

Inversion of band-limited discrete Fourier transforms of binary images: Uniqueness and algorithms

Howard W. Levinson*, Vadim A. Markel†, and Nicholas Triantafillou‡

Abstract. Conventional inversion of the discrete Fourier transform (DFT) requires all DFT coefficients to be known. When the DFT coefficients of a rasterized image (represented as a matrix) are known only within a pass band, the original matrix cannot be uniquely recovered. In many cases of practical importance, the matrix is binary and its elements can be reduced to either 0 or 1. This is the case, for example, for the commonly used QR codes. The *a priori* information that the matrix is binary can compensate for the missing high-frequency DFT coefficients and restore uniqueness of image recovery. This paper addresses, both theoretically and numerically, the problem of recovery of blurred images without any known structure whose high-frequency DFT coefficients have been irreversibly lost by utilizing the binarity constraint. We investigate theoretically the smallest band limit for which unique recovery of a generic binary matrix is still possible. Uniqueness results are proved for images of sizes $N_1 \times N_2$, $N_1 \times N_1$, and $N_1^\alpha \times N_1^\alpha$, where $N_1 \neq N_2$ are prime numbers and $\alpha > 1$ an integer. Inversion algorithms are proposed for recovering the matrix from its band-limited (blurred) version. The algorithms combine integer linear programming methods with lattice basis reduction techniques and significantly outperform naive implementations. The algorithm efficiently and reliably reconstructs severely blurred 29×29 binary matrices with only $11 \times 11 = 121$ DFT coefficients.

Published in *SIAM Journal on Imaging Sciences* **16**, 1338-1369 (2023)

doi: [10.1137/22M1540442](https://doi.org/10.1137/22M1540442)

Key words. Two-dimensional discrete Fourier transform, recovery of binary matrices, inversion, deblurring

MSC codes. 94A08, 68U10, 65T50

1. Introduction. The paper address the problem of reconstruction of binary images from limited sets of discrete Fourier transform (DFT) coefficients. We are interested in exact pixel-by-pixel reconstruction of general images without any structure or known properties, i.e., under the conditions when the methods based on machine learning are not expected to be efficient. Images whose DFT coefficients are lost outside of a given pass band are blurred and therefore the problem we are addressing is that of de-blurring. A typical application is de-blurring of QR codes or similar rasterized images in which only two colors are present. Forms such as Data Matrix codes and QR codes are used in applications ranging from industrial tracking to advertising [19]. If the stored information is lost due to a corrupted signal at high frequencies, the results of this paper allow one to recover the original code. Therefore, the main advance reported below is the ability to reconstruct not very large but seemingly random binary images. The paper builds upon our previous results for the one-dimensional case [38], which were, in turn, related to the work of Tao [61], Tropp [63], and the recent work of Pei and Chang [47].

*Department of Mathematics and Computer Science, Santa Clara University, Santa Clara, CA, USA (hlevinson@scu.edu).

†Department of Radiology, University of Pennsylvania, Philadelphia, PA, USA (vmarkel@upenn.edu).

‡Center for Communications Research, Princeton, NJ, USA (ngtriant@alum.mit.edu).

Images are often blurred as a result of low-pass filtering, either due to physical limitations of the image acquisition process [7, 42], or due to application of various filters for image denoising and compression [22, 30]. In either case, DFT coefficients of the blurred image outside of the pass band are below the noise level and, for practical purposes, lost. If no additional information is available, it is, in principle, impossible to recover the image precisely. However, if it is known *a priori* that the original image is binary (contains only two known values), and enough DFT coefficients are known with sufficient precision, we can utilize the binarity constraint to reconstruct all pixels precisely. This is demonstrated below both theoretically in the form of uniqueness theorems and numerically for severely blurred QR codes with the size of up to 29×29 .

Of course, once an image is recovered, we can also compute all of its DFT coefficients, including those that were not known beforehand. We say that, by retrieving the DFT coefficients located outside of the original pass band, we increase the image resolution. If the loss of resolution occurred due to physical limitations of the image acquisition process (such as exponential decay of evanescent waves), and we have recovered the DFT coefficients outside of the physically-imposed pass band, we say that we have achieved the effect of *super-resolution* – that is, we have resolved computationally the details that are not visible directly under the experimental conditions.

In image de-blurring applications, *a priori* information unrelated to the missing DFT coefficients is often available. In such cases, powerful techniques can be developed to achieve recovery of the exact image. Feasibility of achieving super-resolution with meaningful prior information has been demonstrated in many works [43, 45, 53]. A well-studied example is the case of sparse images, which contain relatively few nonzero pixels. It was shown that the knowledge that the original image is sparse allows for stable recovery with severely under-sampled measurements [15, 16, 11]. Corresponding fast reconstruction algorithms have been extensively developed [64, 6, 4, 5]. The sparsity constraint can be independent of the Fourier bases, but there exist many relevant results specific to the Fourier coefficients, including those applicable to random [51, 54] and deterministic measurements [2]. In particular, sparse fast Fourier transform techniques [49, 48] are used to quickly recover sparse vectors, that may or may not have additional known structure. In these problems, however, sampling of high frequency DFT coefficients is required, which are outside the typical pass band considered in this paper. Additional techniques for achieving super-resolution (non-sparsity regularization frameworks) have also been developed, including nonlinear interpolation [50, 31], Laplacian [33, 39] and total variation [3, 55] regularizations.

However, the above techniques rely on assumptions about the images, which limit generality of application and which we wish to avoid in this paper. Instead, we utilize a different, yet still a fundamental constraint. Namely, we consider the case when each pixel of the image can take only two different, *a priori* known values. As was shown in our previous work [38], the problem can be reduced by a simple transformation to that of recovering an image whose pixel values can be either 0 or 1. We say that such images are binary. We will use no additional assumptions on the spatial distribution of zeros and ones, and will be interested in recovering the original image precisely from a limited set of DFT coefficients. Note that, while there exists some overlap between the conditions of binarity and sparsity, a binary image can contain substantially more nonzero entries (roughly half of the total) than a typical sparse

image. In such cases, sparsity-based recovery methods are not efficient.

Binary images and matrices have been extensively studied in the literature, motivated by applications to imaging [14, 41, 52] and combinatorics [18, 9, 10, 60]. Recovering binary images from incomplete data is closely related to the problem of discrete tomography [20, 21, 25, 29]. Here one tries to reconstruct a binary image from families of parallel line integrals (projections) with a small number of specified angles. This mathematical technique has applications to medical imaging [28]. In this paper, we start with DFT coefficients and show that the knowledge of some small sub-sets of such coefficients is similar to the knowledge of some selected projections, except that the line integrals of this paper are periodic in nature, unlike those that arise in discrete tomography. We note that Fourier transforms [65, 66, 67] as well as specific periodic constraints [13] have been previously used in discrete tomography. However, additional prior information is typically used in these applications (such as connectedness) to find a binary image that is physiologically realizable. We do not apply such constraints and consider a more general problem.

The main theoretical question addressed in this paper is the following: how many DFT coefficients are needed to uniquely determine a binary matrix? We assume that the measurements are deterministic and available within a low spatial frequency region (pass band) as defined more precisely below. We will also be interested in recovering the image numerically. However, even if uniqueness is guaranteed, recovery of the exact binary matrix without any known structure is an NP-hard problem [21, 32]. In the most combinatorially challenging regime wherein roughly half of the entries are ones and the rest are zeros, the binary matrix is not sparse. We therefore cannot use the conventional avenues for improving the computational efficiency of recovery. Instead, we solve the inversion problem using integer linear programming and lattice basis reduction techniques. While naive implementations of integer linear programming quickly hit computational roadblocks and are limited to matrices with $\lesssim 50$ entries (i.e., of the size 7×7 or less), we have developed algorithms specifically tailored to the problem at hand. The largest image size for which the algorithm was successfully tested is 29×29 with 841 pixels. We note that our algorithm allows to recover uniquely any of the 2^{841} distinct binary images of this size using only $11 \times 11 = 121$ DFT coefficients.

We use typewriter-style straight letters to denote matrices (as in \mathbf{X}) and vectors (as in \mathbf{x}). Elements of these structures, as well as other scalar quantities, are denoted by italic letters as in X_{nm} or x_n . Fourier transforms are denoted by overhead tilde. For example, $\tilde{\mathbf{X}}$ is a matrix of complex DFT coefficients of \mathbf{X} and \tilde{X}_{kl} is a particular element of $\tilde{\mathbf{X}}$. The greatest common divisor of two integers n and m is denoted by $\gcd(n, m)$, and we let \mathbb{Z}_N denote the ring of integers modulo N .

2. Theoretical background.

2.1. Statement of the inverse problem. Let \mathbf{X} be an $N_1 \times N_2$ matrix, and assume that its entries X_{mn} can take only two values, either 0 or 1. The DFT of \mathbf{X} is given by

$$(1) \quad \tilde{X}_{kl} = \sum_{m=1}^{N_1} \sum_{n=1}^{N_2} X_{mn} e^{2\pi i (mk/N_1 + nl/N_2)} .$$

The DFT coefficients \tilde{X}_{kl} are periodic in each index, so that $\tilde{X}_{kl} = \tilde{X}_{k+N_1, l+N_2}$. Since we will mainly be considering the cases when both N_1 and N_2 are odd, it is sufficient to restrict the indexes k, l to the symmetric intervals

$$(2) \quad -M_1 \leq k \leq M_1, \quad -M_2 \leq l \leq M_2, \quad \text{where } M_1 = (N_1 - 1)/2, \quad M_2 = (N_2 - 1)/2.$$

Then $\tilde{\mathbf{X}}$ is the $N_1 \times N_2$ matrix of DFT coefficients \tilde{X}_{kl} with the indexes restricted by (2). The inverse DFT is defined as

$$(3) \quad X_{mn} = \frac{1}{N_1 N_2} \sum_{k=-M_1}^{M_1} \sum_{l=-M_2}^{M_2} \tilde{X}_{kl} e^{-2\pi i (mk/N_1 + nl/N_2)},$$

which allows for reconstruction of the original matrix \mathbf{X} from the knowledge of $\tilde{\mathbf{X}}$. Generically, if some of the elements of $\tilde{\mathbf{X}}$ are not known, none of the elements of \mathbf{X} can be reconstructed uniquely. Indeed, it can be seen from (3) that changing only one element of $\tilde{\mathbf{X}}$ changes all elements of \mathbf{X} .

However, with the additional constraint that the elements of \mathbf{X} are binary, we can hope to achieve unique inversion from only partial knowledge of $\tilde{\mathbf{X}}$. We will therefore address the following question: is it possible to reconstruct \mathbf{X} precisely from the knowledge of only a proper subset of its DFT coefficients? The precise problem definition is as follows.

Definition 1. We use the acronym $\text{IP}(N_1, N_2, L_1, L_2)$ to denote the inverse problem of reconstructing a generic binary matrix \mathbf{X} of known dimension $N_1 \times N_2$ from the set of its DFT coefficients \tilde{X}_{kl} with indexes restricted by

$$(4) \quad |k| \leq L_1 \leq M_1, \quad |l| \leq L_2 \leq M_2.$$

We refer to two binary matrices \mathbf{X} and \mathbf{Y} as being (L_1, L_2) -indistinguishable if they have the same DFT coefficients within the band (4). In the case $L_1 = L_2 = L$, we use the shorthand “ L -indistinguishable”.

Note that, since \mathbf{X} is real, we have $\tilde{X}_{-k, -l} = \tilde{X}_{k, l}^*$. Consequently, there are $(L_1 + 1)(L_2 + 1) + L_1 L_2$ independent complex coefficients in the band (4), ignoring the pairs that are known conjugates of each other.

The DFT coefficient that is always accessible in this setup is the popcount, $S \equiv \tilde{X}_{00}$, which gives the total number of ones in \mathbf{X} . We thus assume that the value of S is always known. In general, the problem of recovering a binary matrix \mathbf{X} from a limited set of DFT coefficients is most challenging when $S \sim (N_1 N_2)/2$. This is so because the total number of binary matrices with S nonzero entries is given by $\binom{N_1 N_2}{S}$.

2.2. Cyclotomic Integers. One key tool that we will use to determine whether a binary matrix is uniquely recoverable from a certain subset of DFT coefficients is analysis of sums of complex exponentials with integer coefficients. If two binary matrices \mathbf{X} and \mathbf{Y} have the same (k, l) -DFT coefficient, then, by linearity of the DFT, we have $\tilde{Z}_{kl} = 0$, where $\mathbf{Z} = \mathbf{X} - \mathbf{Y}$. Thus, it is useful to know under what conditions a sum of roots of unity can be zero. This problem has been studied extensively. Some relevant results pertaining to the case when the roots of unity are all of the same order are summarized below.

Consider the N -th roots of unity, which are the N solutions to the equation $z^N = 1$. These solutions are of the form $e^{2\pi i k/N}$, $k = 1, \dots, N$. If $\gcd(k, N) = 1$, then $e^{2\pi i k/N}$ is a primitive root of unity, and it is not a solution to the equation $z^M = 1$ for any integer $M < N$. Let ζ_N be a primitive N -th root of unity, and suppose that

$$(5) \quad \sum_{n=1}^N a_n (\zeta_N)^n = 0,$$

where the coefficients a_n are all integers. The sum appearing on the left-hand side of this expression is known as a *cyclotomic integer* – a linear combination of N -th roots of unity with integer coefficients.

First, consider the case when N is a prime number. Since the cyclotomic polynomial $1 + x + x^2 + \dots + x^{p-1}$ is irreducible, the equality (5) can hold only if $a_n = c$ for all $1 \leq n \leq N$, where c is some constant integer (see proof of Theorem 1 of [38], for example). Thus, an important consequence of irreducibility of the cyclotomic polynomial is that, if a cyclotomic integer of prime order is equal to 0, then all of its coefficients are the same constant integer.

Such a strong condition does not hold if N is not prime. However, one can still obtain conditions depending on the prime factors of N . The main result for integer vanishing sums of roots of unity is given by the following two Lemmas as stated in [36].

Lemma 2. Let M be the product of all distinct primes dividing N , and let ζ_M and ζ_N be primitive M -th and N -th roots of unity, respectively. Then $\{(\zeta_M)^m (\zeta_N)^n : 1 \leq m \leq M, 1 \leq n \leq N/M\}$ is the complete set of N -th roots of unity. Moreover, for $a_{mn} \in \mathbb{Z}$, the following equation holds

$$\sum_{m=1}^M \sum_{n=1}^{N/M} a_{mn} (\zeta_M)^m (\zeta_N)^n = 0$$

if and only if

$$\sum_{m=1}^M a_{mn} (\zeta_M)^m = 0 \quad \text{for all } n \text{ such that } 1 \leq n \leq N/M.$$

Lemma 3. Let $N = pM$, where p is prime and does not divide M , and let ζ_M and ζ_p be primitive M -th and p -th roots of unity, respectively. Then $\{(\zeta_M)^m (\zeta_p)^n : 1 \leq m \leq M, 1 \leq n \leq p\}$ is the complete set of N -th roots of unity. Then, for $a_{nm} \in \mathbb{Z}$, the following equality holds

$$\sum_{m=1}^M \sum_{n=1}^p a_{nm} (\zeta_M)^m (\zeta_p)^n = 0$$

if and only if

$$(6) \quad \sum_{m=1}^M a_{nm} (\zeta_M)^m = \sum_{m=1}^M a_{m1} (\zeta_M)^m \quad \text{for all } n \text{ such that } 1 < n \leq p.$$

Lemma 2 is used to analyze roots of unity of order N where N has at least one prime power as a divisor. **Lemma 3** provides a tractable condition when N has only two prime divisors. In this case, M in (6) is prime; therefore, by subtracting the two sums, we have a vanishing cyclotomic integer as in (5). Thus, we can conclude that, for each fixed n , $(a_{mn} - a_{m1})$ is constant for $1 \leq m \leq M$. If N has more than two prime divisors, it is much harder to analyze (6) due to existence of the so-called asymmetrical sums [12, 34].

Building on these ideas, our previous work [38] developed the theory of recovering binary one-dimensional signals from limited sets of DFT coefficients. Results were obtained for vectors of prime length N , and of length of the form $N = pq$ where p and q are two (possibly, equal) prime factors. Two-dimensional binary DFT requires a separate analysis, but some results can be generalized from the one-dimensional setting. We therefore briefly summarize the pertinent one-dimensional theory below.

2.3. Summary of results on binary vectors. For vectors \mathbf{x} of length N , the one-dimensional DFT is defined as

$$(7) \quad \tilde{x}_m = \sum_{n=1}^N x_n e^{2\pi i m n / N} .$$

When \mathbf{x} is known to be a binary vector of prime length p , inversion is unique with the knowledge of the first two DFT coefficients \tilde{x}_0 and \tilde{x}_1 . This is a consequence of the irreducibility of cyclotomic polynomials (see Theorem 1 of [38]). For binary vectors of length pq (where, possibly, $p = q$), the results are more subtle. Many such vectors are uniquely recoverable from only their first two DFT coefficients, but some vectors, which have a special structure, are not. The result is stated below as **Lemma 4**, which was proved in a rephrased form in [38].

Lemma 4. Let \mathbf{x} be a binary vector of length pq , where p and q are (not necessarily distinct) prime numbers. Then \mathbf{x} is not uniquely determined by its DFT coefficients \tilde{x}_0 and \tilde{x}_1 (that is, there exists a distinct vector $\mathbf{y} \neq \mathbf{x}$ with $\tilde{y}_0 = \tilde{x}_0$ and $\tilde{y}_1 = \tilde{x}_1$) if and only if, for $s = p$ or $s = q$, \mathbf{x} has indexes $a, b \in \{1, \dots, pq\}$ such that the following two conditions hold simultaneously:

$$(8) \quad \left\{ x_\alpha = 1 \text{ for all } \alpha = a \pmod{s} \right\} \quad \text{AND} \quad \left\{ x_\beta = 0 \text{ for all } \beta = b \pmod{s} \right\} .$$

Moreover, if \mathbf{x} is not uniquely determined by \tilde{x}_0 and \tilde{x}_1 , then a distinct binary vector \mathbf{y} is 1-indistinguishable from \mathbf{x} if and only if \mathbf{y} satisfies (8) for the same a and b , except for the permutation $0 \leftrightarrow 1$, that is we write $y_\alpha = 0$ and $y_\beta = 1$.

3. Uniqueness results. In this section, we state and prove uniqueness results for binary matrices of the size $N_1 \times N_2$. Due to the complexity associated with the asymmetric sums of roots of unity, we assume below that the total number of pixels, $N_1 N_2$, has no more than two prime divisors. The cases we cover are not exhaustive, but give a taste for the type of super-resolution one can obtain for binary matrices.

3.1. Row- and column-wise popcounts. As previously mentioned, the global popcount (the total number of ones in \mathbf{X}) is given by $S = \tilde{X}_{00}$. We also define the row- and column-wise

popcounts r_m and c_n as

$$(9) \quad r_m = \sum_{n=1}^{N_2} X_{mn}, \quad c_n = \sum_{m=1}^{N_1} X_{mn}.$$

If the dimensions N_1 and N_2 are both prime, the next two lowest-order DFT coefficients of \mathbf{X} fix all r_m and c_n . For example, the coefficient \tilde{X}_{10} is given by

$$(10) \quad \tilde{X}_{10} = \sum_{m=1}^{N_1} r_m e^{2\pi i m / N_1}.$$

The right-hand side of (10) is a cyclotomic integer – a sum of powers of a primitive root of unity with integer coefficients. Assuming that the global popcount S and \tilde{X}_{10} are known, all r_m 's are also known (as the cyclotomic integers are irreducible). This statement is a slight generalization of the result of [38] where we proved that (10) is uniquely invertible for binary r_m ; here we say that it is uniquely invertible for integer r_m . The proof is a trivial extension of the proof given in [38]. Similarly, the knowledge of \tilde{X}_{01} fixes all column-wise popcounts c_n . Note that this geometric equivalence is only true when N_1 and N_2 are prime.

Thus, the knowledge of \tilde{X}_{00} , \tilde{X}_{01} and \tilde{X}_{10} is sufficient to recover the global and the row- and column-wise popcounts assuming N_1 and N_2 are prime. In some special cases, this information defines uniquely the whole binary matrix (a trivial example is when $S = 1$). In general, this is clearly false. The problem of determining a binary matrix by its row- and column-wise sums has been extensively studied and solved [56, 57]. In particular, two binary matrices \mathbf{X} and \mathbf{Y} have the same row- and column-wise sums if they differ by an *interchange*, where an interchange is defined by a quadruple (k, l, m, n) such that

$$\begin{bmatrix} X_{kl} & X_{kn} \\ X_{ml} & X_{mn} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} Y_{kl} & Y_{kn} \\ Y_{ml} & Y_{mn} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Moreover, any two matrices with equivalent row and column sums can be obtained from one another by a sequence of such interchanges.

These results imply that, except for some very special cases, uniquely determining a binary matrix from its row- and column-wise popcounts is an impossible task. In what follows, we investigate how many additional DFT coefficients are required to make all binary matrices of a given size uniquely recoverable. Below, we study matrices of dimensions $N_1 \times N_2$ and consider the cases (i) when N_1 and N_2 are distinct primes, (ii) square matrices with $N_1 = N_2 = N$ and prime N , and (iii) square matrices with $N = p^\alpha$ where p is prime and $\alpha > 1$ is an integer.

3.2. Matrices of sizes $N_1 \times N_2$ with distinct primes N_1 and N_2 . For rectangular matrices with prime dimensions, we can prove our strongest uniqueness result. With the knowledge of just one additional DFT coefficient (in addition to \tilde{X}_{00} , \tilde{X}_{01} and \tilde{X}_{10}), the binary matrix \mathbf{X} can be uniquely recovered. In line with our assumption of low frequency coefficients becoming available first, this additional DFT coefficient is \tilde{X}_{11} . Note that this is a stronger restriction than the notation $\text{IP}(N_1, N_2, 1, 1)$ conveys, which includes all DFT coefficients in the pass band with $|k|, |l| \leq 1$. However, we will show that uniqueness does not require the knowledge of $\tilde{X}_{1,-1}$ or of its equivalent conjugate pair.

Theorem 5. Consider a generic binary matrix \mathbf{X} of dimension $N_1 \times N_2$, where N_1 and N_2 are prime and $N_1 \neq N_2$. If the four DFT coefficients \tilde{X}_{00} , \tilde{X}_{10} , \tilde{X}_{01} , and \tilde{X}_{11} are known, then the inverse problem of reconstructing \mathbf{X} is uniquely solvable.

Proof. Denote the total number of elements as $T = N_1 N_2$. Let \mathbf{X} and \mathbf{Y} be two distinct $N_1 \times N_2$ binary matrices. Suppose that $\tilde{X}_{kl} = \tilde{Y}_{kl}$ for $0 \leq k, l \leq 1$. Consider the (1,1)-th DFT coefficient of \mathbf{X} ,

$$(11) \quad \tilde{X}_{11} = \sum_{m=1}^{N_1} \sum_{n=1}^{N_2} X_{mn} e^{2\pi i(m/N_1 + n/N_2)} = \sum_{m=1}^{N_1} \sum_{n=1}^{N_2} X_{mn} e^{2\pi i(mN_2 + nN_1)/T}.$$

As N_1 and N_2 are distinct primes, $e^{2\pi i(N_1 + N_2)/T}$ is a primitive root of unity of T -th order, with the complete set of T -th roots of unity given by

$$\{e^{2\pi i(mN_2 + nN_1)/T} : 1 \leq m \leq N_1, 1 \leq n \leq N_2\}.$$

These are the roots that appear in (11), suggesting that the sum is the one-dimensional DFT coefficient of some vector \mathbf{x} . Let \mathbf{x} be the binary vector of length T formed by unrolling the entries of \mathbf{X} according to

$$(12) \quad x_\alpha = X_{mn}, \quad \alpha = mN_2 + nN_1 \pmod{T}.$$

We can thus rewrite (11) as

$$\tilde{X}_{11} = \sum_{n=1}^T x_n e^{2\pi i(n/T)} = \tilde{x}_1,$$

which is equivalent to the first DFT coefficient of the one-dimensional binary vector \mathbf{x} . Similarly define the binary vector \mathbf{y} such that $\tilde{Y}_{11} = \tilde{y}_1$. Thus, we have two distinct one-dimensional binary vectors, \mathbf{x} and \mathbf{y} of length T each, which agree at their first two DFT coefficients. By Lemma 4, \mathbf{x} and \mathbf{y} must agree at all entries, except on at least one pair of indexes $a, b \in \{1, \dots, T\}$ that satisfy (8). Assuming $s = N_1$ in Lemma 4, we have $x_\alpha = 1$ for all $\alpha = a \pmod{N_1}$. Applying this result to (12), there exists a fixed value of $m = m_0$ such that $X_{m_0 n} = 1$ for all $1 \leq n \leq N_2$. We can similarly conclude that $Y_{m_0 n} = 0$ for all $1 \leq n \leq N_2$. However, \tilde{X}_{10} and \tilde{Y}_{10} fix the row sums of the matrices \mathbf{X} and \mathbf{Y} . As $\tilde{X}_{10} = \tilde{Y}_{10}$ by assumption, \mathbf{X} and \mathbf{Y} must have the same row sums. We have, in contradiction, already shown that the m_0 -th row of \mathbf{X} has a row sum of N_2 whereas the same row of \mathbf{Y} sums to 0. Identical logic holds for the case when $s = N_2$ in Lemma 4 by, instead, finding a fixed column index that has differing sums for \mathbf{X} and \mathbf{Y} . This contradicts the assumption that $\tilde{X}_{01} = \tilde{Y}_{01}$. Thus, by Lemma 4, as \mathbf{x} and \mathbf{y} agree on their 1st one-dimensional DFT coefficient, but do not differ at the stated indexes, they must be equal. Hence, by (12), $\mathbf{X} = \mathbf{Y}$, making the solution to the inverse problem unique. \blacksquare

While results for binary one-dimensional vectors were used in the proof of Theorem 5, the conclusion of this theorem is significantly stronger than in the one-dimensional case. Indeed, for vectors of length $T = N_1 N_2$ with $N_1 < N_2$ being both prime, one requires $L = N_2$ to guarantee uniqueness by Lemma 2 of [38]. In contrast, for matrices of the dimension $N_1 \times N_2$, the required number of DFT coefficients does not increase with N_1 or N_2 but rather stays fixed at 4.

3.3. Square matrices of prime order N . While results for binary vectors of length $T = N_1 N_2$ were used in the above proof of [Theorem 5](#) for rectangular matrices, we cannot use the same approach for square matrices. This is so because, for a binary matrix \mathbf{X} of dimension $N \times N$, the expression for \tilde{X}_{11} no longer involves a complete, non-repeating set of roots of unity as in [\(11\)](#). Instead, we have

$$(13) \quad \tilde{X}_{11} = \sum_{m=1}^N \sum_{n=1}^N X_{mn} e^{2\pi i(m+n)/N} .$$

The exponential factors in the right-hand side of [\(13\)](#) are the N -th roots of unity, and each root appears N times (there are N^2 terms in the summation). Albeit different than in the rectangular case, equation [\(13\)](#) contains useful geometric information about the elements of \mathbf{X} , similarly to the coefficients \tilde{X}_{10} and \tilde{X}_{01} , which contain information about the number of nonzero entries in each row and column, respectively. To see that this is the case, we rewrite [\(13\)](#) by grouping the roots of unity as

$$(14) \quad \tilde{X}_{11} = \sum_{j=1}^N \eta_j e^{2\pi i(j/N)} , \quad \text{where } \eta_j = \sum_{\substack{m,n=1 \\ m+n=j \pmod{N}}}^N X_{mn} .$$

Using the fact that the cyclotomic integers are irreducible, we conclude that the knowledge of \tilde{X}_{11} is equivalent to knowing the values of η_j for $j = 1, \dots, N$. This, in turn, tells us how many ones are in each subset (labeled by j) of elements X_{mn} with indexes m, n satisfying the equation

$$(15) \quad m + n = j \pmod{N} .$$

For each fixed j , the N solutions to [\(15\)](#) lie along a line of the slope -1 , which may be periodically extended. This is illustrated in [Figure 1](#). Thus, the value of \tilde{X}_{11} tells us how many ones are in each line of slope -1 . In this sense, \tilde{X}_{11} provides projection information similar to that in \tilde{X}_{10} and \tilde{X}_{01} , but along the lines that are neither horizontal nor vertical but have the slope of -1 .

It is a straightforward extension to show that \tilde{X}_{kl} contains information equivalent to the projection along a periodic line defined by the equation

$$(16) \quad kn + lm = j \pmod{N} .$$

We say that the slope of the line defined by [\(16\)](#) is $-l/k$. Note that the expression [\(16\)](#) is valid for $k \neq 0$. If $k = 0$, \tilde{X}_{0l} counts the number of nonzero entries along the vertical lines. An immediate consequence of the above observation is that \tilde{X}_{kl} and $\tilde{X}_{k'l'}$ provide the same information if the periodic line classes with (k, l) and (k', l') have the same slope. This happens whenever

$$(17) \quad (k, l) \sim (k', l') \iff kl' = k'l \pmod{N} ,$$

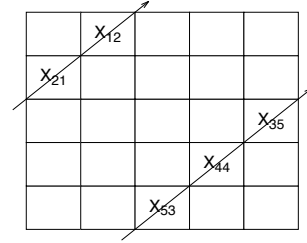


Figure 1. Solutions to [\(15\)](#) with $N = 5$ and $j = 3$. This line is referred to as having slope -1 as the column index increases by 1 as the row index decreases by 1.

where we have stated the condition as an equivalence relation. Note that, in general, this is a valid equivalence relation whenever at least k and k' (or l and l') are relatively prime to the congruent modulo number, which is always true if N is prime, as we assume here.

We are thus considering a periodic extension of the standard problem concerning row and column sums of binary matrices considered in [56, 57]. Instead of asking when a binary matrix can be uniquely determined by its projections along horizontal and vertical lines, we are interested in how many periodic projections (and in which directions) are sufficient to uniquely recover an $N \times N$ binary matrix. The key idea here is that, while in general we need all N^2 DFT coefficients to determine the original matrix (or $(N^2 + 1)/2$ by symmetry when the matrix is known to be real), in this binary setup, many of the DFT coefficients contain the same information as another coefficient. For example, it is easy to see that, for any prime N , $\tilde{X}_{2,0}$ also gives the individual popcount along each row of \mathbf{X} and provides no additional information compared to $\tilde{X}_{1,0}$. As another example, let $N = 23$; then, according to (17), $\tilde{X}_{7,5}$ provides the same information as $\tilde{X}_{1,6}$. Thus, it is clear that we should not need all N^2 DFT coefficients to recover \mathbf{X} as there are fewer than N^2 independent coefficients. The following lemma, originally due to Thue, is the key algebraic result for determining how many coefficients are required for unique recovery.

Lemma 6. *Let N be prime and define $L_0 = \lfloor \sqrt{N} \rfloor$. Let k and l be integers such that $|k|, |l| \leq N - 1$. Then there exist integers k' and l' with $|k'|, |l'| \leq L_0$ such that $kl' = k'l \pmod{N}$.*

A proof can be found in [59]. Equation (17) provides the condition under which two DFT coefficients are dependent. Lemma 6 states that we can always find a solution to (17) with k' and l' both smaller in magnitude than L . These results are combined to obtain the uniqueness result in Theorem 7.

Theorem 7. *Consider a generic binary matrix \mathbf{X} of known dimension $N \times N$, where N is prime. Let $L_0 = \lfloor \sqrt{N} \rfloor$. Then the inverse problem $\text{IP}(N, N, L, L)$ (see Definition 1) is uniquely solvable for any $L \geq L_0$.*

Proof. It is sufficient to prove the theorem for $L = L_0$. By the inverse DFT in (3), knowledge of all DFT coefficients uniquely determines any binary matrix. Suppose that \tilde{X}_{kl} is unknown for some k and l such that $|k|$ or $|l|$ is greater than L_0 . By Lemma 6, there exists a k' and l' satisfying $|k'|, |l'| \leq L_0$ and $(k', l') \sim (k, l)$. By (17), \tilde{X}_{kl} and $\tilde{X}_{k'l'}$ are dependent and provide identical information. As $\tilde{X}_{k'l'}$ is within the assumed pass band, \mathbf{X} is uniquely determined. ■

Approaching this setup geometrically, one can represent the entries of the $N \times N$ matrix as a $N \times N$ grid of points, and consider all the lines that (periodically) connect these points. This is an example of a finite affine plane of order N [26]. It is known that each line in such a geometry contains N points, and each point is on $N + 1$ lines (with N parallel classes for each line for a total of $N^2 + N$ lines). As each DFT coefficient provides the popcount along N lines in a parallel class, there can, in fact, only be $N + 1$ independent DFT coefficients (in addition to the global popcount \tilde{X}_{00}).

This observation implies that the condition provided by Theorem 7 is not a necessary one; it is sufficient but necessary to know *all* DFT coefficients up to order L_0 for unique recovery.

However, the theorem states that at least one of the required $N + 1$ coefficients (in addition to \tilde{X}_{00}) is of the order L_0 . For example, for $N = 17$, we have $L_0 = \lfloor \sqrt{17} \rfloor = 4$, but out of the $N + 1 = 18$ coefficients needed (in addition to the global popcount) to guarantee recovery, $\tilde{X}_{1,4}$ and $\tilde{X}_{4,1}$ are the only independent coefficients of 4th order. All other 4th order DFT coefficients are equivalent to some coefficient of lower order by (17). One can see that, in general, uniqueness requires knowledge of at least one coefficient of the order L_0 . This is so because $\tilde{X}_{L_0,1}$ is independent from all DFT coefficients of lower order. Indeed, there are no solutions to the equation $L_0 l' = k' \pmod{N}$ with $|k'|, |l'| < L_0$.

3.4. Square matrices of non-prime dimension. When the dimension of a square binary matrix is not prime, the geometric interpretation of the coefficients is not as apparent. Consider a binary matrix of the size $N \times N$ where $N = p^\alpha$ with $\alpha > 1$. The DFT coefficients can be expressed in this case as

$$(18) \quad \tilde{X}_{kl} = \sum_{m,n=1}^N X_{mn} e^{2\pi i(mk+nl)/N} = \sum_{j=1}^N \eta_j e^{2\pi i(j/N)}, \quad \text{where } \eta_j = \sum_{\substack{m,n=1 \\ mk+nl=j \pmod{N}}}^N X_{mn}.$$

The last expression partitions the entries of \mathbf{X} according to $mk + nl = j \pmod{N}$ for each integer j in the range $1 \leq j \leq N$. We no longer refer to the entries satisfying $mk + nl = j \pmod{N}$ as a line because this fails the usual geometric definition of two lines intersecting at most once. For example, for $\alpha = 2$, the partition $k = 0, l = j = 1$ and the partition $k = p, l = j = 1$ intersect at $(m, n) = (\mu p, 1)$ for all $0 \leq \mu \leq p - 1$. Moreover, as N is not prime, the DFT coefficients no longer uniquely determine the sums along these partitions. In particular, the DFT coefficient \tilde{X}_{10} no longer uniquely determines the row sums of \mathbf{X} . By Lemma 2, it is possible that $\tilde{X}_{10} = 0$ as long as, for all m , the row sums r_m satisfy $r_m = r_{m+\mu p^{\alpha-1}}$ for $0 \leq \mu \leq p - 1$. It is straightforward to see that \tilde{X}_{k0} yields identical information, as long as k is not a multiple of p . When k is a multiple of p , let $\beta = \log_p(\gcd(k, N))$. Then, for $k' = k/\gcd(k, N)$, we have

$$(19) \quad \tilde{X}_{k0} = \sum_{m,n=1}^N X_{mn} e^{2\pi i(mk/N)} = \sum_{m,n=1}^N X_{mn} e^{2\pi i(mk'/p^{\alpha-\beta})}.$$

The second sum involves roots of unity of the order $p^{\alpha-\beta}$, each root appearing $p^{\alpha+\beta}$ times. Intuitively, this suggests that $\tilde{X}_{k_1,0}$ and $\tilde{X}_{k_2,0}$ contain different information if $\gcd(k_1, N) \neq \gcd(k_2, N)$. For $k < N$, we have the bound $\gcd(k, N) \leq p^{\alpha-1}$. This suggests that $X_{p^{\alpha-1},0}$ contains new information as compared to all the lower-order coefficients and motivates the uniqueness result in Theorem 8.

Theorem 8. Consider a generic binary matrix \mathbf{X} of known dimension $N \times N$ where $N = p^\alpha$, p is prime and $\alpha > 1$ an integer. Define $L_0 = p^{\alpha-1}$. Then the inverse problem $\text{IP}(N, N, L, L)$ (see Definition 1), is uniquely solvable for any $L \geq L_0$.

Before proceeding, we state and prove the following useful lemma:

Lemma 9. Under the conditions of Theorem 8, let $k'l = kl' \pmod{N}$, and suppose that at least one of k and l is relatively prime with p . Then $X_{k'l'} = 0$ implies that $X_{kl} = 0$.

Proof. Suppose that $X_{k'l'} = 0$. Without loss of generality, assume that $\gcd(l, p) = 1$. To employ [Lemma 2](#) we first collect all powers of the N -th primitive root of unity ζ_N . We rewrite this coefficient as

$$(20) \quad 0 = \tilde{X}_{k'l'} = \sum_{m,n=1}^N X_{mn} (\zeta_N)^{mk'+nl'} = \sum_{\mu=1}^N \left[\sum_{\substack{m,n=1 \\ mk'+nl'=\mu \pmod{N}}}^N X_{mn} \right] (\zeta_N)^\mu .$$

By [Lemma 2](#), this implies that, for $1 \leq \mu \leq p^{\alpha-1}$ and for all $0 \leq \nu \leq p-1$,

$$(21) \quad \sum_{\substack{m,n=1 \\ mk'+nl'=\mu+\nu p^{\alpha-1} \pmod{N}}}^N X_{mn} = c_\mu$$

with some integer constant c_μ . We need to prove that an identical expression holds for \tilde{X}_{kl} for all μ and ν and a different set of constants,

$$(22) \quad \sum_{\substack{m,n=1 \\ mk+nl=\mu+\nu p^{\alpha-1} \pmod{N}}}^N X_{mn} = d_\mu .$$

For fixed μ and ν , consider the indexes of terms summed in [\(22\)](#). Using the fact that $k'l = kl' \pmod{N}$ and that l has a multiplicative inverse, we make the following algebraic manipulations:

$$\begin{aligned} mk + nl &= \mu + \nu p^{\alpha-1} \pmod{N} \\ mkl^{-1}l' + nl' &= l^{-1}l'\mu + \nu(l^{-1}l')p^{\alpha-1} \pmod{N} \\ mk' + nl' &= l^{-1}l'\mu + \nu(l^{-1}l')p^{\alpha-1} \pmod{N} . \end{aligned}$$

Thus, letting $\mu' = l^{-1}l'\mu \pmod{N}$ and $\nu' = \nu(l^{-1}l') \pmod{N}$, we have

$$\sum_{\substack{m,n=1 \\ mk+nl=\mu+\nu p^{\alpha-1} \pmod{N}}}^N X_{mn} = \sum_{\substack{m,n=1 \\ mk'+nl'=\mu'+\nu' p^{\alpha-1} \pmod{N}}}^N X_{mn} = c_{\mu'} ,$$

where this last equality holds from [\(21\)](#). Thus $d_\mu = c_{\mu'}$ in [\(22\)](#), which implies that $\tilde{X}_{kl} = 0$. \blacksquare

[Lemma 9](#) implies that $\tilde{X}_{k'l'}$ and \tilde{X}_{kl} are dependent if $k'l = kl' \pmod{N}$. What remains to show is that this condition is satisfied for all DFT coefficients of order larger than $p^{\alpha-1}$. We are now ready to prove [Theorem 8](#).

Proof. [Theorem 8](#) will be proved by showing that, for any DFT coefficient \tilde{X}_{kl} with either $|k|$ or $|l|$ greater than $L_0 = p^{\alpha-1}$, there exists a DFT coefficient $\tilde{X}_{k'l'}$ with $|k'|, |l'| \leq L_0$ that already contains dependent information. We consider three separate cases: both k and l relatively prime with N , only one of k and l relatively prime with N , and neither k nor l relatively prime with N .

1) *Case* $\gcd(k, p) = \gcd(l, p) = 1$. By a small extension of [Lemma 6](#), we can find k' and l' that are relatively prime with p , $|k'|, |l'| \leq \lfloor \sqrt{p^\alpha} \rfloor < L_0$, and $(k, l) \sim (k', l')$, with the equivalence relation denoted by \sim defined in (19). We can now apply [Lemma 9](#) to these pairs of integers to conclude that $\tilde{X}_{k'l'}$ and \tilde{X}_{kl} are dependent.

2) *Case* $\gcd(k, p) \neq 1, \gcd(l, p) = 1$. Without loss of generality, we will assume that l is still relatively prime with p . Let $k' = \gcd(k, N) \leq L_0$. With this choice of k' , we can find an l' such that $|l'| \leq L_0$ and $(k, l) \sim (k', l')$. As $k'l$ can take one of N/k' values in \mathbb{Z}_N , and k is an additive generator of these N/k' values, there is some $l' \leq N/k' \leq L_0$ such that $kl' = k'l \pmod{N}$. As k' was chosen to be the greatest common divisor of k and N , this choice of l' must be relatively prime with N . Thus, [Lemma 9](#) applies, implying that $\tilde{X}_{k'l'}$ and \tilde{X}_{kl} are dependent.

3) *Case* $\gcd(k, p) \neq 1, \gcd(l, p) \neq 1$. In this case, let $k' = \gcd(k, N)$ and $l' = \gcd(l, N)$, and without loss of generality, let $l' \leq k'$. This case can be reduced to Case 1. Setting $\tilde{X}_{k'l'} = 0$, we have a vanishing sum of roots of unity of order N/l'

$$\begin{aligned} 0 = \tilde{X}_{k'l'} &= \sum_{m,n=1}^N X_{mn}(\zeta_N)^{k'm+nl'} = \sum_{m,n=1}^N X_{mn}(\zeta_{N/l'})^{(k'/l')m+n} \\ &= \sum_{\mu=1}^{N/l'} \left[\sum_{\substack{m,n=1 \\ (k'/l')m+n=\mu \pmod{N/l'}}}^N X_{mn} \right] (\zeta_{N/l'})^\mu. \end{aligned}$$

This last equation is exactly the same as (20) in [Lemma 9](#) with the substitutions $N \leftarrow N/l'$, $k' \leftarrow k'/l'$, and $l' \leftarrow 1$. The result of [Lemma 9](#) can now be applied, completing the proof. \blacksquare

The result of [Theorem 8](#) is tight in the sense that there exist matrices that cannot be uniquely recovered with the data bandwidth $L < L_0 = p^{\alpha-1}$. Unfortunately, this implies that we have no universal super-resolution (as defined in this paper) for square matrices of the size $N = 2^\alpha$. By the even version of (2), all DFT coefficients are in the range $[-2^{\alpha-1} + 1, 2^{\alpha-1}]$. With $L_0 = 2^{\alpha-1}$, this range is equivalent to $[-L_0 + 1, L_0]$. Thus, the condition $L = L_0$ is equivalent to the requirement that the complete set of DFT coefficients be known. As an example, consider the checkerboard matrices defined entry-wise by

$$X_{nm} = \begin{cases} 0 & n+m=1 \pmod{2} \\ 1 & n+m=0 \pmod{2} \end{cases}, \quad Y_{nm} = \begin{cases} 0 & n+m=0 \pmod{2} \\ 1 & n+m=1 \pmod{2} \end{cases},$$

where $1 \leq n, m \leq 2^\alpha$. The corresponding DFT coefficients are given by

$$(23) \quad \tilde{X}_{kl} = \begin{cases} 2^{\alpha-1} & k=l=0 \\ 2^{\alpha-1} & k=l=2^{\alpha-1} \\ 0 & \text{otherwise} \end{cases}, \quad \tilde{Y}_{kl} = \begin{cases} 2^{\alpha-1} & k=l=0 \\ -2^{\alpha-1} & k=l=2^{\alpha-1} \\ 0 & \text{otherwise} \end{cases}.$$

These coefficient values can be readily obtained by letting $\mathbf{X} = \frac{1}{2}(\mathbf{J} + \mathbf{A})$ where \mathbf{J} is the matrix of all ones and \mathbf{A} is the matrix with the entries $A_{nm} = (-1)^{n+m}$. The only nonzero DFT

coefficient of \mathbf{J} is $\tilde{J}_{00} = 2^\alpha$. Similarly, we can represent the entries of \mathbf{A} as $A_{nm} = e^{\pi i(n+m)} = e^{2\pi i(m+n)(N/2)/N}$, which shifts the nonzero entry to the position $(N/2, N/2) = (2^{\alpha-1}, 2^{\alpha-1})$. Similar logic applied to $\mathbf{Y} = \frac{1}{2}(\mathbf{J} - \mathbf{A})$ yields the expression given in (23). These two matrices agree on all coefficients except one that requires $L = L + 0 = 2^{\alpha-1}$.

Similarly to the previous checkerboard example, we can show that the square matrices of the size $N = p^\alpha$ (with p being a prime greater than 2) defined as

$$X_{nm} = \begin{cases} 1 & n + m = 1 \pmod{p} \\ 0 & \text{otherwise} \end{cases}, \quad Y_{nm} = \begin{cases} 1 & n + m = 0 \pmod{p} \\ 0 & \text{otherwise} \end{cases}$$

are $(p^{\alpha-1})$ -indistinguishable, implying that the band width $L = p^{\alpha-1}$ is required for unique inversion.

4. Inversion algorithms. We now discuss the algorithms to recover binary matrices for each case considered: (i) rectangular matrices with dimensions $N_1 \times N_2$ where N_1 and N_2 are distinct primes, (ii) square matrices of dimension $N \times N$ where N is prime, and square matrices with N of the form $N = p^\alpha$, where p is prime and $\alpha > 1$ an integer. For each case, we assume access to a large enough bandwidth of DFT coefficients to guarantee uniqueness, as determined by the previous section.

4.1. General strategy. Let, as above, the total number of elements in an $N_1 \times N_2$ matrix be denoted as $T = N_1 N_2$. Even under the conditions when each matrix of given dimension is, theoretically, uniquely determined by the data, finding the inverse solution by exhaustive search requires testing $\binom{T}{S}$ possibilities, where S is the global popcount. Under the condition when $S \sim T/2$, this strategy quickly becomes computationally prohibitive. However, inspired by the theoretical derivations shown above, we can break the inverse problem into more manageable steps and significantly increase the computational efficiency. Before developing algorithms for each case considered, we make an observation on the general form of these subproblems.

Theory suggests that the DFT coefficients often contain information equivalent to how many ones are present in each periodic line. For example, when N_2 is prime, \tilde{X}_{01} (in conjunction with $S = \tilde{X}_{00}$) is equivalent to knowing how many ones are present in each column of \mathbf{X} . We thus consider the related combinatorial problem of placing S ones in N_2 boxes, where we can place no more than N_1 ones in each box. By the inclusion-exclusion principle, one can compute the total number of possibilities as

$$(24) \quad \sum_{n \geq 0} (-1)^n \binom{N_1}{n} \binom{S - nN_2 + N_1 - (n+1)}{N_1 - 1}.$$

This formula gives the complexity of finding by exhaustive search the column-wise sums of \mathbf{X} . If this problem can be solved, the search space for the unique binary image has been significantly reduced to only those matrices with the correct number of ones in each column. We need to find among those the matrix that matches any remaining known but yet unused DFT coefficients. Refer to the correct column sum values as c_n for $1 \leq n \leq N_2$. The unique

binary image that matches the four given DFT coefficients is now within a set of size

$$(25) \quad \prod_{n=1}^{N_2} \binom{N_1}{c_n} .$$

As a concrete example, consider the case $N_1 = 7$, $N_2 = 11$ and $S = 38$. The number of distinct binary matrices with these parameters is 1.36×10^{22} . The problem of determining the c_n values is substantially smaller and is of size 1,528,688 according to (24). With only the c_n values known, the overall search space has been reduced to an upper bound of $35^{11} \approx 9.65 \times 10^{16}$ by (25). As N_1 is also prime here, one could repeat this process to further reduce the search space size by similarly solving for the row-wise popcounts r_m – which has a smaller individual problem size of 443,658,688. The ensuing algorithms make use of these ideas to break down larger problems into more manageable subproblems. However, we still need methods that are more efficient than exhaustive search to solve these subproblems.

4.2. Integer linear programming (ILP) and lattices. Finding an $N_1 \times N_2$ binary matrix \mathbf{X} that agrees with all available DFT coefficients can be phrased as an integer linear programming (ILP) problem of the form

$$(26) \quad \mathbf{Ax} = \mathbf{b} , \quad x_i \in \{0, 1\} .$$

In this formulation, \mathbf{x} is a binary vector of length $N_1 N_2$, which corresponds to stacking the columns of \mathbf{X} . The matrix \mathbf{A} contains the relevant Fourier matrix entries, with \mathbf{b} containing the available DFT coefficients. In line with (1), we can express these entries using multi-indices of the form

$$A_{(k,l),(m,n)} = e^{2\pi i (mk/N_1 + nl/N_2)} , \quad b_{(k,l)} = \tilde{X}_{kl} ,$$

where the multi-index (m, n) varies over $1 \leq m \leq N_1$ and $1 \leq n \leq N_2$, and (k, l) varies over the indexes corresponding to the available DFT coefficients. Note that, in an actual implementation, the entries of \mathbf{A} and \mathbf{b} are split into real and imaginary parts, which forces the entries of \mathbf{x} to be real. Thus, if M DFT coefficients are known in addition to \tilde{X}_{00} , then \mathbf{A} is a $(2M + 1) \times N_1 N_2$ matrix, where we have taken into account that the row corresponding to \tilde{X}_{00} has no imaginary part. For simplicity, we refer to \mathbf{A} and \mathbf{b} as having $M + 1$ rows with complex entries. Additionally, no redundant coefficients (which are known to be conjugates of each other) are needed in an implementation. For larger problems, \mathbf{A} can be efficiently applied by fast Fourier transform techniques.

Solving (26) is a known NP-hard problem. When using ILP techniques, as there is a unique solution, but no objective function to minimize, branch and bound methods do not offer significant improvement over exhaustive search. By defining an arbitrary objective function to minimize, the branch and bound may converge faster or slower, though it is typically difficult to tell *a priori* which is the case [1]. Incorporating cutting planes and other preprocessing steps, however, can restrict the size of the search space [58, 40]. Without an objective function, ILP is reliant on these preprocessing steps to outperform exhaustive search. As solving (26) is NP-hard, the overall runtime is dominated by the size of the search space, as opposed to any cost of applying the matrix \mathbf{A} .

An alternate approach to ILP is to use lattice basis reduction techniques. These techniques aim to reduce a given basis to short, nearly orthogonal vectors, with an end goal of facilitating calculations over the integers. We briefly summarize the celebrated Lenstra-Lenstra-Lovasz (LLL) algorithm [35] for lattice basis reduction, which has many applications in mathematics and cryptography [27].

Consider a linearly independent set of vectors $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ in \mathbb{R}^m , where $n \leq m$. The integer lattice \mathcal{L} with this basis is the set of all linear combinations of the \mathbf{b}_j with integer coefficients

$$\mathcal{L} = \{a_1 \mathbf{b}_1 + \dots + a_n \mathbf{b}_n : a_j \in \mathbb{Z}\} .$$

The LLL algorithm takes this basis of the lattice, \mathbf{B} , and returns a new basis \mathbf{B}^* , which is generally comprised of short, nearly orthogonal vectors. This basis \mathbf{B}^* is called LLL-reduced, and is obtained through a Gram-Schmidt-like process, modified to ensure that the basis vectors stay in the lattice and to prioritize short vectors. Most importantly for our purposes, the first vector \mathbf{b}_1^* in \mathbf{B}^* will be the shortest in the new basis. It will not necessarily be the absolute shortest vector in the lattice [46], but the LLL algorithm returns an approximately shortest vector in polynomial (hopefully, reasonable) time.

To see how we can use lattice reduction to solve (26) with M known DFT coefficients, we first construct the $(T + M + 1) \times (T + 1)$ matrix (as before, $T = N_1 N_2$) with 4 blocks defined as

$$(27) \quad \mathbf{B} = \left(\begin{array}{c|c} \mathbf{I} & \mathbf{0} \\ \hline \beta \mathbf{A} & -\beta \mathbf{b} \end{array} \right) .$$

In this 2×2 block matrix form, \mathbf{A} and \mathbf{b} are defined as in (26), and \mathbf{I} and $\mathbf{0}$ are the identity matrix and zero vector of the length T . The constant β that appears in the lower two blocks is assumed to be large. Again, in an actual implementation, the \mathbf{A} and \mathbf{b} blocks would have $2M + 1$ rows to account for real and imaginary parts.

The LLL algorithm can now be performed on \mathbf{B} , treating the columns of the matrix as the lattice basis elements of length $T + M + 1$. The shortest vector in the resulting LLL-reduced basis, \mathbf{b}_1^* , must necessarily be a linear combination of the original basis vectors. Letting $\mathbf{x} = [a_1, a_2, \dots, a_T]$ be an integer vector, any vector in the lattice \mathcal{L} is of the form

$$\mathbf{b}_1^* = [a_1, a_2, \dots, a_T | \beta(\mathbf{A}\mathbf{x} - \mathbf{b})] .$$

If β is chosen to be sufficiently large, this shortest vector will likely minimize $\mathbf{A}\mathbf{x} - \mathbf{b}$, with the vector \mathbf{x} being the proposed integer solution. Additional details of the algorithm can be found in [8, 17].

Finding the shortest vector in the lattice is also known to be an NP-hard problem. The potential advantages of the LLL algorithm rely on the fact that it is an approximation algorithm, and can be expected to find a solution in polynomial time [37]. However, as an approximation algorithm, there is no guarantee that it will outperform ILP techniques in general. In fact, by changing parameters in LLL, one can trade off between a faster runtime and a higher probability of finding a sufficiently short vector. However, the runtime of the

LLL algorithm is $\mathcal{O}(n^5 m \log^3(B))$, where $B = \max_i \|\mathbf{b}_i\|_2$, which implies that, for practical purposes, the polynomial time still increases quickly in the size of the problem n [37].

One downside to the LLL algorithm is that it does not incorporate known bounds on the integer values. For example, if it is known that the correct integer values are either 0 or 1, the shortest vector in the LLL-reduced basis is not guaranteed to have binary coefficients. In contrast, ILP obeys the integer bounds throughout its search.

Taking into account the relative advantages and disadvantages between these two approaches, we use a combination of ILP and LLL in the following algorithms. In general, the LLL algorithm was found to be much more efficient when running on problems with a smaller number of unknowns, which can take integer values in a possibly large range. This takes advantage of the fact that LLL is independent of the known bound on the integers. In contrast, ILP depends heavily on the range of the integers, and can be more reliable when the integers are known to be binary. ILP can also be effective for large problems (with many constraints) when cutting planes can reduce the overall size. Anecdotally, ILP had slightly more stability than LLL when attempting to reconstruct with only $M = 1$ DFT coefficient.

4.3. Algorithms. We now describe the algorithms for reconstructing the three cases of matrix dimensions. While the three algorithms share many similarities, we consider each case separately.

4.3.1. Case when $N_1 \neq N_2$ are both prime. By the theoretical results for uniqueness, we assume access to only the 4 DFT coefficients $\tilde{X}_{00}, \tilde{X}_{01}, \tilde{X}_{10}$, and \tilde{X}_{11} . As described in the beginning of this section, we first consider the smaller problem of using \tilde{X}_{00} and \tilde{X}_{01} to reconstruct the column sums of \mathbf{X} . Thus we consider the problem

$$(28) \quad \mathbf{A}^{(01)} \mathbf{c} = \mathbf{b}^{(01)}, \quad 0 \leq c_i \leq N_2,$$

where the unknown vector \mathbf{c} represents the column sums of \mathbf{X} , and $\mathbf{A}^{(01)}$ and $\mathbf{b}^{(01)}$ refer to the respective sub-matrix of \mathbf{A} and sub-vector of \mathbf{b} containing only the rows corresponding to \tilde{X}_{00} and \tilde{X}_{01} . The columns of $\mathbf{A}^{(01)}$ are similarly restricted to only have one representative entry from each column of \mathbf{X} . In line with the previous discussion, even though the number of unknowns has been greatly reduced from $N_1 N_2$ to N_1 , and the bound on the integers has been increased to N_2 , solving this problem using ILP was preferable for stability reasons as $M = 1$, where M is the number of DFT coefficients corresponding to this directional sum.

After finding the column sums \mathbf{c} via (28), we solve the corresponding problem

$$(29) \quad \mathbf{A}^{(10)} \mathbf{r} = \mathbf{b}^{(10)} \quad ; \quad 0 \leq r_i \leq N_1,$$

to obtain the corresponding row sum vector \mathbf{r} . In (29), the matrix $\mathbf{A}^{(10)}$ and $\mathbf{b}^{(10)}$ contain only the rows pertaining to \tilde{X}_{00} and \tilde{X}_{10} . With this additional row information in hand, we finally solve the full binary system

$$(30) \quad \begin{bmatrix} \mathbf{A} \\ - \\ \mathbf{C} \\ - \\ \mathbf{R} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ - \\ \mathbf{c} \\ - \\ \mathbf{r} \end{bmatrix} \quad ; \quad x_i \in \{0, 1\},$$

where the $N_2 \times N_1 N_2$ binary matrix \mathbf{C} contains ones appropriately to sum the column entries of \mathbf{x} . As \mathbf{x} is formed by stacking the columns of \mathbf{A} , \mathbf{C} is defined by

$$(31) \quad C_{mn} = \begin{cases} 1 & (m-1)N_1 + 1 \leq n \leq mN_1 \\ 0 & \text{else} \end{cases}.$$

The matrix \mathbf{R} is defined similarly to \mathbf{C} as in (31) to sum the rows of \mathbf{X} based on the ordering of \mathbf{x} . For storage efficiency, one can remove the (00), (01), and (10) rows from \mathbf{A} and \mathbf{b} in (30), as this information is already contained in the \mathbf{C} and \mathbf{R} matrix blocks. This remaining system finds the binary matrix, which matches the DFT coefficient \tilde{X}_{11} in the reduced search space with given column and row sums. As this is a larger system with binary integer bounds, it is generally more efficient to solve by using ILP. This is summarized in Algorithm 1.

Algorithm 1 Reconstruction algorithm for $N_1 \times N_2$ binary matrices where $N_1 \neq N_2$ are primes.

- 1: **Input:** DFT Coefficients $\tilde{X}_{00}, \tilde{X}_{01}, \tilde{X}_{10}, \tilde{X}_{11}$
 - 2: **Output:** $N_1 \times N_2$ binary matrix \mathbf{X}
 - 3: Use ILP to reconstruct column sums \mathbf{c} using (28)
 - 4: Use ILP to reconstruct row sums \mathbf{r} using (29)
 - 5: Use ILP to solve (30) for binary matrix \mathbf{X}
-

We remark that, when $N_1 \gg N_2$, it may be computationally faster to skip solving for the row sums as a separate subproblem. That is, immediately after solving (28), one can solve an equation of the form (30) without the \mathbf{R} block. Similarly, if $N_2 \gg N_1$, it may be prudent to ignore solving for the column sums as its own subproblem. The overall runtime considerations of Algorithm 1 are governed by the size of the search spaces for each subproblem, as discussed in Section 4.1.

4.3.2. Case when $N_1 = N_2 = N$ where N is prime. We take a similar algorithmic approach for reconstructing square $N \times N$ binary matrices. We again reconstruct the row and column sums of the matrix via (28) and (29) but can utilize the additional available DFT coefficients (as required by Theorem 7) to hopefully reconstruct larger matrices in a stable manner.

The matrix $\mathbf{A}^{(01)}$ in (28) was used to solve for the column sums, which were contained in the DFT coefficient \tilde{X}_{01} . For $L = L_0 = \lfloor \sqrt{N} \rfloor$, the corresponding submatrix $\mathbf{A}^{(01)}$ contains additional rows corresponding to the available DFT coefficients $\tilde{X}_{01}, \dots, \tilde{X}_{0L}$, which are all equivalent to the column sum information. These extra equations improve the reconstruction speed and stability of recovery. As we now have a moderately sized system with $M > 1$ DFT coefficients that encode column sum information, this system is efficiently solved using the LLL algorithm. After reconstructing the row and column sums, instead of immediately attempting to match a binary matrix with given row and column sums to the remaining DFT coefficients, we repeat this process for additional directions. For example, an analogous ILP problem can be set up to solve $\mathbf{A}^{(11)}\mathbf{d}^- = \mathbf{b}^{(11)}$ which solves for the sums along the diagonal lines of slope -1 (using the DFT coefficients $\tilde{X}_{11}, \dots, \tilde{X}_{LL}$).

This can be repeated for all $N + 1$ directions. However, while the row, column, and diagonal directions (slopes of ± 1) all have L related coefficients, no other direction will have L coefficients, with possibly many directions only having one related coefficient. This can have an adverse effect on the computational efficiency and stability of recovery. Thus, the LLL algorithm may fail to recover the directional sums for certain directions. As a check, if the resulting shortest vector is not sufficiently short (using a predefined error tolerance), we ignore that direction and only include its information as the DFT coefficient, as was done for \tilde{X}_{11} in [Algorithm 1](#). In our implementation, we used the maximum norm ($\|\mathbf{e}\|_\infty = \max_i |e_i|$) to measure the magnitude of this shortest vector. For improved stability, we do not attempt to reconstruct the directional sums along directions with only $M = 1$ DFT coefficient, and similarly include the DFT coefficient value as a constraint.

After attempting to solve for the directional sums along all $N + 1$ directions (skipping any with $M = 1$), we form an ILP problem of the form [\(30\)](#). A block is added for each successful directional recovery that sums the entries along those directions as in [\(31\)](#). The corresponding rows from the \mathbf{A} block can be removed, with the remaining rows of \mathbf{A} corresponding to directions with unsuccessful recoveries. Pseudocode for this algorithm is provided in [Algorithm 2](#).

Each call of the LLL algorithm roughly scales as N^6 (recall runtime is $\mathcal{O}(n^5 m \log^3(B))$) in [Algorithm 2](#). This rough estimate ignores the B term, and sets $n = m = N$. As [Algorithm 2](#) calls the LLL algorithm up to $N + 1$ times, the total runtime can be proportional to N^7 . In practice, for larger values of M , it is anticipated that this additional data will help the algorithm converge quicker. The main idea of [Algorithm 2](#) is that the final ILP step will run very quickly as the size of the search space will be drastically reduced.

Algorithm 2 Reconstruction algorithm for $N \times N$ binary matrices where N is prime.

- 1: **Input:** DFT Coefficients \tilde{X}_{kl} for all $|k|, |l| \leq L_0 = \lfloor \sqrt{N} \rfloor$
 - 2: **Input:** Error tolerance ϵ
 - 3: **Output:** $N \times N$ binary matrix \mathbf{X}
 - 4: **for** $j = 1$ **to** $N + 1$ **do**
 - 5: Collect M available DFT coefficients corresponding to direction j
 - 6: **if** $M > 1$ **then**
 - 7: Use LLL and the M DFT coefficients to find the shortest vector corresponding to the reconstruction of directional sums \mathbf{d}_j
 - 8: **if** $\|\text{shortest vector}\|_\infty < \epsilon$ **then**
 - 9: Successful Recovery: Create corresponding block \mathbf{D}_j
 - 10: **end if**
 - 11: **end if**
 - 12: **end for**
 - 13: Create matrix with blocks \mathbf{D}_j for all values of j corresponding to a successful recovery and \mathbf{A} containing DFT terms for unsuccessful recoveries.
 - 14: Create right hand side vector with corresponding blocks \mathbf{d}_j and the DFT coefficients.
 - 15: Use ILP to solve for binary matrix \mathbf{X} using this matrix and right hand side.
-

4.3.3. Case when $N_1 = N_2 = p^\alpha$ where p is prime and $\alpha > 1$. For square matrices of the size $N = p^\alpha$, more care is required. We will focus on the case when $\alpha = 2$, but similar ideas hold in theory for $\alpha > 2$. For $N = p^2$, as seen in (19), \tilde{X}_{0p} is equivalent to knowing how many entries in total are in the column numbers that are equal modulo p . Refer to these combined column sums as C_j for $j = 1, \dots, p$. The values of C_j can be solved quickly using the LLL algorithm, as there are only p unknowns as opposed to p^2 (where each C_j is bounded above by p^3). After this information is recovered, the remaining coefficients of the form \tilde{X}_{0k} for $k < p$ are equivalent to knowing the individual column sums. One can set up a linear system of the form (28) to solve for the column sums c_i , with an additional block containing the constraints already obtained from \tilde{X}_{0p} . These additional linear constraints are of the form

$$\sum_{i=j \pmod{p}} c_i = C_j, \quad \text{for } j = 1, \dots, p.$$

Identical results hold for the row sum by first using \tilde{X}_{p0} and subsequently looking at $\tilde{X}_{10}, \dots, \tilde{X}_{p-1,0}$. This is also true for the diagonal sums using \tilde{X}_{pp} and $\tilde{X}_{-p,p}$, which are all in the available DFT coefficient range. However, one cannot simply recover the sums along other directions based on specific DFT coefficients. Consider any DFT coefficient \tilde{X}_{kl} where at least one of k and l is relatively prime with p . By (18), this coefficient is still a sum of roots of unity of order $N = p^2$, where each root corresponds to matrix entries X_{mn} that satisfy $mk + nl = \mu \pmod{N}$, for some integer $1 \leq \mu \leq N$. However, in (18), since the roots of unity are no longer of prime order, the value of this sum does not uniquely determine the integer coefficients. By Lemma 2, the integer coefficients can differ by a fixed constant across entries that are equal modulo p , and still give the same sum.

As an illustrative example, consider a $3^2 \times 3^2$ binary matrix with $S = 40$ nonzero entries. If we are given $\tilde{X}_{1,2} = e^{2\pi i/9}$, we can deduce that there is at least one nonzero entry in the partition of entries with $j = 1$ from (18), which gives the exact value $e^{2\pi i/9}$. However, the remaining 39 nonzero entries still need to be distributed among the 9 partitions. With the knowledge that $\tilde{X}_{1,2} = e^{2\pi i/9}$, this distribution is not unique, but must satisfy the condition that the sum of the 39 corresponding roots of unity is 0, in accordance with Lemma 2. Using the notation in (18), let η_j be the number of ones contained in the j th partition. The four linear constraints for this example $\tilde{X}_{1,2}$ are thus

$$(32) \quad \eta_1 = \eta_4 + 1 = \eta_7 + 1; \quad \eta_2 = \eta_5 = \eta_8; \quad \eta_3 = \eta_6 = \eta_9; \quad \sum_{j=1}^9 \eta_j = S.$$

These constraints ensure that $\tilde{X}_{1,2} = e^{2\pi i/9}$ and that all 40 ones are placed in a partition. However, in these linear constraints, none of the η_j are uniquely determined from just $\tilde{X}_{1,2}$. On the other hand, Theorem 8 indicates that these η_j values will be uniquely determinable in the larger context of all available DFT coefficients.

To solve for these linear constraints, in the general case we use the LLL algorithm without \tilde{X}_{00} , to find a short vector that fits the coefficient. From this possible solution, one can deduce the linear constraints similar to the the form of the first 3 equations of (32).

The proposed algorithm is thus similar to [Algorithm 2](#), but with a modification to take into account that we cannot uniquely determine the sum along lines in all directions. First, reconstruct the sums along the rows, columns, and diagonal directions. These sums are uniquely determinable, and should be reasonably stable since there are p related DFT coefficients. Following this step, instead of finding other directional sums, we find linear constraints that the binary matrix satisfies along these directions. Finally, we search for a binary matrix that matches all these constraints and any remaining DFT coefficients. This algorithm is summarized in [Algorithm 3](#). The runtime considerations of [Algorithm 3](#) are similar to [Algorithm 2](#), where the LLL steps of the algorithm scale like $(p^2)^7$.

Algorithm 3 Reconstruction algorithm for $p^2 \times p^2$ binary matrices with prime p .

- 1: **Input:** DFT Coefficients \tilde{X}_{kl} for all $|k|, |l| \leq p$
 - 2: **Input:** Error tolerance ϵ
 - 3: **Output:** $p^2 \times p^2$ binary matrix \mathbf{X}
 - 4: **for** DFT Direction in $\{\tilde{X}_{0k}, \tilde{X}_{k0}, \tilde{X}_{kk}, \tilde{X}_{k,-k}\}$ **do**
 - 5: Use LLL algorithm with the DFT coefficient $k = p$ to reconstruct sums D along lines modulo p
 - 6: Use LLL algorithm with the DFT coefficients $1 \leq k < p$ and D to reconstruct individual directional sums \mathbf{d}_j
 - 7: **end for**
 - 8: **for** Remaining Direction **do**
 - 9: Collect M available DFT coefficients corresponding to direction j
 - 10: **if** $M > 1$ **then**
 - 11: Use LLL with the M DFT coefficients (without \tilde{X}_{00} to find shortest vector
 - 12: **if** $\|\text{shortest vector}\|_\infty < \epsilon$ **then**
 - 13: Successful Recovery: Create corresponding block \mathbf{D}_j that contains linear constraints which shortest vector obeys
 - 14: **end if**
 - 15: **end if**
 - 16: **end for**
 - 17: Create matrix with blocks \mathbf{D}_j for all j corresponding to a successful recovery and \mathbf{A} containing DFT terms for unsuccessful recoveries
 - 18: Create right hand side vector with corresponding blocks \mathbf{d}_j and the DFT coefficients
 - 19: Use ILP to solve for binary matrix \mathbf{X} using this matrix and right hand side
-

4.4. Stability. The intermediate steps in the algorithms described in the previous section center on finding integer coefficients for a cyclotomic integer to equal a known value, within some precision. For example, in [Algorithm 1](#), one first attempts to reconstruct the column sums by finding a cyclotomic integer of prime order p whose integer coefficients are bounded by prime q , that matches the value of \tilde{X}_{01} . In [Algorithm 2](#), the same problem is considered, although it can be for one of $p + 1$ potential directions, with possibly more than one corresponding DFT coefficient. Therefore, the key question when it comes to stability is how close

can two distinct cyclotomic integers be to one another?

Consider two distinct cyclotomic integers $A = \sum_{j=1}^p a_j (\zeta_p)^j$ and $B = \sum_{j=1}^p b_j (\zeta_p)^j$. Define $e_j = a_j - b_j$ so that $E = A - B = \sum_{j=1}^p e_j (\zeta_p)^j$. We wish to estimate how close E can be to the origin of the complex plane. Finding the exact solution to this problem is difficult [44, 24]. However, we can provide a heuristic estimate. This will yield some insight towards the level of stability we can expect when reconstructing directional sums.

Consider a direction with M available corresponding DFT coefficients. These DFT coefficients are of the form

$$\left[\sum_{j=1}^p a_j (\zeta_p)^j, \sum_{j=1}^p a_j (\zeta_p)^{2j}, \dots, \sum_{j=1}^p a_j (\zeta_p)^{Mj} \right].$$

If the coefficients a_j and b_j are bounded between 0 and K , the coefficients e_j satisfy $-K \leq e_j \leq K$. Moreover, as the total popcount $S = \tilde{X}_{00}$ is known, we have $\sum_{j=1}^p e_j = 0$. If the process of finding integer coefficients that agree with all the available DFT coefficients is unstable, then it is possible that all entries of the vector

$$\mathbf{e} = \left[\sum_{j=1}^p e_j (\zeta_p)^j, \sum_{j=1}^p e_j (\zeta_p)^{2j}, \dots, \sum_{j=1}^p e_j (\zeta_p)^{Mj} \right]$$

are small. For small R , we will determine an approximate condition for which $\|\mathbf{e}\|_\infty \leq R$. Let $\rho(R)$ be the expected number of valid vectors \mathbf{e} satisfying $\|\mathbf{e}\|_\infty \leq R$. We model each term of the form $\sum_{j=1}^p e_j (\zeta_p)^{kj}$ in \mathbf{e} as a sum of $\sum_{j=1}^p |e_j|$ uniform random points on the unit circle. As $n \rightarrow \infty$, the probability that a sum of n random points on the unit circle has length at most R approaches $1 - \exp(-R^2/n)$ [23]. The x - and y -coordinates approach independent normal distributions with the standard deviation $\sqrt{n/2}$ by the central limit theorem. Using this approximation, the probability that all entries of \mathbf{e} are less than or equal to R is $\left[1 - \exp\left(-R^2/\sum_{j=1}^p |e_j|\right)\right]^M$. By linearity of expectations, we can approximate the expectation that $\|\mathbf{e}\|_\infty < R$ as a sum over all valid choices of e_j , viz,

$$(33) \quad \rho(R) \approx \sum_{\substack{|e_j| < K \\ \sum_{j=1}^p e_j = 0}} \left(1 - e^{-R^2/\sum_{j=1}^p |e_j|}\right)^M.$$

Since R is small, we approximate each term inside the sum using the linearization $e^x \approx 1 + x$. Moreover, similar to (24), by the inclusion-exclusion principle, one can compute the total number of terms in this sum to be

$$\nu(p, K) = \sum_{n \geq 0} (-1)^n \binom{p}{n} \binom{Kp - n(2K + 1) + p - 1}{p - 1}.$$

By setting $\sum |e_j| = (2K + 1)p/4$, which is roughly its average value, in (33), and replacing the sum over its $\nu(p, K)$ choices of e_j , we have the reduced approximation

$$(34) \quad \rho(R) \approx \nu(p, K) [4R^2/(2K + 1)p]^M.$$

The only solution to $\sum_{j=1}^p e_j (\zeta_p)^{kj} = 0$ that also satisfies $\sum_{j=1}^p e_j = 0$ is $e_j = 0$. Thus we expect that, for small enough R , $\rho(R) \approx 1$. Setting this equal to our approximation (34) and solving for R , we find

$$R^{2M} \approx (2K + 1)^M (p/4)^M [\nu(p, K)]^{-1},$$

so that we expect to require roughly

$$(35) \quad -\log(R) = \log(2) - \log\left(\sqrt{p(2K + 1)}\right) + \log(\nu(p, K))/2M.$$

digits of precision to distinguish integer coefficients for the M cyclotomic integers.

We emphasize that the result (35) is an approximation, and may not be accurate for small p . A more careful analysis would additionally account for the fact that sums of few points on the unit circle are significantly more likely to be small. However, when K^2 is large compared to p (which is typical in applications), this contribution becomes negligible. So, we content ourselves with the above heuristic, keeping in mind that it may underestimate the precision needed.

5. Numerical examples. We next conduct numerical simulations to test the proposed recovery algorithms. In our implementation of all three algorithms, we use MATLAB's built-in solver for ILP, `intlinprog`, which uses cutting planes and other preprocessing steps to reduce the size of the computational domain. The prescribed stopping condition for any call of `intlinprog` was set to checking 10^7 possible matrices.

The implemented LLL algorithm was programmed in MATLAB. After sufficient testing, the large constant parameter in (27) was set to $\beta = 10^8$. The error tolerance to determine if a vector is sufficiently short was $\epsilon = 0.001$. In our implementations, we made one modification for practical time considerations. Some runs of the LLL-algorithm can take a very long time and ultimately fail to recover a sufficiently short vector. To avoid waiting too long for a failed recovery, we set a time limit on the LLL algorithm to 5 seconds. This stopping criterion was found to be a good balance between minimizing the computation time while not overlooking any feasible reconstructions. All computations were carried out in double precision.

5.1. Algorithm 1 for rectangular matrices. For different prime values of N_1 and N_2 , a model $N_1 \times N_2$ binary matrix generated with $S = \lfloor N_1 N_2 / 2 \rfloor$ nonzero entries, chosen uniformly at random. Algorithm 1 was run given the 4 DFT coefficients \tilde{X}_{00} , \tilde{X}_{10} , \tilde{X}_{01} , and \tilde{X}_{11} , to try to recover the original binary matrix exactly. For fixed values of N_1 and N_2 , this experiment was run 30 times, with the average timing (in seconds) displayed in Table 1.

N_1	N_2	t , sec.	n_a
5	7	0.05	2
5	11	8	4
5	13	10	5
7	11	64	5
7	13	84	6
7	17	111	8
11	13	91	7

Table 1

Time t for reconstructing $N_1 \times N_2$ binary matrices with $S = \lfloor N_1 N_2 / 2 \rfloor$ nonzero entries using Algorithm 1. Averages for 30 randomly-generated model matrices are displayed. The column n_a displays the number of digits needed for stable recovery of the column sums with $M = 1$ according to (35).

Algorithm 1 was able to consistently recover the original binary matrix for dimensions as large as 11×13 . A recovery is considered successful if it reconstructs all elements of the matrix correctly. For all of the dimensions displayed in [Table 1](#), **Algorithm 1** was successful in all 30 trial runs. Matrices of dimension 11×13 were the largest that could be reliably recovered within the prescribed stopping criteria, which took on average one and a half minutes. Note that

7×17 matrices have fewer elements but tended to take longer to be recovered due to the larger column dimension making the row sum recovery more computationally demanding.

As a comparison, we note that naively running ILP on the entire system [\(26\)](#), as opposed to first considering the subproblems of recovering row and column sums, took on average 2.3 seconds (about 40 times longer than by **Algorithm 1**) for 5×7 matrices, and was unable to scale to 5×11 matrices under the prescribed stopping conditions.

A band-limited reconstruction and the reconstruction by **Algorithm 1** of a sample 11×13 binary matrix are shown in [Figure 2](#). The band-limited reconstruction $\mathbf{X}^{(\text{blurred})}$ is given by

$$(36) \quad X_{mn}^{(\text{blurred})} = \frac{1}{N_1 N_2} \sum_{k=-L_1}^{L_1} \sum_{l=-L_2}^{L_2} \tilde{X}_{kl} e^{-2\pi i (mk/N_1 + nl/N_2)},$$

which is identical to [\(3\)](#) except the summation now runs only over the available pass band with parameters L_1 and L_2 . For this $N_1 \times N_2$ case, we set $L_1 = L_2 = 1$ for the band-limited reconstruction in [\(36\)](#), but in accordance with [Theorem 5](#), the terms with $\tilde{X}_{1,-1}$ and $\tilde{X}_{-1,1}$ are removed from the sum (in the reconstruction, these DFT coefficients are not used). It can be seen that, with such few data points, the band-limited reconstruction has little resemblance to the original matrix. However, a reconstruction that takes into account the matrix binarity returns the model exactly

According to [\(35\)](#), the subproblems of reconstructing the row- and column-wise sums is stable when working in double precision, even with only one DFT coefficient (either \tilde{X}_{01} or \tilde{X}_{10}). The number of digits after the decimal place estimated by the heuristic for $M = 1$ for reconstructing the column sums (the more difficult direction) are given in the last column of [Table 1](#). For example, for 11×13 matrices, the heuristic suggests that we need about 7 digits to stably reconstruct the column sums. This corresponds to noise level of the magnitude of $\sim 10^{-8}$ relative to \tilde{X}_{10} , which is of the order of unity. If we reduce the number of known digits to 6 (noise level of $\sim 10^{-7}$), the column sums for the 30 randomly generated models are reconstructed correctly in 6 cases. This strong instability can be rectified by including more DFT coefficients in the data set, beyond the minimum required for theoretical uniqueness. For the band limit parameter $L = 2$, we have $M = 2$ available DFT coefficients for reconstructing the column sums (\tilde{X}_{10} and \tilde{X}_{20}). In this case, the column sums for all 30 matrices are recovered

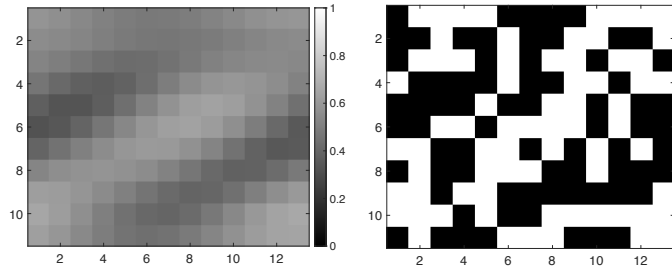


Figure 2. Left: reconstruction with only 4 available DFT coefficients (as required by [Theorem 5](#)) of a randomly generated binary 11×13 matrix with $S = 71$ ones. Right: reconstruction obtained by [Algorithm 1](#) (coincides exactly with the model).

correctly with only three significant digits in the data.

5.2. Algorithm 2 for $N \times N$ matrices with prime N . Algorithm 2 was run on 100 randomly-generated $N \times N$ binary matrices, for $N = 17, 19, 23, 29$. In each case, the global popcount was set to $S = \lfloor N^2/2 \rfloor$, which is the most difficult case. The results of the simulations are summarized in Table 2, which contains the average run time of the algorithm, the percentage of model matrices that were exactly recovered, and how many directional sums (out of $N + 1$) were recovered on average by the LLL algorithm. Additionally the last column displays the stability estimate (35), in terms of the number of digits in the data, for the most unstable directional sum recoveries with $M = 2$ DFT coefficients (as any with $M = 1$ are automatically skipped).

For $N = 17$, when $L = \lfloor \sqrt{17} \rfloor = 4$, the algorithm was able to reconstruct all 100 models in an average of 2 seconds. This is substantially faster than the implementation of Algorithm 1, as we are now using a larger bandwidth of available DFT coefficients in accordance with the theory. The larger bandwidth provides more coefficients than are minimally required for uniqueness, which improves computational speed and stability. For example, we now have access to \tilde{X}_{20} , which provides equivalent information to \tilde{X}_{10} . This increased stability allows us to use LLL algorithm, which runs much faster than ILP. Out of the $N + 1 = 18$ possible directions, 4 directional sums are skipped in the algorithm for having only 1 corresponding DFT coefficient. On average 13.98 (this number is rounded off as 14 in Table 2) of the remaining 14 directions were reconstructed accurately. Note that the final ILP step of the algorithm finds the unique solution quickly as is not a bottleneck.

As we increase the dimensions to $N = 19$, but keep $L = 4$, the average run time increases to about 8 seconds. There are now 8 directions that are skipped due to having only 1 DFT coefficient, and the algorithm reconstructs 11 of the remaining 12 directions on average. Most notably, we have our first instance of failed reconstruction where exactly 1 model matrix was not reconstructed accurately (out of 100). The algorithm in this case fails by ILP reporting that the linear system is inconsistent over the integers. Upon closer investigation, it is seen that the inconsistent system is caused by one of the LLL solves finding an incorrect directional sum due to an instability – it found a sufficiently short vector, but not the correct one. Even though the stability heuristic suggests that 5 digits should be enough for stability, this is not the case for this model. It is not altogether surprising that there is an outlier, as the

N	L	Rec., %	t , sec.	Dir.	n_d
17	4	100	2	14	5
19	4	99	8	11	6
19	5	100	3	20	
23	4	0	–	8	8
23	5	100	3	20	
29	5	96	5	16	11
29	6	100	12	24	

Table 2

Summary of reconstruction results for $N \times N$ binary matrices with global popcount $S = \lfloor N^2/2 \rfloor$ using Algorithm 2. The parameter L indicates the pass band used. The next column displays the percentage of exact recoveries for 100 randomly-generated model matrices. The next two columns display the average timings and the average number of recovered directional sums. The last column displays the number of digits estimated by (35) that are needed for stable recovery of a directional sum with $M = 2$ DFT coefficients. Reconstruction that took longer than the prescribed stopping condition is denoted by a dash.

heuristic was based on statistical arguments. The fast notification of failure by the algorithm is important, as it did not return a misleading answer. This reconstruction could be remedied by removing one of the reconstructed directional sums by trial and error until ILP runs successfully. Another alternative for reconstructing this failed model is to improve stability by increasing the number of available DFT coefficients. When L is increased to 5, which is more than required for uniqueness, all 100 models are reconstructed, in an average of under 3 seconds, where all $N + 1 = 20$ directions are almost always reconstructed.

The $N = 23$ case is an interesting example. With $L = 4$, only 8 out of the 24 directions have more than 1 corresponding DFT coefficient. These 8 directional sums are accurately reconstructed for each model. However, this does not provide enough information for making the final ILP step and finding the unique solution before the prescribed stopping criteria. Stability issues prevent reconstruction of the correct matrix if we remove this restriction on directional sums with only one coefficient ((35) suggests that about 17 digits are required for $M = 1$). If we increase L and take $L = 5$, the algorithm works for all models.

The case $N = 29$ has similar behavior to the $N = 19$ case. At the minimal band limit parameter $L = 5$, the algorithm almost always successfully recovers the model binary matrix, failing for 4 out of the 100 models. A band-limited reconstruction and the reconstruction by Algorithm 2 with $L = 5$ of a sample model 29×29 binary matrix are shown in Figure 3. When the algorithm fails, it fails, as above, by an unstable LLL step that causes an inconsistency in the ILP step. This can, again, be remedied by increasing L to 6. This increases the average time from about 5 seconds to 12 seconds, but recovers more directions on average (24 as opposed to 16).

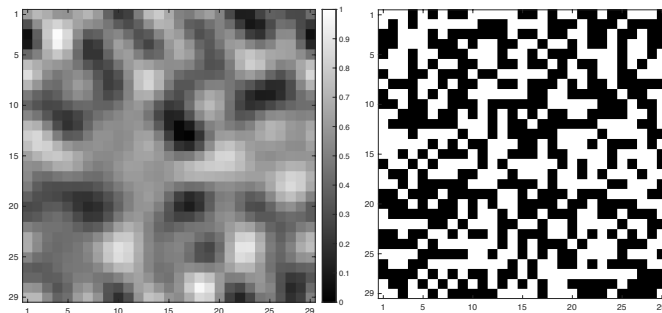


Figure 3. Left: band-limited reconstruction with $L = 5$ of a randomly-generated binary 29×29 matrix with the global popcount $S = 420$. Right: reconstruction by Algorithm 2 (exactly coincides with the model).

As an example of reconstruction with noisy data, we have added Gaussian white noise to the DFT coefficients of the model in Figure 3 with variance 10^{-4} which only corrupted the DFT coefficients beyond 3 digits past the decimal point. Reconstruction failed until L was increased to $L = 9$. At this bandwidth, 21 out of the 30 directional sums were recovered and the model was exactly reconstructed. Importantly, the smallest number of DFT coefficients for any direction is now $M = 4$. This value of M requires 5 digits for stability according to (35). However, the reconstruction in this case outperforms the heuristic.

Based on the success of recovering random 29×29 binary matrices, as a motivated example we seek to recover a blurred QR code. A 29×29 QR code that encodes the phrase “DiscreteFourierTransform” was generated according to the standard format specifications, known as a Version 3 QR code for this size. With the minimum bandwidth required for unique recovery $L = 5$, Algorithm 2 was run on this incomplete set of DFT coefficients. Note that no additional QR code information was used – the image was treated by the algorithm as a

general binary matrix. For example, Version 3 QR codes have fixed patterns, including the recognizable position detector patterns present in three of the corners. Even though these fixed patterns are known based on the size of the QR code, the algorithm treats these as general regions which need to be reconstructed. This QR code information could certainly be added to the algorithm to improve computational speed and stability. The blurred QR code and its reconstruction using [Algorithm 2](#) (which exactly recovers the original code) are displayed in [Figure 4](#). The reconstruction was done in about 6 seconds, with 15 out of the possible 30 directions recovered before the ILP solve.

5.3. Algorithm 3 for $p^2 \times p^2$ matrices with prime p . Finally, [Algorithm 3](#) was tested on 25×25 binary matrices. With the available computational resources, [Algorithm 3](#) was unable to scale to the next prime power of 49×49 . Similar to the experiment performed for [Algorithm 2](#), we tested the algorithm on 100 randomly generated binary matrices in the most computationally difficult regime of $S = \lfloor 25^2/2 \rfloor = 312$ nonzero entries.

With all DFT coefficients within the band limit defined by $L = 5$, [Algorithm 3](#) was able to exactly recon-

struct the randomly generated binary matrix 87 out of 100 times in an average of about 25 seconds. This average timing includes both successful and failed recoveries. It is understandable that [Algorithm 3](#) performed slightly worse than [Algorithm 2](#), as we can only reconstruct certain linear constraints for many of the directions for $p^2 \times p^2$ matrices, as opposed to the directional sum values themselves. In all 100 simulations, the algorithm correctly recovered the only directional sums that are determinable: row, column, and diagonal directions. There were 26 remaining directions, with 12 of these directions automatically skipped for having only one corresponding DFT coefficient. Of the remaining 14 directions, the algorithm successfully found constraints (as measured by finding a corresponding sufficiently short vector) for 10 of these, on average. Whenever the algorithm failed, it was again due to the ILP step finding an inconsistent system, which was caused by an instability (incorrect solve) in finding constraints for one of the directions.

As a final practical test, the phrase “Binary Matrix Recovery” was encoded in a 25×25 Version 2 QR code. The true binary image has $S = 287$ nonzero entries. With access to the DFT coefficients inside the bandwidth of $L = 5$, [Algorithm 3](#) was able to exactly reconstruct the original QR code in about 24 seconds. This reconstruction and the corresponding band-limited (blurred) image are displayed in [Figure 5](#).

6. Discussion. We have shown that prior information that a matrix is binary allows one to reconstruct this matrix exactly from a limited set of DFT coefficients. Theoretically, for $N_1 \times N_2$ matrices with $N_1 \neq N_2$ both prime, only 4 DFT coefficients are needed to guarantee

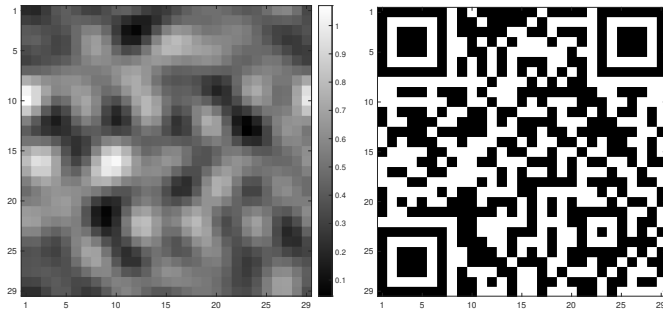


Figure 4. Left: band-limited reconstruction with $L = 5$ of a 29×29 Version 3 QR code with $S = 410$ nonzero entries. Right: reconstruction by [Algorithm 2](#) (exactly coincides with the model).

uniqueness of this reconstruction regardless of the magnitudes of N_1 and N_2 . For $N \times N$ matrices with a prime N , the number of required coefficients grows with N , but at a reasonable rate; the minimum band limit required for unique reconstruction is in this case $L = \lfloor \sqrt{N} \rfloor$. For square matrices of order $N = p^\alpha$, where p is prime and $\alpha > 1$ an integer, the minimum band limit is increased to $L = p^{\alpha-1}$.

However, there exists a sizable gap between the theoretical guarantees of uniqueness and what is practical. The provided stability heuristics, which are supported by numerical examples, indicate that many digits of precision are needed in the data for reconstruction at the theoretical bounds. However, we have shown that it is possible to solve the problem even with a realistic amount of noise or imprecision in the DFT data by increasing the band limit past the theoretical bound while still not making all the coefficients available (in fact,

far from that). This can also be understood by comparing the cases of square and non-square matrices with prime dimensions. In the former case, the band limit required to guarantee stability is significantly larger than in the latter case. However, we can always make a matrix square by making it larger (i.e., by adding rows or columns). Thus the theoretical results are counter-intuitive. For example, more DFT coefficients are required to recover uniquely a 29×29 matrix than a 29×31 matrix. However, with the account of stability, the apparent contradiction disappears. In order to reconstruct the two matrices *stably*, approximately the same number of DFT coefficients is needed.

In the numerical simulations, the algorithms combining integer linear programming (ILP) and Lenstra-Lenstra-Lovasz (LLL) lattice reduction were able to efficiently recover matrices as large as 29×29 . In comparison, naive implementations of the ILP techniques fail for matrices as small as 5×11 . However, even 29×29 matrices are on the smaller side of two-dimensional barcodes. It is therefore an open task to develop improved algorithms to handle larger binary matrix recovery in reasonable time. The current work mainly investigates recovery near the minimal band limit for uniqueness. It is worthwhile to investigate how these algorithms scale for larger matrices when L is significantly larger than the minimum, while still not using all DFT coefficients. Fast Fourier transform (FFT) and sparse FFT techniques are applicable when allowing for sampling of coefficients outside of the pass-band. With even sparse sampling of a few high-frequency DFT coefficients could lead to scalable FFT based algorithms that have a smaller gap between theoretical results and practical reconstruction.

Additional constraints such as sparsity and connectivity can further increase computational feasibility for larger binary matrices, and allow for reconstruction with more significant noise. Sparse matrices with relatively small popcount S can be considered straightforwardly by the algorithms developed here, and smallness of S always entails greatly improved compu-

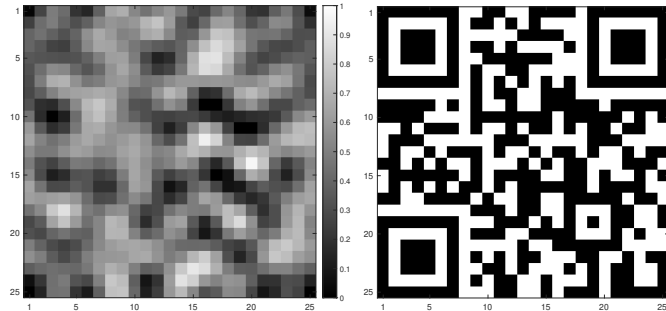


Figure 5. Left: band-limited reconstruction with $L = 5$ of a Version 2 25×25 QR code with $S = 287$ nonzero entries. Right: reconstruction by [Algorithm 3](#) (exactly coincides with the model).

tational efficiency, with potential modifications. For example, repeating the numerical experiment for [Algorithm 2](#) from [Section 5.2](#) for $N = 29$ with smaller popcount $S = 52$ resulted in about a 1 second reduction in average runtime (from 5 seconds to 4 seconds). However, small modifications to the algorithm can increase computational efficiency further. The overall size of the problem is significantly smaller for modest values of S . In these cases, it is likely that fewer explicit directional sums are required to further reduce the overall problem to a manageable size. For this same experiment with $S = 52$, modifying the algorithm to only solve for four directional sums (row, column, and diagonals) resulted in an average run time of about 1.4 seconds, where all 100 randomly generated model matrices were successfully recovered. Optimizing the algorithms for smaller values of S is key ongoing work. Connectivity is a conceptually different constraint, and its application can lead to improvements even for non-sparse matrices.

Lastly, for applications to denoising corrupted QR codes, the algorithm can have improved computational efficiency by including additional prior information based on known QR code features. This includes fixed patterns, as well as masking that promotes disconnected images. QR codes also have built-in error correcting methods [\[62\]](#). Combining this error correction with the proposed algorithms may yield efficient recovery with minimal available DFT coefficients and larger matrix sizes than 29×29 .

REFERENCES

- [1] K. AARDAL, C. A. J. HURKENS, AND A. K. LENSTRA, *Solving a system of linear Diophantine equations with lower and upper bounds on the variables*, [Math. Operations Res.](#), **25** (2000), pp. 427–442.
- [2] J. BAILEY, M. A. IWEN, AND C. V. SPENCER, *On the design of deterministic matrices for fast recovery of Fourier compressible functions*, [SIAM J. Matr. Analysis Appl.](#), **33** (2012), pp. 263–289.
- [3] A. BECK AND M. TEBoulLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, [SIAM J. Imag. Sci.](#), **2** (2009), pp. 183–202.
- [4] T. BLUMENSATH, *Accelerated iterative hard thresholding*, [Sign. Proc.](#), **92** (2012), pp. 752–756.
- [5] T. BLUMENSATH, *Compressed sensing with nonlinear observations and related nonlinear optimization*, [IEEE Trans. Info. Theor.](#), **59** (2013), pp. 3466–3474.
- [6] T. BLUMENSATH AND M. E. DAVIES, *Iterative hard thresholding for compressed sensing*, [Appl. Comp. Harm. Anal.](#), **27** (2009), pp. 265–274.
- [7] M. BORN AND E. WOLF, *Principles of Optics*, Cambridge Univ. Press, 1999.
- [8] J. M. BORWEIN AND P. LISONĚK, *Applications of integer relation algorithms*, [Discr. Math.](#), **217** (2000), pp. 65–82.
- [9] R. A. BRUALDI, *Matrices of zeros and ones with fixed row and column sum vectors*, [Lin. Alg. Appl.](#), **33** (1980), pp. 159–231.
- [10] R. A. BRUALDI AND E. S. SOLHEID, *On the spectral radius of complementary acyclic matrices of zeros and ones*, [SIAM J. Alg. Disc. Meth.](#), **7** (1986), pp. 265–272.
- [11] E. J. CANDÈS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, [IEEE Trans. Info. Theor.](#), **52** (2006), pp. 489–509.
- [12] J. CONWAY AND A. JONES, *Trigonometric diophantine equations (On vanishing sums of roots of unity)*, [Acta Arithmetica](#), **30** (1976), pp. 229–240.
- [13] A. DEL LUNGO, A. FROSINI, M. NIVAT, AND L. VUILLON, *Discrete tomography: Reconstruction under periodicity constraints*, in [International Colloquium on Automata, Languages, and Programming](#), Springer, 2002, pp. 38–56.
- [14] S. DI ZENZO, L. CINQUE, AND S. LEVIALDI, *Run-based algorithms for binary image analysis and processing*, [IEEE Trans. Pattern. Anal. Mach. Intel.](#), **18** (1996), pp. 83–89.
- [15] D. L. DONOHO AND P. B. STARK, *Uncertainty principles and signal recovery*, [SIAM J. Appl. Math.](#), **49**

- (1989), pp. 906–931.
- [16] M. ELAD AND A. M. BRUCKSTEIN, *A generalized uncertainty principle and sparse representation in pairs of bases*, *IEEE Trans. Info. Theor.*, 48 (2002), pp. 2558–2567.
- [17] H. FERGUSON, D. BAILEY, AND S. ARNO, *Analysis of PSLQ, an integer relation finding algorithm*, *Math. Comp.*, 68 (1999), pp. 351–369.
- [18] D. R. FULKERSON, *Zero-one matrices with zero trace*, *Pacific J. Math.*, 10 (1960), pp. 831–836.
- [19] J. Z. GAO, L. PRAKASH, AND R. JAGATESAN, *Understanding 2D-barcode technology and applications in M-commerce-design and implementation of a 2D barcode processing solution*, in *31st Ann. Int. Computer Software and Applications Conference*, vol. 2, IEEE, 2007, pp. 49–56.
- [20] R. GARDNER AND P. GRITZMANN, *Discrete tomography: Determination of finite sets by X-rays*, *Trans. Am. Math. Soc.*, 349 (1997), pp. 2271–2295.
- [21] R. J. GARDNER, P. GRITZMANN, AND D. PRANGENBERG, *On the computational complexity of reconstructing lattice sets from their X-rays*, *Discr. Math.*, 202 (1999), pp. 45–71.
- [22] A. C. GILBERT, P. INDYK, M. IWEN, AND L. SCHMIDT, *Recent developments in the sparse Fourier transform: A compressed Fourier transform for big data*, *IEEE Signal Proc. Mag.*, 31 (2014), pp. 91–100.
- [23] J. A. GREENWOOD AND D. DURAND, *The distribution of length and components of the sum of n random unit vectors*, *Ann. Math. Stat.*, 26 (1955), pp. 233–246.
- [24] P. HABEGGER, *The norm of Gaussian periods*, *Quart. J. Math.*, 69 (2018), pp. 153–182.
- [25] L. HAJDU AND R. TIJDEMAN, *Algebraic aspects of discrete tomography*, *J. Reine Angew. Math.*, 534 (2001), pp. 119–128.
- [26] R. HARTSHORNE, *Geometry: Euclid and Beyond*, Undergraduate Texts in Mathematics, Springer, 2013.
- [27] J. HASTAD, B. JUST, J. C. LAGARIAS, AND C.-P. SCHNORR, *Polynomial time algorithms for finding integer relations among real numbers*, *SIAM J. Computing*, 18 (1989), pp. 859–881.
- [28] G. T. HERMAN AND A. KUBA, *Discrete tomography in medical imaging*, *Proc. IEEE*, 91 (2003), pp. 1612–1626.
- [29] G. T. HERMAN AND A. KUBA, *Discrete Tomography: Foundations, Algorithms, and Applications*, Springer, 2012.
- [30] W. HU, G. CHEUNG, A. ORTEGA, AND O. C. AU, *Multiresolution graph Fourier transform for compression of piecewise smooth images*, *IEEE Trans. Imag. Proc.*, 24 (2014), pp. 419–433.
- [31] C. V. JIJI, P. NEETHU, AND S. CHAUDHURI, *Alias-free interpolation*, in *Computer Vision - ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, eds., Springer, 2006, pp. 255–266.
- [32] R. M. KARP, *Reducibility among combinatorial problems*, in *Complexity of Computer Computations*, Springer, 1972, pp. 85–103.
- [33] R. L. LAGENDIJK AND J. BIEMOND, *Iterative Identification and Restoration of Images*, Springer, 1990.
- [34] T. Y. LAM AND K. H. LEUNG, *On vanishing sums of roots of unity*, *J. Algebra*, 224 (2000), pp. 91–109.
- [35] A. K. LENSTRA, H. W. LENSTRA, AND L. LOVÁSZ, *Factoring polynomials with rational coefficients*, *Mathematische Annalen*, 261 (1982), pp. 515–534.
- [36] H. W. LENSTRA, *Vanishing sums of roots of unity*, in *Proc. Bicentennial Congress Wiskundig Genootschap, Part II*, Vrije Univ. Amsterdam, 1978, pp. 249–268.
- [37] H. W. LENSTRA, *Integer programming with a fixed number of variables*, *Math. Oper. Res.*, 8 (1983), pp. 538–548.
- [38] H. W. LEVINSON AND V. A. MARKEL, *Binary discrete Fourier transform and its inversion*, *IEEE Trans. Sign. Proc.*, 69 (2021), pp. 3484–3499.
- [39] X. LIU, D. ZHAI, D. ZHAO, G. ZHAI, AND W. GAO, *Progressive image denoising through hybrid graph laplacian regularization: A unified framework*, *IEEE Trans. Imag. Proc.*, 23 (2014), pp. 1491–1503.
- [40] H. MARCHAND, A. MARTIN, R. WEISMANTEL, AND L. WOLSEY, *Cutting planes in integer and mixed integer programming*, *Discr. Appl. Math.*, 123 (2002), pp. 397–446.
- [41] S. MARCHAND-MAILLET AND Y. M. SHARAIHA, *Binary Digital Image Processing: A Discrete Approach*, Elsevier, 1999.
- [42] A. A. MAZNEV AND O. B. WRIGHT, *Upholding the diffraction limit in the focusing of light and sound*, *Wave Motion*, 68 (2017), pp. 182–189.
- [43] A. MOITRA, *Super-resolution, extremal functions and the condition number of Vandermonde matrices*, in *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, ACM, 2015,

- pp. 821–830.
- [44] G. MYERSON, *How small can a sum of roots of unity be?*, *The American Mathematical Monthly*, 93 (1986), pp. 457–459.
 - [45] K. NASROLLAHI AND T. B. MOESLUND, *Super-resolution: A comprehensive survey*, *Machine Vision and Applications*, 25 (2014), pp. 1423–1468.
 - [46] P. Q. NGUYEN AND D. STEHLÉ, *Low-dimensional lattice basis reduction revisited*, in *International Algorithmic Number Theory Symposium*, Springer, 2004, pp. 338–357.
 - [47] S.-C. PEI AND K.-W. CHANG, *Binary signal perfect recovery from partial DFT coefficients*, *IEEE Trans. Sign. Proc.*, 70 (2022), pp. 3848–3861.
 - [48] G. PLONKA AND T. VON WULFFEN, *Deterministic sparse sublinear FFT with improved numerical stability*, *Results in Mathematics*, 76 (2021), p. 53.
 - [49] G. PLONKA, K. WANNENWETSCH, A. CUYT, AND W.-S. LEE, *Deterministic sparse FFT for M -sparse vectors*, *Numerical Algorithms*, 78 (2018), pp. 133–159.
 - [50] D. RAJAN AND S. CHAUDHURI, *Generalized interpolation and its application in super-resolution imaging*, *Image and Vision Computing*, 19 (2001), pp. 957–969.
 - [51] H. RAUHUT, *Random sampling of sparse trigonometric polynomials*, *Appl. Comp. Harm. Anal.*, 22 (2007), pp. 16–42.
 - [52] M. REN, J. YANG, AND H. SUN, *Tracing boundary contours in a binary image*, *Image Vision Comp.*, 20 (2002), pp. 125–131.
 - [53] Y. ROMANO, M. ELAD, AND M. PEYMAN, *The little engine that could: Regularization by denoising (RED)*, *SIAM J. Imag. Sci.*, 10 (2017), pp. 1804–1844.
 - [54] M. RUDELSON AND R. VERSHYNIN, *Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements*, in *2006 40th Annual Conference on Information Sciences and Systems*, IEEE, 2006, pp. 207–212.
 - [55] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, *Physica D*, 60 (1992), pp. 259–268.
 - [56] H. J. RYSER, *Combinatorial properties of matrices of zeros and ones*, *Canadian J. Math.*, 9 (1957), pp. 371–377.
 - [57] H. J. RYSER, *Matrices of zeros and ones*, *Bull. Am. Math. Sci.*, 66 (1960), pp. 442–464.
 - [58] A. SCHRIJVER, *On cutting planes*, *Ann. Discr. Math.*, 9 (1980), pp. 291–296.
 - [59] V. SHOUP, *A Computational Introduction to Number Theory and Algebra*, Cambridge Univ. Press, 2009.
 - [60] T. A. B. SNIJDERS, *Enumeration and simulation methods for 0–1 matrices with given marginals*, *Psychometrika*, 56 (1991), pp. 397–417.
 - [61] T. TAO, *An uncertainty principle for cyclic groups of prime order*, *Math. Res. Lett.*, 12 (2005), pp. 121–127.
 - [62] S. TIWARI, *An introduction to QR code technology*, in *International Conference on Information Technology (ICIT)*, IEEE, 2016, pp. 39–44.
 - [63] J. A. TROPP, *On the linear independence of spikes and sines*, *J. Fourier Anal. Appl.*, 14 (2008), p. 838.
 - [64] J. A. TROPP AND A. C. GILBERT, *Signal recovery from random measurements via orthogonal matching pursuit*, *IEEE Trans. Info. Theor.*, 53 (2007), pp. 4655–4666.
 - [65] B.-Y. WANG AND F. ZHANG, *On the precise number of $(0,1)$ -matrices in $A(R,S)$* , *Discr. Math.*, 187 (1998), pp. 211–220.
 - [66] A. E. YAGLE, *An algebraic solution to the 3-D discrete tomography problem*, in *Proc. 1998 Int. Conf. Image Process.*, vol. 2, IEEE, 1998, pp. 714–717.
 - [67] A. E. YAGLE, *A convergent composite mapping Fourier domain iterative algorithm for 3-D discrete tomography*, *Lin. Alg. Appl.*, 339 (2001), pp. 91–109.