Journal of Machine Learning Research 25 (2024) 1-28 Submitted 7/22; Published 9/24

A Statistical Experimental Design Method for Constructing Deterministic Sensing Matrices for Compressed Sensing

Youran Qi YQI28@WISC.EDU
Department of Statistics
University of Wisconsin-Madison
Madison, WI 53706, USA

Xu He HEXU@AMSS.AC.CN
Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing, 100190, China

Tzu-Hsiang Hung THUNG6@WISC.EDU
Department of Statistics
University of Wisconsin-Madison
Madison, WI 53706, USA

Peter Chien PETER.CHIEN@WISC.EDU
Department of Statistics
University of Wisconsin-Madison
Madison, WI 53706, USA

Editor: Ji Zhu

Abstract

Compressed sensing is a signal processing technique used to efficiently acquire and reconstruct signals across various fields, including science, engineering, and business. A critical research challenge in compressed sensing is constructing a sensing matrix with desirable reconstruction properties. For optimal performance, the reconstruction process requires the sensing matrix to have low coherence. Several methods have been proposed to create deterministic sensing matrices. We propose a new statistical method to construct deterministic sensing matrices by intelligently sampling rows of Walsh-Hadamard matrices. Compared to existing methods, our approach yields sensing matrices with lower coherence, accommodates a more flexible number of measurements, and entails lower computational cost.

Keywords: coding theory, coherence, compressed sensing, design of experiment, hadamard matrix, supersaturated design

1. Introduction

The interface between the design of experiments (DOEs), also known as data collection, and data modeling forms the foundation of statistics. This interface manifests in various forms across numerous modern statistical models. Examples include factorial designs for linear models, optimal designs for generalized linear models, space-filling designs for Gaussian process models in computer experiments, and response surface designs for process modeling and optimization. Penalized linear models have garnered significant interest in the field of statistics. This work aims to construct DOEs for compressed sensing, a form of penalized linear model used for acquiring and reconstructing sparse signals in applications such as magnetic resonance imaging, single-pixel cameras, and radar.

The general formulation of DOEs is an optimization problem: selecting a finite number of input configurations among all level combinations according to a desirable criterion. This is not a routine optimization problem that can be readily solved by standard software tools. Addressing this problem requires novel statistical ideas, tools, methods, and algorithms. The foundational tools used in this work include Hadamard matrices, coding theory, and statistical search algorithms. While these tools have traditionally been used for constructing DOEs for linear models, we apply them to the emerging field of compressed sensing.

There are two main categories of DOEs. One consists of model-based optimal designs pioneered by Jack Kiefer and others (Wynn, 1984; Fedorov and Hackl, 2012). The method for low-rank matrix recovery proposed by Mak and Xie (2018) falls into this category. The other consists of model-free DOEs, including *minimum aberration designs* and *supersaturated designs*. In this latter category, tools such as Hadamard matrices (Lin, 1993) and coding theory (Xu, 2005; Xu and Wong, 2007) are used to construct DOEs for standard linear models.

Our proposed DOEs for compressed sensing are related to supersaturated designs and can be viewed as new supersaturated designs for compressed sensing. Supersaturated designs are typical DOEs for penalized linear models. For example, Qi and Chien (2023) used supersaturated designs for a penalized linear model to screen out insignificant factors. Common criteria for selecting supersaturated designs include minimax, $E(s^2)$, $UE(s^2)$, mean square correlation, D-optimality, Bayesian D-optimality, S-optimality, MS-optimality, and others. Extensive research has been conducted to construct supersaturated designs based on these criteria (Shah, 1960; Booth and Cox, 1962; Eccleston and Hedayat, 1974; Lin, 1993; Wu, 1993; Deng and Lin, 1994; Tang and Wu, 1997; Jones et al., 2008; Jones and Majumdar, 2014).

Compared to existing supersaturated designs, our DOEs are related to $UE(s^2)$-optimal supersaturated designs proposed by Jones and Majumdar (2014), particularly in constructing two-level supersaturated designs by sampling the rows of Hadamard matrices and allowing for unbalanced designs. According to Jones and Majumdar (2014), any n rows of a $p \times p$ Hadamard matrix form a $UE(s^2)$-optimal design, termed type T_0 designs in their paper. As we will show below, our proposed DOEs for compressed sensing are formed by particular n rows of a $p \times p$ Hadamard matrix, making our DOEs $UE(s^2)$-optimal as well. Additionally, according to Jones and Majumdar (2014), all type T_0 designs are D-optimal supersaturated designs, a $UE(s^2)$-optimal design is Bayesian D-optimal when the prior variance is sufficiently small, and $UE(s^2)$-optimality is equivalent to S-optimality (Shah, 1960) and MS-optimality (Eccleston and Hedayat, 1974). Thus, as a special type T_0 $UE(s^2)$-optimal design, our design inherits all these advantages, including being $UE(s^2)$-optimal, D-optimal, Bayesian D-optimal for small prior variance, S-optimal, and MS-optimal.

The major difference is that while the $UE(s^2)$-optimal designs randomly sample rows, we intelligently sample the rows of Hadamard matrices using sophisticated tools in coding theory to optimize the *coherence* criterion widely used in compressed sensing. This significant difference makes our DOEs superior to the $UE(s^2)$-optimal designs for compressed sensing. To demonstrate that the coherence of our DOEs is significantly smaller than that of the $UE(s^2)$-optimal designs, we conduct simulations and present the results in Appendix B.

2. Compressed Sensing

A compressed sensing procedure consists of two steps. The first step is to acquire a compressed version of a sparse signal via a small number of measurements. The second step is to reconstruct the original signal from these measurements using a reconstruction algorithm.

Denote the original signal by a p-dimensional vector x with no more than s nonzero entries (i.e., s-sparse). The n-dimensional measurement vector y can be acquired by

$$y = Ax, \tag{1}$$

where A is the $n \times p$ sensing matrix ($n < p$). Without loss of generality, we assume the measurement error is zero. Instead of storing all the values in x and then compressing them to obtain y, we simultaneously sense and compress the signal x to obtain y. This means that in the measurement acquisition stage, x is not observed, and only the measurement vector y is observed and stored, significantly reducing the number of measurements that need to be stored (Eldar and Kutyniok, 2012).

There are two main research tasks in compressed sensing. The first task is developing methods to construct the sensing matrix A to ensure that the measurement vector y captures sufficient information for accurate signal reconstruction. The second task is developing algorithms to reconstruct the original sparse signal x from the measurement vector y given a sensing matrix A. Reconstruction algorithms include Basis Pursuit (Chen et al., 2001) and Iterative Hard Thresholding (Blumensath and Davies, 2009), among others. We focus on the first task.

The following restricted isometry property (RIP) is a popular criterion to evaluate sensing matrices:

Definition 1 *Let A be an $n \times p$ matrix. If there is a constant $0 < \delta_s < 1$ such that*

$$(1 - \delta_s)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_s)\|x\|_2^2$$

holds for any s-sparse signal $x \in \mathbb{R}^p$, then the matrix A satisfies the restricted isometry property (RIP) of order s. The minimum nonnegative integer δ_s is called the restricted isometry constant (RIC) of order s.

The RIP is used to guarantee the exact reconstruction of sparse signals. A sufficient condition for the exact reconstruction of all s-sparse signals is $\delta_{2s} < \sqrt{2} - 1$ (Candes and Tao, 2005).

Coherence is another widely used criterion to evaluate sensing matrices. The coherence of an $n \times p$ matrix A is defined as

$$\mu(A) = \max_{1 \leq i < j \leq p} \frac{|a_i^\top a_j|}{\|a_i\|_2 \|a_j\|_2},$$

where a_i is the ith column of A and $\|\cdot\|_2$ is the ℓ_2-norm (Bourgain et al., 2011). According to Welch (1974), for any $n \times p$ matrix A with $n < p$,

$$\sqrt{\frac{p-n}{n(p-1)}} \leq \mu(A) \leq 1.$$

In compressed sensing, small $\mu(A)$ is preferred. A sufficient condition for the exact reconstruction of all s-sparse signals is $\mu(A) < 1/(2s-1)$ (Eldar and Kutyniok, 2012). Hereinafter, we use coherence as a guiding criterion.

Sensing matrices can be generated randomly or deterministically, and these two classes complement each other. Random sensing matrices can be analyzed probabilistically. For example, random matrices with entries drawn from particular probability distributions satisfy the RIP of order s with high probability (Baraniuk et al., 2008). However, from a practical viewpoint, there is no guarantee that a specific realization of a random matrix will work, and random matrices usually require large storage space. In contrast, the RIP and coherence of deterministic sensing matrices are guaranteed by their constructions, removing variability and significantly reducing uncertainty in decision-making with compressed sensing methods. Moreover, deterministic sensing matrices usually provide significant storage savings because their entries are often integers or they are often sparse matrices.

For these reasons, deterministic sensing matrices are gaining popularity, particularly those based on the coherence criterion. Many methods to construct deterministic sensing matrices have been proposed. Some construction methods are based on finite fields. DeVore (2007) used polynomials over finite fields to construct binary deterministic sensing matrices. Li et al. (2012) generalized DeVore's work by using algebraic curves over finite fields. Wang et al. (2019) provided constructions from optimal codebooks and codes, which generalizes the constructions in DeVore (2007) and Li et al. (2012). Meanwhile, some methods use error-correcting codes. Jafarpour et al. (2008) used the adjacency matrix of an expander graph, obtained from Parvaresh-Vardy codes (Parvaresh and Vardy, 2005), to construct sensing matrices. Howard et al. (2008) used second-order Reed-Muller functions, but their sensing matrices can only be of size $2^m \times 2^{m(m+1)/2}$, which is inflexible. Yu and Zhao (2013) proposed real-valued ternary deterministic sensing matrices using optical orthogonal codes. Additionally, families of sensing matrices, called *equiangular tight frames* (ETFs) (Strohmer and Heath, 2003), achieve the well-known Welch bound (Welch, 1974). Several infinite families of ETFs were given in Sustik et al. (2007) and Fickus et al. (2012). Li and Ge (2014) provided deterministic sensing matrices arising from these near orthogonal systems. Lastly, some construction methods are based on knowledge from domains such as signal processing and lattice theory. Applebaum et al. (2009) used chirps to construct deterministic sensing matrices for Fourier signals. Guo and Liu (2018) constructed deterministic sensing matrices using semi-lattices. We focus on deterministic construction of sensing matrices with real numbers and flexible sizes. The existing deterministic sensing matrices mentioned above often contain complex numbers, have restrictive sizes, and are constructed using non-statistical methods.

Compared to existing deterministic sensing matrices, the sensing matrices constructed from our DOEs have four advantages. First, they have smaller coherence for many values of n and p, leading to better signal reconstruction. Second, our method can construct sensing matrices with any number of rows, offering much more flexibility than existing methods. This flexibility is important in many real applications. Third, they align with the Fast Hadamard Transform technique, enabling the matrix-vector multiplication between a $p \times p$ matrix and a p-dimensional vector to be computed at a lower cost of $\mathcal{O}(p \log p)$ (Wang, 2012). Lastly, the entries in our sensing matrices are ± 1. Compared to floating-point numbers, ± 1 entries require much less storage space and lower computational complexity. Moreover, the corresponding hardware implementation is much easier, especially in the analog domain, where the multiplication between a signal value and a ±1 can be readily implemented as a simple switch.

3. The Proposed Method

The key idea of our method is to intelligently take n rows of the $p \times p$ Walsh Hadamard matrix to form an $n \times p$ ($n < p$) submatrix with small coherence, which will then be used as the sensing matrix in compressed sensing.

When n and p are large, an exhaustive search over all submatrices is computationally prohibitive. Our method overcomes this difficulty in two steps. The first step is to convert the task of finding an $n \times p$ sub-Walsh Hadamard matrix to an equivalent task based on coding theory. The second step is to develop an efficient algorithm to solve the equivalent coding theory task. The details are discussed below.

3.1 Walsh Hadamard Matrix and Binary Linear Code

We establish a connection between Walsh Hadamard matrices and binary linear codes following the well-known link between Hadamard matrices and binary codes (Hedayat et al., 2012).

We first state some definitions. A $p \times p$ matrix H is called a Hadamard matrix if its entries take values -1 and $+1$ and $H^\top H = pI_p$, where I_p is the $p \times p$ identity matrix. The $p \times p$ Walsh Hadamard matrix H_p is a Hadamard matrix recursively constructed by

$$H_p = \begin{pmatrix} H_{p/2} & H_{p/2} \\ H_{p/2} & -H_{p/2} \end{pmatrix},$$

where $p = 2^m$ for some positive integer m and

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The first column and the first row of H_p consist of all 1's. Throughout, let $H_{n,p}$ denote an $n \times p$ submatrix formed by n rows of H_p, where $n < p = 2^m$.

A binary code of length n and size p is a subset of $\{0,1\}^n$ with p elements. Write it as a $p \times n$ binary matrix $C_{p,n} = [c_1, \dots, c_p]^\top$, where each row $c_i \in \{0,1\}^n$ is a codeword. Let $w(c_i)$ denote the number of nonzero entries of c_i, also called the Hamming weight of c_i. The Hamming distance between two codewords c_i and c_j, denoted by $\Delta(c_i, c_j)$, is $w(c_i - c_j)$. Then $d(C_{p,n})$ is defined as the minimum distance of code $C_{p,n}$, i.e., $\min_{i \neq j} \Delta(c_i, c_j)$. A binary linear code is a binary code the codewords of which form a linear space. A binary linear code has full rank if its generator matrix has full row rank. The minimum distance of a full rank binary linear code $C_{p,n}$ equals its minimum nonzero weight, given as (MacWilliams and Sloane, 1977)

$$d(C_{p,n}) = \min_{c \in C_{p,n}, c \neq 0} w(c). \tag{2}$$

The Walsh Hadamard matrix is closely related to a special binary linear code, called the binary first-order Reed-Muller code (MacWilliams and Sloane, 1977). We construct a binary first-order Reed-Muller code[1] $C_{2p,p}$ by

$$C_{2p,p} = \begin{pmatrix} B_p \\ \bar{B}_p \end{pmatrix}, \tag{3}$$

where

$$B_p = \frac{1}{2}(J_{p,p} - H_p^\top), \quad \bar{B}_p = \frac{1}{2}(J_{p,p} + H_p^\top),$$

and hereinafter $J_{p,q}$ denotes a $p \times q$ matrix the entries of which are all 1's (MacWilliams and Sloane, 1977). The construction in (3) indicates a direct one-to-one mapping between the first-order Reed-Muller code $C_{2p,p}$ and the Walsh Hadamard matrix H_p, which holds only for a Walsh Hadamard matrix, but not for a general Hadamard matrix. The same relationship extends to sub-codes and submatrices. For a sub-code $C_{2p,n}$ obtained by taking n columns of $C_{2p,p}$, we have

$$C_{2p,n} = \begin{pmatrix} B_{p,n} \\ \bar{B}_{p,n} \end{pmatrix}, \tag{4}$$

where

$$B_{p,n} = \frac{1}{2}(J_{p,n} - H_{n,p}^\top), \quad \bar{B}_{p,n} = \frac{1}{2}(J_{p,n} + H_{n,p}^\top).$$

We can use (4) to obtain sub-Walsh Hadamard matrices via sub-codes.

The following theorem connects the coherence of a sub-Walsh Hadamard matrix $H_{n,p}$ and the minimum nonzero weight of its corresponding sub first-order Reed-Muller code $C_{2p,n}$.

Theorem 2 *If $\mu(H_{n,p}) < 1$, $C_{2p,n}$ constructed in* (4) *is a full rank binary linear code and*

$$\mu(H_{n,p}) = 1 - \frac{2}{n} \min_{c \in C_{2p,n}, c \neq 0} w(c). \tag{5}$$

Here (5) does not hold when $\mu(H_{n,p}) = 1$, because one can find an $H_{n,p}$ with $\mu(H_{n,p})$ of 1 but the minimum nonzero weight not being 0, using the fact that (2) does not hold for a linear code without full rank.

According to Theorem 2, a sub-code with a large minimum nonzero weight, or equivalently a large minimum distance, leads to a sub-Walsh Hadamard matrix with small coherence. The fact that $C_{2p,n}$ consists of $B_{p,n}$ and $\bar{B}_{p,n}$ stacked together is critical for Theorem 2. For example, consider two codewords of $B_{p,4}$, $c_i = (0,0,1,1)^\top$ and $c_j = (1,1,0,0)^\top$. They have the largest possible distance 4, but the coexistence of these two codewords results in two columns with opposite signs in the sub-Walsh Hadamard matrix $H_{4,p}$, thus yielding the largest possible coherence 1. If we only use $B_{p,4}$ with a large minimum distance, we cannot rule out the coexistence of c_i and c_j since they already have the largest possible distance. However, if we attach $\bar{B}_{p,4}$ under $B_{p,4}$ to obtain $C_{2p,4}$, there will be a pair of codewords $c_i = (0,0,1,1)^\top$ and $\bar{c}_j = (0,0,1,1)^\top$, which has the smallest possible distance 0. This makes it possible to correctly rule out the coexistence of c_i and c_j. Small coherence requires that the distance between any two codewords of $B_{p,n}$ is neither too small nor too large.

[1] It corresponds to a resolution IV two-level fractional factorial design.

Now, our problem is converted into finding a sub first-order Reed-Muller code $C_{2p,n}$ with a large minimum nonzero weight. The computational complexity of finding an $H_{n,p}$ with small coherence is

$$\mathcal{O}\left(\binom{p}{n}np^2\right)$$

since it entails computing the dot product of every two columns for every given sub-Walsh Hadamard matrix. In contrast, the computational complexity of the transferred problem is

$$\mathcal{O}\left(\binom{p}{n}np\right)$$

since it computes the minimum nonzero weight by one pass over the codewords. However, it is still computationally prohibitive. Next, we provide an efficient algorithm in the context of binary linear code to overcome this difficulty.

3.2 Back-Elimination Algorithm

We develop a back-elimination algorithm to intelligently select n columns of the code $C_{2p,p}$. Let D_{-j} be a sub-code removing the column j of a code D and $A_i(D)$ be the number of codewords with weight i in a code D. The algorithm has two versions: the sequential version in Algorithm 1 and the weighted average version in Algorithm 2.

Algorithm 1. Sequential Back-Elimination

```
Input $n$, $p$ $(n < p)$
Set $n_0 = p$ and $D = C_{2p,p}$
While $n_0 > n$
    Compute $A_i(D_{-j})$ for $i = d(D) - 1, \dots, n_0 - d(D)$ and $j = 1, \dots, n_0$
    Select one of the columns $j = j^*$ sequentially minimizing
                    $A_{d(D)-1}(D_{-j}), \dots, A_{n_0-d(D)}(D_{-j})$
    $n_0 \leftarrow n_0 - 1$ and $D \leftarrow D_{-j^*}$
Output $H_{n,p} = J_{n,p} - 2B_{p,n}^\top$, where $B_{p,n}$ consists of the first $p$ rows of $D$
```

Algorithm 1 uses backward elimination to remove $p - n$ columns one by one from the code $C_{2p,p}$ according to the minimum nonzero weight criterion. The resulting $n \times p$ sub-Walsh Hadamard matrix $H_{n,p}$ has guaranteed small coherence and is a suitable deterministic sensing matrix for compressed sensing.

Here are some remarks for Algorithm 1. First, for a tie in selecting j^*, the algorithm picks the column with the largest index. Second, the algorithm obtains the outputs for all integers between n and p with a computational complexity of $\mathcal{O}(p^3)$. These outputs are nested since the columns selected for n are contained in the columns selected for $n + 1$. Third, one can start with any code C_{2p,n_0}, not necessarily $C_{2p,p}$, if C_{2p,n_0} has a relatively large minimum nonzero weight. For example, suppose one has a code C_{2p,n_0} of length n_0 and wants to obtain a code $C_{2p,n}$ of length n, where $n < n_0 \ll p$. Then starting the algorithm with C_{2p,n_0} instead of $C_{2p,p}$ would dramatically reduce the computational cost. If C_{2p,n_0} has a relatively large minimum nonzero weight, then the $C_{2p,n}$ given by the algorithm would have a large minimum nonzero weight as well. Fourth, once computed, the indices of the selected rows can be saved and reused for many compressed sensing problems with no need to rerun the algorithm, which is desirable in practice.

We present the weighted average version of the back-elimination algorithm in Algorithm 2, which simultaneously takes account of all $A_i(D_{-j})$ in deleting a column. The only difference between this version and the previous sequential version is that we now minimize $\tilde{A}(D_{-j}) = \sum_{i=1}^{n_0-1} b_i A_i(D_{-j})$ with n_0 being the length of code D and certain $b_1, \ldots, b_p$ defined below, instead of sequentially minimizing $A_{d(D)-1}(D_{-j}), \ldots, A_{n_0-d(D)}(D_{-j})$. The weighted average version is more computationally expensive but yields smaller coherence for some values of n and p.

Algorithm 2. Weighted Average Back-Elimination

```
Input $n$, $p$ $(n < p)$
Set $b_p = 1$ and $b_{i-1} = b_i(1.5p + i)/i$ for $i = p, \ldots, 2$
Set $n_0 = p$ and $D = C_{2p,p}$
While $n_0 > n$
    $\tilde{A}(D_{-j}) \leftarrow \sum_{i=1}^{n_0-1} b_i A_i(D_{-j})$ for $j = 1, \ldots, n_0$
    Select one of the columns $j = j^*$ minimizing $\tilde{A}(D_{-j})$
    $n_0 \leftarrow n_0 - 1$ and $D \leftarrow D_{-j^*}$
Output $H_{n,p} = J_{n,p} - 2B_{p,n}^\top$, where $B_{p,n}$ consists of the first $p$ rows of $D$
```

Next, we provide examples of sub-Walsh Hadamard matrices $H_{n,p}$ and sub-codes $C_{2p,n}$ constructed by the algorithm. Tables 1 and 2 present results for p = 64, 128, 256, 512, 1024, 2048 and different sampling rates n/p. If Algorithms 1 and 2 yield two submatrices with different coherence, we display the coherence of Algorithm 2 in the parentheses. More results can be found in the Supplementary Materials.

	$p = 64$		$p = 128$		$p = 256$	
n/p	n	$\mu(H_{n,p})$	n	$\mu(H_{n,p})$	n	$\mu(H_{n,p})$
0.80	51	0.098	102	0.078	205	0.054
0.75	48	0.125	96	0.083	192	0.062
0.70	45	0.111	90	0.089 (0.111)	179	0.073
0.65	42	0.143	83	0.108	166	0.084
0.60	38	0.158	77	0.117	154	0.091
0.55	35	0.200	70	0.143	141	0.092
0.50	32	0.250	64	0.156	128	0.109
0.45	29	0.241	58	0.172	115	0.113
0.40	26	0.231	51	0.176	102	0.137
0.35	22	0.273	45	0.200	90	0.156
0.30	19	0.368	38	0.263 (0.211)	77	0.169
0.25	16	0.375	32	0.250	64	0.188
0.20	13	0.385	26	0.308	51	0.216

Table 1: Examples for $p = 64, 128, 256$

	$p = 512$		$p = 1024$		$p = 2048$	
n/p	n	$\mu(H_{n,p})$	n	$\mu(H_{n,p})$	n	$\mu(H_{n,p})$
0.80	410	0.039	819	0.028	1638	0.022
0.75	384	0.042 (0.047)	768	0.031 (0.034)	1536	0.023
0.70	358	0.050	717	0.038	1434	0.028
0.65	333	0.057	666	0.039 (0.042)	1331	0.031
0.60	307	0.068 (0.062)	614	0.046	1229	0.035
0.55	282	0.071	563	0.052	1126	0.037
0.50	256	0.078	512	0.055	1024	0.041
0.45	230	0.087	461	0.063	922	0.046
0.40	205	0.093	410	0.073 (0.068)	819	0.050
0.35	179	0.106	358	0.078	717	0.057
0.30	154	0.130	307	0.088	614	0.065
0.25	128	0.141	256	0.109 (0.102)	512	0.074
0.20	102	0.157 (0.176)	205	0.122	410	0.088

Table 2: Examples for $p = 512, 1024, 2048$

4. Theoretical Analysis

We provide theoretical analysis for our constructed sensing matrices. All proofs are deferred to Appendix A. Our goal is to show how the proposed method makes the minimum distance of the sub-code as large as possible, because (2) and (5) showed that a larger minimum distance of the sub-code leads to smaller coherence of the sensing matrix and Section 2 showed that smaller coherence leads to higher signal reconstruction quality.

Deleting an arbitrary column from a sub-code $C_{2p,n}$ makes the minimum distance either decrease by one or remain the same. Theorem 3 shows that Algorithm 1 will not decrease the minimum distance to zero until $n_0 \leq m$, which means the speed at which the minimum distance decreases is controlled much better than a random deletion algorithm.

Theorem 3 *Let $p = 2^m$. The sub-code $C_{2p,n}$ constructed by Algorithm 1 is a full rank binary linear code for every $n = m+1, \ldots, p-1$ and $d(C_{2p,m+1}) = 1$.*

Theorem 3 guarantees that during the while loop of Algorithm 1, the sub-code has full rank and the connection between coherence and minimum nonzero weight in Theorem 2 holds. More importantly, $d(C_{2p,m+1}) = 1$ indicates that the minimum distance will not decrease to zero until $n_0 \leq m$. This is the best possible decreasing speed, because as shown in the proof of Theorem 3, any sub first-order Reed-Muller code with $n \leq m$ has a minimum distance of exactly zero. The following corollary of Theorem 3 shows that on average Algorithm 1 decreases the minimum distance by one in every two deletions.

Corollary 4 *For Algorithm 1, the decreasing rate of minimum distance is asymptotically 1/2 for large p.*

A stronger result on how Algorithm 1 controls the decreasing speed of minimum distance is given in Theorem 6. First, Lemma 5 gives a simple fact.

Lemma 5 *Suppose that $m+1 < n < p = 2^m$ and $d(C_{2p,n}) = d > 1$. Let $A_d(C_{2p,n}) = a$. If $C_{2p,n-1}$ is a sub-code obtained by deleting a column of $C_{2p,n}$ using Algorithm 1, then $A_{d-1}(C_{2p,n-1}) \leq da/n$.*

Theorem 6 *Suppose t is a positive integer such that $m+t+1 < n < p = 2^m$ and $d(C_{2p,n}) = d > t+1$. Let $A_d(C_{2p,n}) = a$. If $C_{2p,n-t-1}$ is a sub-code obtained by deleting $(t+1)$ columns of $C_{2p,n}$ using Algorithm 1 and*

$$a < \prod_{i=0}^{t} \frac{n-i}{d-i}, \tag{6}$$

then $d(C_{2p,n-t-1}) \geq d-t$.

Theorem 6 indicates that if (6) holds, at least one out of $t+1$ consecutive deletions in the process of Algorithm 1 keeps minimum distance the same. This result is conservative since $d(C_{2p,n-t-1})$ can be much greater than $d-t$. On the contrary, a random deletion algorithm has no such guarantee and decreases the minimum distance much faster. Here is a corollary of Theorem 6.

Corollary 7 *Suppose that t is a positive integer such that $m + t + 1 < n < p = 2^m$ and $d(C_{2p,n}) = d < n/2$. If $C_{2p,n-t-1}$ is a sub-code obtained by deleting $(t+1)$ columns of $C_{2p,n}$ using Algorithm 1 and*

$$p - 1 < \prod_{i=0}^{t} \frac{n-i}{d-i}, \tag{7}$$

then $d(C_{2p,n-t-1}) \geq d - t$.

When $t = m - 1$, (7) holds under the basic assumptions of the corollary, because $p - 1 < p = 2^m = 2^{t+1} < \prod_{i=0}^{t}\{(n-i)/(d-i)\}$. Therefore, Corollary 7 indicates that, as long as $d < n/2$, at least one out of m consecutive deletions in the process of Algorithm 1 keeps minimum distance the same. Moreover, when d/n is smaller, the decreasing speed of minimum distance will be even lower than this.

For Algorithm 1, we provide another lower bound for $d(C_{2p,n})$ in the following theorem.

Theorem 8 *If $C_{2p,n}$ is the sub-code constructed by Algorithm 1, z is a nonnegative integer, and*

$$p - 1 < \prod_{i=0}^{t_y} \frac{p - 1 - \sum_{1\leq j<y}(t_j + 1) - i}{p/2 - 1 - \sum_{1\leq j<y} t_j - i}$$

for all $1 \leq y \leq z$, then $d(C_{2p,n-1-\sum_{1\leq j<z}(t_z+1)}) \geq p/2 - 1 - \sum_{1\leq j<z} t_z$.

Finally, we provide analysis for Algorithm 2 below.

Theorem 9 *If t is an integer such that*

$$\prod_{i=t+1}^{p/2}\left(1 + \frac{1.5p}{i}\right) \geq (2p-2)\prod_{i=n+1}^{p}\left(1 + \frac{1.5p}{i}\right),$$

then $d(C_{2p,n}) \geq t + 1$, where $C_{2p,n}$ is the sub-code constructed by Algorithm 2.

Corollary 10 *If $C_{2p,n}$ is the sub-code constructed by Algorithm 2, then*

$$d(C_{2p,n}) \geq \frac{p}{2} - \frac{2(p-n)}{3} - \frac{m}{2} - \frac{1}{2}.$$

Corollary 10 indicates that, when Algorithm 2 is used, at least one out of three deletions keeps minimum distance the same, which is a stronger guarantee than Algorithm 1.

5. Comparison with Existing Methods

We conduct extensive comparisons with some existing methods to demonstrate the superiority of our method in terms of coherence, probabilities of signal reconstructions and performance of signal reconstructions in real image applications.

5.1 Comparison of Coherence

We compare the coherence of our proposed method with the coherence of four deterministic construction methods proposed by DeVore (2007), Li et al. (2012), Yu and Zhao (2013) and Guo and Liu (2018).

First, we introduce each competing method. For a given prime number q and a positive integer $r < q$, DeVore (2007) constructed an $n \times p$ deterministic sensing matrix A with $n = q^2$, $p = q^{r+1}$ and $\mu(A) = r/q$. For given positive integers r and s, Li et al. (2012) constructed an $n \times p$ deterministic sensing matrix A with $n = 2^r(N_r - 1)$, $p = 2^{rs}$ and $\mu(A) \leq s/(N_r - 1)$, where $n < p$, $s < N_r - 1$, and

$$N_r = \begin{cases} 2^r + 1 & r \equiv 2, 6 \mod 8 \\ 2^r + 1 + 2^{(r+2)/2} & r \equiv 4 \mod 8 \\ 2^r + 1 - 2^{(r+2)/2} & r \equiv 0 \mod 8 \\ 2^r + 1 + 2^{(r+1)/2} & r \equiv 1, 7 \mod 8 \\ 2^r + 1 - 2^{(r+1)/2} & r \equiv 3, 5 \mod 8 \end{cases}.$$

For a given prime number q and a nonnegative integer $\delta \leq q$ with $q + \delta + 1 \equiv 0 \mod 4$, Yu and Zhao (2013) constructed an $n \times p$ deterministic sensing matrix A with $n = q^2 + q + 1$, $p = (q + \delta + 1)n$ and $\mu(A) \leq \max\{1/(q + 1), \delta/(q + 1)\}$. For a given integer $q \geq 2$, Guo and Liu (2018) constructed an $n \times p$ deterministic sensing matrix A with $n = (q^2 + 1)(q + 1)$, $p = (q^2 + 1)(q^2 + q + 1)$ and $\mu(A) = 1/(q + 1)$.

Tables 3 to 6 give the coherence of each of the four methods and the proposed method for some n and p.

			DeVore		Proposed	
q	r	n	p	$\mu(A)$	p	$\mu(H_{n,p})$
3	2	9	27	0.667	32	0.556
5	2	25	125	0.400	128	0.360
5	3	25	625	0.600	1024	0.520
5	4	25	3125	0.800	4096	0.600
7	2	49	343	0.286	512	0.265
7	3	49	2401	0.429	4096	0.388
11	3	121	14641	0.273	16384	0.256

Table 3: Coherence of DeVore (2007)'s method and the proposed method

			Li et al.		Proposed	
r	s	n	p	$\mu(A)$	p	$\mu(H_{n,p})$
2	3	16	64	0.750	64	0.375
3	2	32	64	0.500	64	0.250
3	3	32	512	0.750	512	0.375
4	3	384	4096	0.125	4096	0.104
5	2	768	1024	0.083	1024	0.031

Table 4: Coherence of Li et al. (2012)'s method and the proposed method

			Yu and Zhao		Proposed	
q	δ	n	p	$\mu(A)$	p	$\mu(H_{n,p})$
7	4	57	684	0.500	1024	0.298
11	4	133	2128	0.333	4096	0.203
13	6	183	3660	0.429	4096	0.169
13	10	183	4392	0.714	8192	0.191
17	6	307	7368	0.333	8192	0.140
17	10	307	8596	0.556	16384	0.153
17	14	307	9824	0.778	16384	0.153
19	4	381	9144	0.200	16384	0.134
19	8	381	10668	0.400	16384	0.134
19	12	381	12192	0.600	16384	0.134
19	16	381	13716	0.800	16384	0.134
23	4	553	15484	0.167	16384	0.107

Table 5: Coherence of Yu and Zhao (2013)'s method and the proposed method

		Guo and Liu		Proposed	
q	n	p	$\mu(A)$	p	$\mu(H_{n,p})$
4	85	357	0.200	512	0.176
5	156	806	0.167	1024	0.141
6	259	1591	0.143	2048	0.120
7	400	2850	0.125	4096	0.105
8	585	4745	0.111	8192	0.091
9	820	7462	0.100	8192	0.073
10	1111	11211	0.091	16384	0.068

Table 6: Coherence of Guo and Liu (2018)'s method and the proposed method

As shown in the tables above, the coherence of our method is smaller than that of the four competing methods, even when our sensing matrices have more columns. In addition, given the number of columns p, our method is more flexible as it constructs sensing matrices with any number of rows less than p, whereas the four competing methods all have restrictions on the numbers of rows of the sensing matrices.

5.2 Comparison of Reconstruction Probability

We compare the reconstruction probabilities of our method with those of DeVore (2007)'s method and Guo and Liu (2018)'s method by simulations.

To make our simulations effective, we deviate from the conventional choice of fixing the original signal x to a specific predetermined value. This choice may be potentially unconvincing, as it focuses on a particular instance of the signal. To be more statistical, we randomly generate the original signal x in the simulations, while maintaining fixed values for the parameters p and n. This random approach gives a more comprehensive assessment to capture a broader spectrum of scenarios and provide a better understanding of the performance of different methods under varying signal conditions.

For each sparsity level s, we randomly generate a total of 5000 s-sparse original signals. Then we compress each original signal x by a sensing matrix and use the Orthogonal Matching Pursuit (OMP) algorithm (Pati et al., 1993) to obtain the reconstructed signal $\hat{x}$. Let the successful and unsuccessful reconstructions correspond to the outcomes of 1 and 0 respectively, where success means the Signal-to-Noise Ratio (SNR), calculated as $10\log_{10}(\|x\|^2/\|x-\hat{x}\|^2)$, surpasses a threshold of 100. For each sparsity level s, we compute the reconstruction probability as the mean of the 5000 binary outcomes.

5.2.1 Comparison with DeVore (2007)

Let the dimensions of the original signals and the measurement vectors be 125 and 25, respectively. We follow DeVore's method to construct a sensing matrix with $p = 125$ and $n = 25$, corresponding to $q = 5$ and $r = 2$. We construct our sensing matrix with $p = 128$ and $n = 25$, and append 3 zeros to the end of each original signal to apply our sensing matrix. The probabilities of reconstructions across varying sparsity levels are plotted in Figure 1, where the probabilities of reconstructions of a random Gaussian sensing matrix are provided as a benchmark.

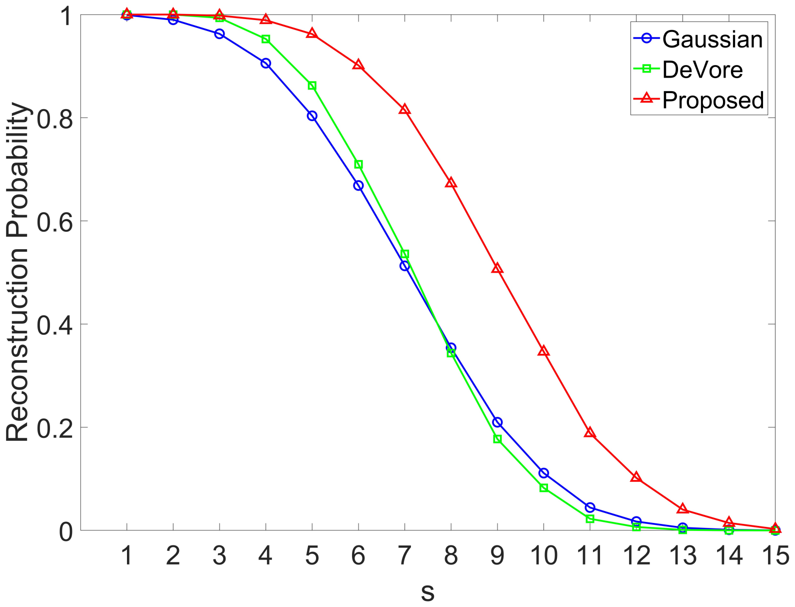


Figure 1: The probabilities of reconstructions of a random Gaussian matrix, a matrix constructed by DeVore's method and a matrix constructed by our method.

Figure 1 shows that for every s, the probability of reconstruction of our sensing matrix is significantly higher than that of DeVore's sensing matrix and the random Gaussian matrix, because of the smaller coherence of our sensing matrix. In addition, DeVore's sensing matrix performs equally well as the random Gaussian matrix but without variability, which demonstrates the superiority of deterministic sensing matrices.

5.2.2 Comparison with Guo and Liu (2018)

Let the dimensions of the original signals and the measurement vectors be 252 and 45 respectively. We follow Guo and Liu's method to construct a sensing matrix with $p = 252$ and $n = 45$ according to Example 3.1 of Guo and Liu (2018), which corresponds to $d = 2$, $l = 5$ and $N = 10$ in the context of their Example 3.1. We construct our sensing matrix with $p = 256$ and $n = 45$, and append 4 zeros to the end of each original signal to apply our sensing matrix. The probabilities of reconstructions across varying sparsity levels are plotted in Figure 2.

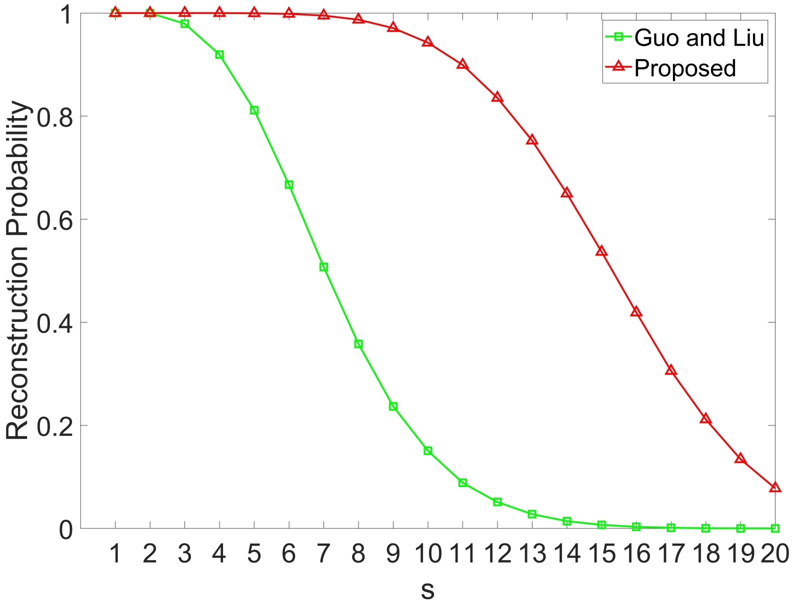


Figure 2: The probabilities of reconstructions of a matrix constructed by Guo and Liu's method and a matrix constructed by our method.

Figure 2 shows a consistent pattern across varying sparsity levels: the probability of reconstruction of our method consistently surpasses that of Guo and Liu's method. This significant difference is due to the smaller coherence of our sensing matrix.

5.3 Real Applications

We compare our method with DeVore (2007)'s method in real image examples. Our setup here follows the standard practice in image processing to provide a meaningful evaluation of the image reconstruction performance.

We use three images *Boat*, *Baboon* and *Peppers*. Each image is represented as a 512×512 matrix of pixels. The Daubechies 9/7 wavelet basis is commonly used in image processing to capture and represent complex image features in a sparse and efficient manner, which balances between localization and frequency representation. We use the Daubechies 9/7 wavelet basis to represent each of the three images as a sparse vector in the wavelet domain. This sparse vector is compressible, which is the original sparse signal we will compress in the next step.

Since the dimension of the original sparse signal $512^2 = 262144$ is large, we use the following block diagonal sensing matrix A to compress it:

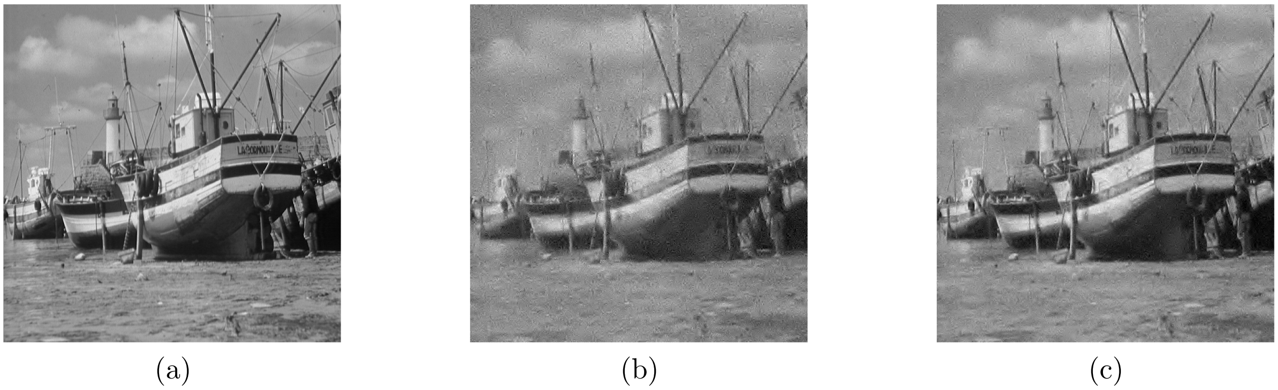
$$A = \begin{bmatrix} M & & & \\ & M & & \\ & & \ddots & \\ & & & M \end{bmatrix},$$

where each block M is an $\ell \times k$ matrix constructed by either DeVore's method or our Algorithm 1, and the remaining entries are all zero. Because $\mu(A) = \mu(M)$ due to this block diagonal structure, the coherence of A remains small. Let $\ell = 9$, $k = 32$ and use 8192 blocks, resulting in a sensing matrix A with $p = 8192 \times 32 = 262144$, $n = 8192 \times 9 = 73728$ and a sampling rate of $73728/262144 \approx 28\%$.

After compressing the original sparse signal by the above block diagonal sensing matrix, we reconstruct the signal by the GPSR software (Figueiredo et al., 2007) and obtain the reconstructed image by applying the inverse wavelet transform on the reconstructed signal. The GPSR software employs gradient-based optimization techniques commonly used for sparse signal recovery in compressed sensing.

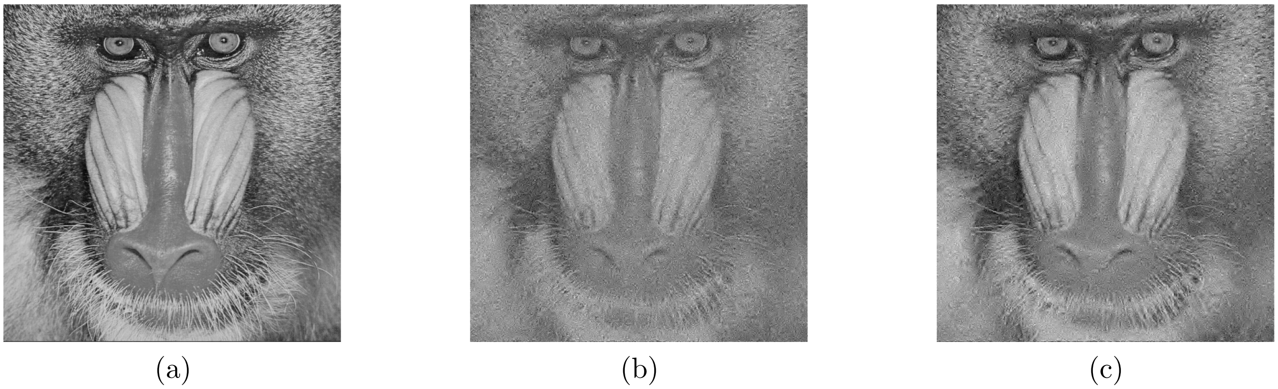
We use the Peak Signal-to-Noise Ratio (PSNR) to evaluate the quality of the reconstructed image. The PSNR is defined as $10 \log_{10}(255^2/\mathrm{MSE}(x, \hat{x}))$, where x is the original image and $\hat{x}$ is the reconstructed image. A higher PSNR indicates a higher quality of the reconstruction.

In Figures 3 to 5, for each image of *Boat*, *Baboon* and *Peppers*, we show the original image and the images reconstructed from the compressions given by DeVore's method and our proposed method. We also provide the PSNRs in the figure captions. For all the three examples, the PSNRs of the images reconstructed from our compressions are significantly higher than those reconstructed from DeVore's compressions. Furthermore, visual inspection of the reconstructed images reveals some significant advantage of our method. The images reconstructed from DeVore's compressions exhibit blurring, which indicates a compromised fidelity, whereas the images reconstructed from our compressions are sharper and more faithful. These findings demonstrate the superiority of our method in preserving the image information during the compression process, and its accuracy and robustness in achieving superior image reconstruction results. In addition, these image reconstruction examples underscore the effectiveness of the block diagonal sensing matrix constructed by our method, particularly in handling scenarios with very large p.



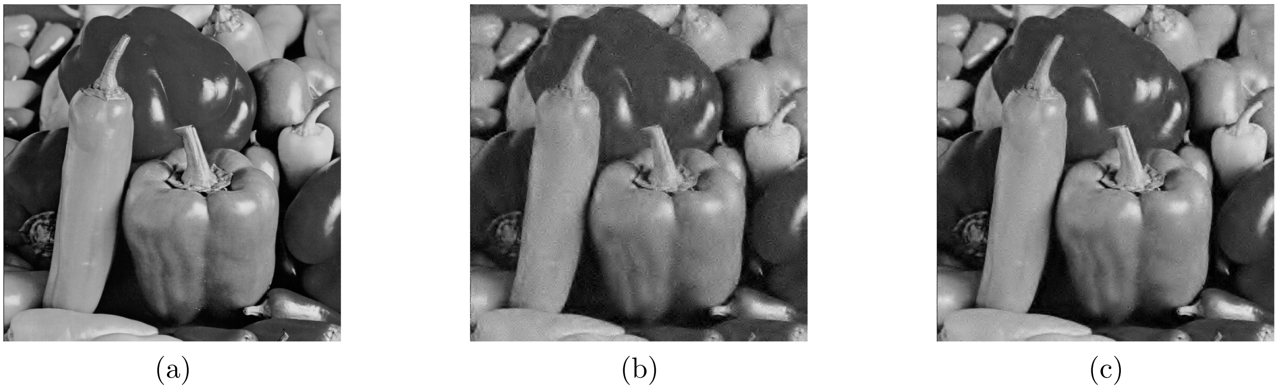
(a) (b) (c)

Figure 3: 512 × 512 Boat image reconstruction. (a) Original Boat image (b) DeVore's method (PSNR: 22.69 dB) (c) Our method (PSNR: 26.42 dB).



(a) (b) (c)

Figure 4: 512 × 512 Baboon image reconstruction. (a) Original Baboon image (b) DeVore's method (PSNR: 18.79 dB) (c) Our method (PSNR: 20.70 dB).



(a) (b) (c)

Figure 5: 512 × 512 Peppers image reconstruction. (a) Original Peppers image (b) DeVore's method (PSNR: 25.88 dB) (c) Our method (PSNR: 29.50 dB).

6. Discussion

We proposed a method for constructing a new class of deterministic compressed sensing matrices by intelligently selecting some rows of a Walsh Hadamard matrix. Compared to existing deterministic sensing matrices, the proposed method constructs sensing matrices with smaller coherence for many values of n and p. It also constructs sensing matrices for any numbers of rows n and columns p with $n < p$. Depending on the required number of columns, one can either choose a subset of columns of our sensing matrices or use a block diagonal version of our sensing matrices, for which the coherence would further decrease or remain the same. Recall that the proposed algorithm has a computational complexity of $\mathcal{O}(p^3)$. It would be computationally expensive for our method to construct a standard non-block diagonal sensing matrix with very large p, e.g., $p = 32768$, 65536. However, as shown in Section 5.3, it is still feasible to apply our method in real applications with very large p, e.g., $p = 262144$ by using a block diagonal sensing matrix. Moreover, our method aligns with the Fast Hadamard Transform technique. In addition, the row indices of the submatrices given by our back-elimination algorithms can be saved and reused without the need to rerun the algorithms. Finally, our sensing matrices, which consist of only ± 1, significantly reduce the cost of storage, computation and hardware implementation. The superiority of our method is demonstrated by numerical simulations and real applications in image reconstruction.

Some possible directions for future work are as follows. First, one can propose a new criterion to delete columns in the back-elimination algorithms. It is of interest to find better criteria to make the algorithms more efficient or yield smaller coherence. Second, one can apply our method to other statistical applications requiring design or model matrices with small coherence.

Our proposed method can also be applied to construct better subsampled Hadamard transforms in other statistical and machine learning problems, including matrix approximation, matrix completion, kernel regression and least square. This line of research sheds new light on how experimental design researchers tackle modern challenges in the big data era. The combined benefits of the new experimental design thinking and the power of large-scale statistical analysis can solve larger and more complex problems.

7. Acknowledgements and Disclosure of Funding

The authors would like to thank the reviewers and Action Editor for useful comments and suggestions that significantly improved the article. Qi, Hung and Chien are supported by U.S. NSF DMS Focus Research Group Grant 1564376. He is supported by National Key R&D Program of China 2021YFA 1000300, 2021YFA 1000301, National Natural Science Foundation of China (NSFC 12022115, NSFC 11971465, NSFC 71988101), and National Center for Mathematics and Interdisciplinary Sciences, CAS

8. Supplementary Materials

Our supplementary materials list the coherence of the sensing matrices constructed by Algorithm 1 for $p = 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384$ and many values of n.

Appendix A. Proofs

Theorem 2 *If $\mu(H_{n,p}) < 1$, $C_{2p,n}$ constructed in* (4) *is a full rank binary linear code and*

$$\mu(H_{n,p}) = 1 - \frac{2}{n} \min_{c \in C_{2p,n}, c \neq 0} w(c). \tag{5}$$

Proof First, since a first-order Reed-Muller code $C_{2p,p}$ in (3) is a binary linear code, $C_{2p,n}$ obtained by taking some columns of $C_{2p,p}$ is binary linear as well.

Second, $\mu(H_{n,p}) < 1$ indicates no two columns of $H_{n,p}$ have completely identical or completely opposite signs. By the construction of $C_{2p,n}$, this implies that $C_{2p,n}$ has no equal codewords. It then follows that the generator matrix of $C_{2p,n}$ has full row rank.

Finally, by Lemma 1 in Cheng and Tang (2001), $h_i^\top h_j = n - 2\Delta(c_i, c_j)$ for any $i \neq j$, where h_i is the ith column of $H_{n,p}$ and c_i is the ith codeword of $B_{p,n}$. Thus,

$$\begin{aligned}
&\mu(H_{n,p}) \\
=&\max_{i \neq j} \frac{|h_i^\top h_j|}{n} \\
=&\max_{i \neq j} \frac{|n - 2\Delta(c_i, c_j)|}{n} \\
=&\max[\max_{i \neq j}\{1 - \frac{2}{n}\Delta(c_i, c_j)\}, -\min_{i \neq j}\{1 - \frac{2}{n}\Delta(c_i, c_j)\}] \\
=&\max\{1 - \frac{2}{n}\min_{i \neq j}\Delta(c_i, c_j), -1 + \frac{2}{n}\max_{i \neq j}\Delta(c_i, c_j)\} \\
=&\max[1 - \frac{2}{n}\min_{i \neq j}\Delta(c_i, c_j), -1 + \frac{2}{n}\max_{i \neq j}\{n - \Delta(c_i, \bar{c}_j)\}] \\
=&\max\{1 - \frac{2}{n}\min_{i \neq j}\Delta(c_i, c_j), 1 - \frac{2}{n}\min_{i \neq j}\Delta(c_i, \bar{c}_j)\} \\
=&1 - \frac{2}{n} d(C_{2p,n}),
\end{aligned}$$

where $\bar{c}_j = J_{n,1} - c_j$ is the counterpart of c_j in $\bar{B}_{p,n}$. Since $C_{2p,n}$ has full rank and is linear,

$$d(C_{2p,n}) = \min_{c \in C_{2p,n}, c \neq 0} w(c),$$

which completes the proof. ■

Theorem 3 *Let $p = 2^m$. The sub-code $C_{2p,n}$ constructed by Algorithm 1 is a full rank binary linear code for every $n = m+1, \ldots, p-1$ and $d(C_{2p,m+1}) = 1$.*

Proof First, since $C_{2p,p}$ is binary linear, the sub-code $C_{2p,n}$ obtained by the algorithm is binary linear as well.

Second, since each step of the algorithm deletes only one column, the minimum distance decreases at most by 1. Hence, there exists an n_0 such that C_{2p,n_0} and C_{2p,n_0-1} constructed by the algorithm have $d(C_{2p,n_0}) = 1$ and $d(C_{2p,n_0-1}) = 0$, respectively. Since C_{2p,n_0} has full rank and is linear, the minimum distance of 1 indicates that it has some rows with weight 1, i.e., $A_1(C_{2p,n_0}) > 0$. Consider the submatrix $C_{2p,n_0}(1)$ consisting of the rows with weight 1. None of its columns can have more than one 1; otherwise it has two equal rows, which contradicts with $d(C_{2p,n_0}) = 1$. None of its columns consists of only 0's; otherwise Algorithm 1 would delete a column consisting of only 0's in $C_{2p,n_0}(1)$ and thus will not decrease the minimum distance to 0, contradicting with the fact $d(C_{2p,n_0-1}) = 0$. Hence, $C_{2p,n_0}(1)$ is an $n_0 \times n_0$ submatrix, each row or column of which has exactly one 1. These n_0 rows form a basis of the linear space $\{0,1\}^{n_0}$. Since C_{2p,n_0} is a linear code, $2^{n_0} \leq 2p$, which means $n_0 \leq m+1$.

Suppose that $n_0 \leq m$. Since any sub-Walsh Hadamard matrix $H_{n_0,p}$ has at most $2^{n_0} \leq 2^m = p$ possible different columns, there will either be two identical columns or two columns with opposite signs, resulting that $d(C_{2p,n_0}) = 0$, which is a contradiction. Therefore, $n_0 = m+1$, implying $d(C_{2p,m+1}) = 1$.

Finally, since the minimum distance will not increase in the process of deleting columns, for every $n = m+1, \ldots, p-1$, $d(C_{2p,n}) \geq d(C_{2p,m+1}) = 1$, which indicates they all have full rank. ■

Corollary 4 *For Algorithm 1, the decreasing rate of minimum distance is asymptotically 1/2 for large p.*

Proof The decreasing rate of minimum distance is the average amount of decrease in minimum distance as n_0 decreases from p to $m+1$. Since $d(C_{2p,p}) = p/2$ (MacWilliams and Sloane, 1977) and $d(C_{2p,m+1}) = 1$, the decreasing rate equals to $(p/2-1)/(p-(\log_2 p+1))$, which approaches to $1/2$ as p goes to infinity. ■

Lemma 5 *Suppose that $m+1 < n < p = 2^m$ and $d(C_{2p,n}) = d > 1$. Let $A_d(C_{2p,n}) = a$. If $C_{2p,n-1}$ is a sub-code obtained by deleting a column of $C_{2p,n}$ using Algorithm 1, then $A_{d-1}(C_{2p,n-1}) \leq da/n$.*

Proof According to Algorithm 1,

$$A_{d-1}(C_{2p,n-1}) = \min\left\{\sum_{i=1}^{a} C_{2p,n}(d)_{i1}, \ldots, \sum_{i=1}^{a} C_{2p,n}(d)_{in}\right\},$$

where $C_{2p,n}(d)$ is the submatrix consisting of the rows with weight d in $C_{2p,n}$, and $C_{2p,n}(d)_{ij}$ is the (i,j)th entry of $C_{2p,n}(d)$. Since

$$\sum_{j=1}^{n}\sum_{i=1}^{a} C_{2p,n}(d)_{ij} = da,$$

we have $A_{d-1}(C_{2p,n-1}) \leq da/n$. ■

Theorem 6 *Suppose t is a positive integer such that $m + t + 1 < n < p = 2^m$ and $d(C_{2p,n}) = d > t + 1$. Let $A_d(C_{2p,n}) = a$. If $C_{2p,n-t-1}$ is a sub-code obtained by deleting $(t + 1)$ columns of $C_{2p,n}$ using Algorithm 1 and*

$$a < \prod_{i=0}^{t} \frac{n-i}{d-i}, \tag{6}$$

then $d(C_{2p,n-t-1}) \geq d - t$.

Proof By Lemma 5 and (6),

$$A_{d-t-1}(C_{2p,n-t-1}) \leq a \prod_{i=0}^{t} \frac{d-i}{n-i} < 1.$$

Hence, $A_{d-t-1}(C_{2p,n-t-1}) = 0$, implying $d(C_{2p,n-t-1}) \geq d - t$. ■

Corollary 7 *Suppose that t is a positive integer such that $m + t + 1 < n < p = 2^m$ and $d(C_{2p,n}) = d < n/2$. If $C_{2p,n-t-1}$ is a sub-code obtained by deleting $(t+1)$ columns of $C_{2p,n}$ using Algorithm 1 and*

$$p - 1 < \prod_{i=0}^{t} \frac{n-i}{d-i}, \tag{7}$$

then $d(C_{2p,n-t-1}) \geq d - t$.

Proof Because $A_0(C_{2p,n}) = A_n(C_{2p,n}) = 1$, $A_d(C_{2p,n}) = A_{n-d}(C_{2p,n})$, $0 < d < n/2 < n-d$, and $\sum_{z=0}^{n} A_z(C_{2p,n}) = 2p$, $A_d(C_{2p,n}) \leq p - 1$. By Theorem 6, $d(C_{2p,n-t-1}) \geq d - t$. ■

Theorem 8 *If $C_{2p,n}$ is the sub-code constructed by Algorithm 1, z is a nonnegative integer, and*

$$p - 1 < \prod_{i=0}^{t_y} \frac{p - 1 - \sum_{1 \leq j < y}(t_j + 1) - i}{p/2 - 1 - \sum_{1 \leq j < y} t_j - i}$$

for all $1 \leq y \leq z$, then $d(C_{2p,n-1-\sum_{1 \leq j < z}(t_z+1)}) \geq p/2 - 1 - \sum_{1 \leq j < z} t_z$.

Proof For $z = 0$, because $d(C_{2p,p}) = p/2$, we have $d(C_{2p,p-1}) \geq p/2 - 1$. Then the result can be proved by performing induction on z and repeatedly applying Corollary 7. ■

Theorem 9 *If t is an integer such that*

$$\prod_{i=t+1}^{p/2} \left(1 + \frac{1.5p}{i}\right) \geq (2p - 2) \prod_{i=n+1}^{p} \left(1 + \frac{1.5p}{i}\right),$$

then $d(C_{2p,n}) \geq t + 1$, where $C_{2p,n}$ is the sub-code constructed by Algorithm 2.

Proof Let $D = C_{2p,n_0}$, the sub-code obtained by deleting $p - n_0$ columns of $C_{2p,p}$. Since

$$\begin{aligned}\sum_{j=1}^{n_0} \tilde{A}(D_{-j}) &= \sum_{j=1}^{n_0} \sum_{i=1}^{n_0-1} b_i A_i(D_{-j}) \\ &\leq \sum_{i=1}^{n_0-1} \{b_i A_i(D)(n_0 - i) + b_{i-1} A_i(D) i\} \\ &= \sum_{i=1}^{n_0-1} \left\{ b_i A_i(D)(n_0 - i) + \left(1 + \frac{1.5p}{i}\right) b_i A_i(D) i \right\} \\ &= \tilde{A}(D)(n_0 + 1.5p),\end{aligned}$$

there exists a j such that

$$\tilde{A}(D_{-j}) \leq (1 + \frac{1.5p}{n_0})\tilde{A}(D).$$

Consequently,

$$\tilde{A}(C_{2p,n_0-1}) \leq \left(1 + \frac{1.5p}{n_0}\right) \tilde{A}(C_{2p,n_0}),$$

where C_{2p,n_0-1} is obtained by deleting a column of C_{2p,n_0} using Algorithm 2. Since $\tilde{A}(C_{2p,p}) = (2p-2)b_{p/2}$,

$$\begin{aligned}\tilde{A}(C_{2p,n}) &\leq \tilde{A}(C_{2p,p}) \prod_{i=n+1}^{p} \left(1 + \frac{1.5p}{i}\right) \\ &= (2p-2) b_{p/2} \prod_{i=n+1}^{p} \left(1 + \frac{1.5p}{i}\right) \\ &= (2p-2) b_t \prod_{i=n+1}^{p} \left(1 + \frac{1.5p}{i}\right) \left\{ \prod_{i=t+1}^{p/2} \left(1 + \frac{1.5p}{i}\right) \right\}^{-1} \\ &\leq b_t,\end{aligned}$$

which implies $d(C_{2p,n}) \geq t + 1$. ■

Corollary 10 *If $C_{2p,n}$ is the sub-code constructed by Algorithm 2, then*

$$d(C_{2p,n}) \geq \frac{p}{2} - \frac{2(p-n)}{3} - \frac{m}{2} - \frac{1}{2}.$$

Proof Since

$$(2p-2r)^2(p-3r)^3 \geq 4(p-r)^3(p-4r)^2 \geq (0.5p-2r)^2(2.5p-3r)^3 \quad (0 \leq r \leq \frac{p}{4}),$$

we have

$$\log \frac{2p-2r}{0.5p-2r} \geq \frac{3}{2} \log \frac{2.5p-3r}{p-3r} \quad (0 \leq r \leq \frac{p}{4}).$$

As a result, for any $r \in [0, p/4]$,

$$\frac{2}{3}\log\frac{1.5p+0.5p-2r-1}{0.5p-2r-1} \geq \log\frac{1.5p+p-3r}{p-3r},$$
$$\frac{1}{3}\log\frac{1.5p+0.5p-2r-1}{0.5p-2r-1} + \frac{1}{3}\log\frac{1.5p+0.5p-2r-2}{0.5p-2r-2} \geq \log\frac{1.5p+p-3r-1}{p-3r-1},$$
$$\frac{2}{3}\log\frac{1.5p+0.5p-2r-2}{0.5p-2r-2} \geq \log\frac{1.5p+p-3r-2}{p-3r-2}.$$

In addition, for any $r \in [0, p/2]$,

$$\frac{1}{2}\log\frac{1.5p+0.5p-r}{0.5p-r} \geq \log 2.$$

We have

$$\log(2p-2) < \log 2^{m+1} = (m+1)\log 2.$$

Therefore, letting $t = \lfloor p/2 - 2(p-n)/3 - m/2 - 1/2 \rfloor$, where $\lfloor x \rfloor$ denotes the largest integer no greater than x, we have

$$\prod_{i=t+1}^{p/2}\left(1+\frac{1.5p}{i}\right) \geq (2p-2)\prod_{i=n+1}^{p}\left(1+\frac{1.5p}{i}\right).$$

Therefore, according to Theorem 9,

$$d(C_{2p,n}) \geq t+1 \geq \frac{p}{2} - \frac{2(p-n)}{3} - \frac{m}{2} - \frac{1}{2},$$

which completes the proof. ■

Appendix B. Comparison with $UE(s^2)$-Optimal Designs

In this section, we compare the coherence of the $UE(s^2)$-optimal designs and the proposed sensing matrices. Since for given n and p, any n rows of the $p \times p$ Walsh Hadamard matrix form a $UE(s^2)$-optimal design, we construct 30 $UE(s^2)$-optimal designs by randomly selecting the n rows, compute the coherence for each of the 30 designs and use the median of the 30 coherence. The proposed sensing matrices are all constructed by Algorithm 1.

For $p = 256, 512, 1024, 2048$ and each n between $0.125p$ and $0.5p$, we plot the median coherence of the $UE(s^2)$-optimal designs and the coherence of the proposed sensing matrices in Figure 6 below. As shown in the figure, the coherence of the proposed sensing matrices is much smaller than the median coherence of the $UE(s^2)$-optimal designs for all p and n.

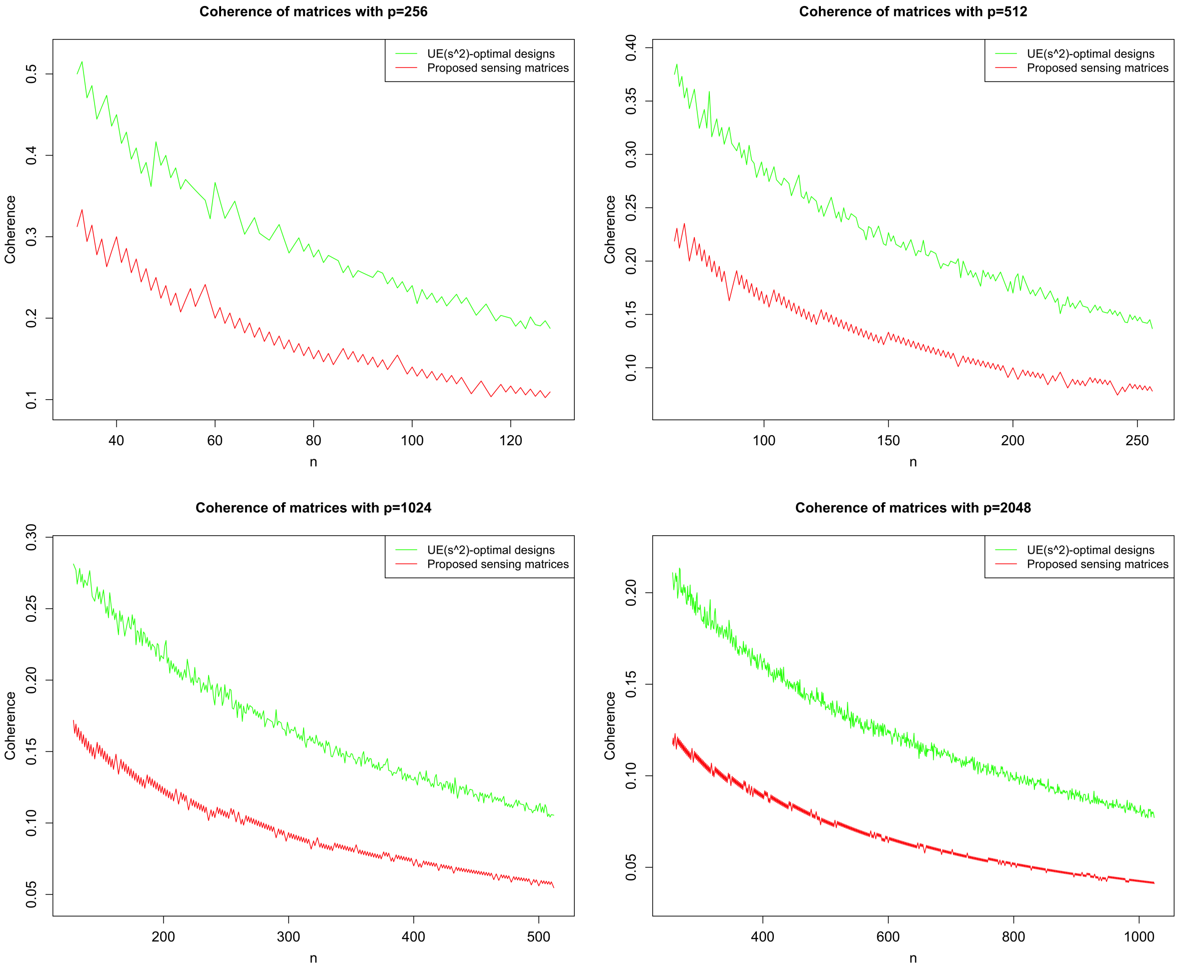


Figure 6: Coherence of the $UE(s^2)$-optimal designs and the proposed sensing matrices for $p = 256, 512, 1024, 2048$

References

Lorne Applebaum, Stephen D Howard, Stephen Searle, and Robert Calderbank. Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery. *Applied and Computational Harmonic Analysis*, 26(2):283–290, 2009.

Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28 (3):253–263, 2008.

Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

Kathleen HV Booth and David R Cox. Some systematic supersaturated designs. *Technometrics*, 4(4):489–495, 1962.

Jean Bourgain, Stephen Dilworth, Kevin Ford, Sergei Konyagin, Denka Kutzarova, et al. Explicit constructions of rip matrices and related problems. *Duke Mathematical Journal*, 159(1):145–185, 2011.

Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

Ching-Shui Cheng and Boxin Tang. Upper bounds on the number of columns in supersaturated designs. *Biometrika*, 88(4):1169–1174, 2001.

Lih-Yuan Deng and Dennis KJ Lin. Criteria for supersaturated designs. In *Proceedings of the Section on Physical and Engineering Sciences, American Statistical Association*, volume 124128, 1994.

Ronald A DeVore. Deterministic constructions of compressed sensing matrices. *Journal of Complexity*, 23(4-6):918–925, 2007.

JA Eccleston and A Hedayat. On the theory of connected designs: characterization and optimality. *The annals of statistics*, pages 1238–1255, 1974.

Yonina C Eldar and Gitta Kutyniok. *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

Valerii V Fedorov and Peter Hackl. *Model-Oriented Design of Experiments*, volume 125. Springer Science & Business Media, 2012.

Matthew Fickus, Dustin G Mixon, and Janet C Tremain. Steiner equiangular tight frames. *Linear algebra and its applications*, 436(5):1014–1027, 2012.

Mário AT Figueiredo, Robert D Nowak, and Stephen J Wright. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.

Jun Guo and Junli Liu. Deterministic construction of compressed sensing matrices based on semilattices. *Journal of Combinatorial Optimization*, 35(1):148–161, 2018.

A Samad Hedayat, Neil James Alexander Sloane, and John Stufken. *Orthogonal Arrays: Theory and Applications*. Springer Science & Business Media, 2012.

Stephen D Howard, A Robert Calderbank, and Stephen J Searle. A fast reconstruction algorithm for deterministic compressive sensing using second order reed-muller codes. In *2008 42nd Annual Conference on Information Sciences and Systems*, pages 11–15. IEEE, 2008.

Sina Jafarpour, Weiyu Xu, Babak Hassibi, and Robert Calderbank. Efficient compressed sensing using high-quality expander graphs. *Preprint*, 2008.

Bradley Jones and Dibyen Majumdar. Optimal supersaturated designs. *Journal of the American Statistical Association*, 109(508):1592–1600, 2014.

Bradley Jones, Dennis KJ Lin, and Christopher J Nachtsheim. Bayesian d-optimal supersaturated designs. *Journal of Statistical Planning and Inference*, 138(1):86–92, 2008.

Shuxing Li and Gennian Ge. Deterministic sensing matrices arising from near orthogonal systems. *IEEE Transactions on Information Theory*, 60(4):2291–2302, 2014.

Shuxing Li, Fei Gao, Gennian Ge, and Shengyuan Zhang. Deterministic construction of compressed sensing matrices via algebraic curves. *IEEE Transactions on Information Theory*, 58(8):5035–5041, 2012.

Dennis KJ Lin. A new class of supersaturated designs. *Technometrics*, 35(1):28–31, 1993.

Florence Jessie MacWilliams and Neil James Alexander Sloane. *The Theory of Error-Correcting Codes*, volume 16). Elsevier, 1977.

Simon Mak and Yao Xie. Maximum entropy low-rank matrix recovery. *IEEE Journal of Selected Topics in Signal Processing*, 12(5):886–901, 2018.

Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the guruswami-sudan radius in polynomial time. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 285–294. IEEE, 2005.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

Youran Qi and Peter Chien. Construction of supersaturated designs with small coherence for variable selection. *The New England Journal of Statistics in Data Science*, 1(3): 323–333, 2023.

KR Shah. Optimality criteria for incomplete block designs. *The Annals of Mathematical Statistics*, pages 791–794, 1960.

Thomas Strohmer and Robert Heath. Grassmannian frames with applications to coding and communication. *Applied and Computational Harmonic Analysis*, 14(3):257–275, 2003.

Mátyás A Sustik, Joel A Tropp, Inderjit S Dhillon, and Robert W Heath Jr. On the existence of equiangular tight frames. *Linear Algebra and its applications*, 426(2-3):619–635, 2007.

Boxin Tang and CFJ Wu. A method for constructing supersaturated designs and its es2 optimality. *Canadian Journal of Statistics*, 25(2):191–201, 1997.

Gang Wang, Min-Yao Niu, and Fang-Wei Fu. Deterministic constructions of compressed sensing matrices based on optimal codebooks and codes. *Applied Mathematics and Computation*, 343:128–136, 2019.

Ruye Wang. *Introduction to Orthogonal Transforms: With Applications in Data Processing and Analysis.* Cambridge University Press, 2012.

Lloyd Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.

CFJ Wu. Construction of supersaturated designs through partially aliased interactions. *Biometrika*, 80(3):661–669, 1993.

Henry P Wynn. Jack Kiefer's Contributions to Experimental Design. *The Annals of Statistics*, 12(2):416–423, 1984.

Hongquan Xu. Some nonregular designs from the nordstrom–robinson code and their statistical properties. *Biometrika*, 92(2):385–397, 2005.

Hongquan Xu and Alan Wong. Two-level nonregular designs from quaternary linear codes. *Statistica Sinica*, pages 1191–1213, 2007.

Nam Yul Yu and Na Zhao. Deterministic construction of real-valued ternary sensing matrices using optical orthogonal codes. *IEEE Signal Processing Letters*, 20(11):1106–1109, 2013.