The Zero Cubes Free and Cubes Unique Multidimensional Constraints

Sagi Marcovich, *Student Member, IEEE*, and Eitan Yaakobi, *Senior Member, IEEE*

***Abstract*— This paper studies two families of constraints for two-dimensional and multidimensional arrays. The first family requires that a multidimensional array will not contain a cube of zeros of some fixed size and the second constraint imposes that there will not be two identical cubes of a given size in the array. These constraints are natural extensions of their one-dimensional counterpart that have been rigorously studied recently. For both of these constraints we present conditions on the size of the cube for which the asymptotic rate of the set of valid arrays approaches 1 as well as conditions for the redundancy to be at most a single symbol. For the first family we present an efficient encoding algorithm that uses a single redundant symbol to encode arbitrary information into a valid array and for the second family we present a similar encoder for the two-dimensional case. The results in the paper are also extended to similar constraints where the sub-array is not necessarily a cube, but a box of arbitrary dimensions and only its volume is bounded.**

***Index Terms*— Constrained codes, multidimensional codes, repeat-free codes, de-Bruijn sequences, zero cubes free, cubes unique, minimal boxes.**

I. Introduction

Coding for two-dimensional and multidimensional arrays is a topic which attracted significant attention in the last three decades due to its various applications in different areas. This includes optical storage such as page-oriented optical memories [13], [21] and holographic storage [12]. Other applications in robotics are robot localization [27], camera localization [30], projected touchscreens [6], just to name a few, and there are several more in structured light; see e.g. [15], [20], [25], [26]. Examples of coding schemes for these applications include error-correction codes [10], constrained codes [23], [24], [29], [31], [32], pseudo random arrays and perfect maps [9], [18], codes for self locating patterns [2], and more.

This paper takes one more step in advancing the theory of coding for multidimensional arrays, and in particular two-dimensional arrays, and studies two special constraint families for multidimensional arrays of any dimension d, over an alphabet of any size q. In the first constraint, it is said that a d-dimensional array is *zero-L-cube free* if it does not contain any d-dimensional L-cube of zeros, where a d-dimensional L-cube is a d-dimensional array that all its sides are of length L (when $d = 2$ we say L-squares). This constraint generalizes the well known one-dimensional *run length limited (RLL)* constraint, which has numerous applications in various areas of information theory [28]. In the second constraint, it is said that a d-dimensional array is *L-cubes unique* if it does not contain any two identical d-dimensional L-cubes. This constraint generalizes the one-dimensional *repeat free* constraint [7], [11], which is an extension to well known family of *de Bruijn sequences* [3]. Hence, similarly, the L-cubes unique constraint extends the very strict family of de Bruijn arrays [18] which exist only for a very specific range of parameters and have several applications on their own such as robotic vision, location detection, pseudo-random arrays and more [14], [16]. Examples of applications of L-cubes unique arrays include *self-location patterns* [2] and *position sensing schemes* [4], which allow for knowing the absoulte positioning of any local sampling of the array for some size. In our case, since any L-cube contained in the array appears only once, each L-cube provides sufficient local information such that its absolute positioning in the array can be attained. See examples of both constraints in Figures 1 and 2.

$$Y = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Fig. 1. 5-square Y contains two zero-2-squares (marked in red). However, Y contains no zero-3-squares and thus Y satisfies the zero-3-square free constraint.

Only little is known on these families of codes and the goal of this paper is to rigorously study them for all values of L and d and in particular for $d = 2$, as well as to construct efficient encoding and decoding algorithms for these constraints. Our research is focused on multidimensional arrays that are cubes and we denote by n the length of the side of the studied arrays for the rest of this paper. As commonly known, generalizing one-dimensional constraints to higher dimensions is a difficult problem, especially if the original constraints are not easy to solve; see for example [2], [29]. Hence, the main novelty of

Manuscript received 23 July 2022; revised 20 March 2023; accepted 13 June 2023. Date of publication 21 June 2023; date of current version 15 September 2023. This work was supported by the United States–Israel Binational Science Foundation (BSF) under Grant 2018048. An earlier version of this paper was presented in part at the 2022 IEEE Information Theory Workshop. *(Corresponding author: Sagi Marcovich.)*

The authors are with the Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel (e-mail: sagimar@cs.technion.ac.il; yaakobi@cs.technion.ac.il).

Communicated by L. Dolecek, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2023.3288291.

Digital Object Identifier 10.1109/TIT.2023.3288291

0018-9448 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

$$Z = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Fig. 2. 5-square Z contains two identical 3-squares (marked in blue). However, Z contains no identical 4-squares and hence Z satisfies the 4-squares unique constraint.

this work is assimilated in the ability to provide various encoding and decoding schemes, as well as cardinality analyses, for various dimensions. Furthermore, all of the encoding and decoding schemes in this paper use just a single redundancy symbol, and thus provide powerful properties for the minimum redundancy cost. This approach follows several other recent works which studied encoding with just a single redundancy symbol [7], [17], [19], [22].

The zero-L-cube free constraint was studied for the one-dimensional sequences, i.e., $d = 1$, in [17], referred by *L-RLL* sequences. It was shown that if $L = \log_q(n) - f(n)$ and it satisfied that $n-2L = \Theta(n)$ then the redundancy of this family of sequences is $\Theta(q^{f(n)})$. The authors of [17] also proposed an encoding and decoding scheme for binary sequences, i.e., $q = 2$, that uses a single redundancy bit and avoids zero-runs of length $L = \lceil \log(n) \rceil + 1$.

The family of L-cubes unique arrays was studied in [7], and it was shown that for L that satisfies $L^d = \lfloor ad \log_q(n) \rfloor$ with $a > 1$, the asymptotic rate of this family of arrays approaches 1 as $n \to \infty$. The authors also proposed two encoding and decoding schemes for the binary one-dimensional sequences, i.e., $d = 1, q = 2$, referred by *L-repeat free sequences*. The first encoding scheme uses a single redundancy symbol and $L = 2\lceil \log_2(n) \rceil + 2$, while the second scheme requires $L = \lceil a \log_2(n) \rceil$ where $1 < a \leqslant 2$ and its asymptotic rate approaches 1 as $n \to \infty$.

In this paper, we study the family of zero-L-cube free arrays for various values of q and d. We show that if L is a non-decreasing function of n then the asymptotic rate of this family of arrays approaches 1 as $n \to \infty$. Moreover, it is shown that if $L \geqslant \sqrt[d]{d \log_q(n) + \log_q(\frac{q}{q-1})}$, the redundancy of this family of arrays is at most a single symbol. Notice that the difference between the two latter values of L is at most $1+\sqrt[d]{2}$. Then, we present an efficient encoding and decoding scheme of L-cubes free arrays that uses a single redundancy symbol for $L = \left\lceil \sqrt[d]{\lceil d \log_q(n) \rceil + 1} \right\rceil$. Moreover, we carefully study the cardinality of this family of arrays for $d = 2$. It is proven that if L satisfies that $n - 2L = \Theta(n)$ then the redundancy of this family of arrays is $\Theta\left(n^2/q^{L^2}\right)$.

We also study the family of zero-L-cube free arrays for various values of q and d. It is shown that for values of L such that $L \geqslant \sqrt[d]{2d \log_q(n) + \log_q(\frac{q}{q-1})}$, then the redundancy of this family of arrays is at most a single symbol. Additionally, we present a novel encoding and decoding scheme for the binary two-dimensional case, i.e., $d = 2, q = 2$. This scheme uses a single binary bit and requires $L = 2\left\lceil \sqrt{\lceil 3 \log(n) \rceil + 2} \right\rceil$. Note that this value of L is far from the lower bound we found only by roughly a factor of $\sqrt{3}$.

Later, the paper tackles novel and interesting extensions of the aforementioned constraints where instead of the sub-arrays being constrained are d-dimensional cubes, they are allowed to be any d-dimensional boxes, i.e., a d-dimensional generalization of rectangles, where only the volume of the boxes is bounded. Namely, a d-dimensional box has d sides of lengths $x_1, \ldots, x_d$ while its volume $\prod_{i=1}^d x_i$ is bounded. We say that a d-dimensional array is *zero-V-box free* if it does not contain any box of zeros with volume of at least V. Similarly, it is said that a d-dimensional array is *V-boxes unique* if it does not contain any two identical boxes with volume of at least V. As far as we know, these constraints were not studied before. In order to study these two constraints, we first bound for any d, V the number of minimal d-dimensional V-boxes, that is, boxes with volume at least V that are not contained in any other box of such volume. It is shown that for fixed d, there are $\Theta(V^{\frac{d-1}{d}})$ minimal V-boxes. Then, we provide for these two family of arrays cardinality analyses for various values of V, q, d. For the zero-V-box free constraint, we also provide an efficient encoding and decoding scheme for any q, d that uses a single redundancy symbol.

The rest of the paper is organized as follows. In Section II, we formally define the constraints studied in this paper, and review several previous results. In Section III, we study the zero-L-cube free constraint and in Section IV we address the L-cubes unique constraint. Then, in Section V we study the extensions of these constraints to the zero-V-box free and V-boxes unique constraints. Section VI refines the study of both zero-L-cube free and zero-V-box free constraints for the two-dimensional case and provides an accurate asymptotic analysis of their cardinality for various values of L. Finally, Section VII concludes this paper.

II. Definitions and Preliminaries

In this section we formally define the notations and constraints studied in this paper. For integers $i, j \in \mathbb{N}$ such that $i \leqslant j$ we denote by $[i, j]$ the set $\{i, i+1, \ldots, j-1, j\}$. We notate by $[i]$ a shorthand for $[0, i-1]$. For a set A, let $|A|$ denote the number of elements in A. Let Σ_q denote a finite alphabet of size $|\Sigma_q| = q$. When $q = 2$, we omit the subscript q from this and from similar notations.

Let $d \in \mathbb{N}$ be an integer, let $\mathbb{N}^d$ be the d-dimensional grid, and let $\boldsymbol{v} = (v_0, v_1, \ldots, v_{d-1}) \in \mathbb{N}^d$ denote a *coordinate vector* of length d. For $A \subseteq \mathbb{N}^d$, a set of coordinate vectors, we denote by $\boldsymbol{v} + A$ the set

$$\{(v_0 + u_0, \ldots, v_{d-1} + u_{d-1}) \mid \boldsymbol{u} = (u_0, \ldots, u_{d-1}) \in A\},$$

and by $c \cdot A$, where $c \in \mathbb{N}$, the set

$$\{(cu_0, \ldots, c\, u_{d-1}) \mid \boldsymbol{u} = (u_0, \ldots, u_{d-1}) \in A\}.$$

The set $\boldsymbol{v} - A$ is defined similarly. Next, for a set $A \subseteq \mathbb{N}^d$ we denote by Σ_q^A the set of all functions from A to Σ_q. We denote by $\bigcup_{A \subseteq \mathbb{N}^d} \Sigma_q^A$ the set of all *d-dimensional arrays*. For an integer $n \in \mathbb{N}$, we denote by $[n]^d$ the set $[n]^d = \otimes_{i=0}^{d-1} [n]$ and say

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

that $\Sigma_q^{[n]^d}$ is the set of all *d-dimensional n-cubes*. Throughout this paper, we sometimes remove the d-dimensional prefix when using those notations when the dimension d is clear from the context. When $d = 2$, we refer to d-dimensional n-cubes as n-squares. Additionally, the redundancy of a set $\mathcal{A} \subseteq \Sigma_q^{[n]^d}$ is defined as $red(\mathcal{A}) = n^d - \log_q(|\mathcal{A}|)$.

Example 1: Let $d = 2$. The 2-dimensional grid $\mathbb{N}^2$ if the set of all coordinate vectors of length 2. For example, $A_1 = [3]^2 = \{(0,0),(0,1)(0,2),(1,0),(1,1),(1,2),(2,0),(2,1),(2,2)\}$ and $A_2 = \{(0,0),(0,1)(0,2),(1,0),(1,1),(2,0),(2,1)\}$ are subsets of $\mathbb{N}^2$.

For any $A \subseteq \mathbb{N}^2$, the set Σ_q^A contains all mapping from each coordinate vector of A to a symbol of Σ_q. For simplicity, when $d = 2$, the elements of Σ_q^A can be represented by a matrix (which can be sometimes sparse). For example, for $q = 2$,

$$W = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \in \Sigma^{A_1}.$$

Notice that in our example, since $A_1 = [3]^2$, the set Σ^{A_1} is the set of all 2-dimensional 3-squares. □

For two non-negative functions f, g of a common variable n, by denoting $L \triangleq \limsup_{n\to\infty} \frac{f(n)}{g(n)}$ (in the wide sense) we say that $f = o_n(g)$ if $L = 0$, $f = \Omega_n(g)$ if $L > 0$, $f = O_n(g)$ if $L < \infty$, and $f = \omega_n(g)$ if $L = \infty$. When it is clear from the context, we omit the subscript from the aforementioned notations.

Let $W \in \Sigma_q^A$ be an array and $A' \subseteq A \subseteq \mathbb{N}^d$ be sets of coordinate vectors. We denote by $W_{A'}$ the restriction of W to the coordinates in A'. When A' contains a single coordinate vector $A' = \{\boldsymbol{v}\}$ we simplify the representation and write $W_{\boldsymbol{v}}$.

Example 2: Let d, A_1, A_2, W from Example 1. Since $A_2 \subset A_1$, we can restrict W to the coordinates of A_2 and write

$$W_{A_2} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & \\ 0 & 0 & \end{pmatrix} \in \Sigma^{A_2}.$$

Notice that since A_2 does not contain the coordinates $(1,2)$ and $(2,2)$, these coordinates correspond with missing entries in the matrix representation of W_{A_2}.

Additionally, another example is given by letting $\boldsymbol{v} = (0,0) \in A_1$, then we have the restriction $W_{\boldsymbol{v}} = 1$. □

Next, we define a total order over $\mathbb{N}^d$.

Definition 1: Let $\boldsymbol{u} = (u_0,\ldots,u_{d-1}), \boldsymbol{v} = (v_0,\ldots,v_{d-1}) \in \mathbb{N}^d$ be two different coordinate vectors. We say that $\boldsymbol{u} < \boldsymbol{v}$ if there exists $0 \leqslant s \leqslant d-1$ such that $u_s < v_s$ and for every $0 \leqslant t < s$, $u_t = v_t$.

For a set $A \subseteq \mathbb{N}^d$ and a vector $\boldsymbol{v} \in A$, the mapping $B_{A,q}(\boldsymbol{v})$ returns a q-ary vector of the index representation of $\boldsymbol{v}$ in A, where the vectors are ordered increasingly according to the total order presented in Definition 1. Note that the size of the mapping output is $\lceil \log_q(|A|) \rceil$.

Example 3: Let d, A_1 from Example 1. The elements of A_1 satisfy the following order:

$$(0,0) < (0,1) < (0,2) < (1,0) < (1,1) < \cdots < (2,2).$$

The size outputs of B_{A_1} is $\lceil \log_2(9) \rceil = 4$. For example, the index of $(0,0)$ in A_1 is 0, and therefore $B_{A_1}((0,0)) = 0000$. The index of $(2,2)$ in A_1 is 8, and therefore $B_{A_1}((2,2)) = 1000$. □

For an integer $i \in [d]$, let $\boldsymbol{e}_i \in \Sigma_2^d$ denote the i-th unit vector, i.e., a vector with *one* at its i-th bit and *zeros* elsewhere. Additionally, we denote the bijection $MD_A : \Sigma_q^{|A|} \to \Sigma_q^A$ which transforms a sequence to its multidimensional representation under the coordinates of A, and its inverse $SD_A : \Sigma_q^A \to \Sigma_q^{|A|}$. MD_A reorders the symbols using the order of Definition 1 over the coordinates of A, i.e., the i-th symbol of the input sequence will transform to the symbol in the i-th coordinate in A. We will sometimes omit A from the notations when it is clear from the context.

Example 4: Let $d = 2, n = 4$, and

$$X = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \in \Sigma^{[n]^2}.$$

Then, $SD_{[n]^2}(X) = 1100101100010001 \in \Sigma^{16}$ (notice that $|[n]^2| = 16$). Moreover, let $\boldsymbol{s} = 0101101000111100 \in \Sigma^{16}$, then

$$MD_{[n]^2}(\boldsymbol{s}) = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \in \Sigma^{[n]^2}.$$

□

Next, the main families of constraints that are studied in the paper are defined.

Definition 2: Let $W \in \Sigma_q^{[n]^d}$ be a d-dimensional array. We say that W contains a **zero-L-cube** (or **zero-L-square** for $d = 2$) at position $\boldsymbol{v} \in [n-L+1]^d$, if $W_{\boldsymbol{v}+[L]^d} = \mathbf{0}$. An array W satisfies the **zero-L-cube free constraint** if it does not contain any zero-L-cube.

Throughout the paper, we sometimes refer to an array that satisfies the constraint in Definition 2 as a *zero-L-cube free* array.

For positive integers n, q, d, L, we denote by $\mathcal{C}_{d,q}(n,L)$ the set of all arrays over $\Sigma_q^{[n]^d}$ that satisfy the zero-L-cube free constraint. The authors of [17] studied the one dimensional variation of this problem and showed that if $L = \log_q(n) - f(n)$, where $f(n)$ is a function that satisfies $n - 2(\log_q(n) - f(n)) = \Theta(n)$, then the redundancy of $\mathcal{C}_{1,q}(n, f(n))$ is $\Theta(q^{f(n)})$. They also proposed an encoding scheme for the binary case that uses a single redundancy bit and avoids zero-runs of length $L = \lceil \log(n) \rceil + 1$.

In Section III, we analyze the cardinality of $\mathcal{C}_{d,q}(n,L)$ for any d, q, and present lower bounds for L for two cases: 1) the asymptotic rate of $\mathcal{C}_{d,q}(n,L)$ is 1, and 2) the redundancy of $\mathcal{C}_{d,q}(n,L)$ is at most a single symbol. Moreover, we present an algorithm that encodes arrays from $\mathcal{C}_{d,q}(n,L)$ using a single redundancy symbol, where L almost achieves the lower bound that we found for this case. In Section VI we revisit this

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

TABLE I
TABLE OF DEFINITIONS AND NOTATIONS

Notation	Meaning	Remarks
n	Length of side of studied arrays	Sec. I
L	Length of side of cube contained in studied arrays	Sec. I
d	Dimension	Sec. I
Σ_q	Alphabet of size q	Sec. II
$[i]=[0,i-1]$	The set $\{0,\ldots,i-1\}$	Sec. II
$\mathbb{N}^d$	The d-dimensional grid	Sec. II
$\boldsymbol{v}=(v_0,v_1,\ldots,v_{d-1})\in\mathbb{N}^d$	A coordinate vector of length d	Sec. II
$A\subseteq\mathbb{N}^d$	A set of coordinate vectors	Sec. II
$\bigcup_{A\subseteq\mathbb{N}^d}\Sigma_q^A$	The set of all d-dimensional arrays	Sec. II
$[n]^d$	The set $\otimes_{i=0}^{d-1}[n]$	Sec. II
$\Sigma_q^{[n]^d}$	The set of all d-dimensional n-cubes	Sec. II
$red(\mathcal{A})=n^d-\log_q(\lvert\mathcal{A}\rvert)$	The redundancy of a set $\mathcal{A}\subseteq\Sigma_q^{[n]^d}$	Sec. II
$W_{A'}$ for $W\in\Sigma_q^A, A'\subseteq A\subseteq\mathbb{N}^d$	Restriction of W to the coordinates of A'	Sec. II
$B_{A,q}(\boldsymbol{v})$	An index representation of $\boldsymbol{v}$ in A	Sec. II
$\boldsymbol{e}_i\in\Sigma_2^d$	The i-th unit vector	Sec. II
MD_A	Multidimensional transformation of a sequence under A	Sec. II
SD_A	One-dimensional transformation of an array under A	Sec. II
$\mathcal{C}_{d,q}(n,L)$	The set of all zero-L-cube free arrays over $\Sigma_q^{[n]^d}$	Sec. II
$\mathcal{D}_{d,q}(n,L)$	The set of all L-cubes unique arrays over $\Sigma_q^{[n]^d}$	Sec. II
An $(n,\boldsymbol{v})$-semi square	A two-dimensional array over $\Sigma^{[n]^2\setminus(\boldsymbol{v}+[n]^2)}$.	Sec. IV
$CR(X)$	The self-concatenation of a semi-square X	Sec. IV
V-box	An array over $\Sigma_q^{[x_0]\times\cdot\times[x_{d-1}]}$ s.t. $x_0\cdots x_{d-1}=V$	Sec. V
$\mathcal{CA}_{d,q}(n,V)$	The set of all zero-V-box free arrays over $\Sigma_q^{[n]^d}$	Sec. V
$\mathcal{DA}_{d,q}(n,V)$	The set of all V-boxes unique arrays over $\Sigma_q^{[n]^d}$	Sec. V
$F_d(V), f_d(V)$	The set, number of minimal boxes with volume of at least V	Sec. V

constraint for the two-dimensional case and give tight bounds for its redundancy for every n, L.

Next, the second constraint studied in the paper is defined.

Definition 3: Let $W\in\Sigma_q^{[n]^d}$ be a d-dimensional array. We say that W contains two **identical L-cubes** (or **identical L-squares** for $d=2$) at positions $\boldsymbol{u}\neq\boldsymbol{v}\in[n-L+1]^d$, if $W_{\boldsymbol{u}+[L]^d}=W_{\boldsymbol{v}+[L]^d}$. An array W satisfies the **L-cubes unique constraint** if it does not contain any two identical L-cubes.

Throughout the paper, we sometimes refer to an array that satisfies the constraint in Definition 3 as an *L-cubes unique* array.

We denote by $\mathcal{D}_{d,q}(n,L)$ the set of all arrays over $\Sigma_q^{[n]^d}$ that satisfy the L-cubes unique constraint. In [7], the authors analyzed the cardinality of $\mathcal{D}_{d,q}(n,L)$ and proved that for L satisfying $L^d=\lfloor ad\log_q(n)\rfloor$ with $a>1$, the asymptotic rate of $\mathcal{D}_{d,q}(n,L)$ approaches 1. Namely,

$$\lim_{n\to\infty}\frac{\log_q(\lvert\mathcal{D}_{d,q}(n,L)\rvert)}{n^d}=1.$$

Additionally, the authors of [7] proposed two encoding schemes for the one dimensional case of the set $\mathcal{D}_1(n,L)$, which is also known as the set of *repeat free strings* [7], [11] or *L-substring unique* sequences [19]. The first scheme is applied for substrings of length $L=2\lceil\log(n)\rceil+2$ with a single bit of redundancy, and the second one works for substrings of length $L=\lceil a\log(n)\rceil$ for any $1<a\leqslant 2$ and its asymptotic rate approaches 1 as $n\to\infty$. In Section IV, we present for all d,q a lower bound for L such that the redundancy of $\mathcal{D}_{d,q}(n,L)$ is at most 1. Then, we present an encoding scheme for the binary multidimensional case that uses a single redundancy bit, while the value of L is far from the lower bound we found only by a factor of $\sqrt{3}$.

Finally, for the convenience of the reader, relevant notations and terminology referred to throughout the paper are summarized in Table I.

III. THE ZERO CUBES FREE CONSTRAINT

In this section we study the zero cube free constraint. We will show in Theorem 4 a lower bound on L for which the asymptotic rate of the set $\mathcal{C}_{d,q}(n,L)$ is 1. Then, in Theorem 5, we find a lower bound on L which implies that the redundancy of the set $\mathcal{C}_{d,q}(n,L)$ is bounded from above by 1. Lastly, we present efficient encoding and decoding scheme that use a single redundancy symbol to encode arrays that are zero-L-cube free for $L=\left\lceil\sqrt[d]{\lceil d\log_q(n)\rceil+1}\right\rceil$. It is important to mention that most of the techniques used to encode the one-dimensional variation of this constraint in [17] could not be applied to higher dimensions. Therefore, our encoding and decoding scheme provides unique and novel approaches for applying this constraint to multidimensional arrays of any dimension d over an alphabet of any size q. Specifically, the ideas of using a *lookup-cube* and invoking the transformations SD, MD (defined in Section II) in order to move symbols within the array (see Algorithm 1) are novel and innovative.

We start this section by presenting a condition for L such that the asymptotic rate of the set $\mathcal{C}_{d,q}(n,L)$ is 1 as $n\to\infty$.

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

Theorem 4: Let $L = f(n)$ be a positive non-decreasing function of n that satisfies $L = \omega_n(1)$. Then, the asymptotic rate of $\mathcal{C}_{d,q}(n, L)$ is 1. Namely,

$$\lim_{n\to\infty} \frac{\log_q(|\mathcal{C}_{d,q}(n, L)|)}{n^d} = 1.$$

Proof: Let A be the set of coordinates

$$A = (L \cdot [1, \left\lceil \frac{n}{L} \right\rceil - 1])^d \subseteq [n]^d,$$

and let $\mathcal{S}$ be the following set of arrays,

$$\mathcal{S} = \{X \in \Sigma_q^{[n]^d} \mid \text{ for every } \boldsymbol{v} \in A, X_{\boldsymbol{v}} = 1\}.$$

For every $X \in \mathcal{S}$, every L-cube contained in X contains a coordinate from A, and thus X is zero-L-cube free and $\mathcal{S} \subseteq \mathcal{C}_{d,q}(n, L)$. The size of $\mathcal{S}$ satisfies $|\mathcal{S}| \geqslant q^{n^d - \frac{n^d}{L^d}}$, and therefore it is deduced that

$$\lim_{n\to\infty} \frac{\log_q(|\mathcal{C}_{d,q}(n, L)|)}{n^d} \geqslant 1 - \frac{1}{L^d},$$

which approaches 1 for $L = \omega_n(1)$. ■

Next, we utilize the union bound to reach the following upper bound on the redundancy of the set $\mathcal{C}_{d,q}(n, L)$.

Theorem 5: For an integer $L \geqslant \sqrt[d]{d \log_q(n) + \log_q(\frac{q}{q-1})}$, it holds that $|\mathcal{C}_{d,q}(n, L)| \geqslant q^{n^d - 1}$. That is, *red*$(\mathcal{C}_{d,q}(n, L)) \leqslant 1$.

Proof: Let $W \in \Sigma_q^{[n]^d}$ be an array. If W is not zero-L-cube free, then it contains at least one zero-L-cube. Therefore, according to the union bound, the number of arrays over $\Sigma_q^{[n]^d}$ that are not zero-L-cube free can be bounded from above by

$$n^d q^{n^d - L^d} = q^{n^d} \cdot \frac{n^d}{q^{L^d}} \leqslant q^{n^d} \cdot \frac{n^d}{q^{d \log_q(n)} \cdot q^{\log_q(\frac{q}{q-1})}} = (q-1)q^{n^d - 1},$$

where the inequality follows from the lower bound on L stated in the theorem. This implies that

$$|\mathcal{C}_{d,q}(n, L)| \geqslant q^{n^d} - (q-1)q^{n^d - 1} = q^{n^d - 1}.$$

■

Our next goal in the paper is to provide an algorithm that encodes d-dimensional arrays over $\Sigma_q^{[n]^d}$ which satisfy the zero-L-cube constraint for

$$L = \left\lceil \sqrt[d]{\lceil d \log_q(n) \rceil + 1} \right\rceil.$$

Note that the difference between this value of L and the lower bound derived in Theorem 5 is at most $1 + \sqrt[d]{2}$. The algorithm uses a single redundancy symbol and its encoding and decoding time complexities is $O(dn^d \log(n))$.

Algorithm 1 receives a d-dimensional array $W \in \Sigma_q^{[n]^d \setminus \{(n-1)\cdot\mathbf{1}\}}$ with a single symbol missing at its corner, and outputs a cube $X \in \mathcal{C}_{d,q}(n, L)$. First, we initialize X with W and set 1 at the missing entry to mark the start of the algorithm. Then, we scan over all L-cubes in X from start to end and look for a zero-L-cube. When such a cube is found, it is replaced with the non-zero cube at the position $(n-L)\cdot\mathbf{1}$ which will be referred as the *lookup-cube*. The lookup-cube is then filled with an encoding of the position of the zero cube that was found and at least one more additional zero symbol to mark the occurrence of the zero cube to the decoding process. In the case which the found cube and the lookup-cube intersect, we backup only the non-intersecting part of the lookup-cube, since we know the rest of it is zero. Notice that in both cases the size of X remains n^d, and therefore the algorithm returns an array of size n^d.

Algorithm 1 Zero-L-Cube Free Encoding

Input: A d-dimensional array $W \in \Sigma_q^{[n]^d \setminus \{(n-1)\cdot\mathbf{1}\}}$
Output: A d-dimensional array $X \in \mathcal{C}_{d,q}(n, L)$

1: Set an array $X \in \Sigma^{[n]^d}$ with $X_{[n]^d \setminus \{(n-1)\cdot\mathbf{1}\}} = W$ and $X_{(n-1)\cdot\mathbf{1}} = 1$
2: **for** every $\boldsymbol{v} \in [n-L+1]^d$ (iterate in an increasing order) **do**
3:     **if** $X_{\boldsymbol{v}+[L]^d} = \mathbf{0}$ **then**
4:         **if** $A = (\boldsymbol{v} + [L]^d) \cap [n-L, n-1]^d = \emptyset$ **then**
5:             Set $X_{\boldsymbol{v}+[L]^d} = X_{[n-L,n-1]^d}$
6:         **else**
7:             Set $\boldsymbol{y} = SD(X_{([n-L,n-1]^d \setminus A)})$
8:             Set $X_{\boldsymbol{v}+[L]^d \setminus A} = MD(\boldsymbol{y})$
9:         **end if**
10:         Set $X_{[n-L,n-1]^d} = MD(B_{[n]^d,q}(\boldsymbol{v} + \boldsymbol{e}_d) \circ 0^{L^d - \lceil d \log_q(n) \rceil})$
11:     **end if**
12: **end for**

The correctness of Algorithm 1 is proved in the next lemma.

Lemma 6: Algorithm 1 successfully outputs an array which is zero-L-cube free.

Proof: Observe that at each iteration of the loop, if the condition at Step 3 is satisfied, the found zero-L-cube is replaced with a non-zero cube, while new zero-L-cube can not be created. The lookup-cube $X_{[n-L,n-1]^d}$ is initialized as non-zero at Step 1, and being kept non-zero after every iteration since $B_{[n]^d,q}(\boldsymbol{v}+\boldsymbol{e}_d) > \mathbf{0}$ for every $\boldsymbol{v} \in [n-L+1]^d$. Thus, it is ensured that at Step 5 we replace a zero cube with a non-zero cube. This also holds for Step 8 in which the found zero cube intersects with the lookup-cube, since $X_{\boldsymbol{v}+[L]^d \setminus A}$ is filled with the non-zero data part of the lookup-cube. Therefore, since we iterate over all the L-cubes in X at Step 2, when the algorithm ends there are no zero-L-cube left. Lastly, note that since $L^d > \lceil d \log_q(n) \rceil$, Step 10 is valid and after every iteration of the algorithm $X_{(n-1)\cdot\mathbf{1}} = 0$. ■

In order to reconstruct $W \in \Sigma_q^{[n]^d \setminus \{(n-1)\cdot\mathbf{1}\}}$ from X, the output of Algorithm 1, we repeatedly inverse the encoding loop. Note that at each iteration that the algorithm encoded a position of an L-cube at Step 10, we have $X_{(n-1)\cdot\mathbf{1}} = 0$. Thus, we execute the following procedure.

We conclude this section with the following lemma.

Lemma 7: The time complexity of Algorithm 1 and Algorithm 2 is $\Theta(dn^d \log(n))$.

Proof: Both the encoding and decoding algorithms have the same number of iterations, which is $O(n^d)$.

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

Algorithm 2 Zero-L-Cube Free Decoding

while $X_{(n-1)\cdot\mathbf{1}} = 0$ **do**
  Extract $\boldsymbol{v} \in [n-L]^d$ from $SD(X_{[n-L,n-1]^d})$
  Set $A = (\boldsymbol{v} + [L]^d) \cap [n-L, n-1]^d$
  Set $X_{[n-L,n-1]^d \setminus A}$ with $MD(SD(X_{\boldsymbol{v}+[L]^d \setminus A}))$
  Set $X_{\boldsymbol{v}+[L]^d \setminus A} = \mathbf{0}$
end while
Return $W = X_{[n]^d \setminus \{(n-1)\cdot\mathbf{1}\}}$

The complexity of the encoding or decoding of $B_{[n]^d}(\boldsymbol{v})$ for some $\boldsymbol{v} \in [n]^d$ is $\Theta(d \log n)$. The actions of reading and writing L-cubes have complexity of $\Theta(L^d) = \Theta(d \log(n))$ as well. Therefore, the time complexity of both algorithms is $O(dn^d \log(n))$. ■

Note that Algorithm 1 works also for the one-dimensional case, which achieves the same value of L as the one achieved by the algorithm presented in [17]. However the complexity of the algorithm in [17] is $\Theta(n)$ while the complexity of Algorithm 1 for the one dimensional case is $\Theta(n \log(n))$.

Example 5: Let $n = 7, L = 3$, and the input array is

$$W = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \end{pmatrix}.$$

The algorithm appends *one* at the missing entry to initialize X. Then, it iterates the coordinates in an increasing order until a zero-3-square is found. In the following figures, the lookup-square and the found zero square are highlighted.

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

A zero-3-square found in $\boldsymbol{v} = (1, 0)$. It is replaced with the lookup-square, and the latter is filled with the encoding of $\boldsymbol{v} + \boldsymbol{e}_2 = (1, 0) + (0, 1) = (1, 1)$ using six bits, which is $B((1, 1)) = 000100$, and appending three more zeros to have a 3-square.

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Next, a zero-3-square found in $\boldsymbol{v} = (2, 3)$. It intersects with the lookup-square at positions $A = \{(4, 4), (4, 5)\}$. Hence, the algorithm fills only the non-intersecting part of $X_{(2,3)+[3]^2}$ with the non-intersecting portion of the lookup-square, $\boldsymbol{y} = 0100000$. The lookup-square is filled with the encoding of $\boldsymbol{v} + \boldsymbol{e}_2 = (2, 3) + (0, 1) = (2, 4)$, that is, $B((2, 4)) = 010010$, appended by zeros.

$$X = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

The algorithm finishes iterating the entries of X without finding additional zero-3-squares. The result is indeed a zero-3-square free array. □

IV. The Cubes Unique Constraint

In this section, we analyze the size of the set $\mathcal{D}_{d,q}(n, L)$ and find a condition on L such that its redundancy is at most a single symbol. Furthermore, we provide an encoding and decoding scheme for the binary two-dimensional case that uses a single redundancy symbol, for $L = 2\left\lceil\sqrt{\lceil 3\log(n)\rceil + 2}\right\rceil$. We believe that this is a powerful property to achieve with only a single symbol of redundancy. The structure of this scheme follows the structure of the encoding scheme of *repeat free* strings from [7]; however, the adaptation of these ideas to the two-dimensional case introduces interesting challenges. This includes for example the removal of a square from a specific position in the array, the insertion of a square at a specific position in the array, and avoiding the creation of new identical squares in the array when invoking these actions. Hence, various new concepts and techniques are introduced in this section in order to meet the challenges of this encoding schemes.

First, we use a union bound argument to derive a lower bound for L which assures that the redundancy of $\mathcal{D}_{d,q}(n, L)$ is at most a single symbol.

Theorem 8: For $L \geqslant \sqrt[d]{2d\log_q(n) + \log_q(\frac{q}{q-1})}$, it holds that $|\mathcal{D}_{d,q}(n, L)| \geqslant q^{n^d - 1}$. That is, $red(\mathcal{D}_{d,q}(n, L)) \leqslant 1$.

Proof: If an array $W \in \Sigma^{[n]^d}$ is not L-cubes unique, then it contains at least two identical L-cubes. The number of possible selections of the identical L-cubes coordinates is bounded from above by n^{2d}. These coordinates can be intersecting or not; in both cases one of the cubes is determined from picking the rest of the $n^d - L^d$ entries of W. Hence, according to the union bound, the number of arrays over $\Sigma^{[n]^d}$ that are not L-cubes unique can be bounded from above by

$$n^{2d} q^{n^d - L^d} = q^{n^d} \cdot \frac{n^{2d}}{q^{L^d}} \leqslant q^{n^d} \cdot \frac{n^{2d}}{q^{2d\log_q(n)} \cdot q^{\log_q(\frac{q}{q-1})}} = (q-1)q^{n^d - 1},$$

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

where the last inequality follows from the lower bound on L. This accordingly implies that

$$|\mathcal{D}_{d,q}(n,L)| \geqslant q^{n^d} - (q-1)q^{n^d-1} = q^{n^d-1}.$$

■

Next, we present a generic encoding algorithm that uses a single redundancy bit in order to encode binary n-squares that are L-squares unique, for

$$L = 2\left\lceil \sqrt{\lceil 3\log(n)\rceil + 2}\right\rceil.$$

Note that this value of L is far from the value derived in Lemma 8 only by roughly a factor of $\sqrt{3}$. For simplicity, we sometimes omit ceiling notations in the rest of this section.

We introduce first a new type of two-dimensional arrays, denoted as *bottom semi squares*, or *semi squares* in short. For a vector $\boldsymbol{v} \in [n]^2$, the set $A = [n]^2 \setminus (\boldsymbol{v} + [n]^2)$ contains coordinates of a semi square with a corner at $\boldsymbol{v}$. Hence, we say that $X \in \Sigma_q^A$ is an $(n, \boldsymbol{v})$-semi square.

Let X be an $(n, \boldsymbol{v})$-semi square for $\boldsymbol{v} \in [n]^2$, let t be an integer, and let Y be a $(t, \boldsymbol{u})$-semi square for $\boldsymbol{u} \in [t]^2$. We denote by $X \circ Y$ the concatenation of X and Y which is defined by placing Y at position $\boldsymbol{v}$ of X, and restricting the result to the coordinates in $[n]^2$. It follows that $X \circ Y$ is a $(n, \boldsymbol{v}+\boldsymbol{u})$-semi square if and only if for every $i \in [2]$, $u_i = 0$ or $v_i + t \geqslant n$.

Example 6:

$$X = \begin{pmatrix} 1 & 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 1 & 1\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & & & \\ 0 & 1 & & & \end{pmatrix} \in \Sigma^{[5]^2\setminus((3,2)+[5]^2)}$$

is a $(5, \boldsymbol{v})$-semi square for $\boldsymbol{v} = (3, 2)$, and

$$Y = \begin{pmatrix} 0 & 0 & 1\\ 1 & & \\ 0 & & \end{pmatrix} \in \Sigma^{[3]^2\setminus((1,1)+[3]^2)}$$

is a $(3, \boldsymbol{u})$-semi square for $\boldsymbol{u} = (1, 1)$. Then, the concatenation $X \circ Y$ is a semi $(5, \boldsymbol{v}+\boldsymbol{u})$-square,

$$X \circ Y = \begin{pmatrix} 1 & 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 1 & 1\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\\ 0 & 1 & 1 & & \end{pmatrix}.$$

□

Definition 9: Let X be an $(n, \boldsymbol{v})$-semi square for $\boldsymbol{v} \in [n]^2$, such that $\boldsymbol{v} \neq \mathbf{0}$. We denote by $CR(X)$ the iterative self-concatenation of X to an n-square, that is,

$$CR(X) = X^{\left\lceil \frac{n}{v_{\min}} \right\rceil},$$

where $v_{\min}$ is the smallest entry of $\boldsymbol{v}$ that is not 0.

It can be shown by induction that after m concatenations, X^m is a $(n, m \cdot \boldsymbol{v})$-semi square, since $v_i + n \geqslant n$ for every $i \in [2]$. Thus, the self-concatenation of an $(n, \boldsymbol{v})$-semi square for every $\boldsymbol{v} \neq \mathbf{0}$ is defined properly, and in fact an n-square since for every $i \in [2]$, $\left\lceil \frac{n}{v_{\min}} \right\rceil \cdot v_i \geqslant n$.

Example 7: Let X, Y from Example 6. Then,

$$CR(X) = \begin{pmatrix} 1 & 1 & 0 & 0 & 1\\ 1 & 0 & 1 & 1 & 1\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1 & 0\\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}, CR(Y) = \begin{pmatrix} 0 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0 \end{pmatrix}.$$

□

Additionally, we define the matching *upper semi squares*, described by a coordinate vectors set of $A = [n]^2 \setminus (\boldsymbol{v} - [n]^2)$ for $\boldsymbol{v} \in [n]^2$. We similarly define concatenation and self-concatenation to an n-square of upper semi squares.

Algorithm 4 receives a two-dimensional array $W \in \Sigma^{[n]^2\setminus\{(0,0)\}}$, an n-square with a single missing entry, and outputs an n-square $X \in \mathcal{D}_2(n, L)$. The algorithm consists of two main procedures, elimination and expansion. First, we initialize X with W and set 0 at the missing entry to mark the start of the elimination. Then, we append to X a marker $(L/2)$-square that will mark the transition between the elimination and the expansion parts of the encoder. At the elimination part, we iteratively shorten X by an $(L/2)$-square at a time by eliminating one of the two occurrences in X: 1. two identical L-squares, 2. two identical rectangles of size $[L/2] \times [L]$ (notated for the rest of this section as $(L/2, L)$-rectangles) where one of them is at the bottom of X. Likewise, we make sure that the marker $(L/2)$-square appears only once in X. Later, at the expansion part, we enlarge X to an n-square by iteratively appending $(L/2)$-squares while making sure that no new identical L-squares are created.

For convenience, we denote for the rest of this section

$$k = L/2 = \left\lceil \sqrt{3\log(n) + 2}\right\rceil.$$

We define the *marker k-square* denoted as P_M as the following square,

$$P_M = \begin{pmatrix} 1 & 0 & \cdots & 0\\ 0 & & & \\ \vdots & & \mathbf{0} & \\ 0 & & & \end{pmatrix} \in \Sigma^{[k]^2}.$$

Before presenting Algorithm 4, we explain the notion of removal and insertion of squares with granularity. We assume that $n \bmod k = 0$, and let $X \in \Sigma^{[n]^2\setminus A_t}$ where A_t contains the coordinates of the last t k-squares in $[n]^2$, i.e., $A_t = ([i_t - k, n-1] \times [j_t, n-1]) \cup ([i_t, n-1] \times [n])$ where $(i_t, j_t) = (n - \lfloor tk/n \rfloor, n - (tk \mod n))$. We look at X as a grid of k-squares, and allow only removals and insertions in granularity of k-square units. Removal or insertion actions on a k-square at an aligned position $(\hat{i} \cdot k, \hat{j} \cdot k)$ are performed by transforming X to a k-rows array with coordinates $[k] \times [|X|/k]$, executing the action on the $(i \cdot k + j)$-th k-square like in a one-dimensional array, and transforming back to a grid of k-squares.

However, during the elimination part of the algorithm we sometimes need to remove a k-square from an unaligned position (i, j). Such an action is done by finding the closest aligned position $(\hat{i} \cdot k, \hat{j} \cdot k)$, then replacing the data of the non-intersecting parts of $X_{(\hat{i}\cdot k, \hat{j}\cdot k)+[k]^2}$ and $X_{(i,j)+[k]^2}$, and finally removing the aligned k-square at position $(\hat{i} \cdot k, \hat{j} \cdot k)$. This is a technical procedure that can be transparent to

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

the reader of the encoder in Algorithm 4. Nonetheless, for completeness of the encoder, this procedure is explained in Algorithm 3.

Algorithm 3 Removing a k-Square With Granularity

Removing a k-square from position (i,j) of $X \in \Sigma^{[n]^2 \setminus A_t}$

Find maximal $\hat{i}, \hat{j}$ such that $\hat{i} \cdot k \leqslant i$, $\hat{j} \cdot k \leqslant j$

Replace $X_{(\hat{i}\cdot k, \hat{j}\cdot k)+[k]^2 \setminus ((i,j)+[k]^2)}$ with the data of $X_{(i,j)+[k]^2 \setminus ((\hat{i}\cdot k, \hat{j}\cdot k)+[k]^2)}$

Transform $\widehat{X} = MD_{[k]\times[|X|/k]}(SD(X))$

Remove $(\hat{i} \cdot k + \hat{j})$ k-square from $\widehat{X}$

Transform $X = MD_{[n]^2 \setminus A_{t+1}}(SD(\widehat{X}))$

All of the above ensures that appended k-squares, and specifically the marker k-square, are not trimmed or modified accidentally as a result of unrelated removal of insertion actions.

First, we show that Algorithm 4 reaches Step 14, i.e., the elimination part terminates, by showing that at each iteration of the elimination loop the length of X decreases or the Hamming weight of X increases. Notice that all removal and insertion actions are done with granularity of k-square units, as allowed. We analyze each case of removal and insertion independently.

Case 1: We remove a square of size k^2 with Hamming weight equals to $w_H(P_M) = 1$ and insert a square of size k^2 with Hamming weight of at least $\log(n)$.

Case 2: We remove a square of size $L^2 = 4k^2$ and insert a smaller rectangle of size $3k^2$.

Case 3: We remove a square of size $L \cdot k = 2k^2$ and insert a smaller rectangle of size $3\log(n) + 2 = k^2$. This follows from the fact that the number of possible indices for (i_2, j_2) satisfies $|I| \leqslant n$.

We prove the correctness of the algorithm in the next few claims.

Claim 10: At Step 14 of Algorithm 4, the two-dimensional array X satisfies the following properties:

(1) X is L-square unique,

(2) X contains no identical (k, L)-rectangles where one of them is at position that belongs to

$$I = (\{i_m - k\} \times [j_m - k, n-1]) \cup (\{i_m\} \times [j_m - k - 1]),$$

(3) X ends with $X_{(i_m,j_m)+[k]^2} = P_M$,

(4) X contains no other k-square equals to P_M.

Proof: First, we prove (3) by showing that throughout the elimination loop, $X_{(i_m,j_m)+[k]^2} = P_M$. This holds at Step 2 before the elimination loop, and during the elimination loop the indices (i_m, j_m) are decremented by a k-square if and only if X was shortened by k^2. Thus, this condition can be violated only if some part of $X_{(i_m,j_m)+[k]^2}$ was removed as part of an elimination procedure in cases 2 or 3. Assume in the contrary that case 2 occurred and there were two identical squares at positions $(i_1, j_1) < (i_2, j_2)$ such that $X_{(i_1,j_1)+[L]^2}$ intersects with $X_{(i_m,j_m)+[k]^2}$. Thus, $X_{(i_1,j_1)+[L]^2}$ contains the 1-bit at the top-left corner of P_M at some position (i_r, j_r). However,

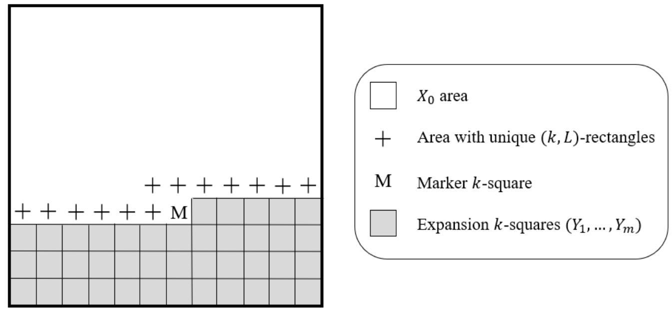

Fig. 3. Structure of X returned by Algorithm 4 at Step 21.

it follows that $X_{(i_2+i_r, j_2+j_r)} = 1$ which is a contradiction since P_M contains a single 1-bit and $(i_1, j_1) \neq (i_2, j_2)$. It can be shown similarly that a part of $X_{(i_m,j_m)+[k]^2}$ can not be removed in case 3. Statements (1), (2), (4) follows from the fact that the elimination loop terminates, using cases 2, 3, 1 of the elimination loop, respectively. ■

Claim 11: For every iteration of the expansion loop, the set $\Sigma^{[k]^2} \setminus \mathcal{S}$ is not empty.

Proof: The size of the set $\mathcal{S}$ satisfies

$$|\mathcal{S}| \leqslant |X| \leqslant n^2,$$

while the alphabet $\Sigma^{[k]^2}$ satisfies

$$|\Sigma^{[k]^2}| \geqslant 2^{3\log(n)+2} > n^2.$$ ■

Let m denote the number of iterations of the expansion loop of Algorithm 4 that were executed. For every $\ell = 1, \ldots, m$, let X_ℓ denote the value of X at the end of the ℓ-th iteration, and let Y_ℓ denote the expansion k-square that the algorithm picked at Step 19. We notate by X_0 the value of X before the first iteration of the expansion loop, i.e., at Step 14 right after the elimination part. Figure 3 presents an example of the structure of X at the end of the expansion part.

Claim 12: For every iteration $\ell = 1, \ldots, m$, the array $X_\ell = X_{\ell-1} \circ Y_\ell$ contains the square Y_ℓ only once, at its end.

Proof: Let i_e, j_e denote the position of Y_ℓ. According to the construction of $\mathcal{S}$, the square Y_ℓ can not appear as a sub-square of $X_{\ell-1}$. Thus, it might appear at some position (i,j) of X, where $(i,j) \in (i_e, j_e) - [k]^2$. Assume in the contrary that such a case occurs. We have

$$(X_\ell)_{(i,j)+[k]^2} = (X_{\ell-1} \circ Y_\ell)_{(i,j)+[k]^2} = Y_\ell.$$

However, it is implied that

$$Y_\ell = CR((X_{\ell-1})_{(i,j)+[k]^2}),$$

which is a contradiction to the construction of $\mathcal{S}$. ■

Claim 13: At Step 21 of Algorithm 4, X is L-square unique.

Proof: Assume in the contrary that X contains two identical L-squares at positions $(i_1, j_1) < (i_2, j_2)$. We prove the claim by examining all different cases for (i_2, j_2) and reaching a contradiction at each case. These cases are also presented graphically at Figure 4.

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

Algorithm 4 L-Squares Unique Encoding

Input: A two-dimensional array $W \in \Sigma^{[n]^2 \setminus \{(0,0)\}}$
Output: An n-square $X \in \mathcal{D}_2(n, L)$

1: Set a square $X \in \Sigma^{[n]^2}$ with $X_{0,0} = 0, X_{[n]^2 \setminus \{(0,0)\}} = W$
2: Denote $(i_m, j_m) = (n, 0)$, append $X_{(i_m,j_m)+[k]^2} = P_M$
First part - Elimination
3: **while** at least one of the occurences in cases 1,2,3 exists **do**
4: **case 1**: A k-square equals to P_M exists at $(i_1, j_1) < (i_m, j_m)$
5: Remove square $X_{(i_1,j_1)+[k]^2}$
6: Set $\boldsymbol{v} = 101 \circ B_{[n]^2}(i_1, j_1) \circ 1^{k^2 - 2\log(n) - 3}$, insert $MD_{[k]^2}(\boldsymbol{v})$ at $X_{0,0}$
7: **case 2**: Identical L-squares exist at $(i_1, j_1) < (i_2, j_2)$
8: Remove square $X_{(i_1,j_1)+[L]^2}$
9: Set $\boldsymbol{v} = 100 \circ B_{[n]^2}(i_1, j_1) \circ B_{[n]^2}(i_2, j_2) \circ 1^{3k^2 - 4\log(n) - 3}$, insert $MD_{[k]\times[3k]}(\boldsymbol{v})$ at $X_{0,0}$
10: **case 3**: Identical (k, L)-rectangles exist at $(i_1, j_1) < (i_2, j_2)$, where

$$(i_2, j_2) \in I = (\{i_m - k\} \times [j_m - k, n-1]) \cup (\{i_m\} \times [j_m - k - 1]),$$

11: Remove rectangle $X_{(i_1,j_1)+[k]\times[L]}$
12: Set $\boldsymbol{v} = 11 \circ B_{[n]^2}(i_1, j_1) \circ B_I(i_2, j_2)$, insert $MD_{[k]^2}(\boldsymbol{v})$ at $X_{0,0}$
13: If cases 2 or 3 were executed, decrement (i_m, j_m) by a k-square
14: **end while**
15: If $|X| \geqslant n^2$, return $X_{[n]^2}$
Second part - Expansion
16: **while** $|X| < n^2$ **do**
17: Set indexes (i_e, j_e) to point to the next missing k-square in X
18: Let $I_e = ((i_e, j_e) - [k]^2) \cap [n]^2$. Set

$$\mathcal{S} = \{X_{(i,j)+[k]^2} \mid (i,j) \notin I_e\} \cup \{CR(X_{(i,j)+[k]^2}) \mid (i,j) \in I_e\}$$

19: Pick $Y \in \Sigma^{[k^2]}/\mathcal{S}$ and set $X_{(i_e,j_e)+[k]^2} = Y$
20: **end while**
21: Return X

(1) If $X_{(i_2,j_2)+[L]^2}$ is contained in X_0, we have a contradiction since X_0 is L-square unique from Claim 10 Statement (1).
(2) If $X_{(i_2,j_2)+[L]^2}$ contains an (k, L)-rectangle which starts at position which belongs to

$$I = (\{i_m - k\} \times [j_m - k, n-1]) \cup (\{i_m\} \times [j_m - k - 1]),$$

it follows that $X_{(i_1,j_1)+[L]^2}$ contains an identical (k, L)-rectangle which is a contradiction to Claim 10 Statement (2).
(3) If $X_{(i_2,j_2)+[L]^2}$ contains at some position (i_r, j_r) the marker k-square $X_{(i_m,j_m)+[k]^2}$, it follows that $X_{(i_2+i_r,j_2+j_r)+[k]^2} = P_M$ from Claim 10 Statement (3). However, therefore $X_{(i_1+i_r,j_1+j_r)+[k]^2} = P_M$ as well which is a contradiction to Claim 10 Statement (4).
(4) Otherwise, $X_{(i_2,j_2)+[L]^2}$ contains an expansion k-square at some position (i_r, j_r). That is, $X_{(i_2+i_r,j_2+j_r)+[L]^2} = Y_\ell$ where Y_ℓ is a k-square that was appended to $X_{\ell-1}$ at the ℓ-th iteration of the expansion loop. It follows that $X_{(i_1+i_r,j_1+j_r)+[k]^2} = Y_\ell$ as well. Thus, Y_ℓ appears twice in X_ℓ which is a contradiction to Claim 12.

■

Finally, observe that if the condition in Step 15 is satisfied, then $X_{[n]^2}$ is an n-square and is also L-square unique from Claim 10 Statement (1). Otherwise, the algorithm reaches

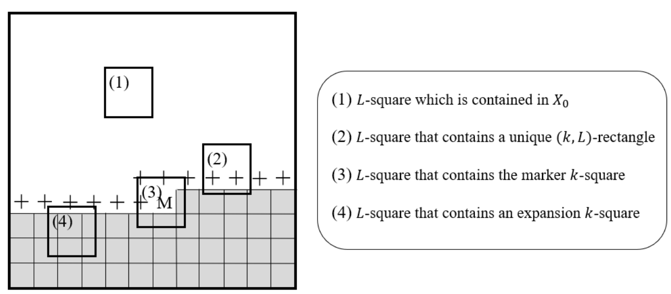

Fig. 4. Different types of L-squares presented in the proof of Claim 13, based on the structure of X that is returned by Algorithm 4 at Step 21, presented in Figure 3. Legend: □ - area of X_0, $+$ - area containing unique (k, L)-rectangles, M - marker k-square, ■ - expansion k-squares.

Step 21 with $X \in \Sigma^{[n]^2}$ which is L-square unique as well from Claim 13.

The decoding scheme receives X which is an output of Algorithm 4 and returns $W \in \Sigma^{[n]^2 \setminus \{(0,0)\}}$. First, we identify the marker square position by looking at the first occurrence of P_M. Using Claim 10 we can remove the part of X after the marker square since it was appended during the expansion procedure. Next, we iteratively inverse the elimination procedure. We identify using the first three entries of X the last elimination case at which data was encoded. If data was encoded at

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

case 2, we decode $\boldsymbol{v} = SD(X_{[k]\times[3k]})$, extract the positions $(i_1, j_1), (i_2, j_2)$ and insert $X_{(i_2,j_2)+[L]^2}$ at position (i_1, j_1) if the L-squares do not intersect. Otherwise, we insert instead the self-concatenation $CR(X_{((i_2,j_2)+[L]^2)\setminus((i_1,j_1)+[L]^2)})$. Similarly, in case 3 we recover $X_{(i_1,j_1)+[k]\times[L]}$ from $\boldsymbol{v} = SD(X_{[k]^2})$. If the data was encoded at case 1, we decode $\boldsymbol{v} = SD(X_{[k]^2})$, extract (i_1, j_1) and insert $X_{(i_1,j_1)+[k]^2} = P_M$. This process is repeated until $X_{(0,0)} = 0$, then we return $X_{[n]^2\setminus\{(0,0)\}}$ as W.

Remark 1: Algorithm 4 requires that $n \bmod k = 0$. However, the algorithm can be altered to support cases where this is not possible, e.g. n is a prime number. In this case, we pad the input array in the right and the bottom with *ones* in order to receive an n'-square, where $n' \geqslant n$ is the closest multiple of k to n. Then, we invoke Algorithm 4 with minor modifications that are described shortly to receive $X' \in \mathcal{D}_2(n', L)$, and return $X = X'_{[n]^2}$. Some information that is valuable for the decoder can be lost when restricting the result to an n-square. In order for the decoder to uniquely identify the marker-square P_M, we pick at Step 19 only squares with $Y_{0,0} = 0$. Additionally, we make sure that the area padded with *ones* that is contained in X_0 (the array X before the expansion part) remains unchanged throughout the elimination. This can be done by adding two cases that are similar to cases 2 and 3 that are specific to when the identical L-square or (k, L)-rectangle found intersects with the padded area. These new cases will encode the special occurrence with a small number of bits and only the non-intersecting part of the sub-array will be removed. Both these modifications will not change the redundancy of the algorithm.

V. Extensions to Multidimensional Boxes of Any Volume

In this section, we introduce a generalization of the zero-L-cubes free and the L-cubes unique constraints to multidimensional arrays where the shape of the sub-array is not necessarily a cube, but a box, where only its volume is bounded from below. A d-dimensional box is a shape that generalizes the shape of a rectangle to any dimension d, and is given by a set of coordinates $A = [x_0] \times \cdots \times [x_{d-1}]$ where the sides $x_0, \ldots, x_{d-1}$ are positive integers that belong to $[1, n]$. The volume of such a box is given by $|A| = x_0 \cdots x_{d-1}$.

Definition 14: Let $W \in \Sigma^{[n]^d}$ be a d-dimensional array. For a positive integer V, we say that W contains a **zero-V-box** (or **zero-V-rectangle** for $d = 2$) with a coordinates set $A = [x_0] \times \cdots \times [x_{d-1}]$ at position $\boldsymbol{v}$ such that $\boldsymbol{v} + A \subseteq [n]^d$, if $W_{\boldsymbol{v}+A} = \mathbf{0}$ and $|A| = V$. An array W satisfies the **zero-V-boxes free constraint** if it does not contain a zero-V'-box, for any positive integer $V' \geqslant V$.

Example 8: Let $n = 5, d = 2$, and

$$Y = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \in \Sigma^{[n]^2}$$

Then, Y contains a zero-6-rectangle, at position $(2, 0)$, marked in red. For any $V > 6$, Y does not contain a zero-V-rectangle; for example, the 8-rectangle at position $(0, 3)$ marked in gray is not a zero rectangle. Therefore, Y satisfies the zero-V-rectangles free constraint. □

Definition 15: Let $W \in \Sigma^{[n]^d}$ be a d-dimensional array. For a positive integer V, we say that W contains **identical V-boxes** (or **identical V-rectangles** for $d = 2$) with a coordinates set $A = [x_0] \times \cdots \times [x_{d-1}]$ at positions $\boldsymbol{u} \neq \boldsymbol{v}$ such that $\boldsymbol{u} + A, \boldsymbol{v} + A \subseteq [n]^d$, if $W_{\boldsymbol{u}+A} = W_{\boldsymbol{v}+A}$ and $|A| = V$. An array W satisfies the **V-boxes unique constraint** if it does not contain two identical V'-boxes, for any positive integer $V' \geqslant V$.

Example 9: Let $n = 5, d = 2$, and

$$Z = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \in \Sigma^{[n]^2}$$

Then, Z contains two identical 6-rectangles at positions $(0, 1)$ and $(2, 3)$, marked in blue. However, Z contains no two identical V-rectangles for $V > 6$ and hence Z satisfies the V-rectangles unique constraint. □

In the rest of this paper, we sometimes refer to an array that satisfies the constraint in Definition 14 as a *zero-V-box free* array, and to an array that satisfies the constraint in Definition 15 as a *V-boxes unique* array. We denote by $\mathcal{CA}_{d,q}(n, V)$ the set of all arrays over $\Sigma^{[n]^d}$ that satisfy the zero-V-box free constraint and by $\mathcal{DA}_{d,q}(n, V)$ the set of all arrays over $\Sigma^{[n]^d}$ that satisfy the V-boxes unique constraint.

To the best of our knowledge, studying multidimensional arrays with constraints on boxes that are bounded only by their volume has received only a little attention; see e.g. [5], and similar problems to the ones studied in this paper were not addressed before. As a consequence, the foundations for studying this family of constraints have to be established first. This includes, for example, enumerating the number of unique minimal boxes of a given volume, which is presented next. Later, we analyze the size of the set $\mathcal{CA}_{d,q}(n, V)$ and find an upper bound on V such that the redundancy of this set is at most a single symbol. We provide an encoding and decoding scheme for any q and d that uses a single redundancy symbol and achieves this bound on V; due to the challenging and flexible structure of these shapes, this scheme uses an entirely different approach compared to Algorithm 1. Finally, we study the cardinality of $\mathcal{DA}_{d,q}(n, V)$ and provide a condition on V such that the asymptotic rate of this set approaches 1 as $n \to \infty$; this is proved based on a nontrivial application of the asymmetric Loàsz local lemma [8].

A. Enumeration of Minimal Boxes

Before analyzing the constraints, we first need to estimate the number of *minimal boxes* for a given volume. For an

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

integer V, Let $F_d(V)$ denote the set of minimal boxes $A = [x_0] \times \cdots \times [x_{d-1}]$ such that $|A| \geqslant V$ and for every other $A' \in F_d(V), A \not\subset A'$. Additionally, let $f_d(V) = |F_d(V)|$.

Example 10: Let $n = 5, d = 2, V = 5$. The set $F_d(V)$ is presented with colors in the following array (notice that the starting position of the rectangle is not important, rather only its shape),

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

We have $A_1 = [1] \times [5]$ (marked in red), $A_2 = [2] \times [3]$ (marked in green), $A_3 = [3] \times [2]$ (marked in blue), and $A_4 = [5] \times [1]$ (marked in gray). Therefore, $f_d(V) = 4$. Notice that for example $A_5 = [2] \times [4]$ does not belong to $F_d(V)$ since $A_2 \in F_d(V)$ is a subset of it. □

First, we examine $f_d(V)$ for small values of d. It is clear that $f_1(V) = 1$. For $d = 2$, there are $\lfloor \sqrt{V} \rfloor$ possibilities for the smaller side of the rectangle, which yields

$$f_2(V) = \begin{cases} 2\lfloor \sqrt{V} \rfloor & \sqrt{V} \notin \mathbb{N} \\ 2\sqrt{V} - 1 & \sqrt{V} \in \mathbb{N} \end{cases},$$

since when $\sqrt{V}$ is an integer, the cube $[\sqrt{V}]^2$ is counted twice. When it is not known if $\sqrt{V}$ is an integer, we can write $f_2(V) \leqslant 2\sqrt{V}$.

After acquiring some intuition regarding the value of $f_d(V)$, we have the following claim in the general case.

Claim 16: For every $d \geqslant 2$ and for every positive integer V,

$$f_d(V) \leqslant \alpha^{d-2} d!(d-1)! V^{\frac{d-1}{d}},$$

where α is a constant that satisfies $1 \leqslant \alpha \leqslant \sqrt{2}$ and approaches 1 as $V \to \infty$.

Proof: We can write $f_d(V)$ as a recursive inequality,

$$f_d(V) \leqslant d \sum_{x=1}^{\lfloor \sqrt[d]{V} \rfloor} f_{d-1}\Big(\Big\lceil \frac{V}{x} \Big\rceil \Big),$$

which follows from having d options for the shortest side, that is at most $\lfloor \sqrt[d]{V} \rfloor$, and letting the remained $d-1$ coordinates determine the volume of the box. Let $\alpha > 1$ be the minimal constant that satisfies $\lceil \frac{V}{x} \rceil \leqslant \frac{\alpha V}{x}$ for every $x \leqslant \lfloor \sqrt{V} \rfloor$. It follows that $\alpha \leqslant \frac{\lceil \sqrt{V} \rceil}{\sqrt{V}} \leqslant 1 + \frac{1}{\sqrt{V}}$. Thus, $\alpha \leqslant \sqrt{2}$ from plugging $V = 2$, and $\lim_{V \to \infty} \alpha \leqslant \lim_{V \to \infty} 1 + \frac{1}{\sqrt{V}} = 1$. We prove the claimed inequality using an induction. From previous calculations, $f_2(V) \leqslant 2\sqrt{V}$ which verifies the claim's inequality for $d = 2$.

Assume next that the inequality holds for $d-1$. Then, we have that

$$\begin{aligned} f_d(V) &\leqslant d \sum_{x=1}^{\lfloor \sqrt[d]{V} \rfloor} f_{d-1}\Big(\Big\lceil \frac{V}{x} \Big\rceil \Big) \\ &\leqslant d \sum_{x=1}^{\lfloor \sqrt[d]{V} \rfloor} f_{d-1}\Big(\frac{\alpha V}{x} \Big) \\ &\leqslant d \sum_{x=1}^{\lfloor \sqrt[d]{V} \rfloor} \alpha^{d-3}(d-1)!(d-2)!\Big(\frac{\alpha V}{x} \Big)^{\frac{d-2}{d-1}} \\ &\overset{(a)}{\leqslant} \alpha^{d-2} d!(d-2)! V^{\frac{d-2}{d-1}} \int_{x=0}^{\sqrt[d]{V}} \frac{dx}{x^{\frac{d-2}{d-1}}} \\ &= \alpha^{d-2} d!(d-2)! V^{\frac{d-2}{d-1}} \Big[(d-1) x^{\frac{1}{d-1}} \Big]_0^{\sqrt[d]{V}} \\ &\leqslant \alpha^{d-2} d!(d-1)! V^{\frac{d-2}{d-1}} V^{\frac{1}{(d-1)d}} \\ &= \alpha^{d-2} d!(d-1)! V^{\frac{d-1}{d}} \end{aligned}$$

where (a) from the inequality $\sum_{i=L}^{U} g(i) \leqslant \int_{L-1}^{U} g(x)dx$ for a nonnegative decreasing function $g(x)$. ■

Claim 17: For every $d \geqslant 2$ and for every positive integer V,

$$f_d(V) \geqslant d(\lfloor \sqrt[d]{V} \rfloor)^{d-1} - d + 1.$$

Proof: Let S denote the set of d-dimensional boxes that are generated by letting the first $d-1$ sides have any value of $[1, \lfloor \sqrt[d]{V} \rfloor]$, and letting the last side fill the remaining volume of the box to V. That is,

$$S = \left\{ [x_0] \times \cdots \times [x_{d-1}] \,\middle|\, \begin{array}{l} x_0, \ldots, x_{d-2} \in [1, \lfloor \sqrt[d]{V} \rfloor], \\ x_{d-1} = \Big\lceil \frac{V}{x_0 \cdots x_{d-2}} \Big\rceil. \end{array} \right\}$$

Let $A_0 = [x_0] \times \cdots \times [x_{d-1}] \in S$ be a box. Clearly, $|A_0| \geqslant V$. Hence, in order to prove that $S \subseteq F_d(V)$, it is left to show that S is minimal. Assume that there exists $A_1 = [y_0] \times \cdots \times [y_{d-1}] \in S$ and w.l.o.g there exists $i \in [d-1]$ such that $y_i > x_i$ and for every other $j \in [d-1] \setminus \{i\}$, $y_j \geqslant x_j$. It follows that

$$\begin{aligned} \frac{V}{x_0 \cdots x_{d-2}} - \frac{V}{y_0 \cdots y_{d-2}} &\geqslant \frac{V}{y_0 \cdots y_{i-1} y_{i+1} \cdots y_{d-2} x_i} - \frac{V}{y_0 \cdots y_{d-2}} \\ &= \frac{V(y_i - x_i)}{y_0 \cdots y_{d-2} x_i} \\ &\geqslant \frac{V(y_i - x_i)}{(\sqrt[d]{V})^d} \\ &= y_i - x_i \\ &\geqslant 1. \end{aligned}$$

Therefore $x_{d-1} > y_{d-1}$ and hence $A_0 \not\subset A_1$. Thus, S is minimal.

From its definition, we have that $|S| = (\lfloor \sqrt[d]{V} \rfloor)^{d-1}$. If $\sqrt[d]{V} \notin \mathbb{N}$, we can shift $d-1$ times the sides of each $A \in S$ in order to generate additional unique boxes that belong to $F_d(V)$. This holds since the remaining side satisfies $x_{d-1} \geqslant \lceil \sqrt[d]{V} \rceil$ where the other sides are at most $\lfloor \sqrt[d]{V} \rfloor$. In the

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

case where $\sqrt[d]{V} \in \mathbb{N}$ we can shift each box of S but the set $[\sqrt[d]{V}]^d$. We can conclude that

$$f_d(V) \geqslant d(\lfloor \sqrt[d]{V} \rfloor)^{d-1} - d + 1.$$

■

The next corollary follows immediately from Claims 16 and 17.

Corollary 18: For every fixed positive $d \in \mathbb{N}$ and for every positive V,

$$f_d(V) = \Theta(V^{\frac{d-1}{d}}).$$

Even though for the results in the paper only an upper bound on the value of $f_d(V)$ would be sufficient, we still found it important to present Corollary 18 for a more comprehensive analysis of the value of $f_d(V)$. In particular, for every $d \in \mathbb{N}$, we can write $f_d(V) \leqslant C_d V^{\frac{d-1}{d}}$ where C_d denotes a positive constant that fulfills Corollary 18. Note that regarding our constraints, the number of minimal boxes of volume V that are contained in $W \in \Sigma_q^{[n]^d}$ and start at position $\boldsymbol{v}$ depends on n and $\boldsymbol{v}$ in addition to the volume V. However, it can be bounded from above by $f_d(V)$.

B. The Zero Boxes Free Constraint

First, we prove the following lemma regarding the cardinality of the set $\mathcal{CA}_{d,q}(n,V)$.

Lemma 19: For $V = d\log_q(n) + \frac{d-1}{d}\log_q(\log_q(n)) + \mathcal{O}(1)$, and for n large enough it holds that $|\mathcal{CA}_{d,q}(n,V)| \geqslant q^{n^d-1}$. That is, *red*$(\mathcal{CA}_{d,q}(n,V)) \leqslant 1$.

Proof: Let $V = d\log_q(n) + \frac{d-1}{d}\log_q(\log_q(n)) + C + \log_q(\frac{q}{q-1})$ for some positive constant C that will be determined later. If an array $W \in \Sigma^{[n]^d}$ is not zero-V-box free, then it contains at least a single zero box with coordinates set $A \in F_d(V)$, such that $|A| \geqslant V$. From Corollary 18 there are at most $C_d V^{\frac{d-1}{d}}$ possible selections of such a coordinates set. Hence, according to the union bound, the number of arrays that are not zero-V-box free can be bounded from above by

$$\begin{aligned} n^d C_d V^{\frac{d-1}{d}} \cdot q^{n^d - V} &= q^{n^d} \cdot \frac{n^d C_d V^{\frac{d-1}{d}}}{q^V} \\ &= (q-1)q^{n^d-1} \cdot \frac{C_d V^{\frac{d-1}{d}}}{q^{\frac{d-1}{d}\log_q(\log_q(n))+C}} \\ &\overset{(a)}{\leqslant} (q-1)q^{n^d-1} \cdot \frac{C_d((d+1)\log_q(n))^{\frac{d-1}{d}}}{\log_q(n)^{\frac{d-1}{d}} q^C} \\ &\overset{(b)}{\leqslant} (q-1)q^{n^d-1}. \end{aligned} \tag{1}$$

Inequality (a) follows from $V \leqslant (d+1)\log_q(n)$ for n large enough and (b) holds by choosing $C \geqslant \log_q(C_d(d+1)^{\frac{d-1}{d}})$. This accordingly implies that $|\mathcal{CA}_{d,q}(n,V)| \geqslant q^{n^d-1}$. ■

When comparing the result of Lemma 19 with the lower bound derived in Theorem 5 for arrays that are zero-L-cube free, it follows that for the same volume $V = L^d$, the minimal volume required for a redundancy of one symbol in the latter case is smaller by $\Delta = \frac{d-1}{d}\log_q(\log_q(n)) + \mathcal{O}(1)$.

Next, we present an encoding algorithm that uses a single redundancy symbol to encode V-boxes free cubes over $\Sigma_q^{[n]^d}$, for

$$V = \lceil d\log_q(n)\rceil + \left\lceil \frac{d-1}{d}\log_q(\log_q(n))\right\rceil + C + 1,$$

where $C = \lceil \log_q(C_d) + \frac{d-1}{d}\log_q(d+1)\rceil$, i.e., the ceiling of the constant from the proof of Lemma 19. Note that this value of V adds at most four redundancy symbols to the lower bound derived in the proof of Lemma 19. For simplicity, we omit the ceiling notation in the rest of this section.

Algorithm 5 receives a d-dimensional array $W \in \Sigma^{[n]^d \setminus \{\mathbf{0}\}}$ with a single symbol missing, and outputs a cube $X \in \mathcal{CA}_{d,q}(n,V)$. First, we initialize X with W and set 0 at the missing entry to mark the start of the algorithm. Then, we iteratively look for a zero-V-box in X. When such a box is found, we remove it from X, and insert at the beginning of X an encoding of the position and the shape of the box, along with additional 1-bits. Thus, we ensure that the Hamming weight of the square increases and the algorithm eventually terminates.

The insertions and deletions in this algorithm are preformed with granularity of 1, i.e., as a one-dimensional sequence. In particular, at Step 3 we remove from X a box with coordinates A at position $\boldsymbol{u}$ by performing $\boldsymbol{x} = SD(X)$ and removing from $\boldsymbol{x}$ the entry at position $\sum_{i=0}^{d-1} n^i u'_{d-i}$ for every $\boldsymbol{u}' = (u'_1, \ldots, u'_d) \in \boldsymbol{u} + A$. Then at Step 5 we insert a length-$|A|$ vector at the beginning of $\boldsymbol{x}$ and retransform it to a cube by $X = MD_{[n]^d}(\boldsymbol{x})$.

Algorithm 5 Zero-V-Box Free Encoding

Input: A d-dimensional array $W \in \Sigma^{[n]^d \setminus \{\mathbf{0}\}}$
Output: A d-dimensional array $X \in \mathcal{CA}_{d,q}(n,V)$

1: Set an array $X \in \Sigma^{[n]^d}$ with $X_{\mathbf{0}} = 0, X_{[n]^d \setminus \{\mathbf{0}\}} = W$
2: **while** there exists a zero box $X_{\boldsymbol{u}+A} = \mathbf{0}$ where $A \in F_d(V)$ **do**
3: Remove box $X_{\boldsymbol{u}+A}$
4: Set $\boldsymbol{v} = 1 \circ B_{[n]^d}(\boldsymbol{u}) \circ b_{F_d(V)}(A)$
5: Insert $\boldsymbol{v} \circ 1^{|A|-|\boldsymbol{v}|}$ at the start of X
6: **end while**
7: Return X

Lemma 20: Algorithm 5 successfully outputs a d-dimensional array that satisfies the zero-V-box free constraint.

Proof: First, the assignment in Step 5 is correctly defined since on one hand,

$$|A| \geqslant V = d\log_q(n) + \frac{d-1}{d}\log_q(\log_q(n)) + C + 1,$$

and on the other hand,

$$\begin{aligned} |\boldsymbol{v}| &\leqslant d\log_q(n) + \log_q(f_d(V)) + 1 \\ &\leqslant d\log_q(n) + \log_q(C_d V^{\frac{d-1}{d}}) + 1 \\ &= d\log_q(n) + \log_q(C_d) + \frac{d-1}{d}\log_q(V) + 1 \\ &\leqslant d\log_q(n) + \log_q(C_d) + \frac{d-1}{d}\log_q((d+1)\log_q(n)) + 1 \\ &\leqslant d\log_q(n) + \frac{d-1}{d}\log_q(\log_q(n)) + C + 1 \\ &= V. \end{aligned}$$

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

Thus, it follows that throughout the while loop of the algorithm the size of X remains exactly n^d. Since we remove a box of zeros at Step 3 and insert data with Hamming weight of at least 1 at Step 5, the Hamming weight of X increases at every iteration. Therefore, the while loop eventually stops and the algorithm reaches Step 7.

Next, assume in the contrary that X, which is returned in Step 7, is not zero-V-box free. Thus, X contains a zero box at position $\boldsymbol{u}$ and a coordinates set $A \in F_d(V)$ which contradicts the condition of the loop in Step 2. ■

The decoder reconstructs $W \in \Sigma^{[n]^d \setminus \{\mathbf{0}\}}$ from X, an output of Algorithm 1, by inverting the encoding loop. Note that at Step 1 we initialized $X_{\mathbf{0}}$ with 0 while at Step 5 we set $X_{\mathbf{0}} = 1$ since $v_1 = 1$. Hence, we execute the following procedure, described in Algorithm 6.

Algorithm 6 Zero-V-Box Free Decoding

1: **while** $X_{\mathbf{0}} = 1$ **do**
2: Extract $\boldsymbol{u}, A$ from the $(d\log_q(n) + \log_q(f_d(V)) + 1)$-prefix of $SD(X)$
3: Remove $|A|$ entries from the start of X
4: Insert zero rectangle $X_{\boldsymbol{u}+A} = \mathbf{0}$
5: **end while**
6: Return $W = X_{[n]^d \setminus \{\mathbf{0}\}}$

C. The Boxes Unique Constraint

First, we use a union bound argument to derive a lower bound for V such that the redundancy of the set of V-boxes unique arrays over $\Sigma_q^{[n]^d}$ is at most 1.

Lemma 21: For $V = 2d\log_q(n) + \frac{d-1}{d}\log_q(\log_q(n)) + \mathcal{O}(1)$, and for n large enough it holds that $|\mathcal{DA}_{d,q}(n,V)| \geqslant q^{n^d-1}$. That is, *red*$(\mathcal{DA}_{d,q}(n,V)) \leqslant 1$.

Proof: Let $V = 2d\log_q(n) + \frac{d-1}{d}\log_q(\log_q(n)) + C + \log_q(\frac{q}{q-1})$ for a positive constant C that will be determined later. If an array $W \in \Sigma^{[n]^d}$ is not V-boxes unique, then it contains at least two identical boxes with a coordinates set $A \in F_d(V)$. Hence, according to the union bound, the number of arrays that are not V-boxes unique can be bounded from above by

$$\begin{aligned} n^{2d}C_dV^{\frac{d-1}{d}} \cdot q^{n^d-V} &= q^{n^d} \cdot \frac{n^{2d}C_dV^{\frac{d-1}{d}}}{q^V} \\ &= (q-1)q^{n^d-1} \cdot \frac{C_dV^{\frac{d-1}{d}}}{q^{\frac{d-1}{d}\log_q(\log_q(n))+C}} \\ &\overset{(a)}{\leqslant} (q-1)q^{n^d-1} \cdot \frac{C_d((2d+1)\log_q(n))^{\frac{d-1}{d}}}{\log_q(n)^{\frac{d-1}{d}}q^C} \\ &\overset{(b)}{\leqslant} (q-1)q^{n^d-1}, \end{aligned}$$

where inequality (a) follows from $V \leqslant (2d+1)\log_q(n)$ for n large enough and (b) holds for n large enough by choosing a constant $C \geqslant \log_q(C_d(2d+1)^{\frac{d-1}{d}})$. This accordingly implies that $|\mathcal{DA}_{d,q}(n,V)| \geqslant q^{n^d-1}$. ■

When comparing the result of Lemma 21 with the lower bound derived in Theorem 8 for arrays that are L-cubes unique, it follows that for the same volume $V = L^d$, the minimal volume required for a redundancy of one symbol in the latter case is smaller by $\Delta = \frac{d-1}{d}\log_q(\log_q(n)) + \mathcal{O}(1)$. Note that this result of Δ is the same as the one achieved for the comparison of the zero-free constraints.

Next, we present a lower bound on the value of V which guarantees that the asymptotic rate of $\mathcal{DA}_{d,q}(n,V)$ approaches 1 as $n \to \infty$. This is done similarly to the proof of Theorem 8 in [7]. The size of $\mathcal{DA}_{q,d}(n,V)$ will be estimated using a probabilistic approach. Consider the uniform distribution over all length-n sequences, then

$$|\mathcal{DA}_{q,d}(n,V)| = q^{n^d} \cdot Pr(W \in \mathcal{DA}_{q,d}(n,V)).$$

The asymptotic rate of $\mathcal{DA}_{q,d}(n,V)$ is given by

$$\begin{aligned} \mathbb{R}_{q,d}(V) &\triangleq \lim_{n\to\infty} \frac{\log_q(|\mathcal{DA}_{q,d}(n,V)|)}{n^d} \\ &= 1 + \lim_{n\to\infty} \frac{1}{n^d}\log_q(Pr(W \in \mathcal{DA}_{q,d}(n,V))). \end{aligned} \tag{2}$$

Theorem 22: Let n be an integer. For fixed d, and

$$V = ad\log_q(n),$$

with $a > 1$, the asymptotic rate of $\mathcal{DA}_{q,d}(n,V)$ approaches 1 as $n \to \infty$.

We prove Theorem 22 using the asymmetric Loàsz local lemma which was first proved in [8] and is stated next as it appears in [1].

Lemma 23 ([1], Lemma 5.1.1): Let $Y_0, \dots, Y_{m-1}$ be events in the arbitrary probability space. Let $G = (V, E)$ be a graph with $V = [m]$ such that for every $i \in [m]$, the event Y_i is mutually independent of all the events $\{Y_j \mid (i,j) \notin E\}$. Suppose that there are real numbers $\alpha_0, \dots, \alpha_{m-1}$ such that $\alpha_i \in [0,1]$ and for all $i \in [m]$,

$$Pr(Y_i) \leqslant \alpha_i \prod_{(i,j)\in E}(1-\alpha_j).$$

Then, it is satisfied that

$$Pr\left(\bigcap_{i\in[m]} \overline{Y}_i\right) \geqslant \prod_{i\in[m]}(1-\alpha_i)$$

where $\overline{Y}_i$ is the complement of Y_i.

Proof of Theorem 22: Let $X \in \Sigma_q^{[n]^d}$ be a random array in which each coordinate is chosen uniformly and independently over Σ_q. For coordinates $\boldsymbol{u}, \boldsymbol{v} \in [n]^d$ and a set $A \in F_d(V)$ such that $\boldsymbol{u} + A \subseteq [n]^d$ and $\boldsymbol{v} + A \subseteq [n]^d$, we notate $\boldsymbol{z} = (\boldsymbol{u}, \boldsymbol{v}, A)$ and denote $I_{\boldsymbol{z}} = \mathbb{1}(X_{\boldsymbol{v}+A} = X_{\boldsymbol{u}+A})$, the indicator function of the event that the V-boxes with coordinates set A that start at positions $\boldsymbol{u}$ and $\boldsymbol{v}$ are identical. Let

$$\mathcal{Z} = \{(\boldsymbol{u}, \boldsymbol{v}, A) \mid \boldsymbol{u} \neq \boldsymbol{v}, A \in F_d(V), \boldsymbol{u}+A \subseteq [n]^d, \boldsymbol{v}+A \subseteq [n]^d\}$$

be the set of all admissible triples, and notice that we are interested in a lower bound on

$$Pr(W \in \mathcal{DA}_{q,d}(n,V)) = Pr\left(\sum_{\boldsymbol{z}\in\mathcal{Z}} I_{\boldsymbol{z}} = 0\right).$$

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

Note that for every $\boldsymbol{z} \in \mathcal{Z}$ it holds that $Pr(I_{\boldsymbol{z}}) = \frac{1}{q^V}$. Let $\boldsymbol{z}_0 = (\boldsymbol{u}_0, \boldsymbol{v}_0, A_0), \boldsymbol{z}_1 = (\boldsymbol{u}_1, \boldsymbol{v}_1, A_1) \in \mathcal{Z}$. It is clear that if the V-boxes $\boldsymbol{u}_0 + A_0, \boldsymbol{v}_0 + A_0$ do not overlap with $\boldsymbol{u}_1 + A_1$ or $\boldsymbol{v}_1 + A_1$, then the indicators $I_{\boldsymbol{z}_0}, I_{\boldsymbol{z}_1}$ are independent. We use Lemma 23 with a graph $G = (V, E)$ such that $V = \mathcal{Z}$ and there is an edge $\boldsymbol{z}_0 \to \boldsymbol{z}_1$ if at least one of $\boldsymbol{u}_0 + A_0, \boldsymbol{v}_0 + A_0$ overlaps with $\boldsymbol{u}_1 + A_1$ or $\boldsymbol{v}_1 + A_1$. Thus, every $\boldsymbol{z}$

For a given minimal V-box with coordinates set $A_0 = [x_0] \times \cdots \times [x_{d-1}]$ at position $\boldsymbol{u}$, an intersecting minimal V-box with coordinates set $A_1 = [y_0] \times \cdots \times [y_{d-1}]$ can only start at position that belongs to

$$U = \otimes_{i=0}^{d-1} [u_i - y_i + 1, u_i + x_i - 1].$$

The size of U can be bounded from above by

$$|U| = \prod_{i=0}^{d-1} x_i + y_i - 1 \leqslant \prod_{i=0}^{d-1} x_i y_i \leqslant A_0 \cdot A_1 \leqslant 4V^2,$$

where the last inequality holds since the size of every minimal V-box is bounded from above by $2V$. Therefore, since the number of minimal V-boxes is at most $f_d(V) \leqslant c_d V^{\frac{d-1}{d}}$, the number of neighbors of each vertex is bounded from above by

$$2 \cdot 4V^2 \cdot f_d(V) \cdot n^d \leqslant c_d 8 V^{\frac{3d-1}{d}} n^d = c'_d V^{\frac{3d-1}{d}} n^d,$$

where $c'_d = 8c_d$.

We set the numbers $\alpha_{\boldsymbol{z}} = \frac{1}{c'_d V^{\frac{3d-1}{d}} n^d}$ for every $\boldsymbol{z} \in \mathcal{Z}$. It holds that

$$\prod_{(\boldsymbol{z}, \boldsymbol{z}_1) \in E} (1 - \alpha_{\boldsymbol{z}_1}) \geqslant \left(1 - \frac{1}{c'_d V^{\frac{3d-1}{d}} n^d}\right)^{c'_d V^{\frac{3d-1}{d}} n^d} \geqslant \frac{1}{e}$$

for every $\boldsymbol{z} \in \mathcal{Z}$ since the last expression approaches e^{-1} from above as $n \to \infty$. Hence, the condition of the lemma holds since for every $\boldsymbol{z} \in \mathcal{Z}$,

$$Pr(I_{\boldsymbol{z}}) = \frac{1}{q^V} = \frac{1}{n^{ad}} \leqslant \frac{1}{c'_d V^{\frac{3d-1}{d}} n^d} \cdot \frac{1}{e} \leqslant \alpha_{\boldsymbol{z}} \prod_{(\boldsymbol{z}, \boldsymbol{z}_1) \in E} (1 - \alpha_{\boldsymbol{z}_1}),$$

where the first inequality holds since $c'_d V^{\frac{3d-1}{d}} n^d = o(n^{ad})$. By applying Lemma 23 we obtain

$$\begin{aligned} Pr(W \in \mathcal{DA}_{q,d}(n, V)) &\geqslant \prod_{\boldsymbol{z} \in \mathcal{Z}} \left(1 - \frac{1}{c'_d V^{\frac{3d-1}{d}} n^d}\right) \\ &\geqslant \left(1 - \frac{1}{c'_d V^{\frac{3d-1}{d}} n^d}\right)^{c_d V^{\frac{d-1}{d}} n^{2d}}, \end{aligned}$$

since $1 - \frac{1}{c'_d V^{\frac{3d-1}{d}} n^d} \leqslant 1$ and $|\mathcal{Z}| \leqslant f_d(V) n^{2d} \leqslant c_d V^{\frac{d-1}{d}} n^{2d}$.

Moreover, since

$$\left(1 - \frac{1}{c'_d V^{\frac{3d-1}{d}} n^d}\right)^{c_d V^{\frac{d-1}{d}} n^{2d}} \approx \exp\left(-\frac{n^d}{8(ad \log_q(n))^2}\right),$$

it follows that $\frac{1}{n^d} \log_q(Pr(W \in \mathcal{DA}_{q,d}(n, V)))$ approaches 0 as $n \to \infty$. By plugging into (2) we conclude that $\mathbb{R}_{q,d}(V) = 1$. ∎

VI. Redundancy Analysis for the Two-Dimensional Zero-Free Constraints

In this section we revisit the zero-L-cube free constraint and the zero-V-box free constraint for the two-dimensional case. We analyze the redundancy of the set of arrays satisfying these constraints and present lower and upper bounds on the redundancy for both constraints. These bounds give an expression that is asymptotically tight for the redundancy of $\mathcal{C}_{2,q}(n, L)$ when $n - 2L = \Theta(n)$ and the redundancy of $\mathcal{CA}_{2,q}(n, L)$ when $n - 2\sqrt{V} = \Theta(n)$.

A. The Redundancy of the Zero-L-square Free Constraint

The result of this section is summarized in the following theorem.

Theorem 24: There exist constants C_1, C_2 such that for any positive integer n it holds that

$$C_2 \frac{(n - 2L)^2}{q^{L^2}} \leqslant \mathit{red}(\mathcal{C}_{2,q}(n, L)) \leqslant C_1 \frac{n^2}{q^{L^2}}.$$

The proof of Theorem 24 is given by lower and upper bounds proved in Claim 26 and Claim 27, respectively. The next corollary follows immediately.

Corollary 25: Let n, L be integers such that $n - 2L = \Theta(n)$. Then,

$$\mathit{red}(\mathcal{C}_{2,q}(n, L)) = \Theta\left(\frac{n^2}{q^{L^2}}\right).$$

An upper bound on the redundancy of $\mathcal{CA}_{2,q}(n, L)$ is proved in the next claim.

Claim 26: There exists a constant C_1 such that for any integer n it holds that

$$\mathit{red}(\mathcal{C}_{2,q}(n, L)) \leqslant C_1 \frac{n^2}{q^{L^2}}.$$

Proof: Let k be an integer, and let $A_q(k, L)$ denote a set of squares over $\Sigma_q^{[k]^2}$ that contain a zero-$(L/2)$-square in one of its corners, or a zero-$(L, L/2)$-rectangle at its right or left side, or a zero-$(L/2, L)$-rectangle at its upper or bottom side. That is,

$$A_q(k, L) = \left\{ X \;\middle|\; \begin{array}{l} \exists i, j \in \{0, k - \frac{L}{2}\} \text{ s.t. } X_{(i,j)+[\frac{L}{2}]^2} = \mathbf{0} \text{ or} \\ \exists i \in \{0, k - \frac{L}{2}\}, j \in [k - L + 1] \text{ s.t. } X_{(i,j)+[\frac{L}{2}] \times [L]} = \mathbf{0} \text{ or} \\ \exists i \in [k - L + 1], j \in \{0, k - \frac{L}{2}\} \text{ s.t. } X_{(i,j)+[L] \times [\frac{L}{2}]} = \mathbf{0}. \end{array} \right\}.$$

Note that

$$|A_q(k, L)| \leqslant 4kq^{k^2 - \frac{L^2}{2}} + 4\, q^{k^2 - \frac{L^2}{4}}.$$

Next, let $B_q(k, L) = \mathcal{C}_{2,q}(k, L) \setminus A_q(k, L)$. From Lemma 8 we know that for $L \geqslant \sqrt{2 \log_q(k) + \log_q(\frac{q}{q-1})}$, then $|\mathcal{C}_{2,q}(k, L)| \geqslant q^{k^2 - 1}$. This applies that for $k \leqslant q^{\frac{L^2 - \log_q(\frac{q}{q-1})}{2}}$ we have

$$\begin{aligned} |B_q(k, L)| &\geqslant q^{k^2 - 1} - 4kq^{k^2 - \frac{L^2}{2}} - 4\, q^{k^2 - \frac{L^2}{4}} \\ &\geqslant q^{k^2 - 1} \left(1 - \frac{4k}{q^{\frac{L^2}{2} - 1}} - \frac{4}{q^{\frac{L^2}{4} - 1}}\right). \end{aligned} \quad (3)$$

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

We choose $k = q^{\frac{L^2-7}{2}}$ which satisfies (3), and assume w.l.o.g that $n \bmod k = 0$. Let $E_q(n, L)$ denote the set of n-squares that are composed of a grid of n^2/k^2 squares from $B_q(k, L)$. We prove next that $E_q(n, L) \subseteq \mathcal{C}_{2,q}(n, L)$. Assume otherwise that $X \in E_q(n, L)$ contains a corner zero-L-square. It is clear that the zero square is not contained in one of the $B_q(k, L)$ k-squares, and therefore it intersects two or four k-squares. If it intersects two k-squares, one of them must contain a zero-$(L, L/2)$-rectangle at its right of left edge or a zero-$(L/2, L)$-rectangle at its upper or bottom edge, which is a contradiction. Otherwise, the zero-L-square intersects four k-squares and hence one of them contains a zero-$(L/2)$-square which contradicts the assumption as well. Thus,

$$
\begin{aligned}
|\mathcal{C}_{2,q}(n, L)| &\geqslant |B_q(q^{\frac{L^2-7}{2}}, L)|^{\frac{n^2}{q^{L^2-7}}} \\
&= \left(q^{q^{L^2-7}-1}\left(1 - \frac{4q^{\frac{L^2-7}{2}}}{q^{\frac{L^2}{2}-1}} - \frac{4}{q^{\frac{L^2}{4}-1}}\right)\right)^{\frac{n^2}{q^{L^2-7}}} \\
&= q^{n^2} \cdot q^{-\frac{n^2}{q^{L^2-7}}}\left(1 - \frac{4}{q^{2.5}} - \frac{4}{q^{\frac{L^2}{4}-1}}\right)^{\frac{n^2}{q^{L^2-7}}} \\
&= q^{n^2}(q^{-1}(1 - \frac{4}{q^{2.5}}))^{\frac{n^2}{q^{L^2-7}}}\left(1 - \frac{4}{q^{\frac{L^2}{4}-1}(1-4q^{-2.5})}\right)^{\frac{n^2}{q^{L^2-7}}}.
\end{aligned}
$$

It is known that for all $x < -1$, $(1 + \frac{1}{x})^{x+1} < e$. We denote $x = -\frac{q^{\frac{L^2}{4}-1}(1-4q^{-2.5})}{4}$. For $L \geqslant 3$ and $q \geqslant 2$ we have that $x < -1$ and hence

$$
\begin{aligned}
\left(1 - \frac{4}{q^{\frac{L^2}{4}-1}(1-4q^{-2.5})}\right)^{\frac{n^2}{q^{L^2-7}}} &= (1 + \frac{1}{x})^{(x+1)(\frac{n^2}{q^{L^2-7}})/(x+1)} \\
&\overset{(a)}{>} \exp\left((\frac{n^2}{q^{L^2-7}})/(x+1)\right) \\
&\overset{(b)}{=} \exp\left((c_2\frac{n^2}{q^{L^2}})/(-c_1 q^{\frac{L^2}{4}})\right) \\
&= \exp\left(-\frac{c_2}{c_1}\frac{n^2}{q^{\frac{5L^2}{4}}}\right)
\end{aligned}
$$

where (a) follows from $x+1 < 0$ and (b) follows from a choice of appropriate constants c_1, c_2. Finally, let $c_3 = \frac{1}{1-4q^{-2.5}}$ for some constant $c_3 > 0$. We conclude that

$$
|\mathcal{C}_{2,q}(n, L)| \geqslant q^{n^2} \cdot (qc_3)^{-c_2\frac{n^2}{q^{L^2}}} \exp(-\frac{c_2}{c_1}\frac{n^2}{q^{\frac{5L^2}{4}}})
$$

and thus

$$
\mathrm{red}(\mathcal{C}_{2,q}(n, L)) \leqslant c_2(1 + \log_q(c_3))\frac{n^2}{q^{L^2}} + \log_q(e)\frac{c_2}{c_1}\frac{n^2}{q^{\frac{5L^2}{4}}}.
$$

It follows that there exists a constant $C_1 > 0$ such that

$$
\mathrm{red}(\mathcal{C}_{2,q}(n, L)) \leqslant C_1\frac{n^2}{q^{L^2}}. \tag{4}
$$

Note that before constructing $E_q(n, k)$, if $n \bmod k \neq 0$ we can pick $n' = n + (n - (n \bmod k))$ and continue the proof for n' instead of n to receive the result of (4) for n'. Since $n' - n \leqslant c_2 q^{\frac{L^2}{2}}$, this affects only the constant C_1 and the claim statement holds for n. ■

Next, a lower bound on $\mathrm{red}(\mathcal{C}_{2,q}(n, L))$ is given.

Claim 27: There exists a constant C_2 such that for any integer n it holds that

$$
\textit{red}(\mathcal{C}_{2,q}(n, L)) \geqslant C_2\frac{(n-2L)^2}{q^{L^2}}
$$

Proof: Let $D_q(n, L)$ denote the set of n-squares that are constructed from a grid of $\mathcal{C}_{2,q}(2L, L)$ squares, i.e., $2L$-squares that are zero-L-square free. The remained $n^2 - (\lfloor\frac{n}{2L}\rfloor L)^2$ entries are filled with any symbols from Σ_q. We have that $\mathcal{C}_{2,q}(n, L) \subseteq D_q(n, L)$ and hence,

$$
|\mathcal{C}_{2,q}(n, L)| \leqslant |\mathcal{C}_{2,q}(2L, L)|^{(\lfloor\frac{n}{2L}\rfloor)^2} \cdot q^{n^2 - (\lfloor\frac{n}{2L}\rfloor L)^2}.
$$

Let $\beta(L)$ denote the set of $2L$-squares that contain a zero-L-square exactly once. We lower bound $|\beta(L)|$ by placing a zero-L-square at some position (i, j) for $X \in \beta(L)$, and adding redundancy symbols to ensure that no other zero-L-squares exist in X. Assume w.l.o.g that $i, j \in [1, L-1]$, i.e., the zero square is in the middle of X. We set four non-zero symbols at positions $I = \{(i, j-1), (i+L, j), (i+L-1, j+L), (i-1, j+L-1)\}$. For example, let $L = 3, i = j = 1$, and

$$
X = \begin{pmatrix}
? & ? & ? & 1 & ? & ? \\
1 & 0 & 0 & 0 & ? & ? \\
? & 0 & 0 & 0 & ? & ? \\
? & 0 & 0 & 0 & 1 & ? \\
? & 1 & ? & ? & ? & ? \\
? & ? & ? & ? & ? & ?
\end{pmatrix} \in \Sigma_q^{[2L]^2},
$$

where each occurrence of '?' denotes any symbol of Σ_q. Note that besides $X_{(i,j)+L^2}$, any other L-square in X contains one of the coordinates of I and hence $X_{(i,j)+L^2}$ is the only zero-L-square in X. If the zero square is next to one of the sides of X, it is enough to set only the valid positions of I in order to eliminate additional zero-L-squares. Hence,

$$
|\beta(L)| \geqslant (L+1)^2(q-1)^4 q^{3L^2-4}.
$$

Since $\beta(L) \cap \mathcal{C}_{2,q}(2L, L) = \emptyset$ we can write

$$
\begin{aligned}
|\mathcal{C}_{2,q}(2L, L)| &\leqslant q^{4L^2} - L^2(q-1)^4 q^{3L^2-4} \\
&= q^{4L^2}\left(1 - \frac{L^2(q-1)^4}{q^{L^2+4}}\right)
\end{aligned}
$$

and by combining the inequalities we have

$$
\begin{aligned}
|\mathcal{C}_{2,q}(n, L)| &\leqslant \left(q^{4L^2}\left(1 - \frac{L^2(q-1)^4}{q^{L^2+4}}\right)\right)^{(\lfloor\frac{n}{2L}\rfloor)^2} \cdot q^{n^2 - (\lfloor\frac{n}{2L}\rfloor L)^2} \\
&= q^{n^2}\left(1 - \frac{L^2(q-1)^4}{q^{L^2+4}}\right)^{(\lfloor\frac{n}{2L}\rfloor)^2} \\
&\leqslant q^{n^2}\left(\exp(-\frac{L^2(q-1)^4}{q^{L^2+4}})\right)^{(\lfloor\frac{n}{2L}\rfloor)^2} \\
&\overset{(a)}{\leqslant} q^{n^2 - \log_q(e)\frac{L^2(q-1)^4}{q^{L^2+4}} \cdot (\frac{n}{2L}-1)^2} \\
&\leqslant q^{n^2 - \log_q(e)\frac{(q-1)^4(n-2L)^2}{4q^{L^2+4}}},
\end{aligned}
$$

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

where (a) follows from the inequality $(1-x) < e^{-x}$ for all x. By denoting $C_2 = \frac{\log_q(e)(q-1)^4}{4q^4}$ we can conclude that

$$\text{red}(\mathcal{C}_{2,q}(n,L)) \geqslant C_2 \frac{(n-2L)^2}{q^{L^2}}.$$

■

B. The Redundancy of the Zero-V-Box Free Constraint

Next, we present tight bounds on the cardinality of $\mathcal{CA}_{2,q}(n,V)$. These bounds use similar methods to those presented in Section VI-A. Nonetheless, we introduce improvements and adaptations to those methods in order to fit the constraint where the zero sub-arrays are rectangles and only their area is known. The main result is summarized in the next theorem.

Theorem 28: There exist constants C'_1, C'_2 such that for any integer n it holds that

$$C'_2 \frac{(n-2\sqrt{V})^2}{q^{V-\log_q(V)}} \leqslant \text{red}(\mathcal{CA}_{2,q}(n,V)) \leqslant C'_1 \frac{n^2}{q^{V-\log_q(V)}}.$$

The proof of Theorem 28 is given by lower and upper bounds proved in Claim 30 and Claim 31, respectively. The next corollary follows immediately.

Corollary 29: Let n, V be integers such that $n - 2\sqrt{V} = \Theta(n)$. Then,

$$\textit{red}(\mathcal{CA}_{2,q}(n,V)) = \Theta\left(\frac{n^2}{q^{V-\log_q(V)}}\right).$$

Claim 30: There exists a constant C'_1 such that for any integer n it holds that

$$\textit{red}(\mathcal{CA}_{2,q}(n,V)) \leqslant C'_1 \frac{n^2}{q^{V-\log_q(V)}}.$$

Proof: Let k be an integer, and let $A_q(k,V)$ denote a set of squares over $\Sigma_q^{[k]^2}$ that contain a zero-$(V/4)$-rectangle at one of its corners, or a zero-$(V/2)$-rectangle at any of its sides. The following upper bound holds for the size of the set $A_q(k,V)$.

$$\begin{aligned} |A_q(k,V)| &\leqslant 4k f_2(\frac{V}{2}) q^{k^2-\frac{V}{2}} + 4\ f_2(\frac{V}{4}) q^{k^2-\frac{V}{4}} \\ &\leqslant 4k\sqrt{2V} q^{k^2-\frac{V}{2}} + 4\sqrt{V} q^{k^2-\frac{V}{4}}. \end{aligned}$$

Let $B_q(k,V) = \mathcal{CA}_{2,q}(k,V) \setminus A_q(k,V)$. From Lemma 19, if k satisfies

$$k \leqslant q^{\frac{1}{2}(V-\frac{1}{2}\log_q(V)+\log_q(\frac{q-1}{2q}))}, \tag{5}$$

then $\mathcal{CA}_{2,q}(k,V) \geqslant q^{k^2-1}$ and therefore,

$$\begin{aligned} |B_q(k,V)| &\geqslant q^{k^2-1} - 4k\sqrt{2V} q^{k^2-\frac{V}{2}} - 4\sqrt{V} q^{k^2-\frac{V}{4}} \\ &\geqslant q^{k^2-1}\left(1 - \frac{4k\sqrt{2V}}{q^{\frac{V}{2}-1}} - \frac{4\sqrt{V}}{q^{\frac{V}{4}-1}}\right). \end{aligned}$$

We pick $k = q^{\frac{V}{2}-\frac{1}{2}\log_q(V)-4}$ which satisfies equation (5), and assume w.l.o.g that $n \bmod k = 0$.

Next, we construct $E_q(n,V)$, which is the set of n-squares that are composed of a grid of n^2/k^2 squares from $B_q(k,V)$. It can be shown similarly to the proof of Claim 26 that $E_q(n,V) \subseteq \mathcal{C}_{2,q}(n,V)$ and thus,

$$\begin{aligned} &|\mathcal{CA}_{2,q}(n,V)| \geqslant |B_q(k,V)|^{\frac{n^2}{k^2}} \\ &= |B_q(q^{\frac{V}{2}-\frac{1}{2}\log_q(V)-4}, V)|^{\frac{n^2}{q^{V-\log_q(V)-8}}} \\ &\geqslant \left(q^{q^{V-\log_q(V)-8}-1}\left(1 - \frac{4\sqrt{2V}q^{\frac{V}{2}-\frac{1}{2}\log_q(V)-4}}{q^{\frac{V}{2}-1}} - \frac{4\sqrt{V}}{q^{\frac{V}{4}-1}}\right)\right)^{\frac{Vn^2}{q^{V-8}}} \\ &= q^{n^2} q^{-\frac{Vn^2}{q^{V-8}}}\left(1 - \frac{4\sqrt{2}}{q^3} - \frac{\sqrt{V}}{q^{\frac{V}{4}-1}}\right)^{\frac{Vn^2}{q^{V-8}}} \\ &= q^{n^2}(q^{-1}(1-\frac{4\sqrt{2}}{q^3}))^{\frac{Vn^2}{q^{V-8}}}\left(1 - \frac{\sqrt{V}}{q^{\frac{V}{4}-1}(1-\frac{4\sqrt{2}}{q^3})}\right)^{\frac{Vn^2}{q^{V-8}}}. \end{aligned}$$

It is known that for all $x < -1$, $(1+\frac{1}{x})^{x+1} < e$. We denote

$$x = -\frac{q^{\frac{V}{4}-1}(1-\frac{4\sqrt{2}}{q^3})}{\sqrt{V}}.$$

For $V \geqslant 2^5$ and $q \geqslant 2$ we have that $x < -1$ and hence

$$\begin{aligned} &\left(1 - \frac{\sqrt{V}}{q^{\frac{V}{4}-1}(1-\frac{4\sqrt{2}}{q^3})}\right)^{\frac{Vn^2}{q^{V-8}}} \\ &= (1+\frac{1}{x})^{(x+1)(\frac{Vn^2}{q^{V-8}})/(x+1)} \\ &\overset{(a)}{>} \exp\left((\frac{n^2}{q^{V-\log_q(V)-8}})/(x+1)\right) \\ &\overset{(b)}{=} \exp\left((c_2\frac{n^2}{q^{V-\log_q(V)}})/(-c_1 q^{\frac{V}{4}-\frac{1}{2}\log_q(V)})\right) \\ &= \exp\left(-\frac{c_2}{c_1}\frac{n^2}{q^{\frac{5V}{4}-\frac{3}{2}\log_q(V)}}\right), \end{aligned}$$

where (a) follows from $x+1 < 0$ and (b) follows from a choice of appropriate constants $c_1, c_2 > 0$. Finally, we denote $c_3 = (1-\frac{4\sqrt{2}}{q^3})^{-1}$ and conclude that

$$|\mathcal{CA}_{2,q}(n,L)| \geqslant q^{n^2}(qc_3)^{-c_2\frac{n^2}{q^{V-\log_q(V)}}} \exp\left(-\frac{c_2}{c_1}\frac{n^2}{q^{\frac{5V}{4}-\frac{3}{2}\log_q(V)}}\right)$$

and thus the redundancy satisfies

$$\begin{aligned} \text{red}(\mathcal{CA}_{2,q}(n,V)) &\leqslant c_2(1+\log_q(c_3))\frac{n^2}{q^{V-\log_q(V)}} \\ &\quad + \log_q(e)\frac{c_2}{c_1}\frac{n^2}{q^{\frac{5V}{4}-\frac{3}{2}\log_q(V)}}. \end{aligned}$$

It follows that there exists a constant $C'_1 > 0$ such that

$$\text{red}(\mathcal{CA}_{2,q}(n,V)) \leqslant C'_1 \frac{n^2}{q^{V-\log_q(V)}}.$$

Similarly to the proof of Claim 26, when $n \bmod k \neq 0$ we can enlarge n to the closest multiple of k, and the claim statement still holds for n. ■

Claim 31: There exists a constant C'_2 such that for any integer n it holds that

$$\textit{red}(\mathcal{CA}_{2,q}(n,V)) \geqslant C'_2 \frac{(n-2\sqrt{V})^2}{q^{V-\log_q(V)}}.$$

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

Proof: Let $D_q(n,V)$ denote the set of n-squares that are constructed from a grid of $\mathcal{CA}_{2,q}(2\sqrt{V},V)$ squares, i.e., $2\sqrt{V}$-squares that are V-rectangles free, where the remained entries are filled with any symbols from Σ_q. We have that $\mathcal{CA}_{2,q}(n,V) \subseteq D_q(n,V)$ and therefore,

$$|\mathcal{CA}_{2,q}(n,V)| \leqslant |\mathcal{CA}_{2,q}(2\sqrt{V},V)|^{(\lfloor \frac{n}{2\sqrt{V}} \rfloor)^2} q^{n^2-(\lfloor \frac{n}{2\sqrt{V}} \rfloor)^2 4V} \tag{6}$$

Let $\beta(V)$ denote the set of $(2\sqrt{V})$-squares that contain a exactly one zero-V-rectangle. Similarly to Claim 27, we lower bound $\beta(V)$ by placing a zero-V-rectangle in some $X \in \beta(V)$ and using redundancy symbols to ensure that no other zero-V-rectangle exist in X. Let $X_{(i,j)+A}$ be such a zero-V-rectangle, and since X is a $(2\sqrt{V})$-square, the shorter side of A is in the range $[\sqrt{V}/2, \sqrt{V}]$ and therefore there are $\sqrt{V}$ possible options for A. Moreover, there are at most $(\sqrt{V}+1)^2$ different possible options for the indexes i,j.

Next, in order to eliminate additional zero-V-rectangles, it is enough to ensure that no additional zero-$(\sqrt{V}/2)$-square exist in X. Assume w.l.o.g that $A = [a]\times[b]$ where $a \geqslant \sqrt{V}/2$, and that the zero square is in the middle of X. By surrounding the zero square with non zero symbols at positions $I_1 = \{(i,j-1),(i+a/2,j-1),(i+a,j),(i+a-1,j+b),(i+a/2-1,j+b)(i-1,j+b-1)\}$, we ensure that no zero-$(\sqrt{V}/2)$-square intersect with $X_{(i,j)+A}$. Additionally, in order to prevent zero-$(\sqrt{V}/2)$-square in the rest of X, we set a non-zero symbol every $\sqrt{V}/2$ entries, as long as those do not intersect with $X_{(i,j)+A}$; that is, $I_2 = ([1,3]\cdot(\sqrt{V}/2))^2 \setminus ((i,j)+A)$. This results in at most $c_1 = |I_1|+|I_2| \leqslant 15$ non-zero symbols. Therefore, we can bound

$$\begin{aligned}|\beta(V)| &\geqslant (\sqrt{V}+1)^2\sqrt{V}q^{3V-c_1}(q-1)^{c_1}\\ &\geqslant V^{1.5}q^{3V-c_1}(q-1)^{c_1}.\end{aligned}$$

Next, we have that $\beta(V)\cap\mathcal{CA}_{2,q}(2\sqrt{V},V)=\emptyset$ and hence,

$$\begin{aligned}|\mathcal{CA}_{2,q}(2\sqrt{V},V)| &\leqslant q^{4V} - V^{1.5}q^{3V-c_1}(q-1)^{c_1}\\ &= q^{4V}\left(1-\frac{V^{1.5}(q-1)^{c_1}}{q^{V+c_1}}\right).\end{aligned} \tag{7}$$

By combining inequalities (6) and (7) we get that $|\mathcal{CA}_{2,q}(n,V)|$ is bounded from above by

$$\begin{aligned}&\left(q^{4V}\left(1-\frac{V^{1.5}(q-1)^{c_1}}{q^{V+c_1}}\right)\right)^{(\lfloor \frac{n}{2\sqrt{V}} \rfloor)^2} q^{n^2-(\lfloor \frac{n}{2\sqrt{V}} \rfloor)^2 4V}\\ &= q^{n^2}\left(1-\frac{V^{1.5}(q-1)^{c_1}}{q^{V+c_1}}\right)^{(\lfloor \frac{n}{2\sqrt{V}} \rfloor)^2}\\ &\leqslant q^{n^2}\exp\left(-\frac{V^{1.5}(q-1)^{c_1}}{q^{V+c_1}}\right)^{(\lfloor \frac{n}{2\sqrt{V}} \rfloor)^2}\\ &\leqslant q^{n^2}\exp\left(\frac{V^{1.5}(q-1)^{c_1}}{q^{V+c_1}}(\frac{n}{2\sqrt{V}}-1)^2\right)\\ &= q^{n^2}\exp\left(\frac{\sqrt{V}(q-1)^{c_1}(n-2\sqrt{V})^2}{4q^{V+c_1}}\right)\\ &= q^{n^2-\log_q(e)\frac{\sqrt{V}(q-1)^{c_1}(n-2\sqrt{V})^2}{4q^{V+c_1}}}\end{aligned}$$

and by denoting $C_2' = \frac{\log_q(e)(q-1)^{c_1}}{4q^{c_1}}$ we have

$$\text{red}(\mathcal{CA}_{2,q}(n,V)) \geqslant C_2'\frac{(n-2\sqrt{V})^2}{q^{V-\log_q(V)}}.$$

■

VII. Conclusion

This paper studied two main families of constraints, referred as the zero-L-cube free constraint and the L-cube unique constraint, for multidimensional arrays that impose conditions on the cubes contained in the array. The paper studied also the extensions of these constraints to the case where the conditions are imposed on sub-arrays that are multidimensional boxes and not necessarily cubes, where only their volume is given as a parameter. For the zero free constraints, we presented a lower bound on the size of the sub-array such that the redundancy of the constraint is at most a single symbol, an efficient encoding algorithm for any dimension that uses a single redundancy symbol and tight bounds on the cardinality of the constraints specifically for the two-dimensional case. As for the cube-unique and box-unique constraints, we presented a lower bound on the size of the sub-array such that the asymptotic rate of the set of valid arrays approaches 1 as $n \to \infty$, as well as conditions for the redundancy to be at most a single symbol. Additionally, we presented an encoder for the two-dimensional L-square unique constraint that uses a single redundancy bit.

References

[1] N. Alon and J. Spencer, *The Probabilistic Method*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000.

[2] A. M. Bruckstein, T. Etzion, R. Giryes, N. Gordon, R. J. Holt, and D. Shuldiner, "Simple and robust binary self-location patterns," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4884–4889, Jul. 2012.

[3] N. G. de Bruijn, "A combinatorial problem," *Koninklijke Nederlandse Akademie Wetenschappen*, vol. 49, pp. 758–764, Jun. 1946.

[4] J. Burns and C. Mitchell, "Coding schemes for two-dimensional position sensing," in *Cryptography and Coding III*. Palo Alto, CA, USA: Hewlett-Packard Laboratories, 1993, pp. 31–66.

[5] B. Chor, C. E. Leiserson, R. L. Rivest, and J. B. Shearer, "An application of number theory to the organization of raster-graphics memory," *J. ACM*, vol. 33, no. 1, pp. 86–104, Jan. 1986.

[6] J. Dai and C. R. Chung, "Touchscreen everywhere: On transferring a normal planar surface to a touch-sensitive display," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1383–1396, Aug. 2014.

[7] O. Elishco, R. Gabrys, E. Yaakobi, and M. Médard, "Repeat-free codes," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 5749–5764, Sep. 2021.

[8] P. Erdos and L. Lovász, "Problems and results on 3-chromatic hypergraphs and some related questions," in *Infinite and Finite Sets*, A. Hajnal, R. Rado, and V. T. Sôs, Eds. Amsterdam, The Netherlands: North Holland, 1975, pp. 609–628.

[9] T. Etzion, "Constructions for perfect maps and pseudorandom arrays," *IEEE Trans. Inf. Theory*, vol. IT-34, no. 5, pp. 1308–1316, Sep. 1988.

[10] T. Etzion and E. Yaakobi, "Error-correction of multidimensional bursts," *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 961–976, Mar. 2009.

[11] R. Gabrys and O. Milenkovic, "Unique reconstruction of coded sequences from multiset substring spectra," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018, pp. 2540–2544.

[12] C. Gu, J. Hong, I. McMichael, R. Saxena, and F. Mok, "Cross-talk-limited storage capacity of volume holographic memory," *J. Opt. Soc. Amer.*, vol. 9, no. 11, pp. 1978–1983, 1992.

[13] J. F. Heanue, M. C. Bashaw, and L. Hesselink, "Volume holographic storage and retrieval of digital data," *Science*, vol. 265, no. 5173, pp. 749–752, Aug. 1994.

[14] V. Horan and B. Stevens, "Locating patterns in the de Bruijn torus," *Discrete Math.*, vol. 339, no. 4, pp. 1274–1282, Apr. 2016.

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.

[15] Y.-C. Hsieh, "Decoding structured light patterns for three-dimensional imaging systems," *Pattern Recognit.*, vol. 34, no. 2, pp. 343–349, Feb. 2001.

[16] G. Hurlbert and G. Isaak, "On the de Bruijn torus problem," *J. Combinat. Theory A*, vol. 64, no. 1, pp. 50–62, Sep. 1993.

[17] M. Levy and E. Yaakobi, "Mutually uncorrelated codes for DNA storage," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3671–3691, Jun. 2019.

[18] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proc. IEEE*, vol. 64, no. 12, pp. 1715–1729, Mar. 1976.

[19] S. Marcovich and E. Yaakobi, "Reconstruction of strings from their substrings spectrum," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4369–4384, Jul. 2021.

[20] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissano, "Structured light using pseudorandom codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 322–327, Mar. 1998.

[21] M. A. Neifeld and M. McDonald, "Error correction for increasing the usable capacity of photorefractive memories," *Opt. Lett.*, vol. 19, no. 18, pp. 1483–1485, 1994.

[22] T. T. Nguyen, K. Cai, H. M. Kiah, K. A. S. Immink, and Y. M. Chee, "Using one redundant bit to construct two-dimensional almost-balanced codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aalto, Finland, Jun. 2022, pp. 3091–3096.

[23] E. Ordentlich and R. M. Roth, "Two-dimensional weight-constrained codes through enumeration bounds," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1292–1301, Jul. 2000.

[24] E. Ordentlich and R. M. Roth, "Low complexity two-dimensional weight-constrained codes," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, pp. 3892–3899, Jun. 2012.

[25] J. Pagès, J. Salvi, C. Collewet, and J. Forest, "Optimised de Bruijn patterns for one-shot shape acquisition," *Image Vis. Comput.*, vol. 23, no. 8, pp. 707–720, Aug. 2005.

[26] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado, "A state of the art in structured light patterns for surface profilometry," *Pattern Recognit.*, vol. 43, no. 8, pp. 2666–2680, Aug. 2010.

[27] E. R. Scheinerman, "Determining planar location via complement-free de Brujin sequences using discrete optical sensors," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 883–889, Mar. 2001.

[28] K. Schouhamer-Immink, *Coding Techniques for Digital Recorders*. Upper Saddle River, NJ, USA: Prentice-Hall, 1991.

[29] A. Sharov and R. M. Roth, "Two-dimensional constrained coding based on tiling," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1800–1807, Apr. 2010.

[30] I. Szentandrási, M. Zachariáš, J. Havel, A. Herout, M. Dubská, and R. Kajan, "Uniform marker fields: Camera localization by orientable de Bruijn tori," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Nov. 2012, pp. 319–320.

[31] I. Tal and R. M. Roth, "Bounds on the rate of 2-D bit-stuffing encoders," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2561–2567, Jun. 2010.

[32] R. Talyansky, T. Etzion, and R. M. Roth, "Efficient code constructions for certain two-dimensional constraints," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 794–799, Mar. 1999.

Sagi Marcovich (Student Member, IEEE) received the B.Sc. degree in software engineering and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2016 and 2021, respectively, where he is currently pursuing the Ph.D. degree with the Computer Science Department. His research interests include algorithms, information theory, and coding theory with applications to DNA-based storage.

Eitan Yaakobi (Senior Member, IEEE) received the B.A. degree in computer science and mathematics and the M.Sc. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California at San Diego, in 2011. from 2011 to 2013, he was a Post-Doctoral Researcher with the Department of Electrical Engineering, California Institute of Technology, and the Center for Memory and Recording Research, University of California at San Diego. Since 2016, he has been with the Center for Memory and Recording Research, University of California at San Diego, and since 2018, he has been with the Institute of Advanced Studies, Technical University of Munich, where he holds a four-year Hans Fischer Fellowship, funded by the German Excellence Initiative and the EU 7th Framework Program. He is currently an Associate Professor at the Computer Science Department, Technion—Israel Institute of Technology. He also holds a courtesy appointment at the Technion's Electrical and Computer Engineering (ECE) Department. His research interests include information and coding theory with applications to non-volatile memories, associative memories, DNA storage, data storage and retrieval, and private information retrieval. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship (2010–2011). He was a recipient of several grants, including the ERC Consolidator Grant. Since 2020, he has been serving as an Associate Editor for Coding and Decoding of IEEE TRANSACTIONS ON INFORMATION THEORY.

Authorized licensed use limited to: Eitan Yaakobi. Downloaded on December 04,2023 at 04:03:42 UTC from IEEE Xplore. Restrictions apply.