Two Deletion Correcting Codes from Indicator Vectors

Jin Sima, Netanel Raviv, and Jehoshua Bruck

Abstract

Construction of capacity achieving deletion correcting codes has been a baffling challenge for decades. A recent breakthrough by Brakensiek *et al.*, alongside novel applications in DNA storage, have reignited the interest in this longstanding open problem. In spite of recent advances, the amount of redundancy in existing codes is still orders of magnitude away from being optimal. In this paper, a novel approach for constructing binary two-deletion correcting codes is proposed. By this approach, parity symbols are computed from indicator vectors (i.e., vectors that indicate the positions of certain patterns) of the encoded message, rather than from the message itself. Most interestingly, the parity symbols and the proof of correctness are a direct generalization of their counterparts in the Varshamov-Tenengolts construction. Our techniques require $7\log(n) + o(\log(n)$ redundant bits to encode an n-bit message, which is near-optimal.

I. Introduction

A *deletion* in a binary sequence $\boldsymbol{c} = (c_1, \ldots, c_n) \in \{0,1\}^n$ is the case where a symbol is removed from $\boldsymbol{c}$, which results in a subsequence length $n-1$. Similarly, the result of a k-deletion is a subsequence of $\boldsymbol{c}$ of length $n-k$. A k-deletion code $\mathcal{C}$ is a set of n-bit sequences, no two of which share a common subsequence of length $n-k$; and clearly, such a code can correct any k-deletion.

It has been proved in [1] that the largest size $L_k(n)$ of a k-deletion code satisfies

$$\frac{2^k(k!)^2 2^n}{n^{2k}} \lesssim L_k(n) \lesssim \frac{k!2^n}{n^k}, \tag{1}$$

which implies the existence of a k-deletion code with at most $2k\log(n) + o(\log n)$ bits of redundancy for a constant k. However, to this day no explicit construction of such code is known beyond the case $k = 1$.

For $k = 1$, the well-known Varshamov-Tenengolts (VT) [2] construction

$$\left\{ \boldsymbol{c} : \sum_{i=1}^{n} ic_i = 0 \bmod (n+1) \right\} \tag{2}$$

can correct one deletion with not more than $\log(n+1)$ bits of redundancy [1]. Several attempts to generalize the VT construction to $k > 1$ have been made. In the construction of [3], a modified Fibonacci sequence is used as weights instead of $(1, 2, \ldots, n)$ in (2). In [4], number-theoretic arguments are used to obtain k-deletion correction in run-length limited sequences. Yet, both [3] and [4] have rates that are asymptotically bounded away from 1.

The problem of finding an explicit k-deletion code of rate that approaches 1 as n grows has long been unsettled. Only recently, a code with $O(k^2 \log k \log n)$ redundancy bits and encoding/decoding complexity[1] of $O_k(n\log^4 n)$ was proposed in [5]. This code is based on a k-deletion code of length $\log n$, which is constructed using computer search. Nevertheless, the constants that are involved in the work of [5] are orders of magnitude away from the

The work was presented in part at the IEEE International Symposium on Information Theory, July 2018. The work was supported in part by NSF grant CCF-1717884. The work of Netanel Raviv was supported in part by the postdoctoral fellowship of the Center for the Mathematics of Information (CMI), Caltech, and in part by the Lester-Deutsch postdoctoral fellowship.

Jin Sima is with the Electrical Engineering Department, California Institute of Technology, Pasadena, CA, 91125, Email: jsima@caltech.edu.

Netanel Raviv is with the Electrical Engineering Department, California Institute of Technology, Pasadena, CA, 91125, Email: netanel.raviv@gmail.com.

Jehoshua Bruck is with the Electrical Engineering Department, California Institute of Technology, Pasadena, CA, 91125, Email: bruck@caltech.edu.

[1]Here O_k denotes *parameterized complexity*, i.e., $O_k(n\log^4 n) = f(k)O(n\log^4 n)$ for some function f.

lower bound in (1) even for $k = 2$, and the code is not systematic. Moreover, finding a k-deletion correcting code with an asymptotic rate 1 as an extension of the VT construction remains widely open[2].

One such potential extension is using higher order parity checks $\sum_{i=1}^{n} i^j c_i = 0 \bmod (n^j + 1)$ for $j = 1, \ldots, t$, but counterexamples are easily constructible even for $k = 2$. In this paper, we find that similar higher order parity checks work when $t = 3$, given that we restrict our attention to sequences with no consecutive ones. Consequently, applying these parity checks on certain *indicator vectors* yields the desired result. For a and b in $\{0, 1\}$ and a binary sequence $\boldsymbol{c}$, the ab-indicator $\mathbb{1}_{ab}(\boldsymbol{c}) \in \{0, 1\}^{n-1}$ of $\boldsymbol{c}$ is

$$\mathbb{1}_{ab}(c)_i = \begin{cases} 1 & \text{if } c_i = a \text{ and } c_{i+1} = b \\ 0 & \text{else.} \end{cases}.$$

Since any two 10 or 01 patterns are at least two positions apart, the 10- and 01-indicators of any n-bit sequence do not contain consecutive ones, and hence higher order parity checks can be applied.

The parity checks in the proposed code rely on the following integer vectors.

$$\begin{aligned} \mathbf{m}^{(0)} &\triangleq (1, 2, \ldots, n-1) \\ \mathbf{m}^{(1)} &\triangleq \left(1, 1+2, 1+2+3, \ldots, \frac{n(n-1)}{2}\right) \\ \mathbf{m}^{(2)} &\triangleq \left(1^2, 1^2+2^2, 1^2+2^2+3^2, \ldots, \frac{n(n-1)(2n-1)}{6}\right). \end{aligned}$$

Further, for $\mathbf{c} \in \{0, 1\}^n$ let

$$\begin{aligned} f(\mathbf{c}) \triangleq (&\mathbb{1}_{10}(\mathbf{c}) \cdot \mathbf{m}^{(0)} \bmod 2n, \\ &\mathbb{1}_{10}(\mathbf{c}) \cdot \mathbf{m}^{(1)} \bmod n^2, \\ &\mathbb{1}_{10}(\mathbf{c}) \cdot \mathbf{m}^{(2)} \bmod n^3), \text{ and} \\ h(\mathbf{c}) \triangleq (&\mathbb{1}_{01}(\mathbf{c}) \cdot \mathbb{1} \bmod 3, \mathbb{1}_{01}(\mathbf{c}) \cdot \mathbf{m}^{(1)} \bmod 2n), \end{aligned} \tag{3}$$

where $\cdot$ denotes inner product over the integers, and $\mathbb{1}$ denotes the all 1's vector.

For any integer k let $B_k(\mathbf{c})$ be the *k-deletion ball* of $\mathbf{c}$, i.e., the set of n-bit sequences that share a common $n-k$ subsequence with $\mathbf{c}$. The main result of this paper, from which a code construction is immediate, is as follows.

Theorem 1. *For any integer $n \geq 3$ and $N = n + 7\log n + o(\log n)$, there exists an encoding function $\mathcal{E} : \{0, 1\}^n \to \{0, 1\}^N$ and a decoding function $\mathcal{D} : \{0, 1\}^{N-2} \to \{0, 1\}^n$ such for any $\mathbf{c} \in \{0, 1\}^n$ and subsequence $\mathbf{c}' \in \{0, 1\}^{N-2}$ of $\mathcal{E}(\mathbf{c})$, we have $\mathcal{D}(\mathbf{c}') = \mathbf{c}$. In addition, functions $\mathcal{E}$ and $\mathcal{D}$ can be computed in $O(n)$ time.*

To prove this, we first show that the parities $f(\mathbf{c})$ and $h(\mathbf{c})$ can be used to correct two deletions.

Theorem 2. *For $\mathbf{c}, \mathbf{c}' \in \{0, 1\}^n$, if $\mathbf{c} \in B_2(\mathbf{c}')$, $f(\mathbf{c}) = f(\mathbf{c}')$, and $h(\mathbf{c}) = h(\mathbf{c}')$, then $\mathbf{c} = \mathbf{c}'$.*

Theorem 2 readily implies that that the functions h and f can serve as the redundancy bits in a 2-deletion code, and that the induced redundancy is at most $7\log(n) + o(\log n)$ (the additional term stems from protecting the redundancy bits). Furthermore, the encoding algorithm is trivial, and the decoding algorithm in Section VI is linear. Most interestingly, the proof of Theorem 2 can be seen as a higher dimensional variant of the proof for the VT construction, as explained in the remainder of this section.

Clearly, a length $n-1$ VT code can be seen as the set of sequences $\mathbf{c}$ for which the values of $\ell(\mathbf{c}) \triangleq \mathbf{c} \cdot \mathbf{m}^{(0)} \bmod n$ coincide. Adopting this point of view, the correctness of the VT construction can be proved by the following lemma, in which $\ell_v(\mathbf{c}) \triangleq \mathbf{c} \cdot \mathbf{v} \bmod (v_{n-1} + 1)$, and $\mathbf{v} = (v_1, \ldots, v_{n-1})$ is a vector in $\mathbb{Z}_+^{n-1}$.

Lemma 1. *For $\mathbf{c}, \mathbf{c}' \in \{0, 1\}^{n-1}$, and $\mathbf{v} \in \mathbb{Z}_+^{n-1}$, if $\mathbf{c} \in B_1(\mathbf{c}')$, $\ell_v(\mathbf{c}) = \ell_v(\mathbf{c}')$, and $v_1 < v_2 < \ldots < v_{n-1}$ then $\mathbf{c} = \mathbf{c}'$.*

[2]For $k = 2$, [6] has very recently improved the redundancy up to $8\log n$ using techniques similar to [5], our techniques incur lower redundancy and complexity, and use a fundamentally different approach.

In turn, the proof of this lemma can be completed by defining the following function. For a vector $\mathbf{v} \in \mathbb{Z}_+^{n-1}$, an integer $r \in [n-1]$, and a binary vector $\mathbf{x} = (x_1, \ldots, x_s)$ with $r + s - 2 \leq n - 1$, let

$$\begin{aligned} g_{\mathbf{v}}(r, \mathbf{x}) &\triangleq \mathbf{x} \cdot ((\mathbf{v}^{(r,r+s-2)}, 0) - (0, \mathbf{v}^{(r,r+s-2)})) \\ &= x_1 v_r - x_s v_{r+s-2} + \sum_{t=2}^{s-1} x_t (v_{t+r-1} - v_{t+r-2}), \end{aligned} \tag{4}$$

where $\mathbf{v}^{(r,r+s-2)} \triangleq (v_r, v_{r+1}, \ldots, v_{r+s-2})$, and '$\cdot$' denotes inner product. Let k_1 and k_2 ($k_1 < k_2$) be the indices of the deletions after which $\mathbf{c}$ and $\mathbf{c}'$ are identical. Then we have

$$\begin{aligned} \mathbf{c}_t &= \mathbb{1}_{10}(\mathbf{c}')_t && \text{if } t < k_1 \\ & && \text{or } t > k_2, \text{ and} \\ \mathbf{c}_{t+1} &= \mathbb{1}_{10}(\mathbf{c}')_t && \text{if } k_1 \leq t \leq k_2 - 1. \end{aligned} \tag{5}$$

One can find that

$$\begin{aligned} \mathbf{c} \cdot \mathbf{v} - \mathbf{c}' \cdot \mathbf{v} &= \sum_{t=1}^{k_1-1} c_t v_t + c_{k_1} v_{k_1} + \sum_{t=k_1+1}^{k_2} c_t v_t + \sum_{t=k_2+1}^{n} c_t v_t \\ &\quad - \sum_{t=1}^{k_1-1} c_t v_t + \sum_{t=k_1}^{k_2-1} c_{t+1} v_t + c'_{k_2} v_{k_2} + \sum_{t=k_2+1}^{n} c_t v_t \\ &= c_{k_1} v_{k_1} + \sum_{t=k_1+1}^{k_2} c_t (v_t - v_{t-1}) - c_{k_2} v_{k-2} \\ &= g_{\mathbf{v}}(k_1, (\mathbf{c}^{(k_1,k_2)}, c'_{k_2})) \end{aligned} \tag{6}$$

Hence, if $\ell_v(\mathbf{c}) = \ell_v(\mathbf{c}')$ then $g_{\mathbf{v}}(k_1, (\mathbf{c}^{(k_1,k_2)}, c'_{k_2})) \equiv 0 \bmod (v_{n-1} + 1)$. Furthermore, since

$$\begin{aligned} -v_{n-1} \leq -v_{r+s-2} \leq x_1 v_r - x_s v_{r+s-2} + \sum_{t=2}^{s-1} x_t (v_{t+r-1} - v_{t+r-2}) \\ \leq v_r + \sum_{t=2}^{s-1} (v_{t+r-1} - v_{t+r-2}) = v_{r+s-2} \leq v_{n-1}, \end{aligned}$$

it follows that $\ell_v(\mathbf{c}) = \ell_v(\mathbf{c}')$ if and only if $g_{\mathbf{v}}(k_1, (\mathbf{c}^{(k_1,k_2)}, c'_{k_2})) = 0$. Therefore, the proof is concluded by the following lemma.

Lemma 2. *For integers r and s such that $r + s - 2 \leq n - 1$, a vector $\mathbf{v} \in \mathbb{Z}_+^{n-1}$, and an s-bit binary vector $\mathbf{x}$, if $g_{\mathbf{v}}(r, \mathbf{x}) = 0$ and $v_1 < \ldots < v_{n-1}$ then $\mathbf{x}$ is a constant vector.*

Proof. We dinstinguish between two cases according to the value of x_s. On the one hand, if $x_s = 0$, then it is readily verified that $g_{\mathbf{v}}(r, \mathbf{x})$ is the sum of nonnegative terms. In which case, the equation $g_{\mathbf{v}}(r, \mathbf{x}) = 0$ holds if and only if $\mathbf{x} = 0$.

On the other hand, if $x_s = 1$, then

$$\begin{aligned} g_{\mathbf{v}}(r, \mathbf{x}) &= v_r x_1 + \sum_{t=2}^{s-1} (v_{t+r-1} - v_{t+r-2}) x_t - v_{r+s-2} \\ &\leq v_r + \sum_{t=2}^{s-1} (v_{t+r-1} - v_{t+r-2}) - v_{r+s-2} = 0. \end{aligned} \tag{7}$$

The equality holds if and only if $\mathbf{x} = 1$. □

Remark 1. *The VT code is the special case when $\mathbf{v} = (1, \ldots, n)$. From Lemma 2 we have that $c_{k_1} = \ldots = c_{k_2} = c'_{k_2}$. According to Equation 5, this implies that $c'_t = c_{t+1} = c_t$ for $k_1 \leq t \leq k_2 - 1$ and $c'_t = c_t$ for $t < k_1$ or $t \geq k_2$.*

$c_{i-1}c_ic_{i+1}$	000	001	010	011	100	101	110	111
$\mathbb{1}_{10}(c)_{i-1}\mathbb{1}_{10}(c)_i$	**00**	**00**	**01**	**00**	**10**	**10**	**01**	**00**
$\mathbb{1}_{01}(c)_{i-1}\mathbb{1}_{01}(c)_i$	**00**	**01**	**10**	**10**	**00**	**01**	**00**	**00**

TABLE I
ALL POSSIBLE CASES OF DELETIONS OF c_i FOR $2 \leq i \leq n-1$ CORRESPOND TO DELETIONS IN $\mathbb{1}_{10}(\boldsymbol{c})$. THE DELETED SYMBOL IS IN BOLD.

The crux of proving Theorem 2 boils down to the following higher dimensional variant of Lemma 2.

Lemma 3. *For integers r_1, r_2, s_1, and s_2 such that $r_2 > r_1 + s_1 - 2$ and $r_2 + s_2 - 2 \leq n-1$, and binary sequences $\mathbf{x}$ and $\mathbf{y}$ of lengths s_1 and s_2, respectively, if*

$$\begin{aligned} g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) + \lambda g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y}) = 0, \text{ and} \\ g_{\mathbf{m}^{(1)}}(r_1, \mathbf{x}) + \lambda g_{\mathbf{m}^{(1)}}(r_2, \mathbf{y}) = 0, \end{aligned} \tag{8}$$

where $\lambda = \pm 1$, then $\mathbf{x}$ and $\mathbf{y}$ are constant vectors.

Additional technical claims, which involve the remaining ingredients of the redundancy bits, are given in the sequel.

II. OUTLINE

The proof of Theorem 2 is separated to the following two lemmas. In a nutshell, it is shown that for two confusable sequences, i.e., that share a common $n-2$ subsequence, if the f redundancies coincide, then so are the 10-indicators. Then, it is shown that confusable sequences with identical 10-indicators and identical h redundancy have identical 01-indicators.

Lemma 4. *For $\mathbf{c}$ and $\mathbf{c}'$ in $\{0,1\}^n$, if $\mathbf{c} \in B_2(\mathbf{c}')$ and $f(\mathbf{c}) = f(\mathbf{c}')$, then $\mathbb{1}_{10}(\mathbf{c}) = \mathbb{1}_{10}(\mathbf{c}')$.*

Lemma 5. *For $\mathbf{c}$ and $\mathbf{c}'$ in $\{0,1\}^n$ such that $\mathbf{c} \in B_2(\mathbf{c}')$, if $\mathbb{1}_{10}(\mathbf{c}) = \mathbb{1}_{10}(\mathbf{c}')$ and $h(\mathbf{c}) = h(\mathbf{c}')$, then $\mathbb{1}_{01}(\mathbf{c}) = \mathbb{1}_{01}(\mathbf{c}')$.*

From these lemmas it is clear that two n-bit sequences that share a common $n-2$ subsequence and agree on the redundancies f and h have identical 10- and 01-indicators, and hence the next simple lemma concludes the proof of Theorem 2.

Lemma 6. *For $\mathbf{c}$ and $\mathbf{c}'$ in $\{0,1\}^n$ such that $\mathbf{c} \in B_2(\mathbf{c}')$, if $\mathbb{1}_{10}(\mathbf{c}) = \mathbb{1}_{10}(\mathbf{c}')$ and $\mathbb{1}_{01}(\mathbf{c}) = \mathbb{1}_{01}(\mathbf{c}')$ then $\mathbf{c} = \mathbf{c}'$.*

Proof. The conditions $\mathbb{1}_{10}(\boldsymbol{c}) = \mathbb{1}_{10}(\boldsymbol{c}')$ and $\mathbb{1}_{01}(\boldsymbol{c}) = \mathbb{1}_{01}(\boldsymbol{c}')$ imply that the ascending and descending transition positions of $\mathbb{1}_{01}(\boldsymbol{c})$ coincide with those of $\mathbb{1}_{01}(\boldsymbol{c}')$ respectively. Hence if transitions happen in $\boldsymbol{c}$ or $\boldsymbol{c}'$, then $\boldsymbol{c} = \boldsymbol{c}'$. If no transitions happen in $\boldsymbol{c}$ or $\boldsymbol{c}'$ and $\boldsymbol{c} \neq \boldsymbol{c}'$, then one of $\boldsymbol{c}$ and $\boldsymbol{c}'$ is all 0's vector and the other is all 1's vector. Since all 0's vector does not share a common subsequence of length $n-2$ with all 1's vector, we conclude that $\boldsymbol{c} = \boldsymbol{c}'$. □

The proofs of Lemma 4 and Lemma 5 make extensive use of the following two technical claims, that are easy to prove.

Lemma 7. *For $\mathbf{c}$ and $\mathbf{c}'$ in $\{0,1\}^n$, if $\mathbf{c} \in B_2(\mathbf{c}')$ then $\mathbb{1}_{10}(\mathbf{c}) \in B_2(\mathbb{1}_{10}(\mathbf{c}'))$ and $\mathbb{1}_{01}(\mathbf{c}) \in B_2(\mathbb{1}_{01}(\mathbf{c}'))$.*

Proof. We first show that if $\boldsymbol{c} \in B_1(\boldsymbol{c}')$ then $\mathbb{1}_{10}(\boldsymbol{c}) \in B_1(\mathbb{1}_{10}(\boldsymbol{c}'))$ and $\mathbb{1}_{01}(\boldsymbol{c}) \in B_1(\mathbb{1}_{01}(\boldsymbol{c}'))$. To this end, it suffices to show that if $\mathbf{d} \in \{0,1\}^{n-1}$ is obtained from $\boldsymbol{c}$ by one deletion, then $\mathbb{1}_{10}(\mathbf{d})$ ($\mathbb{1}_{01}(\mathbf{d})$) is obtained from $\mathbb{1}_{10}(\boldsymbol{c})$ ($\mathbb{1}_{01}(\boldsymbol{c})$) by one deletion (see table I).

Further, it is easy to see that a deletion of c_1 corresponds to a deletion of $\mathbb{1}_{10}(c)_1$ (resp. $\mathbb{1}_{01}(c)_1$) and a deletion of c_n corresponds to a deletion of $\mathbb{1}_{10}(c)_{n-1}$ (resp. $\mathbb{1}_{01}(c)_{n-1}$). Hence, it follows that if

$$\begin{aligned} \mathbf{c} \overset{1 \text{ del'}}{\longrightarrow} \mathbf{d} \overset{1 \text{ del'}}{\longrightarrow} \mathbf{e} \\ \mathbf{c}' \overset{1 \text{ del'}}{\longrightarrow} \mathbf{d}' \overset{1 \text{ del'}}{\longrightarrow} \mathbf{e} \end{aligned}$$

then

$$\begin{aligned}
\mathbb{1}_{10}(\mathbf{c}) &\xrightarrow{1\text{ del'}} \mathbb{1}_{10}(\mathbf{d}) \xrightarrow{1\text{ del'}} \mathbb{1}_{10}(\mathbf{e})\\
\mathbb{1}_{10}(\mathbf{c}') &\xrightarrow{1\text{ del'}} \mathbb{1}_{10}(\mathbf{d}') \xrightarrow{1\text{ del'}} \mathbb{1}_{10}(\mathbf{e})\\
\mathbb{1}_{01}(\mathbf{c}) &\xrightarrow{1\text{ del'}} \mathbb{1}_{01}(\mathbf{d}) \xrightarrow{1\text{ del'}} \mathbb{1}_{01}(\mathbf{e})\\
\mathbb{1}_{01}(\mathbf{c}') &\xrightarrow{1\text{ del'}} \mathbb{1}_{01}(\mathbf{d}') \xrightarrow{1\text{ del'}} \mathbb{1}_{01}(\mathbf{e}),
\end{aligned}$$

which concludes the claim. □

Lemma 8. *For* $\mathbf{c}, \mathbf{c}' \in \{0,1\}^n$*, if* $\mathbf{c} \in B_2(\mathbf{c}')$ *and* $\mathbb{1}_{01}(\mathbf{c}) \cdot \mathbb{1} = \mathbb{1}_{01}(\mathbf{c}') \cdot \mathbb{1} \bmod 3$*, then* $\mathbb{1}_{01}(\mathbf{c}) \cdot \mathbb{1} = \mathbb{1}_{01}(\mathbf{c}') \cdot \mathbb{1}$.

Proof. Since $\boldsymbol{c} \in B_2(\boldsymbol{c}')$ it follows from Lemma 7 that $\mathbb{1}_{10}(\boldsymbol{c}) \in B_2(\mathbb{1}_{10}(\boldsymbol{c}'))$, and thus $\mathbb{1}_{10}(\boldsymbol{c})$ and $\mathbb{1}_{10}(\boldsymbol{c}')$ have a mutual $(n-3)$-bit string $\mathbf{s}$. Clearly,

$$\begin{aligned}
\mathbf{s} \cdot \mathbb{1} \le \mathbb{1}_{10}(\boldsymbol{c}) \cdot \mathbb{1} \le \mathbf{s} \cdot \mathbb{1} + 2, \text{ and}\\
\mathbf{s} \cdot \mathbb{1} \le \mathbb{1}_{10}(\boldsymbol{c}') \cdot \mathbb{1} \le \mathbf{s} \cdot \mathbb{1} + 2,
\end{aligned}$$

and thus $|\mathbb{1}_{10}(\boldsymbol{c}) \cdot \mathbb{1} - \mathbb{1}_{10}(\boldsymbol{c}') \cdot \mathbb{1}| \le 2$. However, since 3 divides $|\mathbb{1}_{10}(\boldsymbol{c}) \cdot \mathbb{1} - \mathbb{1}_{10}(\boldsymbol{c}') \cdot \mathbb{1}|$, we must have that $\mathbb{1}_{10}(\boldsymbol{c}) \cdot \mathbb{1} = \mathbb{1}_{10}(\boldsymbol{c}') \cdot \mathbb{1}$. □

In addition, one of the cases of the proof of Lemma 4 requires a specialized variant of Lemma 3.

Lemma 9. *Let* r_1, r_2, s_1, s_2 *and* s_3 *be positive integers that satisfy* $r_2 = r_1 + s_1$ *and* $r_2 + s_2 + s_3 \le n - 1$*, and let* $\mathbf{x} \in \{0,1\}^{s_1+s_2+1}$ *and* $\mathbf{y} \in \{0,1\}^{1+s_2+s_3}$ *be such that*

$$(x_{s_1+1}, x_{s_1+2}, \ldots, x_{s_1+s_2}) = (y_2, y_3, \ldots, y_{s_2+1}),$$

and $(x_{s_1+1}, x_{s_1+2}, \ldots, x_{s_1+s_2})$ *has no adjacent* 1*'s. If*

$$\begin{aligned}
g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y}) = 0,\\
g_{\mathbf{m}^{(1)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(1)}}(r_2, \mathbf{y}) = 0, \text{ and}\\
g_{\mathbf{m}^{(2)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(2)}}(r_2, \mathbf{y}) = 0,
\end{aligned} \tag{9}$$

then either $x_1 = \ldots = x_{s_1+s_2+1} = y_1 = \ldots = y_{s_2+s_3+1}$ *or*

$$\begin{aligned}
x_1 = x_2 = \ldots = x_{s_1+1} = 1 - y_1,\\
x_t + x_{t+1} = 1, \text{ for } t \in \{s_1 + 1, \ldots, s_1 + s_2 - 1\},\\
x_{s_1+s_2+1} + y_{s_2+1} = 1, \text{ and}\\
y_{s_2+1} = \ldots = y_{s_2+s_3+1}.
\end{aligned} \tag{10}$$

The following lemma shows a property of $g_{\mathbf{v}}(r, \mathbf{x})$, which will be useful in the proof of Lemma 3 and Lemma 9 that are given in Section V.

Lemma 10. *For integers* r *and* s *such that* $r + s - 2 \le n - 1$*, a vector* $\mathbf{v}$*, and an* s*-bit binary vector* $\mathbf{x}$*, if* $g_{\mathbf{v}}(r, \mathbf{x}) = 0$*, then* $g_{\mathbf{v}}(r, \overline{\mathbf{x}}) = 0$*, where* $\overline{\mathbf{x}} \triangleq \mathbb{1} - \mathbf{x}$.

Proof. Since

$$\begin{aligned}
g_{\mathbf{v}}(r, \mathbf{x}) &= v_r x_1 + \sum_{t=2}^{s-1} (v_{t+r-1} - v_{t+r-2}) x_t - v_{r+s-2} x_s\\
&= v_r x_1 + \sum_{t=2}^{s-1} (v_{t+r-1} - v_{t+r-2}) x_t - v_{r+s-2} x_s - v_r - \sum_{t=2}^{s-1} (v_{t+r-1} - v_{t+r-2}) + v_{r+s-2}\\
&= v_r (x_1 - 1) + \sum_{t=2}^{s-1} (v_{t+r-1} - v_{t+r-2})(x_t - 1) - v_{r+s-2}(x_s - 1) = -g_{\mathbf{v}}(r, \overline{\mathbf{x}})
\end{aligned} \tag{11}$$

Hence if $g_{\mathbf{v}}(r,\mathbf{x})=0$, we have $g_{\mathbf{v}}(r,\overline{\mathbf{x}})=0$. □

Lemma 5 is proved in Section III, and its more involved counterpart Lemma 4 is proved in Section IV. Finally, Lemma 3 and Lemma 9 are proved in Section V.

III. Proof of Lemma 5

We now show that for any $\mathbf{c}$ and $\mathbf{c}'$ in $\{0,1\}^n$ that satisfy $\mathbf{c}\in B_2(\mathbf{c}')$, if $\mathbb{1}_{10}(\mathbf{c})=\mathbb{1}_{10}(\mathbf{c}')$ and $h(\mathbf{c})=h(\mathbf{c}')$ (see (3) for definition of the h function), then $\mathbb{1}_{01}(\mathbf{c})=\mathbb{1}_{01}(\mathbf{c}')$. Since $\mathbf{c}$ and $\mathbf{c}'$ have an identical 10-indicator, they can be written as

$$\begin{aligned}\mathbf{c}&=0^{\pi_0}1^{\pi_1}0^{\pi_2}1^{\pi_3}\cdots 0^{\pi_{2\ell}}1^{\pi_{2\ell+1}},\\ \mathbf{c}'&=0^{\tau_0}1^{\tau_1}0^{\tau_2}1^{\tau_3}\cdots 0^{\tau_{2\ell}}1^{\tau_{2\ell+1}},\end{aligned}\tag{12}$$

where $\{\pi_i\}_{i=0}^{2\ell+1}$ and $\{\tau_i\}_{i=0}^{2\ell+1}$ are nonnegative integers such that π_i and τ_i are strictly positive for every $i\notin\{0,2\ell+1\}$, and such that $\pi_{2i}+\pi_{2i+1}=\tau_{2i}+\tau_{2i+1}$ for all $i\in\{0,1,\ldots,\ell\}$. In addition, since $h(\mathbf{c})_1=h(\mathbf{c}')_1$ it follows from Lemma 8 that $\mathbb{1}_{01}(\mathbf{c})\cdot\mathbb{1}=\mathbb{1}_{01}(\mathbf{c}')\cdot\mathbb{1}$. Hence, we have

$$\begin{aligned}\mathbb{1}_{01}(\mathbf{c})\cdot\mathbb{1}=\mathbb{1}_{01}(\mathbf{c}')\cdot\mathbb{1}=\ell+1 \text{ if } &\quad \pi_0>0,\ \pi_{2\ell+1}>0\\ \mathbb{1}_{01}(\mathbf{c})\cdot\mathbb{1}=\mathbb{1}_{01}(\mathbf{c}')\cdot\mathbb{1}=\ell \text{ if } &\quad \pi_0>0,\ \pi_{2\ell+1}\le 0\\ &\quad \text{or } \pi_0=0,\ \pi_{2\ell+1}>0\\ \mathbb{1}_{01}(\mathbf{c})\cdot\mathbb{1}=\mathbb{1}_{01}(\mathbf{c}')\cdot\mathbb{1}=\ell-1 \text{ if } &\quad \pi_0<0,\ \pi_{2\ell+1}<0\end{aligned}$$

if π_0 and $\pi_{2\ell+1}$ (resp. τ_0 and $\tau_{2\ell+1}$) are both positive then this number is $\ell+1$, if precisely one of them is positive then it is ℓ, and if they are both zero it is $\ell-1$.

Let $\mathbf{d}=0^{\gamma_0}1^{\gamma_1}0^{\gamma_2}1^{\gamma_3}\cdots 0^{\gamma_{2\ell}}1^{\gamma_{2\ell+1}}\in\{0,1\}^{n-2}$ be a common subsequence of $\mathbf{c}$ and $\mathbf{c}'$ which is obtained by deleting two bits from either $\mathbf{c}$ or $\mathbf{c}'$, where $\gamma_i\ge 0$ for all i. Then, it is readily verified that

$$\begin{aligned}&\sum_{i=0}^{2\ell+1}(\pi_i-\gamma_i)=2,\ \sum_{i=0}^{2\ell+1}(\tau_i-\gamma_i)=2,, \text{ and hence}\\ &\sum_{i=1}^{2\ell+1}|\pi_i-\tau_i|\le\sum_{i=1}^{2\ell+1}|\pi_i-\gamma_i|+\sum_{i=1}^{2\ell+1}|\tau_i-\gamma_i|=4.\end{aligned}$$

Moreover, since $\pi_{2i}+\pi_{2i+1}=\tau_{2i}+\tau_{2i+1}$ for all $i\in\{0,1,\ldots,\ell\}$, it follows that $|\pi_{2i}-\tau_{2i}|=|\pi_{2i+1}-\tau_{2i+1}|$. Assuming for contradiction that the 01-indicators do not coincide implies either of the following cases.

Case (a). There exists an integer $j\in[\ell]$ such that $|\pi_{2j}-\tau_{2j}|$ is either 1 or 2 and $\pi_{2i}=\tau_{2i}$ for $i\neq j$.

Case (b). There exist two integers m and r (where $m<r$) such that $|\pi_{2m}-\tau_{2m}|=|\pi_{2r}-\tau_{2r}|=1$, and $\pi_{2i}=\tau_{2i}$ for $i\notin\{m,r\}$.

In Case (a), since $\pi_{2i}+\pi_{2i+1}=\tau_{2i}+\tau_{2i+1}$ for every i and $\pi_{2i}=\tau_{2i}$ for every $i\neq j$, it follows that $\mathbb{1}_{01}(\mathbf{c})$ and $\mathbb{1}_{01}(\mathbf{c}')$ differ in precisely two positions s and t such that $1\le s-t\le 2$. Hence, since the number of 1's in the 01-indicators is equal, it follows that $\mathbb{1}_{01}(\mathbf{c})_s=\mathbb{1}_{01}(\mathbf{c}')_t$, $\mathbb{1}_{01}(\mathbf{c})_t=\mathbb{1}_{01}(\mathbf{c}')_s$, and $\mathbb{1}_{01}(\mathbf{c})_s\neq\mathbb{1}_{01}(\mathbf{c})_t$, and therefore

$$\begin{aligned}h(\mathbf{c})_2-h(\mathbf{c}')_2&=(\mathbb{1}_{01}(\mathbf{c})_s-\mathbb{1}_{01}(\mathbf{c}')_s)\binom{s+1}{2}+\\ &\quad(\mathbb{1}_{01}(\mathbf{c})_t-\mathbb{1}_{01}(\mathbf{c}')_t)\binom{t+1}{2}\\ &=\pm\left(\binom{s+1}{2}-\binom{t+1}{2}\right).\end{aligned}\tag{13}$$

Since $1\le s-t\le 2$, it follows that (13) equals either $\pm(t+1)$ or $\pm(2t+3)$, and a contradiction follows since neither of which is 0 modulo $2n$, .

Similarly, in Case (b), if non of $\pi_{2m}, \tau_{2m}, \pi_{2m+1}, \tau_{2m+1}, \pi_{2r}, \tau_{2r}, \pi_{2r+1}, \tau_{2r+1}$ is zero, then $\mathbb{1}_{01}(\mathbf{c})$ and $\mathbb{1}_{01}(\mathbf{c}')$ differ in four positions $s, s+1, t$, and $t+1$, and hence

$$\begin{aligned} h(\mathbf{c})_2 - h(\mathbf{c}')_2 = &(\mathbb{1}_{01}(\mathbf{c})_s - \mathbb{1}_{01}(\mathbf{c}')_s)\binom{s+1}{2} + \\ &(\mathbb{1}_{01}(\mathbf{c})_{s+1} - \mathbb{1}_{01}(\mathbf{c}')_{s+1})\binom{s+2}{2} + \\ &(\mathbb{1}_{01}(\mathbf{c})_t - \mathbb{1}_{01}(\mathbf{c}')_t)\binom{t+1}{2} + \\ &(\mathbb{1}_{01}(\mathbf{c})_{t+1} - \mathbb{1}_{01}(\mathbf{c}')_{t+1})\binom{t+2}{2}. \end{aligned} \tag{14}$$

Once again, since $\mathbb{1}_{01}(\mathbf{c})$ and $\mathbb{1}_{01}(\mathbf{c}')$ have an identical number of 1's, we have that

$$\begin{aligned} \mathbb{1}_{01}(\mathbf{c})_s &= \mathbb{1}_{01}(\mathbf{c}')_{s+1} & \qquad \mathbb{1}_{01}(\mathbf{c})_{s+1} &= \mathbb{1}_{01}(\mathbf{c}')_s \\ \mathbb{1}_{01}(\mathbf{c})_t &= \mathbb{1}_{01}(\mathbf{c}')_{t+1} & \qquad \mathbb{1}_{01}(\mathbf{c})_{t+1} &= \mathbb{1}_{01}(\mathbf{c}')_t \\ \mathbb{1}_{01}(\mathbf{c})_s &\neq \mathbb{1}_{01}(\mathbf{c}')_s & \qquad \mathbb{1}_{01}(\mathbf{c})_t &\neq \mathbb{1}_{01}(\mathbf{c}')_t. \end{aligned}$$

This readily implies that (14) equals either $\pm(s-t)$ or $\pm(s+t+2)$, and since non of which is 0 modulo $2n$, another contradiction is obtained. If $\pi_{2m} = 0$ (resp. $\tau_{2m} = 0$), by the discussion after Eq. (12) it follows that $\tau_{2r+1} = 0$ (resp. $\pi_{2r+1} = 0$), and hence $\mathbb{1}_{01}(\mathbf{c})$ and $\mathbb{1}_{01}(\mathbf{c}')$ differ in the first and last positions. Hence, (14) becomes $\pm(1 - \frac{n(n-1)}{2})$, which is nonzero modulo $2n$, and the claim follows.

IV. Proof of Lemma 4

Since $c \in B_2(c')$ it follows that there exist integers i_1, i_2, j_1, and j_2 such that

$$\begin{aligned} c &\xrightarrow{\text{del' } i_1} d \xrightarrow{\text{del' } j_1} e \\ c' &\xrightarrow{\text{del' } i_2} d' \xrightarrow{\text{del' } j_2} e \end{aligned}$$

and by Lemma 7 it follows that there exist integers ℓ_1, ℓ_2, k_1, and k_2 such that

$$\begin{aligned} \mathbb{1}_{10}(c) &\xrightarrow{\text{del' } \ell_1} \mathbb{1}_{10}(d) \xrightarrow{\text{del' } k_1} \mathbb{1}_{10}(e) \\ \mathbb{1}_{10}(c') &\xrightarrow{\text{del' } \ell_2} \mathbb{1}_{10}(d') \xrightarrow{\text{del' } k_2} \mathbb{1}_{10}(e). \end{aligned}$$

Due to symmetry between $\mathbf{c}$ and $\mathbf{c}'$, we distinguish between the following three cases. In each case, the difference between the f values of $\mathbf{c}$ and $\mathbf{c}'$ are given in terms of the function g (Eq. (4)). Further, the computation of these three differences, which is tedious but straightforward, is deferred to the appendices.

Case (a). If $\ell_1 \leq \ell_2 < k_2 \leq k_1$ (Fig. 1), then

$$\begin{aligned} \mathbb{1}_{10}(\mathbf{c})_t &= \mathbb{1}_{10}(\mathbf{c}')_t & &\text{if } t < \ell_1 \\ & & &\text{or } \ell_2 < t < k_2 \\ & & &\text{or } t > k_1, \\ \mathbb{1}_{10}(\mathbf{c})_{t+1} &= \mathbb{1}_{10}(\mathbf{c}')_t & &\text{if } \ell_1 \leq t \leq \ell_2 - 1, \\ \mathbb{1}_{10}(\mathbf{c})_t &= \mathbb{1}_{10}(\mathbf{c}')_{t+1} & &\text{if } k_2 \leq t \leq k_1 - 1, \end{aligned}$$

Thus, for $e \in \{0, 1, 2\}$,

$$\begin{aligned} (\mathbb{1}_{10}(\mathbf{c}) - \mathbb{1}_{10}(\mathbf{c}')) \cdot \mathbf{m}^{(e)} = &g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) - \\ &g_{\mathbf{m}^{(e)}}(k_2, (\mathbb{1}_{10}(\mathbf{c}')^{(k_2,k_1)}, \mathbb{1}_{10}(\mathbf{c})_{k_1})). \end{aligned} \tag{15}$$

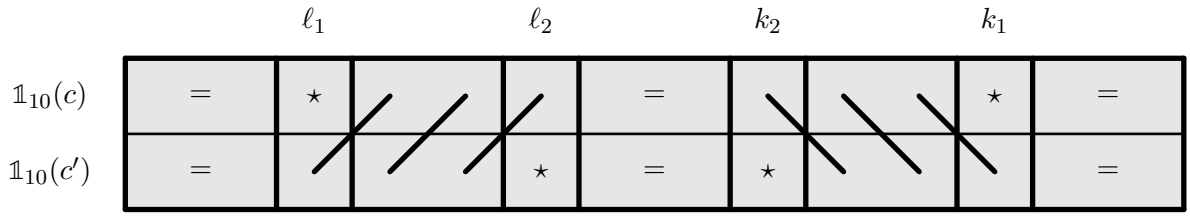

Fig. 1. Case (a)

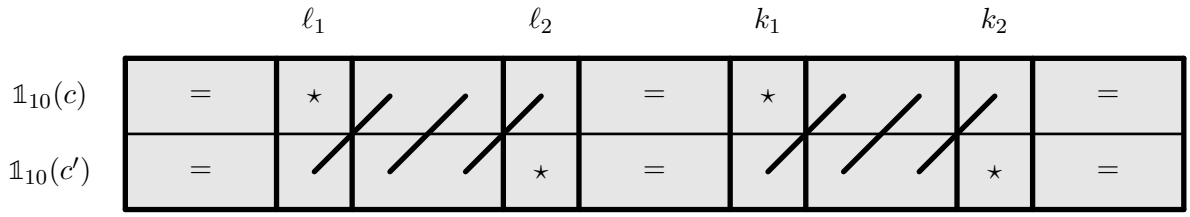

Fig. 2. Case (b)

Case (b). If $\ell_1 \leq \ell_2 < k_1 \leq k_2$ (Fig. 2), then

$$\begin{aligned} \mathbb{1}_{10}(\mathbf{c})_t = \mathbb{1}_{10}(\mathbf{c}')_t \quad & \text{if } t < l_1 \\ & \text{or } l_2 < t < k_1. \\ & \text{or } t > k_2. \\ \mathbb{1}_{10}(\mathbf{c})_{t+1} = \mathbb{1}_{10}(\mathbf{c}')_t \quad & \text{if } \ell_1 \leq t \leq \ell_2 - 1 \\ & \text{or } k_1 \leq t \leq k_2 - 1. \end{aligned}$$

Thus, for $e \in \{0, 1, 2\}$,

$$\begin{aligned} (\mathbb{1}_{10}(\mathbf{c}) - \mathbb{1}_{10}(\mathbf{c}')) \cdot \mathbf{m}^{(e)} = g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) + \\ g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})). \end{aligned} \tag{16}$$

Case (c). If $\ell_1 < k_1 \leq \ell_2 < k_2$ (Fig. 3), then

$$\begin{aligned} \mathbb{1}_{10}(\mathbf{c})_t = \mathbb{1}_{10}(\mathbf{c}')_t \quad & \text{if } t < l_1 \\ & \text{or } t > k_2, \\ \mathbb{1}_{10}(\mathbf{c})_{t+1} = \mathbb{1}_{10}(\mathbf{c}')_t \quad & \text{if } \ell_1 \leq t \leq k_1 - 2 \\ & \text{or } \ell_2 + 1 \leq t \leq k_2 - 1, \\ \mathbb{1}_{10}(\mathbf{c})_{t+2} = \mathbb{1}_{10}(\mathbf{c}')_t \quad & \text{if } k_1 - 1 \leq t \leq \ell_2 - 1. \end{aligned}$$

Thus, for $e \in \{0, 1, 2\}$,

$$\begin{aligned} (\mathbb{1}_{10}(\mathbf{c}) - \mathbb{1}_{10}(\mathbf{c}')) \cdot \mathbf{m}^{(e)} = g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,k_1-1)}, \\ \mathbb{1}_{10}(\mathbf{c})^{(k_1+1,\ell_2+1)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) + \\ g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})). \end{aligned} \tag{17}$$

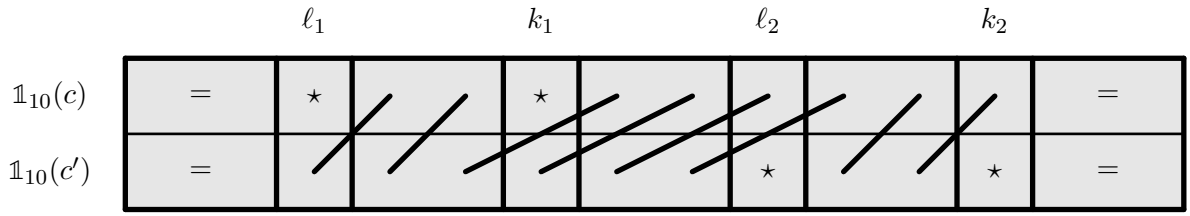

Fig. 3. Case (c)

Note that if $f(\mathbf{c}) = f(\mathbf{c}')$, then $\mathbb{1}_{10}(\mathbf{c}) \cdot \mathbf{m}^{(e)} \equiv \mathbb{1}_{10}(\mathbf{c}) \cdot \mathbf{m}^{(e)} \bmod n_e$, where $n_0 = 2n, n_1 = n^2$, and $n_2 = n^3$. Hence, from (15)-(17) we have that

$$\begin{aligned}
&g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) - g_{\mathbf{m}^{(e)}}(k_2, (\mathbb{1}_{10}(\mathbf{c}')^{(k_2,k_1)}, \mathbb{1}_{10}(\mathbf{c})_{k_1})) \equiv 0 \bmod 2n,\\
&g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) + g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})) \equiv 0 \bmod n^2, \text{ and}\\
&g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,k_1-1)}, \mathbb{1}_{10}(\mathbf{c})^{(k_1+1,\ell_2+1)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}))\\
&+ g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})) \equiv 0 \bmod n^3.g
\end{aligned} \tag{18}$$

In what follows, we show that these equalities also hold in their non modular version. On the other hand, we have

$$-\mathbf{m}^{(e)}_{r+k-2} \le g_{\mathbf{m}^{(e)}}(r, \mathbf{x}) \le \mathbf{m}^{(e)}_{r+k-2}$$

for any $\mathbf{x} \in \{0,1\}^{n-1}$ and any integer r that satisfies $r + k - 2 \le n - 1$. Therefore,

$$\begin{aligned}
-\mathbf{m}^{(e)}_{\ell_2} - \mathbf{m}^{(e)}_{k_2} \le& g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) - g_{\mathbf{m}^{(e)}}(k_2, (\mathbb{1}_{10}(\mathbf{c}')^{(k_2,k_1)}, \mathbb{1}_{10}(\mathbf{c})_{k_1})) \le \mathbf{m}^{(e)}_{\ell_2} + \mathbf{m}^{(e)}_{k_2},\\
-\mathbf{m}^{(e)}_{\ell_2} - \mathbf{m}^{(e)}_{k_2} \le& g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) + g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})) \le \mathbf{m}^{(e)}_{\ell_2} + \mathbf{m}^{(e)}_{k_2}, \text{ and}\\
-\mathbf{m}^{(e)}_{\ell_2} - \mathbf{m}^{(e)}_{k_2} \le& g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,k_1-1)}, \mathbb{1}_{10}(\mathbf{c})^{(k_1+1,\ell_2+1)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}))\\
&+ g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})) \le \mathbf{m}^{(e)}_{\ell_2} + \mathbf{m}^{(e)}_{k_2}.
\end{aligned} \tag{19}$$

Further note that

$$\mathbf{m}^{(0)}_{\ell_2} + \mathbf{m}^{(0)}_{k_2} < 2n, \mathbf{m}^{(1)}_{\ell_2} + \mathbf{m}^{(1)}_{k_2} < n^2, \mathbf{m}^{(2)}_{\ell_2} + \mathbf{m}^{(2)}_{k_2} < n^3 \tag{20}$$

Combining (18), (19), and (20), we conclude that if $f(\mathbf{c}) = f(\mathbf{c}')$, then

$$g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) - g_{\mathbf{m}^{(e)}}(k_2, (\mathbb{1}_{10}(\mathbf{c}')^{(k_2,k_1)}, \mathbb{1}_{10}(\mathbf{c})_{k_1})) = 0, \tag{21}$$

$$g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,\ell_2)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) + g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})) = 0, \text{ and} \tag{22}$$

$$g_{\mathbf{m}^{(e)}}(\ell_1, (\mathbb{1}_{10}(\mathbf{c})^{(\ell_1,k_1-1)}, \mathbb{1}_{10}(\mathbf{c})^{(k_1+1,\ell_2+1)}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2})) + g_{\mathbf{m}^{(e)}}(k_1, (\mathbb{1}_{10}(\mathbf{c})^{(k_1,k_2)}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})) = 0. \tag{23}$$

For Case (a), Equation (21) and Lemma 3 implies that

$$\begin{aligned}
\mathbb{1}_{10}(\mathbf{c})_{\ell_1} = \ldots = \mathbb{1}_{10}(\mathbf{c})_{\ell_2} = \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}\\
\mathbb{1}_{10}(\mathbf{c}')_{k_2} = \ldots = \mathbb{1}_{10}(\mathbf{c}')_{k_1} = \mathbb{1}_{10}(\mathbf{c})_{k_1},
\end{aligned}$$

which readily implies that

$$\mathbb{1}_{10}(\mathbf{c}')_t = \mathbb{1}_{10}(\mathbf{c})_{t+1} = \mathbb{1}_{10}(\mathbf{c})_t$$

for $\ell_1 \le t < \ell_2$ and

$$\mathbb{1}_{10}(\mathbf{c})_t = \mathbb{1}_{10}(\mathbf{c}')_{t+1} = \mathbb{1}_{10}(\mathbf{c}')_t$$

for $k_2 \le t < k_1$. Together with $\mathbb{1}_{10}(\mathbf{c})_{\ell_2} = \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}$ and $\mathbb{1}_{10}(\mathbf{c}')_{k_1} = \mathbb{1}_{10}(\mathbf{c})_{k_1}$, we have that $\mathbb{1}_{10}(\mathbf{c}) = \mathbb{1}_{10}(\mathbf{c}')$.

For Case (b), Equation (22) and Lemma 3 implies that

$$\mathbb{1}_{10}(\mathbf{c})_{\ell_1} = \ldots = \mathbb{1}_{10}(\mathbf{c})_{\ell_2} = \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}$$
$$\mathbb{1}_{10}(\mathbf{c}')_{k_1} = \ldots = \mathbb{1}_{10}(\mathbf{c}')_{k_2} = \mathbb{1}_{10}(\mathbf{c})_{k_2}$$

and hence

$$\mathbb{1}_{10}(\mathbf{c}')_t = \mathbb{1}_{10}(\mathbf{c})_{t+1} = \mathbb{1}_{10}(\mathbf{c})_t$$

for $\ell_1 \leq t < \ell_2$ and $k_1 \leq t < k_2$. $\mathbb{1}_{10}(\mathbf{c}) = \mathbb{1}_{10}(\mathbf{c}')$.

For Case (c), Equation (23) and Lemma 9 imply that either

$$\mathbb{1}_{10}(\mathbf{c})_{\ell_1} = \ldots = \mathbb{1}_{10}(\mathbf{c})_{k_2} = \mathbb{1}_{10}(\mathbf{c}')_{\ell_2} = \mathbb{1}_{10}(\mathbf{c}')_{k_2} \tag{24}$$

or

$$\begin{aligned}
&\mathbb{1}_{10}(\mathbf{c})_{\ell_1} = \ldots = \mathbb{1}_{10}(\mathbf{c})_{k_1-1} = \mathbb{1}_{10}(\mathbf{c})_{k_1+1},\\
&\mathbb{1}_{10}(\mathbf{c})_i + \mathbb{1}_{10}(\mathbf{c})_{i+1} = 1 \text{ for } i \in \{k_1, \ldots, \ell_2\},\\
&\mathbb{1}_{10}(\mathbf{c}')_{\ell_2} + \mathbb{1}_{10}(\mathbf{c}')_{k_2} = 1, \text{ and}\\
&\mathbb{1}_{10}(\mathbf{c})_{\ell_2+1} = \ldots = \mathbb{1}_{10}(\mathbf{c})_{k_2} = \mathbb{1}_{10}(\mathbf{c}')_{k_2}.
\end{aligned} \tag{25}$$

If (24) is true, we can obtain $\mathbf{c} = \mathbf{c}'$ by following similar steps as above.

If (25) is true, we have

$$\mathbb{1}_{10}(\mathbf{c}')_t = \mathbb{1}_{10}(\mathbf{c})_{t+1} = \mathbb{1}_{10}(\mathbf{c})_t$$

for $\ell_1 \leq t \leq k_1 - 2$ and $\ell_2 + 1 \leq t \leq k_2 - 1$. Further more, we have

$$\mathbb{1}_{10}(\mathbf{c}')_t = \mathbb{1}_{10}(\mathbf{c})_{t+2} = 1 - \mathbb{1}_{10}(\mathbf{c})_{t+1} = \mathbb{1}_{10}(\mathbf{c})_t$$

for $k_1 \leq t \leq \ell_2 - 1$. In addition, we have $\mathbb{1}_{10}(\mathbf{c}')_{k_1-1} = \mathbb{1}_{10}(\mathbf{c})_{k_1+1} = \mathbb{1}_{10}(\mathbf{c})_{k_1-1}$, $\mathbb{1}_{10}(\mathbf{c}')_{\ell_2} = 1 - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2+1} = \mathbb{1}_{10}(\mathbf{c})_{\ell_2}$ and $\mathbb{1}_{10}(\mathbf{c}')_{k_2} = \mathbb{1}_{10}(\mathbf{c})_{k_2}$. Therefore, we conclude that $\mathbf{c} = \mathbf{c}'$.

V. Proofs of g-lemmas

Proof. (of Lemma 3) According to Eq. (11), if $\lambda = 1$, then Eq. (8) can be written as

$$g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) - g_{\mathbf{m}^{(0)}}(r_2, \overline{\mathbf{y}}) = 0, \text{ and}$$
$$g_{\mathbf{m}^{(1)}}(r_1, \mathbf{x}) - g_{\mathbf{m}^{(1)}}(r_2, \overline{\mathbf{y}}) = 0.$$

Therefore, it suffices to prove the claim for $\lambda = -1$. We distinguish between four cases according to the value of (y_1, y_{s_2}).

Case (1)**.** $(y_1, y_{s_2}) = (0, 1)$

we have that

$$\begin{aligned}
&g_{\mathbf{m}^{(e)}}(r_1, \mathbf{x}) - g_{\mathbf{m}^{(e)}}(r_2, \mathbf{y})\\
=&\mathbf{m}^{(e)}_{r_1} x_1 + \sum_{t=2}^{s_1-1} (\mathbf{m}^{(e)}_{t+r_1-1} - \mathbf{m}^{(e)}_{t+r_1-2}) x_t -\\
&\mathbf{m}^{(e)}_{r_1+s_1-2} x_{s_1} - \mathbf{m}^{(e)}_{r_2} y_1 - \sum_{t=2}^{s_2-1} (\mathbf{m}^{(e)}_{t+r_2-1} - \mathbf{m}^{(e)}_{t+r_2-2}) y_t + \mathbf{m}^{(e)}_{r_2+s_2-2} y_{s_2}\\
\geq& - \mathbf{m}^{(e)}_{r_1+s_1-2} - \sum_{t=2}^{s_2-1} (\mathbf{m}^{(e)}_{t+r_2-1} - \mathbf{m}^{(e)}_{t+r_2-2}) + \mathbf{m}^{(e)}_{r_2+s_2-2}\\
=&\mathbf{m}^{(e)}_{r_2} - \mathbf{m}^{(e)}_{r_1+s_1-2} > 0,
\end{aligned}$$

a contradiction.

Case (2)**.** $(y_1, y_{s_2}) = (1, 0)$

From Lemma 10 and (8) we have $g_{\mathbf{m}^{(e)}}(r_1, \overline{\mathbf{x}}) + g_{\mathbf{m}^{(e)}}(r_2, \overline{\mathbf{y}}) = 0$ for $e \in \{0, 1\}$, where $\overline{\mathbf{x}} \triangleq \mathbb{1} - \mathbf{x}$ and $\overline{\mathbf{y}} \triangleq \mathbb{1} - \mathbf{y}$. Since $(\overline{y}_1, \overline{y}_{s_2}) = (1, 0)$, from the previous case we have that $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ are constant vectors. So are $\mathbf{x}$ and $\mathbf{y}$.

Case (3)**.** $(y_1, y_{s_2}) = (1, 1)$

Let

$$S_1 \triangleq \{j : y_{j-r_2+1} = 1, r_2 + 1 \leq j \leq r_2 + s_2 - 2\}, \text{ and}$$
$$S_1^c \triangleq \{j : y_{j-r_2+1} = 0, r_2 + 1 \leq j \leq r_2 + s_2 - 2\},$$

and notice that

$$\begin{aligned} g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y}) &= \mathbf{m}^{(0)}_{r_2} - \mathbf{m}^{(0)}_{r_2+s_2-2} + \sum_{j=2}^{s_2-1} (\mathbf{m}^{(0)}_{j+r_2-1} - \mathbf{m}^{(0)}_{j+r_2-2}) y_j \\ &= -\sum_{j=r_2+1}^{r_2+s_2-2} (\mathbf{m}^{(0)}_j - \mathbf{m}^{(0)}_{j-1}) + \sum_{j=2}^{s_2-1} (\mathbf{m}^{(0)}_{j+r_2-1} - \mathbf{m}^{(0)}_{j+r_2-2}) y_j \\ &= -\sum_{j=r_2+1}^{r_2+s_2-2} (\mathbf{m}^{(0)}_j - \mathbf{m}^{(0)}_{j-1})(1 - y_j) \\ &= -\sum_{j \in S_1^c} (\mathbf{m}^{(0)}_j - \mathbf{m}^{(0)}_{j-1}) = -\sum_{j \in S_1^c} 1, \text{ and similarly} \\ g_{\mathbf{m}^{(1)}}(r_2, \mathbf{y}) &= -\sum_{j \in S_1^c} (\mathbf{m}^{(1)}_j - \mathbf{m}^{(1)}_{j-1}) = -\sum_{j \in S_1^c} j. \end{aligned} \tag{26}$$

Now, on the one hand if $x_{s_1} = 0$ we have

$$g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) = \mathbf{m}^{(0)}_{r_1} x_1 + \sum_{t=2}^{s_1-1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2}) x_t \geq 0, \tag{27}$$

and hence, (26) and (27) imply that $g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) - g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y}) \geq 0$, and equality holds only when $g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x})$ and $g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y})$ are both 0, which by Lemma 2 implies that $\mathbf{x}$ and $\mathbf{y}$ are constant vectors. On the other hand, if $x_{s_1} = 1$ let $S_2 = \{j : x_{\max\{j-r_1+1,1\}} = 0, 1 \leq j \leq r_1 + s_1 - 2\}$, and notice that

$$\begin{aligned} g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) &= \mathbf{m}^{(0)}_{r_1} x_1 + \sum_{t=2}^{s_1-1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2}) x_t - \mathbf{m}^{(0)}_{r_1+s_1-2} \\ &= \mathbf{m}^{(0)}_{r_1}(x_1 - 1) + \sum_{t=2}^{s_1-1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2})(x_t - 1) \\ &= -\sum_{t \in S_2} 1, \text{ and similarly} \\ g_{\mathbf{m}^{(1)}}(r_1, \mathbf{x}) &= -\sum_{t \in S_2} t. \end{aligned} \tag{28}$$

Inserting (26) and (28) into (8), we have

$$-\sum_{t \in S_2} 1 + \sum_{j \in S_1^c} 1 = 0,$$
$$-\sum_{t \in S_2} t + \sum_{j \in S_1^c} j = 0.$$

This implies that the sets S_1^c and S_2 have the same cardinality and the same sum of elements. However, the maximum element in S_2 is smaller than the minimum element in S_1^c. Therefore S_1^c and S_2 are empty, which implies that $\mathbf{x}$ is the 0 vector and $\mathbf{y}$ is the all 1's vector.

Case (4)**.** $(y_1, y_{s_2}) = (0, 0)$

From Lemma 10 and Eq. (8) we have $g_{\mathbf{m}^{(e)}}(r_1, \overline{\mathbf{x}}) + g_{\mathbf{m}^{(e)}}(r_2, \overline{\mathbf{y}}) = 0$ for $e \in \{0, 1\}$, where $\overline{\mathbf{x}} \triangleq \mathbb{1} - \mathbf{x}$ and $\overline{\mathbf{y}} \triangleq \mathbb{1} - \mathbf{y}$. Since $(\overline{y}_1, \overline{y}_{s_2}) = (1, 1)$, from the previous case $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ are constant vectors, and thus so are $\mathbf{x}$ and $\mathbf{y}$. □

Proof. (of Lemma 9) We distinguish between four cases according to the value of $(x_{s_1+s_2+1}, y_{s_2+s_3+1})$.

Case (1)**.** $(x_{s_1+s_2+1}, y_{s_2+s_3+1}) = (0, 0)$

Similar to (27), we have that $g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y}) \geq 0$, where equality holds only if $\mathbf{x}$ and $\mathbf{y}$ are constant 0 vectors.

Case (2)**.** $(x_{s_1+s_2+1}, y_{s_2+s_3+1}) = (1, 1)$

From Lemma 10 and Eq. (9) we have $g_{\mathbf{m}^{(0)}}(r_1, \overline{\mathbf{x}}) + g_{\mathbf{m}^{(0)}}(r_2, \overline{\mathbf{y}}) = 0$. On the other hand, since $(\overline{x}_{s_1+s_2+1}, \overline{y}_{s_2+s_3+1}) = (0, 0)$, it follows that $g_{\mathbf{m}^{(0)}}(r_1, \overline{\mathbf{x}}) + g_{\mathbf{m}^{(0)}}(r_2, \overline{\mathbf{y}}) \geq 0$ where equality holds when $\mathbf{x}$ and $\mathbf{y}$ are constant 1 vectors.

Case (3)**.** $(x_{s_1+s_2+1}, y_{s_2+s_3+1}) = (0, 1)$

On the one hand, for $y_1 = 0$ we have

$$
\begin{aligned}
& g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y}) = \\
&= \mathbf{m}^{(0)}_{r_1} x_1 + \sum_{t=2}^{s_1+1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2}) x_t + \sum_{t=s_1+1}^{s_1+s_2-1} (\mathbf{m}^{(0)}_{t+r_1} - \mathbf{m}^{(0)}_{t+r_1-1}) x_{t+1} \\
&\quad + \sum_{t=2}^{s_2} (\mathbf{m}^{(0)}_{t+r_2-1} - \mathbf{m}^{(0)}_{t+r_2-2}) y_t + \sum_{t=s_2+1}^{s_2+s_3} (\mathbf{m}^{(0)}_{t+r_2-1} - \mathbf{m}^{(0)}_{t+r_2-2}) y_t - \mathbf{m}^{(0)}_{r_2+s_2+s_3-1} \\
&= \mathbf{m}^{(0)}_{r_1} x_1 + \sum_{t=2}^{s_1+1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2}) x_t + \sum_{t=s_1+1}^{s_1+s_2-1} (\mathbf{m}^{(0)}_{t+r_1} - \mathbf{m}^{(0)}_{t+r_1-1}) (x_t + x_{t+1}) \\
&\quad + \sum_{t=s_2+1}^{s_2+s_3} (\mathbf{m}^{(0)}_{t+r_2-1} - \mathbf{m}^{(0)}_{t+r_2-2}) y_t - \mathbf{m}^{(0)}_{r_2+s_2+s_3-1} \\
&\leq \mathbf{m}^{(0)}_{r_1} + \sum_{t=2}^{s_1+1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2}) + \sum_{t=s_1+1}^{s_1+s_2-1} (\mathbf{m}^{(0)}_{t+r_1} - \mathbf{m}^{(0)}_{t+r_1-1}) \\
&\quad + \sum_{t=s_2+1}^{s_2+s_3} (\mathbf{m}^{(0)}_{t+r_2-1} - \mathbf{m}^{(0)}_{t+r_2-2}) - \mathbf{m}^{(0)}_{r_2+s_2+s_3-1} = 0,
\end{aligned}
$$

where equality equality holds when

$$
\begin{aligned}
& x_t = 1 \text{ for } t \in \{1, \ldots, s_1 + 1\}, \\
& x_t + x_{t+1} = 1 \text{ for } t \in \{s_1 + 1, \ldots, s_1 + s_2 - 1\}, \text{ and} \\
& y_t = 1 \text{ for } t \in \{s_2 + 1, \ldots, s_2 + s_3\},
\end{aligned}
$$

and hence (10) holds. On the other hand, when $y_1 = 1$, let

$$
\begin{aligned}
S_1 &= \{t : x_{\max\{t-r_1+1,1\}} = 1, 1 \leq t \leq s_1 + r_1\}, \\
S_2 &= \{t : x_{t-r_1} + x_{t-r_1+1} = 0, r_2 + 1 \leq t \leq r_2 + s_2 - 1\}, \\
S_3 &= \{t : y_{t-r_2+1} = 0, r_2 + s_2 \leq t \leq r_2 + s_2 + s_3 - 1\},
\end{aligned}
$$

and notice that

$$\begin{aligned}
&g_{\mathbf{m}^{(0)}}(r_1,\mathbf{x})+g_{\mathbf{m}^{(0)}}(r_2,\mathbf{y})\\
&\quad=\mathbf{m}^{(0)}_{r_1}x_1+\sum_{t=2}^{s_1+1}(\mathbf{m}^{(0)}_{t+r_1-1}-\mathbf{m}^{(0)}_{t+r_1-2})x_t+\\
&\quad\mathbf{m}^{(0)}_{s_1+r_1}+\sum_{t=s_1+1}^{s_1+s_2-1}(\mathbf{m}^{(0)}_{t+r_1}-\mathbf{m}^{(0)}_{t+r_1-1})(x_t+x_{t+1})+\\
&\quad\sum_{t=s_2+1}^{s_2+s_3}(\mathbf{m}^{(0)}_{t+r_2-1}-\mathbf{m}^{(0)}_{t+r_1-2})y_t-\mathbf{m}^{(0)}_{r_2+s_2+s_3-1}\\
&\quad=\sum_{t\in S_1}(\mathbf{m}^{(0)}_t-\mathbf{m}^{(0)}_{t-1})-\sum_{t\in S_2}(\mathbf{m}^{(0)}_t-\mathbf{m}^{(0)}_{t-1})-\sum_{t\in S_3}(\mathbf{m}^{(0)}_t-\mathbf{m}^{(0)}_{t-1})\\
&\quad=\sum_{t\in S_1}1-\sum_{t\in S_2}1-\sum_{t\in S_3}1
\end{aligned}\tag{29}$$

Similarly, we have

$$g_{\mathbf{m}^{(1)}}(r_1,\mathbf{x})+g_{\mathbf{m}^{(1)}}(r_2,\mathbf{y})=\sum_{t\in S_1}t-\sum_{t\in S_2}t-\sum_{t\in S_3}t.\tag{30}$$

Equations (9), (29), and (30) imply that the cardinality of S_1 equals the sum of cardinalities of S_2 and S_3, and in addition, the sum of elements of S_1 equals the sum of elements of S_2 and S_3. Note that the minimum element of $S_2\cup S_3$ is larger than the maximum element of S_1. This is impossible, unless S_1,S_2, and S_3 are empty, which implies that $x_t=0$ for $t\in\{1,\dots,s_1+1\}$, $x_t+x_{t+1}=1$ for $t\in\{s_1+1,\dots,s_1+s_2-1\}$, and $y_t=1$ for $t\in\{s_2+1,\dots,s_2+s_3\}$, and hence (10) holds.

Case (4)**.** $(x_{s_1+s_2+1},y_{s_2+s_3+1})=(1,0)$

On the one hand, for $y_1=0$, let

$$\begin{aligned}
S_1&=\{t:x_{\max\{t-r_1+1,1\}}=0,1\le t\le s_1+r_1\},\\
S_2&=\{t:x_{t-r_1}+x_{t-r_1+1}=0,r_2+1\le t\le r_2+s_2-1\},\\
S_3&=\{t:y_{t-r_2+1}=1,r_2+s_2\le t\le r_2+s_2+s_3-1\}.
\end{aligned}$$

We have

$$\begin{aligned}
&g_{\mathbf{m}^{(0)}}(r_1,\mathbf{x})+g_{\mathbf{m}^{(0)}}(r_2,\mathbf{y})\\
&=\mathbf{m}^{(0)}_{r_1}x_1+\sum_{t=2}^{s_1+1}(\mathbf{m}^{(0)}_{t+r_1-1}-\mathbf{m}^{(0)}_{t+r_1-2})x_t+\sum_{t=s_1+1}^{s_1+s_2-1}(\mathbf{m}^{(0)}_{t+r_1}-\mathbf{m}^{(0)}_{t+r_1-1})(x_t+x_{t+1})-\\
&\quad\mathbf{m}^{(0)}_{r_1+s_1+s_2-1}+\sum_{t=s_2+1}^{s_2+s_3}(\mathbf{m}^{(0)}_{t+r_2-1}-\mathbf{m}^{(0)}_{t+r_1-2})y_t\\
&=-\mathbf{m}^{(0)}_{r_1}(1-x_1)-\sum_{t=2}^{s_1+1}(\mathbf{m}^{(0)}_{t+r_1-1}-\mathbf{m}^{(0)}_{t+r_1-2})(1-x_t)-\\
&\quad\sum_{t=s_1+1}^{s_1+s_2-1}(\mathbf{m}^{(0)}_{t+r_1}-\mathbf{m}^{(0)}_{t+r_1-1})(1-x_t-x_{t+1})+\sum_{t=s_2+1}^{s_2+s_3}(\mathbf{m}^{(0)}_{t+r_2-1}-\mathbf{m}^{(0)}_{t+r_1-2})y_t\\
&=-\sum_{t\in S_1}(\mathbf{m}^{(0)}_t-\mathbf{m}^{(0)}_{t-1})-\sum_{t\in S_2}(\mathbf{m}^{(0)}_t-\mathbf{m}^{(0)}_{t-1})+\sum_{t\in S_3}(\mathbf{m}^{(0)}_t-\mathbf{m}^{(0)}_{t-1})\\
&=-\sum_{t\in S_1}1-\sum_{t\in S_2}1+\sum_{t\in S_3}1=0.
\end{aligned}$$

Then similar to the previous case, we obtain sets with identical cardinalities and sum of elements, and yet the smallest element in one is greater than the largest element in the others. Therefore, it follows that S_1,S_2, and S_3

are empty. Then we have $x_t = 1$ for $t \in \{1, \ldots, s_1 + 1\}$, $x_t + x_{t+1} = 1$ for $t \in \{s_1 + 1, \ldots, s_1 + s_2 - 1\}$, and $y_t = 0$ for $t \in \{s_2 + 1, \ldots, s_2 + s_3\}$, and hence (10) holds.

On the other hand, for $y_1 = 1$, let

$$\begin{aligned} S_1 &= \{t : x_{\max\{t-r_1+1,1\}} = 1, 1 \le t \le s_1 + r_1\}, \\ S_2 &= \{t : x_{t-r_1} + x_{t-r_1+1} = 0, r_2 + 1 \le t \le r_2 + s_2 - 1\}, \\ S_3 &= \{t : y_{t-s_2+1} = 1, r_2 + s_2 \le t \le r_2 + s_2 + s_3 - 1\}. \end{aligned}$$

We have

$$\begin{aligned} & g_{\mathbf{m}^{(0)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(0)}}(r_2, \mathbf{y}) \\ &= \mathbf{m}^{(0)}_{r_1} x_1 + \sum_{t=2}^{s_1+1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2}) x_t + \mathbf{m}^{(0)}_{s_1+r_1} + \sum_{t=s_1+1}^{s_1+s_2-1} (\mathbf{m}^{(0)}_{t+r_1} - \mathbf{m}^{(0)}_{t+r_1-1})(x_t + x_{t+1}) - \\ & \quad \mathbf{m}^{(0)}_{r_1+s_1+s_2-1} + \sum_{t=s_2+1}^{s_2+s_3} (\mathbf{m}^{(0)}_{t+r_2-1} - \mathbf{m}^{(0)}_{t+r_1-2}) y_t \\ &= \mathbf{m}^{(0)}_{r_1} x_1 + \sum_{t=2}^{s_1+1} (\mathbf{m}^{(0)}_{t+r_1-1} - \mathbf{m}^{(0)}_{t+r_1-2}) x_t - \\ & \quad \sum_{t=s_1+1}^{s_1+s_2-1} (\mathbf{m}^{(0)}_{t+r_1} - \mathbf{m}^{(0)}_{t+r_1-1})(1 - x_t - x_{t+1}) + \sum_{t=s_2+1}^{s_2+s_3} (\mathbf{m}^{(0)}_{t+r_2-1} - \mathbf{m}^{(0)}_{t+r_1-2}) y_t \\ &= \sum_{t \in S_1} (\mathbf{m}^{(0)}_t - \mathbf{m}^{(0)}_{t-1}) - \sum_{t \in S_2} (\mathbf{m}^{(0)}_t - \mathbf{m}^{(0)}_{t-1}) + \sum_{t \in S_3} (\mathbf{m}^{(0)}_t - \mathbf{m}^{(0)}_{t-1}) \\ &= \sum_{t \in S_1} 1 - \sum_{t \in S_2} 1 + \sum_{t \in S_3} 1 = 0. \end{aligned} \tag{31}$$

Similarly, we have

$$\begin{aligned} g_{\mathbf{m}^{(1)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(1)}}(r_2, \mathbf{y}) &= \sum_{t \in S_1} t - \sum_{t \in S_2} t + \sum_{t \in S_3} t \\ g_{\mathbf{m}^{(2)}}(r_1, \mathbf{x}) + g_{\mathbf{m}^{(2)}}(r_2, \mathbf{y}) &= \sum_{t \in S_1} t^2 - \sum_{t \in S_2} t^2 + \sum_{t \in S_3} t^2 \end{aligned} \tag{32}$$

According to (31) and (32), the following linear equation

$$A\boldsymbol{x} = \begin{bmatrix} \sum_{t \in S_1} 1 & \sum_{t \in S_2} 1 & \sum_{t \in S_3} 1 \\ \sum_{t \in S_1} t & \sum_{t \in S_2} t & \sum_{t \in S_3} t \\ \sum_{t \in S_1} t^2 & \sum_{t \in S_2} t^2 & \sum_{t \in S_3} t^2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0 \tag{33}$$

has a nonzero solution $(x_1, x_2, x_3) = (1, -1, 1)^\top$. However, according to the linearity of the determinant, sharethe determinant

$$\begin{aligned} \det(A) &= \sum_{i \in S_1, j \in S_2, k \in S_3} \det \begin{pmatrix} 1 & 1 & 1 \\ i & j & k \\ i^2 & j^2 & k^2 \end{pmatrix} \\ &= \sum_{i \in S_1, j \in S_2, k \in S_3} (j - i)(k - i)(k - j) \end{aligned} \tag{34}$$

is strictly positive since $\max_{i \in S_1} i < \min_{j \in S_2} j < \min_{k \in S_3} k$. Thus, Eq. (33) has no nonzero solution unless $A = 0$, which implies that S_1, S_2, and S_3 are empty. Therefore, $x_t = 0$ for $t \in \{1, \ldots, s_1 + 1\}$, $x_t + x_{t+1} = 1$ for $t \in \{s_1 + 1, \ldots, s_1 + s_2 - 1\}$, and $y_t = 0$ for $t \in \{s_2 + 1, \ldots, s_2 + s_3\}$, which implies (10). □

VI. Encoding and Decoding Algorithms

We now show how to use Theorem 2 to construct an encoding algorithm and a decoding algorithm. Similar to the two layer encoding method described in [5], we use the $f(\mathbf{c})$ and $h(\mathbf{c})$ redundancies (3) to protect the sequence $\mathbf{c}$ from two deletions in the first layer. In the second layer, the $f(\mathbf{c})$ and $h(\mathbf{c})$ redundancies are protected again by their corresponding $f(f(\mathbf{c}),h(\mathbf{c}))$ and $h(f(\mathbf{c}),h(\mathbf{c}))$ redundancies. Since $f(f(\mathbf{c}),h(\mathbf{c}))$ and $h(f(\mathbf{c}),h(\mathbf{c}))$ are short, they can be protected by an inefficient 3-fold repetition code. Specifically, for any sequence $\mathbf{c}\in\{0,1\}^n$, the encoding function is

$$\mathcal{E}(\mathbf{c})=(\mathbf{c},f(\mathbf{c}),h(\mathbf{c}),r_3(f(f(\mathbf{c}),h(\mathbf{c}))),r_3(h(f(\mathbf{c}),h(\mathbf{c})))), \tag{35}$$

where r_3 is a 3-fold repetition encoding function. The length of the first layer redundancy $f(\mathbf{c}),h(\mathbf{c})$ is $N_1=7\log n+2$. The length of the 3-fold repetition of the second layer redundancy $r_3(f(f(\mathbf{c}),h(\mathbf{c}))),r_3(h(f(\mathbf{c}),h(\mathbf{c})))$ is $N_2=21\log(7\log n+2)+6$. The length of the codeword $\mathcal{E}(\mathbf{c})$ is

$$N=n+N_1+N_2=n+7\log n+2+21\log(7\log n+2)+6=n+7\log n+o(\log n).$$

Clearly, the computation of the function $\mathcal{E}(\mathbf{c})$ can be done in linear time.

To conveniently describe the decoding algorithm, two building blocks are needed. The first is a 3-fold repetition decoding function

$$\mathcal{D}_1:\{0,1\}^{3N_2-2}\to\{0,1\}^{N_2}$$

that takes a subsequence $\mathbf{d}_1\in\{0,1\}^{3N_2-2}$ of a 3-fold repetition codeword $r_3(\mathbf{s}_1)\in\{0,1\}^{3N_2}$ for some $\mathbf{s}_1\in\{0,1\}^{N_2}$ as input, and outputs an estimate $\tilde{\mathbf{s}}_1$ of the sequence $\mathbf{s}_1$. The second is a decoding function

$$\mathcal{D}_2:\{0,1\}^{n-2}\times\{0,1\}^{7\log n+2}\to\{0,1\}^n$$

that takes a subsequence $\mathbf{d}_2\in\{0,1\}^{n-2}$ of some $\mathbf{s}_2\in\{0,1\}^n$, redundancy $f(\mathbf{s}_2)$, and redundancy $h(\mathbf{s}_2)$ as input, and outputs an estimate $\tilde{\mathbf{s}}_2$ of the sequence $\mathbf{s}_2$. The 3-fold repetition decoding $\mathcal{D}_1$ can be implemented by adding two bits to $\mathbf{d}_1$ such that the length of each run is a multiple of 3, which can obviously be done in linear time. According to Theorem 2, there exists a decoding function $\mathcal{D}_2$ that recovers the original sequence correctly given its f and h redundancy. The linear complexity of $\mathcal{D}_2$ will be shown later in this section.

The functions $\mathcal{D}_1$ and $\mathcal{D}_2$ are used as subroutines to describe the decoding procedure that is given in Algorithm 1. First, we use the function $\mathcal{D}_1$ to recover the second layer redundancy $f(f(\mathbf{c}),h(\mathbf{c}))$ and $h(f(\mathbf{c}),h(\mathbf{c}))$ from the 3-fold repetition code. Then, by applying $\mathcal{D}_2$ and using the second layer redundancy $f(f(\mathbf{c}),h(\mathbf{c}))$ and $h(f(\mathbf{c}),h(\mathbf{c}))$, the first layer redundancy $f(\mathbf{c})$ and $h(\mathbf{c})$ can be recovered. Finally and similarly, the first layer redundancy $f(\mathbf{c})$ and $h(\mathbf{c})$ can be used to recover the original sequence $\mathbf{c}$, with the help of $\mathcal{D}_2$. In the case of single deletion, Algorithm 1 outputs the orginal sequence $\mathbf{c}$. One can also use a VT decoder (see [1]), which has a simper implementation and $O(n)$ time complexity.

Algorithm 1: Decoding

Input: Subsequence $\mathbf{d}\in\{0,1\}^{N-2}$ of $\mathcal{E}(\mathbf{c})$
Output: The sequence $\mathbf{c}$.
layer2_redundancy $=\mathcal{D}_1(\mathbf{d}^{(N-N_2+1,N-2)})$;
if detect two deletions after the first run in $\mathbf{d}_{N-N_2+1,N-2}$ **then**
  return $\mathbf{d}^{(1,n)}$;
else
  $L\triangleq$ the length of the longest suffix of $\mathbf{d}$ that is a subsequence of r_3(*layer2_redundancy*);
  layer1_redundancy $=\mathcal{D}_2(\mathbf{d}^{(N-N_1+1-L,N-2-L)},$ *layer2_redundancy*$)$;
  $\mathbf{c}=\mathcal{D}_2(\mathbf{d}^{(1,n-2)},$ *layer1_redundancy*$)$;
  return $\mathbf{c}$.

Theorem 3. *If the functions $\mathcal{D}_1$ and $\mathcal{D}_2$ provide the correct estimates in $O(n)$ time, then given a $N-2$ subsequence of $\mathcal{E}(\mathbf{c})$, Algorithm 1 returns the original sequence $\mathbf{c}$ in $O(n)$ time.*

Proof. To prove the correctness of Algorithm 1, it suffices to show the following
(1). $\mathbf{d}^{(N-N_2+1,N-2)}$ is a length N_2-2 subsequence of the repetition code $r_3(f(f(\mathbf{c}),h(\mathbf{c}))), r_3(h(f(\mathbf{c}),h(\mathbf{c}))))$.
(2). $\mathbf{d}^{(N-N_1+1-L,N-2-L)}$ is a length N_1-2 subsequence of the $f(\mathbf{c}),h(\mathbf{c})$ redundancy.
(3). $\mathbf{d}^{(1,n-2)}$ is a length $n-2$ subsequence of the sequence $\mathbf{c}$.
Since $\mathbf{d}$ is a length $N-2$ subsequence of $\mathcal{E}(\mathbf{c})$, d_{n-2} must be either the $(n-2)$-th, the $(n-1)$-th or the n-th bits of $\mathcal{E}(\mathbf{c})$, and hence (3) must hold. Similarly, (1) holds by looking at $\mathbf{d}$ and $\mathcal{E}(\mathbf{c})$ in reverse order. By the definition of L, d_{N-2-L} is the i_1-th bit of $\mathcal{E}(\mathbf{c})$ for some $i_1 \leq n+N_1$. Since (1) holds, we have that L is either the N_2-th, the (N_2-1)-th, or the (N_2-2)-th bits of $\mathcal{E}(\mathbf{c})$. Therefore, d_{N-N_1+1-L} is the i_2-th bit of $\mathcal{E}(\mathbf{c})$ for some $i_2 \geq N-N_1+1-L > n$. Since $(f(\mathbf{c}),h(\mathbf{c})) = \mathcal{E}(\mathbf{c})^{(n+1,n+N_1)}$, (2) must hold.

Since finding L has $O(N_2)$ complexity, the complexity of Algorithm 1 is $O(N) = O(n)$, given that the complexities of the functions $\mathcal{D}_1$ and $\mathcal{D}_2$ are linear. □

We are left to implement $\mathcal{D}_2$ with linear complexity. In particular, we need to recover the sequence $\mathbf{c} \in \{0,1\}^n$ from its length $n-2$ subsequence $\mathbf{d}$ in time $O(n)$, given the redundancy $f(\mathbf{c})$ and $h(\mathbf{c})$. Note that there are $O(n^2)$ supersuquences of $\mathbf{d}$ of length n, and f and h can be computed on each of them in $O(n)$. Hence, the brute force approach would require $O(n^3)$.

To achieve linear time complexity, we first recover $\mathbb{1}_{10}(\mathbf{c})$, which is an $(n-3)$-subsequence of $\mathbb{1}_{10}(\mathbf{c}) \in \{0,1\}^{n-1}$, and then use it to recover $\mathbf{c}$. In particular, we find the positions of the deleted bits by an iterative updating algorithm, rather than by exhaustive search, and hence linear complexity is obtained. Furthermore, the uniqueness of the obtained sequence is guaranteed by Lemma 4.

After recovering $\mathbb{1}_{10}(\mathbf{c})$, We can find all length n supersequences $\mathbf{c}'$ of $\mathbf{d}$ such that $\mathbb{1}_{10}(\mathbf{c}') = \mathbb{1}_{10}(\mathbf{c})$. It is shown that there are at most 4 such possible supersequences, and since Theorem 2 guarantees uniqueness, the right $\mathbf{c}$ is found by computing and comparing h.

A. *Recovering* $\mathbb{1}_{10}(\mathbf{c})$

For $1 \leq i \leq 2n-2$, let

$$p_i \triangleq \begin{cases} n-i & \text{if } 1 \leq i \leq n-1 \\ i-n+1 & \text{if } n \leq i \leq 2n-2 \end{cases}, \text{ and} \tag{36}$$

$$b_i \triangleq \begin{cases} 1 & \text{if } 1 \leq i \leq n-1 \\ 0 & \text{if } n \leq i \leq 2n-2 \end{cases}. \tag{37}$$

Given a subsequence $\mathbf{d} \in \{0,1\}^{n-2}$ of $\mathbf{c}$, let $\mathbb{1}_{10}(\mathbf{d}) = (r_1,\ldots,r_{n-3})$, and let $\mathbf{d} : [2n-2]\times[2n-2] \to \{0,1\}^n \cup \{\star\}$ be defined as

$$\mathbf{d}(i,j) = \begin{cases} (r_1,r_2,\ldots,r_{p_i-1},b_i,r_{p_i},\ldots,r_{p_j-2},b_j,r_{p_j-1},\ldots,r_{n-3}) & \text{if } p_i < p_j \\ (r_1,r_2,\ldots,r_{p_j-1},b_j,r_{p_j},\ldots,r_{p_i-2},b_i,r_{p_i-1},\ldots,r_{n-3}) & \text{if } p_i > p_j \\ \star & \text{if } p_i = p_j \end{cases},$$

that is, $\mathbf{d}(i,j)$ results from $\mathbb{1}_{10}(\mathbf{d})$ inserting b_i at position p_i and b_j in position p_j of $\mathbb{1}_{10}(\mathbf{d})$, if $p_i \neq p_j$. Notice that $\mathbf{d}(i,j)$ is one possible way of correcting two deletions in the sequence $\mathbb{1}_{10}(\mathbf{d})$.

For $e \in \{0,1,2\}$ define matrices $\{A^{(e)}\}_{e=0}^2$ as follows.

$$A_{i,j}^{(e)} = \begin{cases} \mathbf{d}(i,j)\cdot\mathbf{m}^{(e)} - \sum_{i=1}^{n-3} \mathbf{m}_i^{(e)} \mathbb{1}_{10}(\mathbf{d})_i & \text{if } \mathbf{d}(i,j) \neq \star. \\ \star & \text{if } \mathbf{d}(i,j) = \star. \end{cases}.$$

Notice that $A_{i,j}^{(e)}$ is the difference in entry e of the f redundancies of $\mathbf{d}(i,j)$ and $\mathbb{1}_{10}(\mathbf{d})$, i.e., $A_{i,j}^{(e)} = f(\mathbf{d}(i,j))_e - f(\mathbb{1}_{10}(\mathbf{d}))_e$.

We prove the following properties of $A^{(e)}$. In the first property, we give an explicit expression for the matrices $A_{i,j}^{(e)}$ in terms of $\mathbb{1}_{10}(\mathbf{d})$, p_i, p_j, b_i, and b_j. The expression will be used for calculating $A_{i,j}^{(e)}$ in constant time from its neighboring entries during $\mathcal{D}_2$. In the following we use $\delta(x)$ to denote the indicator of the event x, where $\delta(x) = 1$ if and only if x is true.

Proposition 1. *If* $A^{(e)}_{i,j} \neq \star$ *then*

$$A^{(e)}_{i,j} = b_i \mathbf{m}^{(e)}_{p_i} + b_j \mathbf{m}^{(e)}_{p_j} + \sum_{k=1}^{n-3} \mathbb{1}_{10}(\mathbf{d})_k [(k+1)^e \delta(\min\{p_i, p_j\} < k+1) + (k+2)^e \delta(\max\{p_i, p_j\} < k+2)]. \tag{38}$$

Proof. The difference between $\sum_{k=1}^{n-3} \mathbf{m}^{(e)}_k \mathbb{1}_{10}(\mathbf{d})_k$ and $\mathbf{d}(i,j) \cdot \mathbf{m}^{(e)}$ consists of two parts. The first part follows from the two inserted bits, and can be written as

$$b_i \mathbf{m}^{(e)}_{p_i} + b_j \mathbf{m}^{(e)}_{p_j} \tag{39}$$

The second part follows from the shift of bits in $\mathbb{1}_{10}(\mathbf{d})_k$ that is caused by the insertions of two bits b_i and b_j. Each bit $\mathbb{1}_{10}(\mathbf{d})_k$ shifts from position k to position $k+1$ if one insertion occurs before $\mathbb{1}_{10}(\mathbf{d})_k$, i.e., $\min\{p_i, p_j\} < k+1$ and $\max\{p_i, p_j\} \geq k+2$. The resulting difference is given by

$$\begin{aligned} &\sum_{k=1}^{n-3} \mathbb{1}_{10}(\mathbf{d})_k \delta(\min\{p_i, p_j\} < k+1) \delta(\max\{p_i, p_j\} \geq k+2)(\mathbf{m}^{(e)}_{k+1} - \mathbf{m}^{(e)}_k) \\ =&\sum_{k=1}^{n-3} \mathbb{1}_{10}(\mathbf{d})_k \delta(\min\{p_i, p_j\} < k+1) \delta(\max\{p_i, p_j\} \geq k+2)(k+1)^e. \end{aligned} \tag{40}$$

The bit $\mathbb{1}_{10}(\mathbf{d})_k$ shifts from position k to $k+2$ if two insertions occur before $\mathbb{1}_{10}(\mathbf{d})_k$, i.e., $\max\{p_i, p_j\} < k+2$. The corresponding difference is given by

$$\begin{aligned} &\sum_{k=1}^{n-3} \mathbb{1}_{10}(\mathbf{d})_k \delta(\min\{p_i, p_j\} < k+1) \delta(\max\{p_i, p_j\} < k+2) \mathbb{1}_{10}(\mathbf{d})_k (\mathbf{m}^{(e)}_{k+2} - \mathbf{m}^{(e)}_k) \\ =&\sum_{k=1}^{n-3} \mathbb{1}_{10}(\mathbf{d})_k \delta(\max\{p_i, p_j\} < k+2)[(k+1)^e + (k+2)^e]. \end{aligned} \tag{41}$$

Combining (40) and (41), we have that the difference that results from the second part is given by

$$\sum_{k=1}^{n-3} \mathbb{1}_{10}(\mathbf{d})_k [(k+1)^e \delta(\min\{p_i, p_j\} < k+1) + (k+2)^e \delta(\max\{p_i, p_j\} < k+2)],$$

that together with (39), implies (38). □

The following shows that the entries of each $A^{(e)}$ are non-decreasing in rows and columns, and that the respective sequences $\mathbf{d}(i,j)$ that lie in the same column or the same row, are unique given each entry value. This property guarantees a simple algorithm for finding a sequence $\mathbf{d}(i,j)$ with a given value $A^{(e)}_{i,j}$ by decreasing i or increasing j by 1 in each step.

Proposition 2. *For every* i, j *and* $i_1 < i_2$, $j_1 < j_2$, *if neither of* $\mathbf{d}(i_1, j), \mathbf{d}(i_2, j), \mathbf{d}(i, j_1)$, *and* $\mathbf{d}(i, j_2)$ *equals* $\star$, *then* $A^{(e)}_{i_1,j} \leq A^{(e)}_{i_2,j}$ *and* $A^{(e)}_{i,j_1} \leq A^{(e)}_{i,j_2}$. *Moreover, if* $A^{(e)}_{i_1,j} = A^{(e)}_{i_2,j}$ *(resp.* $A^{(e)}_{i,j_1} = A^{(e)}_{i,j_2}$*), then* $\mathbf{d}(i_1, j) = \mathbf{d}(i_2, j)$ *(resp.* $\mathbf{d}(i, j_1) = \mathbf{d}(i, j_2)$*).*

Proof. By symmetry we only need to prove that the matrix $A^{(e)}$ is non-decreasing in each column, for which it suffices to prove that:

(1). $A^{(e)}_{i_1,j} \leq A^{(e)}_{i_2,j}$ for $1 \leq i_1 < i_2 \leq n-1$.
(2). $A^{(e)}_{n-1,j} \leq A^{(e)}_{n,j}$.
(3). $A^{(e)}_{i_1,j} \leq A^{(e)}_{i_2,j}$ for $n \leq i_1 < i_2 \leq 2n-2$.

For (2), the only difference between $\mathbf{d}(n-1, j)$ and $\mathbf{d}(n, j)$ is that their first bits are 0 and 1 respectively, and hence $A^{(e)}_{n-1,j} + 1 = A^{(e)}_{n,j}$. We are left to show (1) and (3).

(1): For $1 \leq i_1 < i_2 \leq n-1$, we have $b_{i_1} = b_{i_2} = 0$ and $p_{i_1} > p_{i_2}$. Let $\mathbf{d}'(i_1, j) \in \{0,1\}^{n-2}$ and $\mathbf{d}'(i_2, j) \in \{0,1\}^{n-2}$ be two subsequences of $\mathbf{d}(i_1, j)$ and $\mathbf{d}(i_2, j)$ respectively after deleting the p_j-th bit from both $\mathbf{d}(i_1, j)$

and $\mathbf{d}(i_2,j)$, and similarly, let $\mathbf{m}^{(e),p_j} = (\mathbf{m}_1^{(e)}, \mathbf{m}_2^{(e)}, \ldots, \mathbf{m}_{p_j-1}^{(e)}, \mathbf{m}_{p_j+1}^{(e)}, \ldots, \mathbf{m}_{n-1}^{(e)})$ be a subsequence of $\mathbf{m}^{(e)}$ after deleting the p_j-th entry. Then, according to (5) and (6), we have

$$\begin{aligned} A_{i_2,j}^{(e)} - A_{i_1,j}^{(e)} &= \mathbf{d}(i_2,j)\cdot\mathbf{m}^{(e)} - \mathbf{d}(i_1,j)\cdot\mathbf{m}^{(e)} \\ &= \mathbf{d}'(i_2,j)\cdot\mathbf{m}^{(e),p_j} - \mathbf{d}'(i_1,j)\cdot\mathbf{m}^{(e),p_j} \\ &= g(k_1, \mathbf{d}'(i_2,j)^{(k_1,k_2)}, \mathbf{d}'(i_1,j)_{k_2}) \\ &\geq 0, \end{aligned} \tag{42}$$

where $k_1 = p_{i_2} - \delta(p_{i_2} > p_j)$ and $k_2 = p_{i_1} - \delta(p_{i_1} > p_j)$ are the indices whose deletion from $\mathbf{d}'(i_2,j)$ and $\mathbf{d}'(i_1,j)$, respectively, results in $\mathbb{1}_{10}(\mathbf{d})$. Similarly, as in the proof in Lemma 2, the last inequality follows from the fact that $\mathbf{d}'(i_1,j)_{k_2} = b_{i_1} = 0$. Furthermore, equality holds when $\mathbf{d}'(i_2,j)^{(k_1,k_2)} = 0$ and $\mathbf{d}'(i_1,j)_{k_2} = 0$, which implies that $\mathbf{d}'(i_1,j) = \mathbf{d}'(i_2,j)$, and hence $\mathbf{d}(i_1,j) = \mathbf{d}(i_2,j)$.

(3): For $n \leq i_1 < i_2 \leq 2n-2$, we have $b_{i_1} = b_{i_2} = 1$ and $p_{i_1} < p_{i_2}$. Similar to (42), we have that

$$A_{i_1,j}^{(e)} - A_{i_2,j}^{(e)} = g(k_1, \mathbf{d}'(i_1,j)^{(k_1,k_2)}, \mathbf{d}'(i_2,j)_{k_2}) \leq 0,$$

where $k_1 = p_{i_1} - \delta(p_{i_1} > p_j)$ and $k_2 = p_{i_2} - \delta(p_{i_2} > p_j)g$ are the indices whose deletion from $\mathbf{d}'(i_1,j)$ and $\mathbf{d}'(i_2,j)$, respectively, results in $\mathbb{1}_{10}(\mathbf{d})$. The last inequality follows from the fact that $\mathbf{d}'(i_2,j)_{k_2} = b_{i_2} = 1$, and equality holds when $\mathbf{d}(i_1,j) = \mathbf{d}(i_2,j)$. □

Remark 2. *From proposition 2, we have that*

$$0 = A_{1,2}^{(e)} \leq A_{i,j}^{(e)} \leq A_{2n-2,2n-3}^{(e)} \leq \mathbf{m}_{n-1}^{(e)} + \mathbf{m}_{n-2}^{(e)} \leq n_e, \ , 1 \leq i,j \leq 2n-2, \ A_{i,j}^{(e)} \neq \star$$

where $n_0 = 2n, \ n_1 = n^2, \ n_2 = n^3$.

Our goal is to find a sequence $\mathbf{d}(i,j) \neq \star$ for which

$$A_{i,j}^{(e)} \equiv f_1(\mathbf{c}) - \sum_{i=1}^{n-3} \mathbf{m}_i^{(e)} \mathbb{1}_{10}(\mathbf{d})_i \bmod n_e \tag{43}$$

for every $e \in \{0,1,2\}$. In addition, the sequence $\mathbf{d}(i,j)$ cannot contain adjacent 1's, i.e.,

$$\begin{aligned} \mathbf{d}(i,j)_{p_i-1}\cdot\mathbf{d}(i,j)_{p_i} &= \mathbf{d}(i,j)_{p_i}\cdot\mathbf{d}(i,j)_{p_i+1} = 0 \\ \mathbf{d}(i,j)_{p_j-1}\cdot\mathbf{d}(i,j)_{p_j} &= \mathbf{d}(i,j)_{p_j}\cdot\mathbf{d}(i,j)_{p_j+1} = 0, \end{aligned} \tag{44}$$

and from Lemma 4, such $\mathbf{d}(i,j)$ equals $\mathbb{1}_{10}(\mathbf{c})$. Moreover, since Remark 2 implies that $0 \leq A_{i,j}^{(e)} \leq n_e$, it follows that the modular equality in (43) is unnecessary, i.e., it suffices to find a sequence $\mathbf{d}(i,j) \neq \star$ that satisfies (44) and

$$A_{i,j}^{(e)} = a_e \triangleq f_e(\mathbf{c}) - \sum_{k=1}^{n-3} \mathbf{m}_k^{(e)} \mathbb{1}_{10}(\mathbf{d})_k \bmod n_e, \tag{45}$$

where a_e is the target value to be found in matrix $A^{(e)}$. Eq. (45) implies that $\mathbf{d}(i,j)$ satisfies the f redundancy.

The procedure to find such $\mathbf{d}(i,j)$ is given in Algorithm 2. We search for all sequences $\mathbf{d}(i,j) \neq \star$ with no adjacent 1's (satisfies (44)) such that $A_{i,j}^{(0)} = a_0$. This clearly amounts to a binary search in a sorted matrix[3]. We start from the bottom left corner of the matrix, proceed to the right in each step until reaching the rightmost entry such that $A_{i,j}^{(0)} \leq a_0$, and then go one step up. Figure 4 illustrates an example of how Algorithm 2 runs on matrix $A^{(0)}$.

To avoid the computation of the entire matrix, that would require $O(n^2)$ time, each entry is computed from previously seen ones *only* upon its discovery. To this end we prove the following lemma, that alongside Proposition 1, provides a way of computing a newly discovered entry.

[3]The two $\star$ entries in each row or column can simply be ignored.

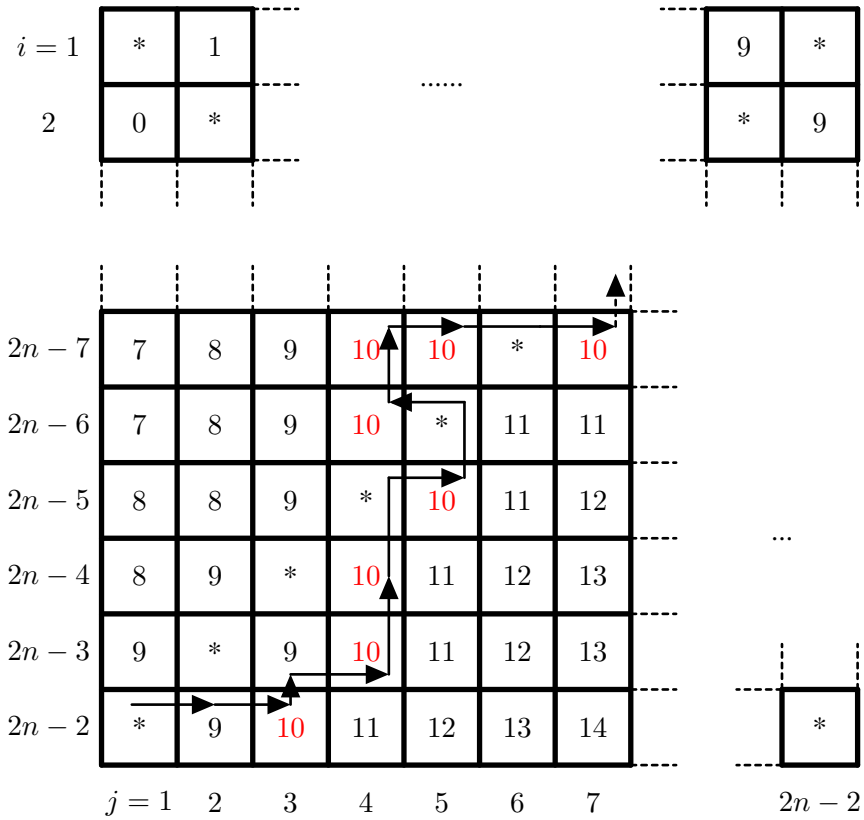

Fig. 4. The path of Algorithm 2 on the matrix $A^{(0)}$. The algorithm searches for all i, j pairs such that $A^{(0)}_{i,j} = 10$ that appear in the lowest position (with maximum i) of each column. The algorithm proceeds right until the next term $A^{(0)}_{i,j}$ is greater than 10. Then, it proceeds up one step and repeats the process in the same manner.

Lemma 11. *Whenever the (i,j)-th and $(i+1,j)$-th (resp. $(i,j+1)$) entries of $A^{(e)}$ are not $\star$, we have that*

$$\begin{aligned} A^{(e)}_{i,j} - A^{(e)}_{i+1,j} &= b_i \mathbf{m}^{(e)}_{p_i} - b_{i+1}\mathbf{m}^{(e)}_{p_{i+1}} \\ &+ \sum_{k=\min\{p_i,p_{i+1}\}-1}^{\min\{p_1,p_{i+1}\}} \mathbb{1}_{10}(\mathbf{d})_k [(k+1)^e(\delta(\min\{p_i,p_j\} < k+1) - \delta(\min\{p_{i+1},p_j\} < k+1)) \\ &+ (k+2)^e(\delta(\max\{p_i,p_j\} < k+2) - \delta(\max\{p_{i+1},p_j\} < k+2))], \text{ and} \end{aligned} \tag{46}$$

$$\begin{aligned} A^{(e)}_{i,j} - A^{(e)}_{i,j+1} &= b_j \mathbf{m}^{(e)}_{p_j} - b_{j+1}\mathbf{m}^{(e)}_{p_{j+1}} \\ &+ \sum_{k=\min\{p_j,p_{j+1}\}-1}^{\min\{p_j,p_{j+1}\}} \mathbb{1}_{10}(\mathbf{d})_k [(k+1)^e(\delta(\min\{p_i,p_j\} < k+1) - \delta(\min\{p_i,p_{j+1}\} < k+1)) \\ &+ (k+2)^e(\delta(\max\{p_i,p_j\} < k+2) - \delta(\max\{p_i,p_{j+1}\} < k+2))] \end{aligned} \tag{47}$$

Proof. Note that if i increases by 1 or if j decreases by 1, then p_i or p_j changes by at most 1 (See (36)). Hence,

$$\begin{aligned} \delta(\min\{p_i,p_j\} < k+1) &= \delta(\min\{p_{i+1},p_j\} < k+1), \\ \delta(\max\{p_i,p_j\} < k+2) &= \delta(\max\{p_{i+1},p_j\} < k+2) \end{aligned}$$

for $k \le \min\{p_j, p_{i+1}\} - 2$ and $k \ge \min\{p_i, p_{i+1}\} + 1$. According to (38), we have that (46) holds, and similarly, (47) holds as well. □

We first show that Algorithm 2 outputs the (i,j) pair such that $\mathbf{d}(i,j) = \mathbb{1}_{10}(\mathbf{c})$. Note that by Lemma 4 there exists a unique sequence $\mathbf{d}(i,j) = \mathbb{1}_{10}(\mathbf{c})$ for which $\mathbf{d}(i,j)$ satisfies Eq. (44) and for which (i,j) satisfies Eq.

Algorithm 2: Finding $\mathbb{1}_{10}(\mathbf{c})$.

Input: Subsequence $\mathbf{d} \in \{0,1\}^{n-2}$ of $\mathbf{c}$, and $f(\mathbf{c})$
Output: i and j such that $\mathbf{d}(i,j) = \mathbb{1}_{10}(c)$
Initialization: $i = 2n-2, j = 1$;
$x_e = A^{(e)}_{1,2n-2}$ for $e \in \{0,1,2\}$;
$a_e = f_e(\mathbf{c}) - \sum_{k=1}^{n-3} \mathbf{m}^{(e)}_k \mathbb{1}_{10}(\mathbf{d})_k \bmod n_e$ for $e \in \{0,1,2\}$;
while $i \geq 0$ **do**
  if $x_e = a_e$ *for every* $e \in \{0,1,2\}$ *and* $\mathbf{d}(i,j) \neq \star$ *and has no adjacent* 1*s' (satisfies* (44)*)* **then**
    return i, j;
  else
    Find the maximum j for which $A^{(0)}_{i,j} \leq a_0$.
    if $p_i = p_j$ *or* $(x_0 > a_0)$ **then**
      $temp\_x_e = x_e + A^{(e)}_{i,j-1} - A^{(e)}_{i,j}$ (using (47)), for $e \in \{0,1,2\}$;
      $temp\_j = j - 1$;
      while $p_{temp\_j} = p_i$ **do**
        $temp\_x_e = x_e + A^{(e)}_{i,temp\_j-1} - A^{(e)}_{i,temp\_j}$ (using (47)) for $e \in \{0,1,2\}$;
        $temp\_j = temp\_j - 1$;
      if $temp\_j \geq 1$ **then**
        $j = temp\_j$;
        $x_e = temp\_x_e$ for $e \in \{0,1,2,\}$;
    else
      $temp\_x_e = x_e + A^{(e)}_{i,j+1} - A^{(e)}_{i,j}$ (using (47)), for $e \in \{0,1,2\}$;
      $temp\_j = j + 1$;
      while $p_{temp\_j} = p_i$ **do**
        $temp\_x_e = x_e + A^{(e)}_{i,temp\_j+1} - A^{(e)}_{i,temp\_j}$ (using (47)) for $e \in \{0,1,2\}$;
        $temp\_j = temp\_j + 1$;
      if $temp\_x_0 \leq a_0$ **then**
        $j = temp\_j$;
        $x_e = temp\_x_e$ for $e \in \{0,1,2,\}$;
      else
        $x_e = x_e + A^{(e)}_{i-1,j} - A^{(e)}_{i,j}$ (using (46));
        $i = i - 1$;
return $(0,0)$;

(45). Since the algorithm terminates either when such a sequence $\mathbf{d}(i,j) = \mathbb{1}_{10}(\mathbf{c})$ is found or no such sequence is found and i reaches 0, it suffices to show that the latter case does not occur. We prove this by contradiction. Assuming that the latter case occurs, we show that $\mathbf{d}(i,j) \neq \mathbb{1}_{10}(\mathbf{c})$ for all (i,j) pairs, which is a contradiction. For each $i \in \{1,2,\ldots,2n-2\}$, let j_i be the maximum $j = j_i$ for which $A^{(0)}_{i,j_i} \leq a_0$. If $A^{(0)}_{i,j} > a_0$ for all j, then $j_i = 1$. Note that each pair (i,j_i) is visited in Algorithm 2 and by assumption we have that $\mathbf{d}(i,j_i) \neq \mathbb{1}_{10}(\mathbf{c})$. We consider the following two cases

(1). $j > j_i$
(2). $j < j_i$

and conclude that no (i,j) pairs in these cases result in $\mathbf{d}(i,j) = \mathbb{1}_{10}(\mathbf{c})$. For $j > j_i$, by Proposition 2 we have that $A^{(0)}_{i,j} \geq A^{(0)}_{i,j_i}$ or that $\mathbf{d}(i,j) = \star$. Hence by definition of j_i we have that $A^{(0)}_{i,j} > a_0$ or that $\mathbf{d}(i,j) = \star$ and hence $\mathbf{d}(i,j) \neq \mathbb{1}_{10}(\mathbf{c})$. For $j < j_i$, by Proposition 2 we have that $A^{(0)}_{i,j} \leq A^{(0)}_{i,j_i}$ or that $\mathbf{d}(i,j) = \star$. If $A^{(0)}_{i,j} < A^{(0)}_{i,j_i}$,

then $A^{(0)}_{i,j} \neq a_0$. If $A^{(0)}_{i,j} = A^{(0)}_{i,j_i}$, then according to Proposition 2, we have that $\mathbf{d}(i,j) = \mathbf{d}(i,j_i) \neq \mathbb{1}_{10}(\mathbf{c})$.

We now show that Algorithm 2 terminates in $O(n)$ time. From (46) and (47) the (i,j)-th entry of $A^{(e)}, e \in \{0,1,2\}$, can be computed by using the update rule $x_e + A^{(e)}_{i-1,j} - A^{(e)}_{i,j}$ and $x_e + A^{(e)}_{i,j\pm1} - A^{(e)}_{i,j}$ (see Algorithm 2), that can be computed in constant time. In addition, one can verify in constant time that (44) holds.

Note that in each round, either i decreases by 1 or j increases by 1, with the exception that j decreases when $A^{(0)}_{i,j} = \star$ or $A^{(0)}_{i,j} > a_0$. We prove by contradiction that the latter case, in which $A^{(0)}_{i,j} > a_0$ and $j > 1$ is impossible. Notice that for each current pair (i,j), the value of next pair (i^*, j^*) falls into either one of the following three case:

(1). $(i^*, j^*) = (i, j')$ for some $j' > j$ with $A^{(0)}_{i^*,j^*} \leq a_0$
(2). $(i^*, j^*) = (i-1, j)$
(3). $(i^*, j^*) = (i-1, j')$ for some $j' < j$ when $A^{(0)}_{i-1,j} = \star$.

Assume by contradiction that $A^{(0)}_{i^*,j^*} > a_0$ and $j^* > 1$, and (i^*, j^*) is the first pair for which this statement is true. In Case (1), we have that $A^{(0)}_{i^*,j^*} \leq a_0$, in contradiction to $A^{(0)}_{i^*,j^*} > a_0$. In Case (2) or Case (3), Proposition 2 implies that $a_0 < A^{(0)}_{i^*,j^*} \leq A^{(0)}_{i,j}$, contradicting the assumption that (i^*, j^*) is the first visited pair which satisfies $A^{(0)}_{i^*,j^*} > a_0$.

Having proved that $A^{(0)}_{i,j} \leq a_0$ whenever $j > 1$, we have the Algorithm 2 proceeds to left only when it encounters a $\star$-entry. We now show that the algorithm terminates in $O(n)$ time. Notice that unless Algorithm 2 encounters a $\star$-entry, it proceeds either up or to the right, for which case, it is clear that only $O(n)$ many steps occur. In cases where Algorithm 2 encounters a $\star$-entry, it proceeds to the *left* until a non $\star$-entry is found. Since the number of $\star$-entries is $4n-4$, the number of left strides of the algorithm is at most this quantity, and therefore the algorithm terminates in at most $O(n)$ time. In the following, we provide a running example of Algorithm 2.

Example 1. *Consider a sequence* $\mathbf{c} = (1,1,0,0,1,0,1,0)$*, where the first and the* 6*-th bits are deleted, resulting in* $\mathbf{d} = (1,0,0,1,1,0)$*. Then* $n = 8$*,* $\mathbb{1}_{10}(\mathbf{c}) = (0,1,0,0,1,0,1)$*,* $f(\mathbf{c}) = (14, 46, 200)$*, and* $\mathbb{1}_{10}(\mathbf{d}) = (1,0,0,0,1)$*. Hence* $a_0 = 8, a_1 = 30, a_2 = 144$*.*

Then, Algorithm 2 proceeds in the following manner.

$$
\begin{aligned}
&i = 1, j = 14, p_i = p_j, x_0 = 7, x_1 = 28, x_2 = 140\\
\rightarrow &i = 2, j = 14, \mathbf{d}(i,j) = (1,0,0,0,1,\underline{0},\underline{1}), x_0 = 7, x_1 = 28, x_2 = 140\\
\rightarrow &i = 3, j = 14, \mathbf{d}(i,j) = (1,0,0,0,\underline{0},1,\underline{1}), x_0 = 8, x_1 = 34, x_2 = 176,\\
\rightarrow &i = 4, j = 14, \mathbf{d}(i,j) = (1,0,0,\underline{0},0,1,\underline{1}), x_0 = 8, x_1 = 34, x_2 = 176,\\
\rightarrow &i = 5, j = 14, \mathbf{d}(i,j) = (1,0,\underline{0},0,0,1,\underline{1}), x_0 = 8, x_1 = 34, x_2 = 176,\\
\rightarrow &i = 6, j = 14, \mathbf{d}(i,j) = (1,\underline{0},0,0,0,1,\underline{1}), x_0 = 8, x_1 = 34, x_2 = 176,\\
\rightarrow &i = 7, j = 14, \mathbf{d}(i,j) = (\underline{0},1,0,0,0,1,\underline{1}), x_0 = 9, x_1 = 36, x_2 = 180\\
\rightarrow &i = 7, j = 13, \mathbf{d}(i,j) = (\underline{0},1,0,0,0,\underline{1},1), x_0 = 9, x_1 = 36, x_2 = 180\\
\rightarrow &i = 7, j = 12, \mathbf{d}(i,j) = (\underline{0},1,0,0,\underline{1},0,1), x_0 = 8, x_1 = 30, x_2 = 144
\end{aligned}
$$

B. Recover the original sequence $\mathbf{c}$

Let (i,j) be the output of Algorithm 2, for which we have that $\mathbf{d}(i,j) = \mathbb{1}_{10}(\mathbf{c})$. Let $\mathbf{c}'$ be a length n supersequence after two insertions to $\mathbf{d}$ such that $\mathbb{1}_{10}(\mathbf{c}') = \mathbb{1}_{10}(\mathbf{c})$. If $b_i = 1$, then inserting b_i to $\mathbb{1}_{10}(\mathbf{d})$ corresponds to either inserting a 0 to $\mathbf{d}$ as the $p_i + 1$-th bit in $\mathbf{c}'$ or inserting a 1 to $\mathbf{d}$ as the p_i-th bit in $\mathbf{c}'$ (see Table I). If $b_i = 0$, then inserting b_i to $\mathbb{1}_{10}(\mathbf{d})$ corresponds to inserting a 0 or 1 in the first 0 run or 1 run respectively after the k'-th bit in $\mathbf{c}'$, where $k' = \max_k\{\mathbf{d}(i,j)_k = 1, k < p_i\}$. The same arguments hold for the insertion of b_j.

Therefore, given the (i,j) pair that Algorithm 2 returns, there are at most four possible $\mathbf{c}'$ supersequences of $\mathbf{d}$ such that $\mathbb{1}_{10}(\mathbf{c}') = \mathbb{1}_{10}(\mathbf{c})$. One can check if the $\mathbf{c}'$ sequences satisfy $h(\mathbf{c})$. According to Theorem 2, there is a unique such sequence, the original sequence $\mathbf{c}$ that satisfies both $f(\mathbf{c})$ and $h(\mathbf{c})$ simultaneously.

REFERENCES

[1] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[2] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors," in *Autom. Remote Control*, vol. 26, no. 2, 1965, pp. 286–290.

[3] A. S. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes," *IEEE Trans. on Inf. Th.*, vol. 48, no. 1, pp. 305–308, 2002.

[4] F. Paluncic, K. A. Abdel-Ghaffar, H. C. Ferreira, and W. A. Clarke, "A multiple insertion/deletion correcting code for run-length limited sequences," *IEEE Trans. on Inf. Th.*, vol. 58, no. 3, pp. 1809–1824, 2012.

[5] J. Brakensiek, V. Guruswami, and S. Zbarsky, "Efficient low-redundancy codes for correcting multiple deletions," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms* (SODA), pp. 1884–1892. 2016

[6] R. Gabrys and F. Sala, "Codes correcting two deletions." *arXiv:1712.07222* [cs.IT], 2017.

APPENDIX

Proof of (15) (Case (a)):

$$
\begin{aligned}
&(\mathbb{1}_{10}(\mathbf{c}) - \mathbb{1}_{10}(\mathbf{c}')) \cdot \mathbf{m}^{(e)} \\
&= \sum_{t=\ell_1}^{\ell_2} \left(\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c}')_t\right) \cdot (\mathbf{m}^{(e)})_t + \sum_{t=k_2}^{k_1} \left(\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c}')_t\right) \cdot (\mathbf{m}^{(e)})_t \\
&= (\mathbb{1}_{10}(\mathbf{c})_{\ell_2} - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_1} - \mathbb{1}_{10}(\mathbf{c}')_{k_1}) \cdot (\mathbf{m}^{(e)})_{k_1} + \\
&\sum_{t=\ell_1}^{\ell_2-1} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c})_{t+1}) \cdot (\mathbf{m}^{(e)})_t + \sum_{t=k_2}^{k_1-1} \left(\mathbb{1}_{10}(\mathbf{c}')_{t+1} - \mathbb{1}_{10}(\mathbf{c}')_t\right) \cdot (\mathbf{m}^{(e)})_t \\
&= (\mathbb{1}_{10}(\mathbf{c})_{\ell_2} - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_1} - \mathbb{1}_{10}(\mathbf{c}')_{k_1}) \cdot (\mathbf{m}^{(e)})_{k_1} + \\
&\sum_{t=\ell_1}^{\ell_2-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_t - \sum_{\ell_1+1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_{t-1} \\
&\quad + \sum_{t=k_2+1}^{k_1} \mathbb{1}_{10}(\mathbf{c}')_t \cdot (\mathbf{m}^{(e)})_{t-1} - \sum_{t=k_2}^{k_1-1} \mathbb{1}_{10}(\mathbf{c}')_t \cdot (\mathbf{m}^{(e)})_t \\
&= (\mathbb{1}_{10}(\mathbf{c})_{\ell_2} - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_1} - \mathbb{1}_{10}(\mathbf{c}')_{k_1}) \cdot (\mathbf{m}^{(e)})_{k_1} + \\
&\mathbb{1}_{10}(\mathbf{c})_{\ell_1} \cdot (\mathbf{m}^{(e)})_{\ell_1} - \mathbb{1}_{10}(\mathbf{c})_{\ell_2} \cdot (\mathbf{m}^{(e)})_{\ell_2-1} + \sum_{t=\ell_1+1}^{\ell_2-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e + \\
&\mathbb{1}_{10}(\mathbf{c}')_{k_1} \cdot (\mathbf{m}^{(e)})_{k_1-1} - \mathbb{1}_{10}(\mathbf{c}')_{k_2} \cdot (\mathbf{m}^{(e)})_{k_2} - \sum_{t=k_2+1}^{k_1-1} \mathbb{1}_{10}(\mathbf{c}')_t \cdot t^e \\
&= (-\mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_1}) \cdot (\mathbf{m}^{(e)})_{k_1} + \\
&\mathbb{1}_{10}(\mathbf{c})_{\ell_1} \cdot (\mathbf{m}^{(e)})_{\ell_1} + \sum_{t=\ell_1+1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e - \mathbb{1}_{10}(\mathbf{c}')_{k_2} \cdot (\mathbf{m}^{(e)})_{k_2} - \sum_{t=k_2+1}^{k_1} \mathbb{1}_{10}(\mathbf{c}')_t \cdot t^e \\
&= \mathbb{1}_{10}(\mathbf{c})_{\ell_1} \cdot (\mathbf{m}^{(e)})_{\ell_1} + \mathbb{1}_{10}(\mathbf{c})_{k_1} \cdot (\mathbf{m}^{(e)})_{k_1} + \sum_{t=\ell_1+1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e \\
&\quad - \sum_{t=k_2+1}^{k_1} \mathbb{1}_{10}(\mathbf{c}')_t \cdot t^e - \left(\mathbb{1}_{10}(\mathbf{c}')_{\ell_2} \cdot (\mathbf{m}^{(e)})_{\ell_2} + \mathbb{1}_{10}(\mathbf{c}')_{k_2} \cdot (\mathbf{m}^{(e)})_{k_2}\right) \\
&= g_{\mathbf{m}^{(e)},\ell_1}(\mathbb{1}_{10}(\mathbf{c})_{\ell_1}, \ldots, \mathbb{1}_{10}(\mathbf{c})_{\ell_2}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) - g_{\mathbf{m}^{(e)},k_2}(\mathbb{1}_{10}(\mathbf{c}')_{k_2}, \ldots, \mathbb{1}_{10}(\mathbf{c}')_{k_1}, \mathbb{1}_{10}(\mathbf{c})_{k_1})
\end{aligned}
$$

Proof of (16) (Case (b)):

$$
\begin{aligned}
&(\mathbb{1}_{10}(\mathbf{c}) - \mathbb{1}_{10}(\mathbf{c}')) \cdot \mathbf{m}^{(e)} \\
&= \sum_{t=\ell_1}^{\ell_2} \left(\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c}')_t\right) \cdot (\mathbf{m}^{(e)})_t + \sum_{t=k_1}^{k_2} \left(\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c}')_t\right) \cdot (\mathbf{m}^{(e)})_t \\
&= (\mathbb{1}_{10}(\mathbf{c})_{\ell_2} - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_2} - \mathbb{1}_{10}(\mathbf{c}')_{k_2}) \cdot (\mathbf{m}^{(e)})_{k_2} + \\
&\sum_{t=\ell_1}^{\ell_2-1} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c})_{t+1}) \cdot (\mathbf{m}^{(e)})_t + \sum_{t=k_1}^{k_2-1} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c})_{t+1}) \cdot (\mathbf{m}^{(e)})_t \\
&= (\mathbb{1}_{10}(\mathbf{c})_{\ell_2} - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_2} - \mathbb{1}_{10}(\mathbf{c}')_{k_2}) \cdot (\mathbf{m}^{(e)})_{k_2} + \\
&\sum_{t=\ell_1}^{\ell_2-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_t - \sum_{\ell_1+1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_{t-1} + \sum_{t=k_1}^{k_2-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_t \\
&- \sum_{t=k_1+1}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_{t-1} \\
&= (\mathbb{1}_{10}(\mathbf{c})_{\ell_2} - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_2} - \mathbb{1}_{10}(\mathbf{c}')_{k_2}) \cdot (\mathbf{m}^{(e)})_{k_2} + \\
&\mathbb{1}_{10}(\mathbf{c})_{\ell_1} \cdot (\mathbf{m}^{(e)})_{\ell_1} - \mathbb{1}_{10}(\mathbf{c})_{\ell_2} \cdot (\mathbf{m}^{(e)})_{\ell_2-1} + \sum_{t=\ell_1+1}^{\ell_2-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e + \\
&\mathbb{1}_{10}(\mathbf{c})_{k_1} \cdot (\mathbf{m}^{(e)})_{k_1} - \mathbb{1}_{10}(\mathbf{c})_{k_2} \cdot (\mathbf{m}^{(e)})_{k_2-1} + \sum_{t=k_1+1}^{k_2-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e \\
&= (-\mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (-\mathbb{1}_{10}(\mathbf{c}')_{k_2}) \cdot (\mathbf{m}^{(e)})_{k_2} + \\
&\mathbb{1}_{10}(\mathbf{c})_{\ell_1} \cdot (\mathbf{m}^{(e)})_{\ell_1} + \sum_{t=\ell_1+1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e + \mathbb{1}_{10}(\mathbf{c})_{k_1} \cdot (\mathbf{m}^{(e)})_{k_1} + \sum_{t=k_1+1}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e \\
&= \mathbb{1}_{10}(\mathbf{c})_{\ell_1} \cdot (\mathbf{m}^{(e)})_{\ell_1} + \mathbb{1}_{10}(\mathbf{c})_{k_1} \cdot (\mathbf{m}^{(e)})_{k_1} - \left(\mathbb{1}_{10}(\mathbf{c}')_{\ell_2} \cdot (\mathbf{m}^{(e)})_{\ell_2} + \mathbb{1}_{10}(\mathbf{c}')_{k_2} \cdot (\mathbf{m}^{(e)})_{k_2}\right) + \\
&\sum_{t=\ell_1+1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e + \sum_{t=k_1+1}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e \\
&= g_{\mathbf{m}^{(e)},\ell_1}(\mathbb{1}_{10}(\mathbf{c})_{\ell_1}, \ldots, \mathbb{1}_{10}(\mathbf{c})_{\ell_2}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) + g_{\mathbf{m}^{(e)},k_1}(\mathbb{1}_{10}(\mathbf{c})_{k_1}, \ldots, \mathbb{1}_{10}(\mathbf{c})_{k_2}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})
\end{aligned}
$$

Proof of (17) (Case (c)):

$$
\begin{aligned}
&(\mathbb{1}_{10}(\mathbf{c}) - \mathbb{1}_{10}(\mathbf{c}')) \cdot \mathbf{m}^{(e)} \\
&= \sum_{t=\ell_1}^{k_1-2} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c}')_t) \cdot (\mathbf{m}^{(e)})_t + \sum_{t=k_1-1}^{\ell_2-1} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c}')_t) \cdot (\mathbf{m}^{(e)})_t \\
&\quad + \sum_{t=\ell_2}^{k_2} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c}')_t) \cdot (\mathbf{m}^{(e)})_t \\
&= \sum_{t=\ell_1}^{k_1-2} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c})_{t+1}) \cdot (\mathbf{m}^{(e)})_t + \sum_{t=k_1-1}^{\ell_2-1} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c})_{t+2}) \cdot (\mathbf{m}^{(e)})_t + \\
&\quad (\mathbb{1}_{10}(\mathbf{c})_{\ell_2} - \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (\mathbb{1}_{10}(\mathbf{c})_{k_2} - \mathbb{1}_{10}(\mathbf{c}')_{k_2}) \cdot (\mathbf{m}^{(e)})_{k_2} + \\
&\quad \sum_{t=\ell_2+1}^{k_2-1} (\mathbb{1}_{10}(\mathbf{c})_t - \mathbb{1}_{10}(\mathbf{c})_{t+1}) \cdot (\mathbf{m}^{(e)})_t \\
&= \sum_{t=\ell_1}^{k_1-2} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_t - \sum_{t=\ell_1+1}^{k_1-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_{t-1} + \sum_{t=k_1-1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_t \\
&\quad - \sum_{t=k_1+1}^{\ell_2+1} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_{t-2} + (-\mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (-\mathbb{1}_{10}(\mathbf{c}')_{k_2}) \cdot (\mathbf{m}^{(e)})_{k_2} + \\
&\quad \sum_{t=\ell_2+1}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_t - \sum_{t=\ell_2+2}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t \cdot (\mathbf{m}^{(e)})_{t-1} \\
&= \mathbb{1}_{10}(\mathbf{c})_{\ell_1} (\mathbf{m}^{(e)})_{\ell_1} - \mathbb{1}_{10}(\mathbf{c})_{k_1-1} (\mathbf{m}^{(e)})_{k_1-2} + \sum_{t=\ell_1+1}^{k_1-2} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e + \\
&\quad \mathbb{1}_{10}(\mathbf{c})_{k_1-1} (\mathbf{m}^{(e)})_{k_1-1} + \mathbb{1}_{10}(\mathbf{c})_{k_1} (\mathbf{m}^{(e)})_{k_1} - \mathbb{1}_{10}(\mathbf{c})_{\ell_2+1} (\mathbf{m}^{(e)})_{\ell_2-1} \\
&\quad + \sum_{t=k_1+1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_t (t^e + (t-1)^e) + (-\mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) \cdot (\mathbf{m}^{(e)})_{\ell_2} + (-\mathbb{1}_{10}(\mathbf{c}')_{k_2}) \cdot (\mathbf{m}^{(e)})_{k_2} + \\
&\quad \mathbb{1}_{10}(\mathbf{c})_{\ell_2+1} (\mathbf{m}^{(e)})_{\ell_2+1} + \sum_{t=\ell_2+2}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t t^e \\
&= \mathbb{1}_{10}(\mathbf{c})_{\ell_1} (\mathbf{m}^{(e)})_{\ell_1} + \mathbb{1}_{10}(\mathbf{c})_{k_1} (\mathbf{m}^{(e)})_{k_1} - (\mathbb{1}_{10}(\mathbf{c}')_{\ell_2} \cdot (\mathbf{m}^{(e)})_{\ell_2} + \mathbb{1}_{10}(\mathbf{c}')_{k_2} \cdot (\mathbf{m}^{(e)})_{k_2}) + \\
&\quad \sum_{t=\ell_1+1}^{k_1-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e + \sum_{t=k_1+1}^{\ell_2+1} \mathbb{1}_{10}(\mathbf{c})_t (t^e + (t-1)^e) + \sum_{t=\ell_2+2}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t t^e \\
&= \mathbb{1}_{10}(\mathbf{c})_{\ell_1} (\mathbf{m}^{(e)})_{\ell_1} + \mathbb{1}_{10}(\mathbf{c})_{k_1} (\mathbf{m}^{(e)})_{k_1} - (\mathbb{1}_{10}(\mathbf{c}')_{\ell_2} \cdot (\mathbf{m}^{(e)})_{\ell_2} + \mathbb{1}_{10}(\mathbf{c}')_{k_2} \cdot (\mathbf{m}^{(e)})_{k_2}) + \\
&\quad \sum_{t=\ell_1+1}^{k_1-1} \mathbb{1}_{10}(\mathbf{c})_t \cdot t^e + \sum_{t=k_1}^{\ell_2} \mathbb{1}_{10}(\mathbf{c})_{t+1} t^e + \sum_{t=k_1+1}^{k_2} \mathbb{1}_{10}(\mathbf{c})_t t^e \\
&= g_{\mathbf{m}^{(e)},\ell_1}(\mathbb{1}_{10}(\mathbf{c})_{\ell_1}, \ldots, \mathbb{1}_{10}(\mathbf{c})_{k_1-1}, \mathbb{1}_{10}(\mathbf{c})_{k_1+1}, \ldots, \mathbb{1}_{10}(\mathbf{c})_{\ell_2+1}, \mathbb{1}_{10}(\mathbf{c}')_{\ell_2}) + \\
&\quad g_{\mathbf{m}^{(e)},k_1}(\mathbb{1}_{10}(\mathbf{c})_{k_1}, \ldots, \mathbb{1}_{10}(\mathbf{c})_{k_2}, \mathbb{1}_{10}(\mathbf{c}')_{k_2})
\end{aligned}
$$