.

| $\boldsymbol{s}_1^{\frac{n}{2}-2}$ | 0 | $s_+$ | $s_-$ | 1 | $\boldsymbol{s}_{\frac{n}{2}+3}^n$ |
|---|---|---|---|---|---|

| $\boldsymbol{v}_1^{\frac{n}{2}-2}$ | 1 | $v_+$ | $v_-$ | 0 | $\boldsymbol{v}_{\frac{n}{2}+3}^n$ |
|---|---|---|---|---|---|

Figure 3: Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\frac{n}{2}-2}, \boldsymbol{s}_{\frac{n}{2}+3}^n) = (\boldsymbol{v}_1^{\frac{n}{2}-2}, \boldsymbol{v}_{\frac{n}{2}+3}^n)$, where $v_+ + v_- = s_+ + s_-$.

*Subcase (ii): $\boldsymbol{\sigma}_s \neq \boldsymbol{\sigma}_v$*
All of the possible combinations of $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}})$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}})$ that comprehensively cover this subcase are:

- $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (1, 2b)$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (2b, 1)$.
- $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (2, 0)$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (1, 1)$.
- $(\sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (0, 2)$ and $(\sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (1, 1)$.

where $b \in \mathbb{F}_2$. For the sake of brevity, we only prove the first instance. The remaining proofs run in a similar fashion.
To reiterate our objective, we check for the existence of a string $\boldsymbol{v} \in \mathcal{S}_R(n)$, for a given $\boldsymbol{s} \in \mathcal{S}_R(n)$, which are characterized as per the depiction in Fig. 4. Since $\boldsymbol{s}$ and $\boldsymbol{v}$ may only differ

| $\boldsymbol{s}_1^{\frac{n}{2}-2}$ | $s_+$ | $b$ | $b$ | $s_-$ | $\boldsymbol{s}_{\frac{n}{2}+3}^n$ |
|---|---|---|---|---|---|

| $\boldsymbol{v}_1^{\frac{n}{2}-2}$ | $b$ | $v_+$ | $v_-$ | $b$ | $\boldsymbol{v}_{\frac{n}{2}+3}^n$ |
|---|---|---|---|---|---|

Figure 4: Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are such that $(\boldsymbol{s}_1^{\frac{n}{2}-2}, \boldsymbol{s}_{\frac{n}{2}+3}^n) = (\boldsymbol{v}_1^{\frac{n}{2}-2}, \boldsymbol{v}_{\frac{n}{2}+3}^n)$, where $s_+ + s_- = v_+ + v_- = 1$.

in their respective composition multisets of substring lengths

$\frac{n}{2}$ and $\frac{n}{2}+1$ alone, we endeavor to find the conditions that allow for the set equality of $C_{\frac{n}{2}-1}(\boldsymbol{s})$ and $C_{\frac{n}{2}-1}(\boldsymbol{v})$. More explicitly, we require:

$$
\begin{Bmatrix}
\{c(\boldsymbol{s}_1^{\frac{n}{2}-2}),s_+\} \\
\{c(\boldsymbol{s}_2^{\frac{n}{2}-2}),s_+,b\} \\
\{c(\boldsymbol{s}_3^{\frac{n}{2}-2}),s_+,b^2\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+3}^n),1-s_+\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-1}),1-s_+,b\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+3}^{n-2}),1-s_+,b^2\}
\end{Bmatrix}
=
\begin{Bmatrix}
\{c(\boldsymbol{v}_1^{\frac{n}{2}-2}),b\} \\
\{c(\boldsymbol{v}_2^{\frac{n}{2}-2}),b,v_+\} \\
\{c(\boldsymbol{v}_3^{\frac{n}{2}-2}),b,01\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+3}^n),b\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-1}),b,1-v_+\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+3}^{n-2}),b,01\}
\end{Bmatrix}.
$$

When $s_+ = v_+ = 0$, we may proceed under the assumption that $\mathrm{wt}(\boldsymbol{s}_2^{n/2-2}) = \mathrm{wt}(\boldsymbol{s}_{n/2+3}^{n-1})$ to account for the worst case. In this situation, either $\{c(\boldsymbol{s}_1^{\frac{n}{2}-2}),s_+\}$ or $\{c(\boldsymbol{s}_{\frac{n}{2}+3}^n),1-s_+\}$ fails to be matched, depending on the chosen value of $b$. Else when either $s_+$ or $v_+$ equals 1, we infer that (24) holds true. Again, we choose to proceed with the worst case, i.e. $\mathrm{wt}(\boldsymbol{s}_2^{n/2-2})+3 = \mathrm{wt}(\boldsymbol{s}_{n/2+3}^{n-1})$, and an exhaustive examination of each possibility reveals that the previous set equality cannot be satisfied. Thus, we conclude that $\boldsymbol{v}$ does not exist. $\qquad\square$

The previous result reveals that the codebook $\mathcal{S}_R(n)$ is sufficiently robust to correct the deletion of a single pair of symmetric multisets, i.e.,

**Theorem 4.** *The code $\mathcal{S}_R(n)$ is a single symmetric multiset deletion correcting code.*

Consequently, if a single composition is substituted in $C(\boldsymbol{s})$ where $\boldsymbol{s} \in \mathcal{S}_R(n)$, then there occurs a mismatch between the cumulative weights of the specific multiset affected, say $C_i(\boldsymbol{s})$, and its symmetric counterpart $C_{n-i+1}(\boldsymbol{s})$. Now if both $C_i(\boldsymbol{s})$ and $C_{n-i+1}(\boldsymbol{s})$ are deleted, Lemma 7 tells us that $\boldsymbol{s}$ is still uniquely recoverable. Thus, we conclude that $\mathcal{S}_R(n)$ is capable of correcting a single composition error just like $S_{CA}^{(1)}(n)$, as pointed out previously in Section III.

We now investigate further along this direction and seek to determine if the absence of multiple pairs of such multisets impacts reconstructability. The deletion of two or more pairs of symmetric multisets, as shown in Lemma 14 (Appendix), no longer guarantees unique reconstruction of codewords drawn from $\mathcal{S}_R(n)$. To remedy this, we propose the code $\mathcal{S}_{DS}^{(2)}(n)$ [see Construction 6], capable of correcting deletions of two pairs of symmetric sets.

**Lemma 8.** *Consider a string $\boldsymbol{s} \in \mathcal{S}_{DS}^{(2)}(n)$. Given only the composition multisets $\bigcup_{i \in [n] \setminus \{k-1,k,n-k+1,n-k+2\}} C_i(\boldsymbol{s})$, one can uniquely recover $\boldsymbol{s}$.*

*Proof.* **Case 1.** $n$ is even and the deleted multisets are neighboring, i.e. $\{C_{n/2-1}(\boldsymbol{s}), \ldots, C_{n/2+2}(\boldsymbol{s})\}$

We recall from the proof of Lemma 14, that for some $\boldsymbol{s} \in \mathcal{S}_R(n)$ characterized by $\boldsymbol{\sigma}_s = (\sigma_1, \ldots \sigma_{n/2})$, there may exist

some $\boldsymbol{v} \in \mathcal{S}_R(n)$ with $\boldsymbol{\sigma_v} = (\sigma'_1, \ldots, \sigma'_{n/2})$, such that:

$$\sigma_i = \sigma'_i, \qquad \forall \ 1 \le i \le \frac{n}{2} - 3 \qquad (25)$$
$$\sigma_{\frac{n}{2}-2} + \sigma_{\frac{n}{2}-1} + \sigma_{\frac{n}{2}} = \sigma'_{\frac{n}{2}-2} + \sigma'_{\frac{n}{2}-1} + \sigma'_{\frac{n}{2}}.$$

The difference of the sum of their respective cumulative weights for composition multisets containing substrings of lengths from 1 to $\frac{n}{2}$, can be simplified to:

$$\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - \sum_{i=1}^{n/2} w_i(\boldsymbol{v})$$
$$= \sum_{i=n/2-1}^{n/2} w_i(\boldsymbol{s}) - \sum_{i=n/2-1}^{n/2} w_i(\boldsymbol{v})$$
$$= 3(\sigma'_{n/2-2} - \sigma_{n/2-2}) + (\sigma'_{n/2-1} - \sigma_{n/2-1}). \quad (26)$$

The above difference is maximized when either:

$$(\sigma_{\frac{n}{2}-2}, \sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (0, 1, 2),$$
$$(\sigma'_{\frac{n}{2}-2}, \sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (2, 1, 0).$$

or:

$$(\sigma_{\frac{n}{2}-2}, \sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) = (0, 2, 2),$$
$$(\sigma'_{\frac{n}{2}-2}, \sigma'_{\frac{n}{2}-1}, \sigma'_{\frac{n}{2}}) = (2, 2, 0).$$

In either case, (25) is upheld. Hence we can write that:

$$\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - \sum_{i=1}^{n/2} w_i(\boldsymbol{v}) \le 6.$$

**Case 2.** $n$ may be odd/even and the deleted multisets are not all consecutive, i.e. $k + 1 < n - k + 1$
From the proof of Lemma 14, we note that when the multisets $\{C_{k-1}(\boldsymbol{s}), C_k(\boldsymbol{s}), C_{n-k+1}(\boldsymbol{s}), C_{n-k+2}(\boldsymbol{s})\}$ are deleted, there may exist an alternate $\boldsymbol{v} \in \mathcal{S}_R(n)$ such that:

$$\boldsymbol{s}_1^{k-3} = \boldsymbol{v}_1^{k-3},$$
$$\boldsymbol{s}_{n-k+4}^n = \boldsymbol{v}_{n-k+4}^n,$$
$$\sigma_i = \sigma'_i, \qquad \forall i \in I$$
$$\sigma_k + 2\sigma_{k-1} + 3\sigma_{k-2} = \sigma'_k + 2\sigma'_{k-1} + 3\sigma'_{k-2},$$
$$\sigma_{k+1} + \sigma_k + \sigma_{k-1} + \sigma_{k-2} = \sigma'_{k+1} + \sigma'_k + \sigma'_{k-1} + \sigma'_{k-2}.$$

where $I = \left[\left[\frac{n}{2}\right]\right] \backslash \{k - 2, \ldots, k + 1\}$. As before, we bound the difference of the sum of cumulative weights of $\boldsymbol{s}$ and $\boldsymbol{v}$:

$$\sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{s}) - \sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{v}) = \sum_{i=k-1}^{k} w_i(\boldsymbol{s}) - \sum_{i=k-1}^{k} w_i(\boldsymbol{v})$$
$$= (\sigma'_{k-1} - \sigma_{k-1})$$
$$\quad + 3(\sigma'_{k-2} - \sigma_{k-2}). \quad (27)$$

We find through numerical verification that this quantity cannot exceed 5, and it precisely occurs when:

$$(\sigma_{k-2}, \sigma_{k-1}, \sigma_k, \sigma_{k+1}) = (0, 2, 0, 0),$$
$$(\sigma'_{k-2}, \sigma'_{k-1}, \sigma'_k, \sigma'_{k+1}) = (1, 0, 1, 0).$$

As a result, in both cases the additional constraint $\sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(\boldsymbol{s}) \bmod 7 = a$ in (12) ensures unique reconstruction when the aforementioned multisets are lost. $\qquad \square$

The previous result permits us to conclude that

**Theorem 5.** *The code $\mathcal{S}_{DS}^{(2)}(n)$ is a 2-symmetric multiset deletion correcting code.*

We now seek to generalize the coding constraints in $\mathcal{S}_{DS}^{(2)}(m)$ in (12) by examining how the required redundancy scales as more consecutive multiset pairs go missing. This is accomplished by $\mathcal{S}_{DS}^{\prime(t)}(n)$ [see Construction 7]. Theorem 11 demonstrates that $\mathcal{S}_{DS}^{\prime(t)}(n)$ is a $t$-symmetric consecutive multiset deletion composition code. The proof commences with the following lemma.

**Lemma 9.** *Consider a string $\boldsymbol{s} \in \mathcal{S}_{DS}^{\prime(t)}(n)$, where $t \ge 2$ and $n \ge 2t + 4$. If one is given a corrupted composition multiset $C'(\boldsymbol{s}) = \bigcup_{i \in [\lceil n/2 \rceil] \backslash \{k-t, \ldots, k-1\}} C_i(\boldsymbol{s}) \cup C_{n-i+1}(\boldsymbol{s})$ for any $t < k - 1 \le \lfloor n/2 \rfloor$, i.e. $t$ consecutive symmetric multiset pairs are missing, $\boldsymbol{s}$ can be uniquely reconstructed.*

*Proof.* **Case 1.** $n$ may be odd/even and the $2t$ deleted multisets are not adjacent, i.e. $k < n - k + 2$.
Since the multiset pairs $(C_i(\boldsymbol{s}), C_{n-i+1}(\boldsymbol{s}))$ have been eliminated, for $k - t \le i \le k - 1$, we also do not know their respective cumulative weights. Thus, the values of $\sigma_{k-t-1}, \ldots \sigma_{k-2}$ are also unknown. Furthermore, we note from (6) that $\sigma_{k-1}$ and $\sigma_k$ are also not deducible. However, the sum of these missing values can be inferred from

$$w_{k+1} - w_k = (k+1)w_1 - \sum_{i=1}^{k} i\sigma_{k+1-i} - kw_1 + \sum_{i=1}^{k-1} i\sigma_{k-i}$$
$$= w_1 - \sigma_k \ldots - \sigma_1.$$

To test if $\boldsymbol{s}$ is uniquely recoverable, we attempt to find a suitable $\boldsymbol{v} \in \mathcal{S}_R(n)$, characterized by $\sigma'_1, \ldots, \sigma'_{\lceil n/2 \rceil}$, such that

$$\widetilde{C}_i(\boldsymbol{s}) = \widetilde{C}_i(\boldsymbol{v}). \quad \forall i \in \left[\left[\frac{n}{2}\right]\right] \backslash \{k - t, \ldots, k - 1\}$$

These equations also imply that:

$$(\boldsymbol{s}_1^{k-t-2}, \boldsymbol{s}_{n-k+t+3}^n) = (\boldsymbol{v}_1^{k-t-2}, \boldsymbol{v}_{n-k+t+3}^n),$$
$$\sigma_i = \sigma'_i, \qquad \forall i \in \left[\left[\frac{n}{2}\right]\right] \backslash I$$
$$\sum_{j \in I} \sigma_j = \sum_{j \in I} \sigma'_j.$$

where $I = \{k - t - 1, \ldots, k\}$. Alike the approach undertaken in prior proofs, we now attempt to compute the maximum difference between the sum of cumulative weights of $\boldsymbol{s}$ and $\boldsymbol{v}$:

$$\sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v}) = \sum_{i=k-t}^{k-1} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v})$$
$$= \sum_{i=k-t}^{k-1} \left( iw_1(\boldsymbol{s}) - \sum_{j=1}^{i-1} j\sigma_{i-j} \right)$$

11

$$- \sum_{i=k-t}^{k-1} \left( i w_1(\boldsymbol{v}) - \sum_{j=1}^{i-1} j \sigma'_{i-j} \right)$$

$$= \frac{t(t+1)}{2} (\sigma'_{k-t-1} - \sigma_{k-t-1})$$

$$+ \ldots + 3(\sigma'_{k-3} - \sigma_{k-3})$$

$$+ (\sigma'_{k-2} - \sigma_{k-2}). \tag{28}$$

The final equality follows from $w_1(\boldsymbol{s}) = w_1(\boldsymbol{v})$, which always holds since the premise of this error model states that $k-t > 1$, suggesting that the multisets $C_1$ and $C_n$ are always preserved.

*Subcase (i): $t$ is even.*

In this case, the quantity in (28) is maximized when we have:

$$(\sigma'_{k-t-1}, \ldots, \sigma'_k) = (\overbrace{2, \ldots 2}^{\frac{t}{2}+1}, \overbrace{0, \ldots 0}^{\frac{t}{2}+1}),$$

$$(\sigma_{k-t-1}, \ldots, \sigma_k) = (\overbrace{0, \ldots 0}^{\frac{t}{2}+1}, \overbrace{2, \ldots 2}^{\frac{t}{2}+1}).$$

It is worth pointing out that these configurations may not always be valid, since the available multisets may not allow for them. However, they certainly embody the worst possible case. Now applying this to (28), we obtain the following bound:

$$\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v}) \leq \frac{t(t+2)^2}{4}. \tag{29}$$

*Subcase (ii): $t$ is odd.*

When $t$ is odd, the difference between the cumulative weights of $\boldsymbol{s}$ and $\boldsymbol{v}$ is maximized when:

$$(\sigma'_{n/2-t}, \ldots, \sigma'_{n/2}) = (\overbrace{2, \ldots 2}^{\frac{t-1}{2}}, p, \overbrace{0, \ldots 0}^{\frac{t-1}{2}}),$$

$$(\sigma_{n/2-t}, \ldots, \sigma_{n/2}) = (\overbrace{0, \ldots 0}^{\frac{t-1}{2}}, p, \overbrace{2, \ldots 2}^{\frac{t-1}{2}}).$$

where $p \in \{0, 1, 2\}$. By further manipulating (28), we get

$$\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v}) \leq \frac{t(t+1)(t+3)}{4}. \tag{30}$$

**Case 2.** $n$ is even and all of the deleted multisets are consecutive, i.e. $C_{n/2-t+1}(\boldsymbol{s}), \ldots, C_{n/2+t}(\boldsymbol{s})$.

Much like the previous case, we attempt to find a $\boldsymbol{v} \in \mathcal{S}_R(n)$, characterized by $\sigma'_1, \ldots \sigma'_{\frac{n}{2}}$, such that for $1 \leq i \leq n/2 - t$:

$$\widetilde{C}_i(\boldsymbol{s}) = \widetilde{C}_i(\boldsymbol{v}),$$

As a consequence, the following equalities also hold:

$$(\boldsymbol{s}_1^{n/2-t-1}, \boldsymbol{s}_{n/2+t+2}^n) = (\boldsymbol{v}_1^{n/2-t-1}, \boldsymbol{v}_{n/2+t+2}^n)$$

$$\sigma_i = \sigma'_i, \qquad \forall i \in [n/2 - t - 1]$$

$$\sum_{j=n/2-t}^{n/2} \sigma_j = \sum_{j=n/2-t}^{n/2} \sigma'_j.$$

Corresponding to (28), we arrive at:

$$\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v}) = \frac{t(t+1)}{2} (\sigma'_{\frac{n}{2}-t} - \sigma_{\frac{n}{2}-t}) + \ldots$$

$$+ 3(\sigma'_{\frac{n}{2}-2} - \sigma_{\frac{n}{2}-2}) + (\sigma'_{\frac{n}{2}-1} - \sigma_{\frac{n}{2}-1}).$$

By appropriately assigning the vectors $(\sigma_{n/2-t}, \ldots, \sigma_{n/2})$ and $(\sigma'_{n/2-t}, \ldots, \sigma'_{n/2})$, we can upper-bound the preceding quantity as follows:

$$\sum_{i=1}^{n/2} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v}) \leq \begin{cases} \frac{(t+1)^3}{4}, & \text{if } t \text{ is even.} \\ \frac{t(t+1)(t+2)}{4}, & \text{otherwise.} \end{cases} \tag{31}$$

The definition of $\mathcal{S}_{DS}'^{(t)}(n)$ in (13) along with the bounds provided in (29), (30) and (31) directly imply the statement. $\square$

**Lemma 10.** *Consider a string $\boldsymbol{s} \in \mathcal{S}_{DS}'^{(t)}(n)$, where $t \geq 2$ and $n \geq 2t + 4$. If one is given a corrupted composition multiset $C'(\boldsymbol{s}) = \bigcup_{i \in [\lceil n/2 \rceil] \setminus [t]} C_i(\boldsymbol{s}) \cup C_{n-i+1}(\boldsymbol{s})$, $\boldsymbol{s}$ can be uniquely reconstructed.*

*Proof.* Unlike Lemma 9, this proof is dedicated to the specific case where the multisets $C_1 \cup C_n, \ldots, C_t \cup C_{n-t+1}$ have been deleted. Since multisets $C_{t+1}(\boldsymbol{s})$ and $C_{t+2}(\boldsymbol{s})$ are available, we can obtain:

$$w_{t+2}(\boldsymbol{s}) - w_{t+1}(\boldsymbol{s}) = w_1(\boldsymbol{s}) - \sigma_{t+1} - \ldots - \sigma_1. \tag{32}$$

Similar to the prior analyses, we check for the existence of some $\boldsymbol{v} \in \mathcal{S}_R(n)$, specified by $\boldsymbol{\sigma_v} = (\sigma'_1, \ldots, \sigma'_{\lceil n/2 \rceil})$, that satisfies:

$$\widetilde{C}_i(\boldsymbol{s}) = \widetilde{C}_i(\boldsymbol{v}). \tag{33}$$

where $t < i \leq \lceil n/2 \rceil$. From (32) and (33), we infer that for $1 \leq i \leq \lceil n/2 \rceil - t - 1$:

$$w_{t+i+1}(\boldsymbol{s}) - w_{t+i}(\boldsymbol{s}) = w_{t+i+1}(\boldsymbol{v}) - w_{t+i}(\boldsymbol{v})$$

$$\implies w_1(\boldsymbol{s}) - \sigma_{t+i} - \ldots - \sigma_1 = w_1(\boldsymbol{v}) - \sigma'_{t+i} - \ldots - \sigma'_1.$$

The preceding relation now allows us to deduce that:

$$\sigma_j = \sigma'_j. \quad \forall \ t + 2 \leq j \leq \lceil n/2 \rceil$$

Also by construction of $\mathcal{S}_R(n)$, we observe that $\sigma_1 = \sigma'_1$. As before, we inspect the difference of the sum of cumulative weights of $\boldsymbol{s}$ and $\boldsymbol{v}$:

$$(\boldsymbol{s}_1^{k-t-2}, \boldsymbol{s}_{n-k+t+3}^n) = (\boldsymbol{v}_1^{k-t-2}, \boldsymbol{v}_{n-k+t+3}^n),$$

$$\sigma_i = \sigma'_i, \qquad \forall i \in [\lceil n/2 \rceil] \setminus I$$

$$\sum_{j \in I} \sigma_j = \sum_{j \in I} \sigma'_j.$$

where $I = \{k - t - 1, \ldots, k\}$. The sum of cumulative weights of $\boldsymbol{s}$ and $\boldsymbol{v}$ differ by:

$$\sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v}) = \sum_{i=1}^{t} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v})$$

$$= \sum_{i=1}^{t} \left( i w_1(\boldsymbol{s}) - \sum_{j=1}^{i-1} j \sigma_{i-j} \right)$$

$$- \sum_{i=1}^{t} \left( i w_1(\boldsymbol{v}) - \sum_{j=1}^{i-1} j \sigma'_{i-j} \right)$$

$$= \frac{t(t+1)}{2}(w_1(\boldsymbol{s}) - w_1(\boldsymbol{v}))$$
$$+ \frac{(t-2)(t-1)}{2}(\sigma_2' - \sigma_2)$$
$$+ \ldots + 3(\sigma_{t-2}' - \sigma_{t-2})$$
$$+ (\sigma_{t-1}' - \sigma_{t-1}). \tag{34}$$

Since $w_{t+1}(\boldsymbol{s}) = w_{t+1}(\boldsymbol{v})$ and $w_{t+2}(\boldsymbol{s}) = w_{t+2}(\boldsymbol{v})$, we rewrite (32) as:

$$\begin{aligned} w_1(\boldsymbol{s}) - w_1(\boldsymbol{v}) &= (\sigma_{t+1} - \sigma_{t+1}') + \ldots + (\sigma_1 - \sigma_1') \\ &= (\sigma_{t+1} - \sigma_{t+1}') + \ldots + (\sigma_2 - \sigma_2') \\ &\leq 2t. \end{aligned} \tag{35}$$

We now attempt to design the vectors $\boldsymbol{\sigma}_s$ and $\boldsymbol{\sigma}_v$ such that for a fixed value of $w_1(\boldsymbol{s}) - w_1(\boldsymbol{v})$, the following quantity is maximized:

$$\frac{(t-2)(t-1)}{2}(\sigma_2' - \sigma_2) + \ldots + (\sigma_{t-1}' - \sigma_{t-1}).$$

while bearing in mind that:

$$w_1(\boldsymbol{s}) - w_1(\boldsymbol{v}) = \sum_{i=2}^{t+1}(\sigma_i - \sigma_i').$$

Clearly, we must set $\sigma_i' - \sigma_i = 2$ for $i = 2, \ldots$, due to the higher weights of these terms, and $\sigma_i' - \sigma_i = -2$ for $i = t-1, t-2, \ldots$ on account of the minor influence of these terms on (34). Additionally, we set $(\sigma_t, \sigma_t') = (\sigma_{t+1}, \sigma_{t+1}') = (2, 0)$, thus allowing us to reduce the quantity $\sum_{i=2}^{t-1}(\sigma_i - \sigma_i')$, i.e.

$$\sum_{i=2}^{t-1}(\sigma_i - \sigma_i') = a - 4.$$

where $a = w_1(\boldsymbol{s}) - w_1(\boldsymbol{v})$. Hence, to proceed with the maximization of (34), we perform the following assignment when $a$ is odd:

$$\begin{aligned} (\sigma_2', \ldots, \sigma_{t-1}') &= (2, \ldots 2, p', 0, \ldots, 0), \\ (\sigma_2, \ldots, \sigma_{t-1}) &= (0, \ldots 0, p, 2, \ldots, 2). \end{aligned} \tag{36}$$

where $p + p' = 1$. Here $p$ and $p'$ may be assigned interchangeably, depending on $t$. In a similar fashion, when $a$ is even, we again reuse this assignment while setting either $(p, p') = (0, 2)$ or $p = p' = 0$. Further noting that the term $w_1(\boldsymbol{s}) - w_1(\boldsymbol{v})$ has the highest weight in (34), we combine (34), (35) and (36) to arrive at the following upper bound:

$$\sum_{i=1}^{\lceil n/2 \rceil} w_i(\boldsymbol{s}) - w_i(\boldsymbol{v}) \leq \left\lceil \frac{4t^3}{3} + \frac{2t}{3} - \frac{35}{4} \right\rceil. \tag{37}$$

$\square$

Upon combining Lemmas 9 and 10, we arrive at:

**Lemma 11.** *Consider a string $\boldsymbol{s} \in \mathcal{S}_{DS}'^{(t)}(n)$ [see Construction 7], where $t \geq 2$ and $n \geq 2t+4$. If one is given a corrupted composition multiset $C'(\boldsymbol{s}) = \bigcup_{i \in [\lceil n/2 \rceil] \setminus \{k-t, \ldots, k-1\}} C_i(\boldsymbol{s}) \cup C_{n-i+1}(\boldsymbol{s})$ for any $t \leq k-1 \leq \lfloor n/2 \rfloor$, i.e. $t$ consecutive symmetric multiset pairs are missing, $\boldsymbol{s}$ can be uniquely reconstructed.*

**Theorem 6.** $\mathcal{S}_{DS}'^{(t)}(n)$ *is a $t$-symmetric consecutive multiset deletion composition code.*

*Remark:* Experimentally, it is found that an appropriate modulo constraint corresponding to (31) is sufficient to allow the correction of deletion of any $t$ symmetric multiset pairs, consecutive or otherwise. An intuitive interpretation for this result follows from the fact that when the missing multiset pairs are consecutive, the least number of constraints are imposed on $\boldsymbol{\sigma}$. A rigorous proof for the same is yet to be found. It is also worth mentioning that though the constraint in (37) is stricter than that of (31), the order of the required redundancy remains identical.

## VIII. SKEWED SUBSTITUTION-CORRECTING CODES

In this section, we confine our focus to the correction of skewed substitution errors [see Definition 6].

**Lemma 12.** *Consider any $\boldsymbol{s} \in \mathcal{S}_R(n)$. Given that there occurs a single skewed substitution error in its composition set, one can uniquely recover $\boldsymbol{s}$.*

*Proof.* In the following, we let the corrupted composition set be denoted by $C'(\boldsymbol{s}) = \bigcup_{i \in [n]} C_i'(\boldsymbol{s})$.
**Case 1.** $n$ is even.
Given $C'(\boldsymbol{s})$, it is easy to identify the corrupted composition multiset $C_k'(\boldsymbol{s})$, since the following relation only holds for $k$:

$$w_k' < w_{n-k+1}'. \tag{38}$$

If we now delete all elements of $C_k'(\boldsymbol{s})$ from $C'(\boldsymbol{s})$, Lemma 4 tells us that $\boldsymbol{s}$ is still uniquely recoverable.
**Case 2.** $n$ is odd.
Using the arguments of the preceding case, we can reach the same conclusion for an odd $n$, when the affected multiset is $C_k'(\boldsymbol{s})$, where $\lceil n/2 \rceil < k \leq n$, because in these cases, there exists an uncorrupted distinct symmetric multiset $C_{n-k+1}'(\boldsymbol{s})$, which gives us the true cumulative weight and thus allows us to accurately recover $\boldsymbol{\sigma}_s$.
If $k = \lceil n/2 \rceil$, this is no longer true since the multiset $C_{\lceil n/2 \rceil}(\boldsymbol{s})$ is its own symmetric counterpart. Noting that this normally helps us determine the bits $(s_{\lceil n/2 \rceil - 1}, s_{\lceil n/2 \rceil + 1})$, we recall from Lemma 2 that when these bits are assigned incorrectly, inconsistencies with the multiset $C_{\lceil n/2 \rceil - 1}$ would arise, which are not permitted under the considered error model. Hence, we conclude that $\boldsymbol{s}$ can be recovered uniquely. $\square$

We now consider a more general error model involving multiple asymmetric skewed substitution errors, wherein each multiset pair $\widetilde{C}_i$, for any $i \in [n]$, may contain at most one skewed substitution and the total number of errors does not exceed $t$. It is found that the asymmetric $t$-multiset deletion-correcting code $\mathcal{S}_{DA}^{(t)}(n)$ is also robust to $t$ asymmetric skewed substitutions and in the following, we prove the same.

**Lemma 13.** *Consider any $\boldsymbol{s} \in \mathcal{S}_{DA}^{(t)}(n)$. Given that there occurs $t$ skewed asymmetric substitution errors in its composition set, such that for all $1 \leq i \leq n$, $\widetilde{C}_i(\boldsymbol{s})$ contains at most one skewed substitution error, then one can uniquely recover $\boldsymbol{s}$.*

*Proof.* Since the error model only allows at most one skewed substitution in a pair of symmetric multisets, the cumulative weights of all sets can be determined accurately. This is due to the fact that if multiset $C_k(\boldsymbol{s})$ has been corrupted, we may write:

$$w_k < w_{n-k+1}. \tag{39}$$

As a consequence, all cumulative weights can be correctly re-assigned and in turn the $\boldsymbol{\sigma}_s$ sequence can be recovered. The preceding inequality also allows to identify the affected multisets, the deletion of which would transform our problem of correcting $t$ asymmetric skewed substitutions into reconstruction under the absence of $t$ multisets. According to Theorem 2, unique reconstruction of $\boldsymbol{s}$ is perfectly possible, thus concluding our proof. $\square$

The aforementioned result naturally leads to the following theorem.

**Theorem 7.** $\mathcal{S}_{DA}^{(t)}(n)$ *is a $t$-asymmetric skewed composition code.*

## IX. Conclusion

In this work, we propose and investigate error models involving insertion and deletion of substring compositions in the context of polymer-based data storage. In particular, we examine the robustness of the composition-reconstructable code introduced in [11], [12], and identify the situations which do not guarantee unique reconstruction of codewords from this construction. For these cases, new codes are proposed. Notably, an equivalence between codes correcting multiset deletions and insertions is established. We also examine a special asymmetric variant of substitution errors, namely skewed substitution errors, which manifest in polymer-based storage.

Several problems pertaining to string construction under this data storage paradigm still remain open:

- The error model involving skewed substitutions under a symmetric setting is yet to be investigated. It would be interesting to know if there exists a suitable codebook offering a lower redundancy than that designed to correct standard substitution errors under the symmetric setting, as stated in [11].
- The problem of reconstructing strings from composition multisets, error-free or otherwise, could be extended to larger alphabets.
- Though some bounds on the maximum number of mutually equicomposable strings were stated in [10], bounds on the error ball sizes under the error models involving substitutions, insertions or deletions are still unknown. These could allow us to infer if the proposed code constructions are indeed optimal.
- One could also extend this research to the construct wherein bits are arranged in a circular fashion, on a ring.
- As pointed out in [10], a polynomial-time algorithm for the string reconstruction problem is yet to be found.

## References

[1] A. Al Ouahabi, J.-A. Amalian, L. Charles, and J.-F. Lutz, "Mass spectrometry sequencing of long digital polymers facilitated by programmed inter-byte fragmentation," *Nature communications*, vol. 8, no. 1, p. 967, 2017.

[2] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, p. 77, 2013.

[3] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on DNA in silica with error-correcting codes," *Angewandte Chemie International Edition*, vol. 54, no. 8, pp. 2552–2555, 2015.

[4] R. Heckel, G. Mikutis and R. N. Grass, "A characterization of the DNA data storage channel," *Scientific Reports*, vol. 9, no. 1, pp. 9663, 2019.

[5] C.N. Takahashi, B.H. Nguyen, K. Strauss and L. Ceze, "Demonstration of End-to-End Automation of DNA Data Storage," *Scientific Reports*, vol. 9, no. 1, pp. 4998, 2019.

[6] S. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Scientific reports*, vol. 5, p. 14138, 2015.

[7] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific reports*, vol. 7, no. 1, p. 5011, 2017.

[8] S. K. Tabatabaei, B. Wang, N. B. M. Athreya, B. Enghiad, A. G. Hernandez, J.-P. Leburton, D. Soloveichik, H. Zhao, and O. Milenkovic, "DNA punch cards: Encoding data on native dna sequences via topological modifications," *bioRxiv*, p. 672394, 2019.

[9] S. Tabatabaei, B. Wang, N. Athreya, B. Enghiad, A. Hernandez, C. Fields, J.-P. Leburton, D. Soloveichik, H. Zhao, and O. Milenkovic, "DNA punch cards for storing data on native DNA sequences via enzymatic nicking," *Nature Communications*, vol. 11, 12, 2020.

[10] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, and S. Pan, "String reconstruction from substring compositions," *SIAM Journal on Discrete Mathematics*, vol. 29, no. 3, pp. 1340–1371, 2015.

[11] S. Pattabiraman, R. Gabrys and O. Milenkovic, "Coding for polymer-based data storage", *arXiv:2003.02121*, 2020.

[12] S. Pattabiraman, R. Gabrys, and O. Milenkovic, "Reconstruction and error-correction codes for polymer-based data storage," in *IEEE Information Theory Workshop*, Visby, Sweden, pp. 1–5, Aug. 2019.

[13] R. Gabrys, S. Pattabiraman, and O. Milenkovic, "Mass error-correction codes for polymer-based data storage," *IEEE International Symposium on Information Theory*, Los Angeles, CA, USA, pp. 25–30, Jun. 2020.

[14] R. Gabrys, S. Pattabiraman and O. Milenkovic, "Reconstructing mixtures of coded strings from prefix and suffix compositions," *2020 IEEE Information Theory Workshop (ITW)*, pp. 1–5, 2021.

## Appendix

**Lemma 14.** *Consider a string $\boldsymbol{s} \in \mathcal{S}_R(n)$. Given $C'(\boldsymbol{s}) = \bigcup_{i \in [n] \setminus \{k-1, k, n-k+1, n-k+2\}} C_i(\boldsymbol{s})$ for any $1 \leq k < \lceil \frac{n-1}{2} \rceil$, $\boldsymbol{s}$ may no longer be uniquely determined.*

*Proof.* **Case 1.** $n$ is even and deleted sets are: $\{C_{\frac{n}{2}-1}(\boldsymbol{s}), \ldots, C_{\frac{n}{2}+2}(\boldsymbol{s})\}$.

To demonstrate that $\mathcal{S}_R(n)$ does not necessarily preserve unique reconstructability when the multisets $\{C_{\frac{n}{2}-1}, \ldots, C_{\frac{n}{2}+2}\}$ go missing, we consider two codewords $\boldsymbol{s}, \boldsymbol{v} \in \mathcal{S}_R(n)$, such that:

$$\bigcup_{i \in \{n, \ldots, \frac{n}{2}+3\}} C_i(\boldsymbol{s}) = \bigcup_{i \in \{n, \ldots, \frac{n}{2}+3\}} C_i(\boldsymbol{v}). \tag{40}$$

From our knowledge of the reconstruction algorithm [Section II], we can also infer the following:

$$(\boldsymbol{s}_1^{n/2-3}, \boldsymbol{s}_{n/2+4}^n) = (\boldsymbol{v}_1^{n/2-3}, \boldsymbol{v}_{n/2+4}^n),$$
$$\sigma_i = \sigma_i'. \qquad 1 \leq i \leq \frac{n}{2} - 3, \tag{41}$$
$$\sigma_{\frac{n}{2}-2} + \sigma_{\frac{n}{2}-1} + \sigma_{\frac{n}{2}} = \sigma_{\frac{n}{2}-2}' + \sigma_{\frac{n}{2}-1}' + \sigma_{\frac{n}{2}}'.$$

where $\boldsymbol{\sigma}_s = (\sigma_1, \ldots, \sigma_{n/2})$ and $\boldsymbol{\sigma}_v = (\sigma_1', \ldots, \sigma_{n/2}')$ correspond to $\boldsymbol{s}$ and $\boldsymbol{v}$ respectively. Additionally, we set:

$$
\begin{aligned}
(\sigma_{\frac{n}{2}-2}, \sigma_{\frac{n}{2}-1}, \sigma_{\frac{n}{2}}) &= (0, 0, 1), \\
(\sigma_{\frac{n}{2}-2}', \sigma_{\frac{n}{2}-1}', \sigma_{\frac{n}{2}}') &= (1, 0, 0), \\
v_{n/2-2} &= 1, \\
s_{n/2} &= 1, \\
s_{n-3} &= 0, \\
\mathrm{wt}(\boldsymbol{s}_2^{n/2-3}) &= \mathrm{wt}(\boldsymbol{s}_{n/2+4}^{n-4}).
\end{aligned}
\tag{42}
$$

The relations between $\boldsymbol{s}$ and $\boldsymbol{v}$ as described by (41) and (42) are depicted in Fig. 5. Evidently, $\boldsymbol{s}$ and $\boldsymbol{v}$ differ in their respective multisets $C_{n/2+2}$ and $C_{n/2+1}$ according Lemma 2. Additionally, since their cumulative weights $w_{n/2+2}$ and $w_{n/2}$ also differ, as one may verify from (6) and (42), we deduce that the multisets $C_{n/2}$ and $C_{n/2-1}$ also do not match for $\boldsymbol{s}$ and $\boldsymbol{v}$. We now proceed to examine if $C_{n/2-2}(\boldsymbol{s}) = C_{n/2-2}(\boldsymbol{v})$ holds:

$$
\left\{
\begin{array}{l}
\{c(\boldsymbol{s}_1^{\frac{n}{2}-3}), 0\} \\
\{c(\boldsymbol{s}_2^{\frac{n}{2}-3}), 0^2\} \\
\{c(\boldsymbol{s}_3^{\frac{n}{2}-3}), 0^2 1\} \\
\{c(\boldsymbol{s}_4^{\frac{n}{2}-3}), 0^3 1\} \\
\{c(\boldsymbol{s}_5^{\frac{n}{2}-3}), 0^4 1\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n}), 0\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-1}), 0^2\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-2}), 0^3\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-3}), 0^3 1\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-4}), 0^4 1\}
\end{array}
\right\}
=
\left\{
\begin{array}{l}
\{c(\boldsymbol{v}_1^{\frac{n}{2}-3}), 1\} \\
\{c(\boldsymbol{v}_2^{\frac{n}{2}-3}), 01\} \\
\{c(\boldsymbol{v}_3^{\frac{n}{2}-3}), 0^2 1\} \\
\{c(\boldsymbol{v}_4^{\frac{n}{2}-3}), 0^3 1\} \\
\{c(\boldsymbol{v}_5^{\frac{n}{2}-3}), 0^4 1\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n}), 0\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-1}), 0^2\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-2}), 0^3\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-3}), 0^4\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-4}), 0^5\}
\end{array}
\right\}.
\tag{43}
$$

Using (42) to simplify this set equality relation, we arrive at:

$$
\left\{
\begin{array}{l}
\{c(\boldsymbol{s}_1^{\frac{n}{2}-3}), 0\} \\
\{c(\boldsymbol{s}_2^{\frac{n}{2}-3}), 0^2\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-3}), 0^3 1\} \\
\{c(\boldsymbol{s}_{\frac{n}{2}+4}^{n-4}), 0^4 1\}
\end{array}
\right\}
=
\left\{
\begin{array}{l}
\{c(\boldsymbol{v}_1^{\frac{n}{2}-3}), 1\} \\
\{c(\boldsymbol{v}_2^{\frac{n}{2}-3}), 01\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-3}), 0^4\} \\
\{c(\boldsymbol{v}_{\frac{n}{2}+4}^{n-4}), 0^5\}
\end{array}
\right\}.
\tag{44}
$$

Since the construction of $\mathcal{S}_R(n)$ in () requires $s_1 = 0$ and (42) mandates that $s_{n-3} = 0$ and $\mathrm{wt}(\boldsymbol{s}_2^{n/2-3}) = \mathrm{wt}(\boldsymbol{s}_{n/2+4}^{n-4})$, we are led to the following relation:

$$
\mathrm{wt}(\boldsymbol{s}_1^{n/2-3}) = \mathrm{wt}(\boldsymbol{s}_2^{n/2-3}) = \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+4}^{n-3}) = \mathrm{wt}(\boldsymbol{s}_{\frac{n}{2}+4}^{n-4}).
\tag{45}
$$

This allows us to conclude that (43) indeed holds, and further bit specifications in $\boldsymbol{s}$ and $\boldsymbol{v}$ can lead us to similar set equality relations for the multisets $C_{n/2-3}, \ldots, C_1$. Hence, $\boldsymbol{s}$ and $\boldsymbol{v}$ become confusable under the deletion of multisets $\{C_{\frac{n}{2}-1}(\boldsymbol{s}), \ldots, C_{\frac{n}{2}+2}(\boldsymbol{s})\}$.

**Case 2.** $n$ may be odd/even and the four deleted sets are not consecutive: $\{C_{k-1}(\boldsymbol{s}), C_k(\boldsymbol{s}), C_{n-k+1}(\boldsymbol{s}), C_{n-k+2}(\boldsymbol{s})\}$, where $k+1 < n-k+1$.

| $\boldsymbol{s}_1^{\frac{n}{2}-3}$ | 0 | 0 | 1 | 0 | 0 | 0 | $\boldsymbol{s}_{\frac{n}{2}+4}^{n}$ |
|---|---|---|---|---|---|---|---|
| $\boldsymbol{v}_1^{\frac{n}{2}-3}$ | 1 | 0 | 0 | 0 | 0 | 0 | $\boldsymbol{v}_{\frac{n}{2}+4}^{n}$ |

Figure 5: Strings $\boldsymbol{s}$ and $\boldsymbol{v}$ are specified by (41) and (42).

In the following, we once again proceed by checking if $\boldsymbol{s}$ is uniquely recoverable, by probing the existence of some $\boldsymbol{v} \in \mathcal{S}_R(n)$, characterized by $\sigma_1', \ldots, \sigma_{\lceil \frac{n}{2} \rceil}'$ such that for all $i \in [n] \setminus \{k-1, k-n-k+1, n-k+2\}$:

$$
C_i(\boldsymbol{s}) = C_i(\boldsymbol{v}).
\tag{46}
$$

*Subcase (i): $k = 2$*

This situation corresponds to the deletion of multisets $C_1(\boldsymbol{s})$, $C_2(\boldsymbol{s})$, $C_{n-1}(\boldsymbol{s})$ and $C_n(\boldsymbol{s})$. When this happens, for any $3 \le i \le \lceil n/2 \rceil - 1$, the following values are recoverable:

$$
w_{i+1}(\boldsymbol{s}) - w_i(\boldsymbol{s}) = \sigma_{i+1} + \ldots + \sigma_{\lceil n/2 \rceil}.
$$

This can be used to recover the values of $\sigma_4, \ldots, \sigma_{\lceil n/2 \rceil}$. In other words,

$$
\sigma_i = \sigma_i'. \quad \forall \ 4 \le i \le \lceil n/2 \rceil
\tag{47}
$$

Furthermore, since $w_3(\boldsymbol{s}) = w_3(\boldsymbol{v})$, we can infer from (5) and (47) that:

$$
\begin{aligned}
\sigma_1 + 2\sigma_2 + 3\sigma_3 &= \sigma_1' + 2\sigma_2' + 3\sigma_3' \\
\implies 2\sigma_2 + 3\sigma_3 &= 2\sigma_2' + 3\sigma_3'.
\end{aligned}
$$

The second equality follows from the construction of $\mathcal{S}_R(n)$. Given the above relation, we conclude that (47) also holds for $i \in \{2, 3\}$. Moreover, we cannot have $(s_2, s_{n-1}) \ne (v_2, v_{n-1})$ even when $\sigma_2 = \sigma_2' = 1$, since the Catalan-Bertrand structure would automatically imply that $(s_2, s_{n-1}) = (v_2, v_{n-1}) = (0, 1)$. This inference combined with Lemma 2, lead us to the conclusion that no suitable $\boldsymbol{v}$ exists.

*Subcase (ii): $k = 3$*

When multisets $C_2(\boldsymbol{s}), C_3(\boldsymbol{s}), C_{n-2}(\boldsymbol{s})$ and $C_{n-1}(\boldsymbol{s})$ have been deleted, the availability of cumulative weights $w_1, w_4, \ldots w_{\lceil n/2 \rceil}$ allow us to retrieve $\sigma_1, \sigma_5, \ldots, \sigma_{\lceil n/2 \rceil}$ as in the previous subcase, i.e.

$$
\sigma_i = \sigma_i'. \quad \forall \ i \in [\lceil n/2 \rceil] \setminus \{2, 3, 4\}
\tag{48}
$$

We also observe from (6) and (46) that:

$$
\begin{aligned}
w_4(\boldsymbol{s}) - w_1(\boldsymbol{s}) &= w_4(\boldsymbol{v}) - w_1(\boldsymbol{v}) \\
&= 3w_1(\boldsymbol{s}) - \sigma_3 - 2\sigma_2 - 3\sigma_1, \\
\implies \sigma_2 + 2\sigma_3 &= \sigma_2' + 2\sigma_3'.
\end{aligned}
\tag{49}
$$

Similarly, since $w_5(\boldsymbol{s}) = w_5(\boldsymbol{v})$, we obtain:

$$
\sigma_2 + 2\sigma_3 + 3\sigma_4 = \sigma_2' + 2\sigma_3' + 3\sigma_4.
$$

As a consequence, (48) also holds for $i = 4$. This, along with (4) hint that:

$$
\sigma_2 + \sigma_3 = \sigma_2' + \sigma_3'.
\tag{50}
$$

Equations (49) and (50) together insinuate that $(\sigma_2, \sigma_3) = (\sigma_2', \sigma_3')$. Hence, we may argue as before,

that no suitable $v$ distinct from $s$ actually exists.

*Subcase (iii): $k \geq 4$*

Similar to the approach used in Case 1, we attempt to show that there exist two codewords $s, v \in \mathcal{S}_R(n)$, such that for all $i \in [n]\backslash\{k-1, k, n-k+1, n-k+2\}$:

$$C_i(s) = C_i(v). \tag{51}$$

To this end, we construct a specific pair of strings $s$ and $v$ as follows:

$$(s_1^{k-3}, s_{n-k+4}^n) = (v_1^{k-3}, v_{n-k+4}^n),$$
$$(\sigma_{k-2}, \sigma_{k-1}, \sigma_k, \sigma_{k+1}) = (1,1,1,0),$$
$$(\sigma'_{k-2}, \sigma'_{k-1}, \sigma'_k, \sigma'_{k+1}) = (2,0,0,1),$$
$$\sigma_i = \sigma'_i, \qquad \forall\, k+2 \leq i \leq \lceil \tfrac{n}{2} \rceil \tag{52}$$
$$(s_{k-1}, s_k, s_{k+1}, s_{k+2}) = (0,0,1),$$
$$s_2 = 1,$$
$$v_{k-2} = 0.$$

| $s_1^{k-3}$ | 0 | 0 | 0 | 1 | $s_{k+2}^{n-k-1}$ | 0 | 1 | 1 | 0 | $s_{n-k+4}^n$ |
|---|---|---|---|---|---|---|---|---|---|---|

| $v_1^{k-3}$ | 0 | 0 | 0 | 1 | $v_{k+2}^{n-k-1}$ | 1 | 0 | 0 | 1 | $v_{n-k+4}^n$ |
|---|---|---|---|---|---|---|---|---|---|---|

Figure 6: Strings $s$ and $v$ are related such that $(s_1^{k-3}, s_{n-k+4}^n) = (v_1^{k-3}, v_{n-k+4}^n)$ and $c(s_{k+2}^{n-k-1}) = c(v_{k+2}^{n-k-1})$

These relations have been illustrated in Fig. 6. The preceding equalities also imply that:

$$\sigma_i = \sigma'_i, \qquad \forall\, 1 \leq i \leq k-3$$
$$\sum_{i=k-2}^{k+1} \sigma_i = \sum_{i=k-2}^{k+1} \sigma'_i, \tag{53}$$
$$\sigma_k + 2\sigma_{k-1} + 3\sigma_{k-2} = \sigma'_k + 2\sigma'_{k-1} + 3\sigma'_{k-2},$$
$$c(s_{k+2}^{n-k-1}) = c(v_{k+2}^{n-k-1}).$$

In turn, these relations help ensure that:

$$w_i(s) = w_i(v), \qquad \forall\, 1 \leq i \leq k-2$$
$$w_{k+1}(s) - w_{k-2}(s) = w_{k+1}(v) - w_{k-2}(v), \tag{54}$$
$$w_{k+i+1}(s) - w_{k+i}(s) = w_{k+i+1}(v) - w_{k+i}(v).$$

for $1 \leq i \leq n-k-1$. One may verify this with the assistance of (4) and (6).

From Fig. 6, it is fairly evident that $s$ and $v$ do not match in their corresponding multisets $C_{n-k+2}$ and $C_{n-k+1}$. Now as done in case 1, we check if multisets $C_{n-k}(s)$ and $C_{n-k}(v)$ match:

$$\begin{Bmatrix} \{c(s_1^{k-3}), 0^4 1, c\} \\ \{c(s_2^{k-3}), 0^4 1^2, c\} \\ \{c(s_3^{k-3}), 0^4 1^3, c\} \\ \{c(s_{n-k+4}^n), 0^2 1^3, c\} \\ \{c(s_{n-k+4}^{n-1}), 0^3 1^3, c\} \\ \{c(s_{n-k+4}^{n-2}), 0^4 1^3, c\} \end{Bmatrix} = \begin{Bmatrix} \{c(v_1^{k-3}), 0^3 1^2, c\} \\ \{c(v_2^{k-3}), 0^4 1^2, c\} \\ \{c(v_3^{k-3}), 0^5 1^2, c\} \\ \{c(v_{n-k+4}^n), 0^2 1^3, c\} \\ \{c(v_{n-k+4}^{n-1}), 0^3 1^3, c\} \\ \{c(v_{n-k+4}^{n-2}), 0^4 1^3, c\} \end{Bmatrix}. \tag{55}$$

where $c = c(s_{k+2}^{n-k-1}) = c(v_{k+2}^{n-k-1})$. By applying (52) to this, we deduce that this equality is indeed upheld, thus implying that $s$ and $v$ are confusable under the absence of multisets $C_{k-1}, C_k, C_{n-k+1}, C_{n-k+2}$.

$\square$