Recovery of Integer Images from Limited DFT Measurements with Lattice Methods

Howard W. Levinson[a,*], Isaac Viviano[b]

[a] *Department of Computer Science, Oberlin College, 10 N Professor St, Oberlin 44074, Ohio, USA*
[b] *Oberlin College, 10 N Professor St, Oberlin 44074, Ohio, USA*

Abstract

Exact reconstruction of an image from measurements of its Discrete Fourier Transform (DFT) typically requires all DFT coefficients to be available. However, incorporating the prior assumption that the image contains only integer values enables unique recovery from a limited subset of DFT coefficients. This paper develops both theoretical and algorithmic foundations for this problem. We use algebraic properties of the DFT to define a reduction from two-dimensional recovery to several well-chosen one-dimensional recoveries. Our reduction framework characterizes the minimum number and location of DFT coefficients that must be sampled to guarantee unique reconstruction of an integer-valued image. Algorithmically, we develop reconstruction procedures which use dynamic programming to efficiently recover an integer signal or image from its minimal set of DFT measurements. While the new inversion algorithms still involve NP-hard subproblems, we demonstrate how the divide-and-conquer approach drastically reduces the associated search space. To solve the NP-hard subproblems, we employ a lattice-based framework which leverages the LLL approximation algorithm to make the algorithms fast and practical. We provide an analysis of the lattice method, suggesting approximate parameter choices to ensure correct inversion. Numerical results for the algorithms support the parameter analysis and demonstrate successful recovery of large integer images.

Keywords: 2D DFT, Minimal sampling, Reconstruction, Lattice algorithms

1. Introduction

Reconstructing a vector or matrix from limited discrete Fourier transform (DFT) measurements is a classic problem in signal processing and imaging [1]. This challenge arises in many applications where noise corrupts parts of the data, or measuring the full spectrum is either impractical or prohibitively expensive. Whether or not we control the sampling strategy, successful recovery from a limited spectrum often requires some form of *a priori* knowledge of the underlying signal or image. For example, sparsity constraints have revolutionized the field through the framework of compressed sensing, enabling dramatic reductions in the number of measurements with corresponding faster computation [2–8]. Related approaches study optimal sampling for sparse signals and can be used to develop sparse fast Fourier transform methods which are faster than their standard counterparts [9–13].

While sparsity has dominated much of this research, another type of prior information has also been studied: that the unknown signal or image is restricted to a small set of discrete values [14, 15]. This line of work often leverages the algebraic structure of the DFT in combination with the discrete nature of the signal values. However, to our knowledge, the case of integer-valued data has not yet been comprehensively addressed. In this paper, we study the recovery problem for integer matrices in general. Our first fundamental question is:

*Corresponding author
Email addresses: hlevinson@oberlin.edu (Howard W. Levinson), iviviano@oberlin.edu (Isaac Viviano)

Problem 1. Let $\mathtt{X}$ be an $N_1 \times N_2$ integer matrix. What is a minimal set of DFT coefficients required for $\mathtt{X}$ to be uniquely recoverable?

Our results both characterize the minimal number of Fourier samples needed and describe their locations in the frequency domain.

Even when the sampled spectrum guarantees uniqueness, the reconstruction problem itself is highly nontrivial. Standard continuous optimization methods are often ineffective over the integers, since small errors in the solution can correspond to large errors in the data [16, 17]. Even for the simplest case of binary signals, many discrete optimization methods reduce to solving NP-hard problems [18]. This motivates our second fundamental question:

Problem 2. How can an integer matrix $\mathtt{X}$ be efficiently recovered from a subset of its DFT coefficients?

To address this, we develop a lattice-based framework that leverages approximation algorithms for efficient recovery. Lattice methods have seen successful applications in various domains, including cryptography, integer optimization, and algebraic number theory, making them a natural tool for our setting [19]. Our approach efficiently recovers integer matrices with a moderate range of values and scales to reasonably sized images. We also provide an analysis of the lattice method, which aligns closely with empirical results from numerical simulations.

Beyond the theoretical interest, these questions arise naturally in many practical settings. Integer-valued models readily appear in signal processing [20, 21], tomography [14, 22–26], and binary codes and image processing [27–30]. In all of these applications, data are often quantized or inherently integer-valued.

This work makes four main contributions: We establish a minimal sampling theorem for integer matrices, identifying both the number and location of DFT coefficients required for unique recovery. We then develop a dynamic programming algorithmic framework for efficient reconstruction of integer matrices from partial Fourier data. We next propose a lattice-based implementation of the algorithms to leverage approximation algorithms. Lastly, we analyze conditions for the lattice method to succeed, and show close agreement between the theory and numerical simulations.

1.1. Related Works

The current work directly extends the results of Pei and Chang [31], which studied the problem of recovering binary signals from sampled DFT measurements. Their work proved that a binary signal of length N requires at least $\tau(N)$ DFT coefficients for unique inversion, where $\tau(N)$ is the number of divisors of N. This result readily generalizes to integer signals. Their paper also show this bound is tight when considering the set of all binary signals of length N. While the authors provide an example illustrating how the problem can extend to two dimensions, no general formalization was developed. Our work fully develops and solves this two-dimensional generalization.

Pei and Chang [31] also introduced a divide-and-conquer algorithm for recovering binary signals of length $N = 2^\alpha$. In our current work, we extend this framework to handle signals of arbitrary length N in one dimension before further generalizing to two dimensions. Our approach incorporates dynamic programming for efficiency, which is required by the presence of additional prime factors in N. Furthermore, while their method relied on integer linear programming, our use of lattice-based techniques enables faster recovery and removal of the binary constraint, allowing the new method to scale to larger signals and images.

Our previous works [32, 33] studied related partial data recovery problems for binary vectors and matrices in the bandlimited setting, which restricts the known DFT coefficients to a frequency band defined by $|k|, |l| \leq L$ for a bandwidth parameter L. This bandlimited sampling models blurring of a signal or image with a low-pass filter. The restrictive shape of the band makes it challenging to identify simple formulas for bandwidths L that guarantee uniqueness for general N_1, N_2. Therefore, these works were limited to the case where $N_1 = N_2 = p^\alpha$ for a prime p, or N_1 and N_2 were different primes. While the algorithm framework developed in the current work builds on these methods, the presence of multiple prime factors in N_1 and N_2 enables a new dynamic programming approach. Moreover, in the bandlimited setting, the minimal band required for uniqueness includes more coefficients than are strictly required. In contrast, this work focuses on recovery from a minimal set of DFT coefficients. The pass-band shape introduces an asymmetry between the

number of DFT coefficients available to each dynamic programming subproblem. Subproblems with more data have improved stability, so the bandlimited setup can facilitate more stable algorithms which leverage both lattice methods and integer linear programming.

Further related works address binary reconstruction from incomplete Fourier data, both in the bandlimited setting [34, 35] and with unrestricted sampling [36]. Other approaches exploit discrete-valued models in alternative bases to enhance reconstruction and guide sampling strategies [37]. Our uniqueness results focus on the algebraic structure of the DFT, and many works similarly leverage algebraic properties to improve reconstruction methods and establish uncertainty principles [38–40].

Lastly, our lattice methods use the LLL approximation algorithm, which has applications across a wide range of fields. Originally formulated as a polynomial-time algorithm to factor polynomials with rational coefficients into irreducible factors [41], LLL has since played a key role in cryptography. The polynomial factorization can be used, for example, to obtain a polynomial-time algorithm for RSA encryption [42]. Lenstra [43] additionally demonstrated that application of LLL to integer linear programming yields a polynomial time algorithm when the dimension is fixed. Similar to our use case, the LLL algorithm has been applied as a polynomial-time algorithm for solving the integer relation problem of whether there exists a non-trivial zero integer linear combination of a given set of real numbers [44, 45]. Moreover, Aardal et al. [46] apply the LLL algorithm to a system of linear diophantine equations, and give an accompanying parameter analysis that is analogous to our analysis approach.

1.2. Outline and Notation

We begin in Section 2 with background on the problem, including a summary of key results for one-dimensional signals. This section also develops additional one-dimensional framework used in the extension to two dimensions. Section 3 addresses Problem 1, deriving the number and location of DFT coefficients required for unique recovery of any integer matrix. In Section 4, we present the algorithm framework for recovery from a minimal set of DFT coefficients, including its reformulation as a lattice problem. Section 5 analyzes the lattice method, providing estimates of key parameter values needed for successful recovery. Finally, Section 6 contains numerical simulations that support algorithms and analysis.

Throughout the paper, we use boldface letters to denote vectors (as in $\mathbf{x}$), with entries written as x_n. Typewriter-style letters denote matrices (as in $\mathtt{X}$), with entries X_{mn}. We denote the $n \times n$ identity matrix by $\mathtt{I}_n$. The complex conjugate of a complex number x is denoted by x^*. We use the standard number-theoretic notation $\gcd(m, n)$ and $\operatorname{lcm}(m, n)$ for the greatest common divisor and least common multiple of two integers m and n, respectively. If an integer m is a divisor of n, we write $m \,|\, n$, and otherwise $m \nmid n$. We also write $\tau(n)$ for the number of divisors of n, $\phi(n)$ for Euler's totient function (the number of integers less than N that are relatively prime with n), and $\omega(n)$ for the number of distinct prime factors of n. Lastly, we let $\eta_n = e^{-2\pi \mathrm{i}/n}$ be an nth primitive root of unity.

2. Problem Setup and Background

Let $\mathtt{X}$ be an $N_1 \times N_2$ matrix. The two-dimensional discrete Fourier transform (DFT) of $\mathtt{X}$ is defined by

$$\tilde{X}_{kl} = \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} X_{mn} e^{-2\pi \mathrm{i}\left(\frac{mk}{N_1} + \frac{nl}{N_2}\right)} \,. \tag{2.1}$$

The Fourier coefficients $\tilde{X}_{kl}$ are periodic in each index independently, satisfying $\tilde{X}_{k,l} = \tilde{X}_{k+sN_1,l+tN_2}$ for any integers s and t. A complete set of N_1N_2 DFT coefficients consists of taking all $\tilde{X}_{kl}$ with indices k and l in the ranges $s \le k < s + N_1$ and $t \le l < t + N_2$, for some choice of s and t. We denote this $N_1 \times N_2$ matrix of DFT coefficients by $\tilde{\mathtt{X}}$, and typically choose $s = t = 0$ so that the index ranges begin at zero.

While the DFT maps $\mathtt{X}$ to $\tilde{\mathtt{X}}$, the inverse DFT recovers $\mathtt{X}$ from $\tilde{\mathtt{X}}$. This inverse transformation is defined by

$$X_{mn} = \frac{1}{N_1N_2} \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} \tilde{X}_{kl} e^{2\pi \mathrm{i}\left(\frac{mk}{N_1} + \frac{nl}{N_2}\right)} \,, \tag{2.2}$$

and can equivalently be expressed using any indexing that forms a complete set of DFT coefficients.

We are interested in partial data question of Problem 1. If only some of the DFT coefficients $\tilde{X}_{kl}$ are known, can $\mathtt{X}$ be uniquely reconstructed? In general, this is impossible as seen in Eq. (2.2), where changing the value of only one DFT coefficient $\tilde{X}_{kl}$ will change all elements of $\mathtt{X}$. However, incorporating the *a priori* knowledge that $\mathtt{X}$ is integer-valued can make this inverse problem uniquely solvable. Existing results on this problem focus primarily on the one-dimensional case for binary signals. Since several aspects of the one-dimensional setting carry over to the two-dimensional problem studied here, we will first summarize the relevant prior results. Later in this section, we derive additional properties of one-dimensional signals that will be useful for the two-dimensional theory.

2.1. Summary of Previous Results

The question of uniqueness posed in Problem 1 has been fully answered in one dimension [31]. The one-dimensional DFT and inverse DFT are defined by

$$\tilde{x}_k = \sum_{n=0}^{N-1} x_n e^{2\pi i n k/N} \quad , \quad x_n = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{x}_k e^{-2\pi i n k/N} \ .$$

If $\mathbf{x}$ is a vector of length N consisting of integer entries, then $\mathbf{x}$ can be uniquely determined by measurements of the $\tau(N)$ DFT coefficients,

$$S = \{\tilde{x}_d : d \,|\, N\}. \tag{2.3}$$

The fact that S is a sufficient set of DFT coefficients for uniqueness comes from the irreducibility of cyclotomic polynomials, which is used to prove the following result.

Lemma 2.1 ([31, Lemma 4]). *Let* $\mathbf{x}$ *be an integer signal of length* N*. If* $\gcd(k, N) = d$ *and* $\tilde{x}_k = 0$*, then* $\tilde{x}_{k'} = 0$ *for all* k' *such that* $\gcd(k', N) = d$*.*

Consider two integer signals $\mathbf{y}$ and $\mathbf{z}$ of length N. Since the DFT is bijective, $\mathbf{y} = \mathbf{z}$ if and only if $\tilde{\mathbf{y}} = \tilde{\mathbf{z}}$. Let $\mathbf{x} = \mathbf{y} - \mathbf{z}$. Applying Lemma 2.1 to the integer signal $\mathbf{x} = \mathbf{y} - \mathbf{z}$ implies that $\tilde{\mathbf{x}} = \mathbf{0}$ only if $\tilde{x}_k = 0$ for each $k \in S$. Therefore, two integer signals $\mathbf{y}$ and $\mathbf{z}$ are equal only if $\tilde{y}_d = \tilde{z}_d$ for each divisor d of N.

Pei and Cheng also prove that every DFT coefficient in the set S from Eq. (2.3) is required to guarantee uniqueness in general, even with further restrictions on the integer values. This is not obvious, as many vectors may be uniquely determinable from a smaller set of DFT coefficients. For example, consider the class of binary signals $\mathbf{x}$ where all entries are either 0 or 1. The only binary signal satisfying $\tilde{x}_0 = 0$ is $\mathbf{x} = \mathbf{0}$, so only the $d = 0$ coefficient is required to distinguish $\mathbf{0}$ from all other binary signals. However, for any N and divisor d of N, there exist binary signals $\mathbf{y}$ and $\mathbf{z}$ such that $\tilde{y}_k \neq \tilde{z}_k$ if and only if $\gcd(k, N) = d$. Therefore, the signals $\mathbf{y}$ and $\mathbf{z}$ may only be distinguished by a DFT coefficient of frequency k satisfying $\gcd(k, N) = d$, preventing the inverse problem from being solved uniquely if $\tilde{x}_d$ is removed from S. We will require a weaker version of this statement, which only considers the divisor $d = 1$. This relaxation allows for a simpler proof, which is provided below. Note that while the discussion has focused on distinguishing the pair of binary vectors $\mathbf{y}$ and $\mathbf{z}$, the following lemma relates to the vector $\mathbf{x} = \mathbf{y} - \mathbf{z}$ which has entries in $\{-1, 0, 1\}$.

Lemma 2.2 (Special case of [31, Lemma 2]). *For any positive integer* N*, there exists a signal* $\mathbf{x}$ *of length* N *with entries in* $\{-1, 0, 1\}$ *such that* $\tilde{x}_k \neq 0$ *if and only if* $\gcd(k, N) = 1$*.*

Proof. Decompose N into its prime factors, $N = \prod_{t=1}^{\omega} p_t^{\alpha_t}$. For each subset of indices $T \subseteq \{1, \ldots, \omega\}$, define $n_T = N \sum_{t\in T} \frac{1}{p_t}$. We construct a vector $\mathbf{x}$ with entries in $\{-1, 0, 1\}$ as follows. For every possible value of n_T, let $x_{n_T} = (-1)^{|T|}$, For all other indices n, set $x_n = 0$. Note that this construction is well defined. If $T_1 \neq T_2$, then the uniqueness of prime factorizations implies that

$$\frac{n_{T_1}}{N} = \sum_{t\in T_1} \frac{1}{p_t} = \frac{\displaystyle\sum_{t\in T_1} \prod_{t'\in T_1\setminus\{t\}} t'}{\displaystyle\prod_{t\in T_1} p_t} \qquad \text{and} \qquad \frac{n_{T_2}}{N} = \sum_{t\in T_2} \frac{1}{p_t} = \frac{\displaystyle\sum_{t\in T_2} \prod_{t'\in T_2\setminus\{t\}} t'}{\displaystyle\prod_{t\in T_2} p_t} \ ,$$

are fractions in lowest terms with different denominators, so $n_{T_1} \neq n_{T_2}$. With this choice of $\mathbf{x}$, we can compute any DFT coefficient as

$$\tilde{x}_k = \sum_{n=0}^{N-1} x_n e^{-2\pi \mathrm{i} n k/N} = \sum_{T \subseteq \{1,\ldots,\omega\}} x_{n_T} e^{-2\pi \mathrm{i} n_T k/N} = \sum_{T \subseteq \{1,\ldots,\omega\}} (-1)^{|T|} \exp\left(-2\pi \mathrm{i} k \sum_{t \in T} 1/p_t\right) . \tag{2.4}$$

If $\gcd(k, N) \neq 1$, then there exists $1 \leq t' \leq \omega$ such that $p_{t'} \mid k$. Continuing from Eq. (2.4), we can further decompose the DFT summation over index sets based on membership of t',

$$\begin{aligned}\tilde{x}_k &= \sum_{T \not\ni t'} (-1)^{|T|} \left[\exp\left(-2\pi \mathrm{i} k \sum_{t \in T} 1/p_t\right) - \exp\left(-2\pi \mathrm{i} k \left(\sum_{t \in T} 1/p_t + 1/p'_t\right)\right)\right] \\ &= \sum_{T \not\ni t'} (-1)^{|T|} \left[\exp\left(-2\pi \mathrm{i} k \sum_{t \in T} 1/p_t\right) - \exp\left(-2\pi \mathrm{i} k \sum_{t \in T} 1/p_t\right) \exp\left(-2\pi \mathrm{i} k/p'_t\right)\right] . \end{aligned} \tag{2.5}$$

This latter expression sums over all subsets T of indices that do not contain index t', adding both the term corresponding to that subset T and the term corresponding to the subset $T \cup \{t'\}$. As we have assumed that $p_{t'} \mid k$, we have that $k/p_{t'}$ is an integer, making $e^{-2\pi \mathrm{i} k/p_{t'}} = 1$. Therefore, the expression in Eq. (2.5) reduces to

$$\tilde{x}_k = \sum_{T' \not\ni t'} (-1)^{|T'|} \left[\exp\left(-2\pi \mathrm{i} k \sum_{t \in T'} 1/p_t\right) - \exp\left(-2\pi \mathrm{i} k \sum_{t \in T'} 1/p_t\right)\right] = 0 ,$$

as desired. For the reverse direction, since $\mathbf{x}$ is nonzero, there exists k such that $\tilde{x}_k \neq 0$. Our previous work implies that $\gcd(k, N)$ must be 1. Thus, Lemma 2.1 implies that $\tilde{x}_{k'} \neq 0$ for each k' such that $\gcd(k', N) = 1$.

To clarify the construction outlined in the proof of Lemma 2.2, consider the following illustrative example. For convenience, we adopt the compact notation η_N for the Nth primitive root of unity,

$$\eta_N = e^{-2\pi \mathrm{i}/N} .$$

This notation will be used in the rest of this work.

Example 2.1. Let $N = 6$. The prime factorization of 6 is $6 = p_1^1 \cdot p_2^1$, where $p_1 = 2$ and $p_2 = 3$. The power set of $\{1, 2\}$ is

$$\mathcal{P}(\{1, 2\}) = \{\varnothing, \{1\}, \{2\}, \{1, 2\}\} .$$

Computing all possible value of $n_T = 6 \sum_{i \in T} \frac{1}{p_i}$ yields,

$$n_\varnothing = 0 , \qquad n_{\{1\}} = 6 \cdot \frac{1}{p_1} = 3 ,$$
$$n_{\{2\}} = 6 \cdot \frac{1}{p_2} = 2 , \qquad n_{\{1,2\}} = 6 \cdot \frac{1}{p_1} + 6 \cdot \frac{1}{p_2} = 5 .$$

This corresponds to the signal $\mathbf{x} = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 1 \end{bmatrix}$, where the appropriate value of ± 1 is placed at the calculated indices. Computing its DFT yields,

$$\tilde{\mathbf{x}} = \begin{bmatrix} 0 & 2 - 2\eta_3 & 0 & 0 & 0 & 2 - 2\eta_3^* \end{bmatrix} ,$$

where we have made the simplification $\eta_6^2 = \eta_3$. Note that $\mathbf{x}$ exactly satisfies $\tilde{x}_k \neq 0$ for only $k = 1$ and $k = 5$, as desired.

The current work generalizes the results of Lemmas 2.1 and 2.2 to two-dimensional integer signals of any size. Limited previous results pertaining to similar problems are available for two dimensions. For example, the results of [33, 47] show that if $N_1 = N_2$ is prime, then an integer signal $\mathtt{X}$ may be recovered from a minimal set of $N + 2$ DFT coefficients. This result will follow as a special case of our Theorems 3.1 and 3.2. We conclude this section with a few results about the one-dimensional DFT that will be useful for generalizing Lemmas 2.1 and 2.2 to two dimensions and developing efficient inversion algorithms.

2.2. The DFT of Frequency Decimated Signals

This section contains some properties regarding the DFT of decimated signals in the frequency domain, which will play an important role in our later results. Note that these properties do not depend on integer constraints and hold for any one-dimensional complex-valued signal. We begin with a definition.

Definition 2.1. Let $\mathbf{x}$ be a signal of length N. If $d \,|\, N$, then the frequency decimated signal $\mathbf{x}^{(N/d)}$ of length N/d is defined entrywise by

$$x_m^{(N/d)} = \sum_{n=0}^{d-1} x_{m+nN/d}, \qquad \text{for } 0 \le m < \frac{N}{d}.$$

The process of constructing $\mathbf{x}^{(N/d)}$ from $\mathbf{x}$ corresponds to decimating by a factor of d in the frequency domain. The following brief lemma justifies this terminology, as the decimated signal keeps every dth DFT coefficient of $\mathbf{x}$.

Lemma 2.3. *Let $\mathbf{x}$ be a signal of length N. If d divides N, then for $k = k'd$, $\tilde{x}_k = \tilde{x}_{k'}^{(N/d)}$.*

Proof. Let $k = dk'$. Then the kth DFT coefficient can be written as

$$\tilde{x}_{dk'} = \sum_{n=0}^{N-1} x_n \eta_N^{nk'd} = \sum_{n=0}^{N-1} x_n \eta_{N/d}^{nk'} = \sum_{m=0}^{N/d-1} \left[\sum_{\substack{n \\ n=m \pmod{N/d}}} x_n \right] \eta_{N/d}^{mk'} = \sum_{m=0}^{N/d-1} x_m^{(N/d)} \eta_{N/d}^{mk'},$$

which shows that $\tilde{x}_k = \tilde{x}_{k'}^{(N/d)}$.

The signal $\mathbf{x}^{(N/d)}$ may be alternatively defined by the matrix equation,

$$\mathbf{x}^{(N/d)} = \begin{bmatrix} \mathtt{I}_{N/d} & \cdots & \mathtt{I}_{N/d} \end{bmatrix} \mathbf{x}\,, \tag{2.6}$$

where the $(N/d) \times N$ matrix consists of d blocks of the $(N/d) \times (N/d)$ identity matrix. Lemma 2.3 implies that if $\begin{bmatrix} \mathtt{I}_{N/d} & \cdots & \mathtt{I}_{N/d} \end{bmatrix} \mathbf{y} = \mathbf{x}^{(N/d)}$ for some other signal $\mathbf{y}$, then it must hold that $\tilde{y}_d = \tilde{x}_d$. This consequence of Lemma 2.3 will be used later to optimize the reconstruction algorithms.

We can also analyze the reverse construction. Starting with a signal $\mathbf{x}$ of length N/d, we can construct a signal $\mathbf{y}$ of length N such that $\mathbf{y}^{(N/d)} = d \cdot \mathbf{x}$. While there is clearly no unique way to do this, the construction we choose in Lemma 2.4 has the nice property that the DFT coefficients of $\mathbf{y}$ at frequencies which are not multiples of d are necessarily zero. The proof of this lemma is based on the orthogonality relationship between Fourier modes, which can be stated as

$$\sum_{n=0}^{N-1} \eta_N^{nk} \eta_N^{nl} = N\delta_{k,l}\,, \qquad \text{for } 0 \le k, l < N\,, \tag{2.7}$$

where $\delta_{k,l}$ is the Kronecker-delta defined by

$$\delta_{k,l} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{if } k \neq l\,. \end{cases}$$

Lemma 2.4. *Let $\mathbf{x}$ be a signal of length N. Let $\mathbf{y}$ be the signal of length Nd formed by stacking d copies of $\mathbf{x}$,*

$$\mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{x} & \cdots & \mathbf{x} \end{bmatrix}}_{d \text{ times}}\,. \tag{2.8}$$

If k is not a multiple of d, then $\tilde{y}_k = 0$. Additionally, for all $0 \le k < N$, $\tilde{y}_{dk} = d\tilde{x}_k$.

PROOF. Fix $0 \leq k < N$ and $0 \leq j < d$. Then we can write any DFT coefficient as

$$\tilde{y}_{dk+j} = \sum_{n=0}^{dN-1} y_n \eta_{dN}^{(dk+j)n} = \sum_{n=0}^{N-1} \sum_{t=0}^{d-1} y_{n+tN} \eta_{dN}^{(dk+j)(n+tN)} \ .$$

By the construction of $\mathbf{y}$, in this last expression $y_{n+tN} = x_n$ for any value of t. We can also expand the exponent on η_{dN} to obtain

$$\tilde{y}_{dk+j} = \sum_{n=0}^{N-1} x_n \sum_{t=0}^{d-1} \eta_{dN}^{dkn+dkNt+jn+jtN} = \sum_{n=0}^{N-1} x_n \eta_N^{kn} \eta_{dN}^{jn} \sum_{t=0}^{d-1} \eta_1^{kt} \eta_d^{jt} \ ,$$

where we have reduced each power of η_{dN} to its lowest primitive order. As $\eta_1 = 1$, we can apply the orthogonality relation in Eq. (2.7) to the inner sum to obtain

$$\tilde{y}_{dk+j} = \sum_{n=0}^{N-1} x_n \eta_N^{kn} \eta_{dN}^{jn} \sum_{t=0}^{d-1} \eta_d^{jt} = \sum_{n=0}^{N-1} x_n \eta_N^{kn} \eta_{dN}^{jn} (d \cdot \delta_{j,0}) \ .$$

Therefore, if $j \neq 0$, we have $\tilde{y}_{dk+j} = 0$. If $j = 0$, then $\eta_{dN}^{jn} = 1$ which simplifies the expression to

$$\tilde{y}_{dk} = d \sum_{n=0}^{N-1} x_n \eta_N^{kn} = d \cdot \tilde{x}_k.$$

Lemma 2.4 can be used to prove the general version of Lemma 2.2 for any divisor d of N. Consider the signal $\mathbf{x}$ in Lemma 2.2 of length N/d and define $\mathbf{y}$ as in Eq. (2.8). By Lemma 2.4, $\tilde{y}_k = 0$ if $d \nmid k$. If $d \,|\, k$, we have $\tilde{y}_k = \tilde{x}_{k/d}$. Therefore, Lemma 2.2 implies that $\tilde{y}_k = 0$ if and only if $\gcd(k/d, d) = 1$ which is equivalent to $\gcd(k, d) = d$.

While we will not use Lemma 2.4 again, we do need the analogous construction in frequency space. The proof of the following result is identical to Lemma 2.4 and is omitted.

Lemma 2.5. *Let* $\mathbf{x}$ *be a signal of length* N*. Let* $\mathbf{y}$ *be the signal of length* dN *formed by stacking* d *copies of* $\tilde{\mathbf{x}}$ *in frequency space,*

$$\tilde{\mathbf{y}} = \big[\underbrace{\tilde{\mathbf{x}} \quad \cdots \quad \tilde{\mathbf{x}}}_{d \text{ times}}\big] \ .$$

If n *is not a multiple of* d*, then* $y_n = 0$*. Additionally, for all* $0 \leq n < N$*,* $y_{dn} = dx_n$*.*

3. Uniqueness Results

We now develop the tools needed to address the uniqueness question of Problem 1. We begin by introducing an equivalence relation on the DFT coefficients of integer matrices. Sampling at least one DFT coefficient from each equivalence class is sufficient to ensure the uniqueness of recovery. Next, we show that this sampling strategy is optimal in the sense that if any equivalence class is not sampled, then there exist distinct integer matrices that cannot be distinguished apart from each other. We conclude with results that explicitly count the minimum number of DFT coefficients required for uniqueness.

3.1. DFT Coefficient Equivalence

Let $\mathtt{X}$ be an $N_1 \times N_2$ matrix and let $N = \operatorname{lcm}(N_1, N_2)$. For every $0 \leq k < N_1$ and $0 \leq l < N_2$, finding the least common denominator yields,

$$\frac{mk}{N_1} + \frac{nl}{N_2} = \frac{mkN_2' + nkN_1'}{N} \ ,$$

where $N_1' = N_1/\gcd(N_1, N_2)$ and $N_2' = N_2/\gcd(N_1, N_2)$. This directly implies that $\eta_{N_1}^{mk}\eta_{N_2}^{nl} = \eta_N^{mkN_2'+nkN_1'}$ Thus, we may express the DFT coefficient $\tilde{X}_{kl}$ as a one-dimensional DFT,

$$\tilde{X}_{kl} = \sum_{m=0}^{N_1-1}\sum_{n=0}^{N_2-1} X_{mn}\eta_{N_1}^{mk}\eta_{N_2}^{nl} = \sum_{m=0}^{N_1-1}\sum_{n=0}^{N_2-1} X_{mn}\eta_N^{mkN_2'+nlN_1'} = \sum_{j=0}^{N-1}\left[\sum_{\substack{m,n\\ mkN_2'+nlN_1'=j \pmod N}} X_{mn}\right]\eta_N^j \ .$$

Additionally, for every $0 \le \lambda < N$, the DFT coefficient $\tilde{X}_{\lambda k,\lambda l}$ takes the form

$$\tilde{X}_{\lambda k,\lambda l} = \sum_{j=0}^{N-1}\left[\sum_{\substack{m,n\\ mkN_2'+nlN_1'=j \pmod N}} X_{mn}\right]\eta_N^{\lambda j} \ . \tag{3.1}$$

Therefore, by letting

$$y_j^{(k,l)} = \sum_{\substack{m,n\\ mkN_2'+nlN_1'=j \pmod N}} X_{mn} \quad , \quad \text{for } 0 \le j < N \ , \tag{3.2}$$

we see that $\tilde{X}_{\lambda k,\lambda l}$ is the λ-th DFT coefficient of the integer signal $\mathbf{y}^{(k,l)} = \begin{bmatrix} y_0^{(k,l)} & \cdots & y_{N-1}^{(k,l)}\end{bmatrix}$. Recall that the length of this signal is $N = \operatorname{lcm}(N_1, N_2)$. However, in general, the sum in Eq. (3.2) may be empty for some choices of j as there may be no m and n that satisfy the congruence condition. We will reduce the length of the vector $\mathbf{y}$ by discarding the empty sums, which carry no information about $\mathtt{X}$ since they are zero for any matrix. To this end, define D_1, D_2, and D by

$$D_1 = N_1/\gcd(k, N_1) \quad , \quad D_2 = N_2/\gcd(l, N_2) \quad , \quad D = \operatorname{lcm}(D_1, D_2) \ . \tag{3.3}$$

We will show that discarding the empty sums in Eq. (3.2) reduces the length of $\mathbf{y}^{(k,l)}$ from N to D. We first observe that

$$\tilde{X}_{(\lambda+D)k,(\lambda+D)l} = \tilde{X}_{\lambda k,\lambda l} \ ,$$

as $kD_1 = 0 \pmod{N_1}$ and $lD_2 = 0 \pmod{N_2}$. Therefore, the signal $\mathbf{y}^{(k,l)}$ defined by Eq. (3.2) takes the form in frequency space

$$\tilde{\mathbf{y}}^{(k,l)} = [\underbrace{\tilde{\mathbf{x}}^{(k,l)} \quad \cdots \quad \tilde{\mathbf{x}}^{(k,l)}}_{d \text{ times}}] \ ,$$

where $\tilde{\mathbf{x}}^{(k,l)} = \begin{bmatrix}\tilde{X}_{0\cdot k,0\cdot l} & \cdots & \tilde{X}_{(D-1)k,(D-1)l}\end{bmatrix}$, and $d = N/D$.

Applying Lemma 2.5 to $\mathbf{y}^{(k,l)}$ implies that $y_j^{(k,l)} = 0$ if j is not a multiple of d. As this holds for any matrix $\mathtt{X}$, the sum in Eq. (3.2) must be empty for all such j. This reasoning shows that

$$\tilde{X}_{\lambda k,\lambda l} = \tilde{x}_\lambda^{(k,l)} = \sum_{j=0}^{N-1}\left[\sum_{\substack{m,n\\ mkN_2'+nlN_1'=j \pmod N}} X_{mn}\right]\eta_N^{\lambda j} = \sum_{j=0}^{D-1}\left[\sum_{\substack{m,n\\ mkN_2'+nlN_1'=jd \pmod N}} X_{mn}\right]\eta_D^{\lambda j} \ , \tag{3.4}$$

as we can rewrite the summation in Eq. (3.1) to only consider the D indices j which are multiples of d. This structure motivates the following definition, which allows us to apply the techniques from one-dimensional integer signals to integer matrices.

Definition 3.1 (Subsignal). Let $\mathtt{X}$ be an $N_1 \times N_2$ integer matrix. Let $N_1' = N_1/\gcd(N_1, N_2)$, $N_2' = N_2/\gcd(N_1, N_2)$ and $N = \operatorname{lcm}(N_1, N_2)$. For any k and l, we define the (k, l)-subsignal $\mathbf{x}^{(k,l)}$ of $\mathtt{X}$ entrywise by,

$$x_j^{(k,l)} = \sum_{\substack{m,n\\ mkN_2'+nlN_1'=jd \pmod N}} X_{mn} \quad , \quad \text{for } 0 \le j < D \ ,$$

where D is defined as in Eq. (3.3) and $d = N/D$.

As the (k,l)-subsignal is a one-dimensional integer signal of length D, Lemma 2.1 states that if $\tilde{x}_{\lambda}^{(k,l)} = 0$, then $\tilde{x}_{\lambda'}^{(k,l)} = 0$ for all λ' such that $\gcd(\lambda', D) = \gcd(\lambda, D)$. Therefore, if $\tilde{x}_{1}^{(k,l)} = \tilde{X}_{kl}$ is zero, then for all λ such that $\gcd(\lambda, D) = 1$, we have $\tilde{x}_{\lambda}^{(k,l)} = \tilde{X}_{\lambda k, \lambda l} = 0$. This implies that if we sample $\tilde{X}_{kl}$, then there is no additional information provided by any $\tilde{X}_{\lambda k, \lambda l}$ with $\gcd(\lambda, D) = 1$. This proves the ensuing Theorem 3.1. Before stating the theorem, we first express the relationship between dependent coefficients as an equivalence relation to simplify its formulation.

Definition 3.2. Let $\mathtt{X}$ be an $N_1 \times N_2$ integer matrix. Two DFT coefficients $\tilde{X}_{kl}$ and $\tilde{X}_{k'l'}$ are equivalent, denoted by $(k,l) \sim (k',l')$, if there exists a nonzero integer λ that is relatively prime with D satisfying

$$k' = \lambda k \pmod{N_1} \qquad \text{and} \qquad l' = \lambda l \pmod{N_2}.$$

To verify that the equivalence relation $\sim$ is symmetric, fix k and define D_1 as in Eq. (3.3). Suppose $k' = \lambda k \pmod{N_1}$ with $\gcd(\lambda, D) = 1$, as in Definition 3.2. As λ is relatively prime with D, λ^{-1} exists $\pmod{D}$. Since $D_1 \mid D$, λ^{-1} is also the multiplicative inverse of λ modulo D_1. Let $d_1 = \gcd(k, N_1)$. Note that we must have $d_1 \mid k'$, which justifies that

$$\begin{aligned} k' = \lambda k \pmod{N_1} &\iff (k'/d_1) = \lambda(k/d_1) \pmod{N_1/d_1} \\ &\iff (k'/d_1) = \lambda(k/d_1) \pmod{D_1} \\ &\iff \lambda^{-1}(k'/d_1) = (k/d_1) \pmod{D_1} \\ &\iff \lambda^{-1}k' = k \pmod{N_1}\ . \end{aligned}$$

Performing an identical computation for l, l' shows that $l = \lambda^{-1} l' \pmod{N_2}$, so $(k',l') \sim (k,l)$.

We can now state the main result of this section.

Theorem 3.1. *Let $\mathtt{X}$ be an $N_1 \times N_2$ integer matrix. For every $0 \le k < N_1, 0 \le l < N_2$, and every frequency $(k',l') \sim (k,l)$, $\tilde{\mathtt{X}}_{kl} = 0$ if and only if $\tilde{\mathtt{X}}_{k'l'} = 0$.*

Theorem 3.1 provides an upper bound for Problem 1. It implies that a single representative from each equivalence class in $\tilde{\mathtt{X}}/\sim$ forms a sufficient set of given DFT coefficients for unique recovery of an integer matrix $\mathtt{X}$. Therefore, the number of DFT coefficients required for unique recovery is at most the number of elements of $\tilde{\mathtt{X}}/\sim$.

The next section will focus on proving that this upper bound provided by Theorem 3.1 is tight. To do so, for any given frequency (k,l), we produce a pair of integer $N_1 \times N_2$ matrices $\mathtt{X}$ and $\mathtt{Y}$ such that $\tilde{\mathtt{X}}$ and $\tilde{\mathtt{Y}}$ differ only at frequencies $(k',l') \sim (k,l)$, and are equal at all other frequencies. This shows that a representative of the equivalence class (k,l) is required to uniquely reconstruct an integer matrix.

3.2. Minimal DFT Sampling

We begin by showing that the equation which determines the jth entry of the (k,l)-subsignal,

$$mkN_2' + nlN_1' = jd \pmod{N}\ , \tag{3.5}$$

evenly partitions the entries of $\mathtt{X}$.

Lemma 3.1. *Let $\mathtt{X}$ be an $N_1 \times N_2$ integer matrix. For any frequency (k,l), each entry $x_j^{(k,l)}$ of the (k,l)-subsignal is a sum of $M = N_1N_2/D$ entries of $\mathtt{X}$.*

Proof. We first show the existence of an m and n that satisfy Eq. (3.5) for each $0 \le j < D$. This will show that each entry $x_j^{(k,l)}$ contains a contribution from at least one entry of $\mathtt{X}$. Let $d_1 = \gcd(k, N_1) = N_1/D_1$ and

$d_2 = \gcd(l, N_2) = N_2/D_2$, and recall the definitions from Eq. (3.3). Repeatedly applying the relationship $ab = \gcd(a,b)\operatorname{lcm}(a,b)$ and using the definition of d_1 and d_2 yields,

$$d = \frac{N}{D} = \frac{\operatorname{lcm}(N_1, N_2)}{\operatorname{lcm}(D_1, D_2)} = \frac{N_1 N_2 \gcd(D_1, D_2)}{D_1, D_2 \gcd(N_1, N_2)} = \frac{d_1 d_2 \gcd(N_1/d_1, N_2/d_2)}{\gcd(N_1, N_2)} \ .$$

Next, repeatedly applying the fact, $c\gcd(a,b) = \gcd(ca, cb)$, to the expression above, first with $c = d_1 d_2$ and subsequently with $c = \gcd(N_1, N_2)$ yields,

$$d = \frac{\gcd(N_1 d_2, N_2 d_1)}{\gcd(N_1, N_2)} = \frac{\gcd(N_1, N_2)}{\gcd(N_1, N_2)} \gcd\left(\frac{N_1 d_2}{\gcd(N_1, N_2)}, \frac{N_2 d_1}{\gcd(N_1, N_2)}\right) = \gcd(N_1' d_2, N_2' d_1) \ ,$$

where in the last equality we recall that $N_t' = N_t/\gcd(N_1, N_2)$. Thus, Bézout's identity gives integers p and q such that

$$p(N_2' d_1) + q(N_1' d_2) = d \ . \tag{3.6}$$

Since $d_1 = \gcd(k, N_1)$ and $d_2 = \gcd(l, N_2)$, k/d_1 and l/d_2 must have multiplicative inverses modulo N_1 and N_2, respectively. Thus, set $m_j = j(k/d_1)^{-1} p \bmod N_1$ for the row index, and set $n_j = h(l/d_2)^{-1} q \bmod N_2$ for the column index. These values imply that

$$\begin{aligned} m_j(k/d_1) = jp \pmod{N_1} \quad &\Longrightarrow \quad m_j k = jpd_1 \pmod{N_1} \quad \Longrightarrow \quad m_j k N_2' = jpd_1 N_2' \pmod{N_1 N_2'} \ , \\ n_j(l/d_2) = jq \pmod{N_2} \quad &\Longrightarrow \quad n_j l = jqd_2 \pmod{N_2} \quad \Longrightarrow \quad n_j l N_1' = jqd_2 N_1' \pmod{N_2 N_1'} \ . \end{aligned}$$

Since $N = N_1 N_2/\gcd(N_1, N_2) = N_1' N_2 = N_1 N_2'$, we can substitute these expressions into the left hand side of Eq. (3.5) to obtain

$$m_j k N_2' + n_j l N_1' = jpd_1 N_2' + jqd_2 N_1' = j(pd_1 N_2' + qd_2 N_1') = jd \pmod{N} \ ,$$

where we have used Eq. (3.6) in the last equality. This shows that (m_j, n_j) satisfies Eq. (3.5)

We next show that the same number of entries of $\mathtt{X}$ are summed for each choice of j. Equivalently stated, for any j there are the same number of solutions (m,n) to Eq. (3.5). Let $H = \{(m,n) \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} : mkN_2' + nlN_1' = 0 \pmod{N}\}$, and suppose $(m,n), (m',n') \in H$. Since $(0,0) \in H$, and

$$(m - m')kN_2' + (n - n')lN_1' = 0 \pmod{N} \ ,$$

the set H is a subgroup of $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$. For any fixed j, since

$$(m_j + m)kN_2' + (n_j + n)lN_1' = jd \pmod{N} \iff mkN_2' + nlN_1' = 0 \pmod{N} \ ,$$

the set of indices which satisfy the congruence in Eq. (3.5) is given by the coset,

$$\{(m,n) \in \mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} : mkN_2' + nlN_1' = jd \pmod{N}\} = (m_j, n_j) + H \ .$$

Since the set of cosets partitions the group evenly and there are D cosets of H (as there are D choices of j), we have shown that each entry of the subsignal is a sum of $|H| = N_1 N_2/D$ entries of $\mathtt{X}$.

The following lemma uses Lemma 3.1 to generalize Lemma 2.2 to the two-dimensional setting. In particular, we construct a matrix $\mathtt{X}$ with entries in $\{-1, 0, 1\}$ such that only its DFT coefficients in fixed equivalence class are nonzero. For frequency (k,l), the constructed subsignal $\mathbf{x}^{(k,l)}$ is proportional to the signal in Lemma 2.2. Moreover, every DFT frequency (k',l') that does not appear in the (k,l)-subsignal must be 0.

Lemma 3.2. *Let N_1 and N_2 be any positive integers. Fix (k,l) and define D as in Eq. (3.3). Let $\mathbf{x}$ be the signal of length D defined in Lemma 2.2. For the matrix $\mathtt{X}$ defined by,*

$$X_{mn} = \begin{cases} x_j & \text{if } mkN_2' + nlN_1' = jd \pmod{N} \end{cases} \ , \tag{3.7}$$

the DFT coefficients satisfy $\tilde{X}_{k',l'} \neq 0$ if and only if $(k',l') \sim (k,l)$.

Note that Eq. (3.7) defines all entries of $\mathtt{X}$. This is true as for any m and n, there exists a j such that $mkN_2' + nlN_1' = jd \pmod N$, as demonstrated in the proof of Lemma 3.1.

PROOF. For the constructed matrix $\mathtt{X}$ in Eq. (3.7), the (k,l)-subsignal takes the form,

$$x_j^{(k,l)} = \sum_{\substack{m,n \\ mkN_2'+nlN_1'=jd \pmod N}} X_{mn} = \sum_{\substack{m,n \\ mkN_2'+nlN_1'=jd \pmod N}} x_j = Mx_j \ , \tag{3.8}$$

where the last equality uses Lemma 3.1 to count the $M = N_1N_2/D$ elements in the sum. In Fourier space, Eqs. (3.4) and (3.8) give $\tilde{X}_{\lambda k,\lambda l} = \tilde{x}_\lambda^{(k,l)} = M\tilde{x}_\lambda$. Therefore, if $k' = \lambda k \pmod{N_1}$ and $l' = \lambda l \pmod{N_2}$, the definition of $\mathbf{x}$ in Lemma 2.2 implies that

$$\tilde{X}_{k'l'} = \tilde{x}_\lambda \neq 0 \ \text{ if and only if } \ \gcd(\lambda, N) = 1 \ .$$

Thus, we have shown the desired behavior for frequencies where $k' = \lambda k \pmod{N_1}$ and $l' = \lambda k \pmod{N_2}$ (which includes all $(k',l') \sim (k,l)$). Next, we handle the other case, where we must show that $\tilde{X}_{k'l'} = 0$ for the remaining frequencies (k',l').

Consider the Frobenius entrywise 2-norm of $\mathtt{X}$, which can be computed as

$$\|\mathtt{X}\|_F^2 = \sum_{m=0}^{N_1-1}\sum_{n=0}^{N_2-1} |X_{mn}|^2 = \sum_{j=0}^{D-1} \sum_{\substack{m,n \\ mkN_2'+nlN_1'=jd \pmod N}} |X_{mn}|^2 = \sum_{j=0}^{D-1} M|x_j|^2 = M\|\mathbf{x}\|_2^2 \ ,$$

where we have again used the definition of $\mathtt{X}$ to replace within the sum the M entries of $\mathtt{X}$ that equal x_j. Therefore, applying the discrete Parseval relation yields,

$$\|\tilde{\mathtt{X}}\|_F^2 = N_1N_2\|\mathtt{X}\|_F^2 = N_1N_2M\|\mathbf{x}\|_2^2 = \frac{N_1N_2M}{D}\|\tilde{\mathbf{x}}\|_2^2 = M^2\|\tilde{\mathbf{x}}\|_2^2 \ . \tag{3.9}$$

As $\tilde{X}_{\lambda k,\lambda l} = M\tilde{x}_\lambda$, we also have that

$$M^2\|\tilde{\mathbf{x}}\|_2^2 = M^2\sum_{\lambda=0}^{D-1}|\tilde{x}_\lambda|^2 = \sum_{\lambda=0}^{D-1}|\tilde{X}_{\lambda k,\lambda l}|^2 \ ,$$

which in combination with Eq. (3.9) shows that

$$\|\tilde{\mathtt{X}}\|_F^2 = \sum_{\lambda=0}^{D-1}|\tilde{X}_{\lambda k,\lambda l}|^2 \ .$$

This implies that all nonzero DFT coefficients are contained in the set $\{\tilde{X}_{\lambda k,\lambda l} : 0 \le \lambda < D\}$. Therefore, we have $\tilde{X}_{k'l'} = 0$ if there is no λ such that $k' = \lambda k \pmod{N_1}$ and $l' = \lambda l \pmod{N_2}$, as desired.

We now provide two examples that illustrate the application of Lemma 3.2. Both examples build on Example 2.1. The first example demonstrates a straightforward case where $D = N$. The second example exhibits a more involved setup with $D \neq N$.

Example 3.1. Let $N_1 = 2, N_2 = 3$ and $k = l = 1$. Note that $N = \operatorname{lcm}(N_1, N_2) = 6$, and as N_1, N_2, k, and l are all relatively prime, $D = N$. Therefore, by Example 2.1, the relevant signal from Lemma 2.2 is $\mathbf{x} = \begin{bmatrix}1 & 0 & -1 & -1 & 0 & 1\end{bmatrix}$. We define the following indexing matrix $\mathtt{S}$ by $S_{mn} = (kmN_2' + lnN_1')/d \bmod D$. As $d = N/D = 1$, we have $S_{mn} = 3m + 2n \bmod 6$, which gives,

$$\mathtt{S} = \begin{bmatrix}0 & 2 & 4 \\ 3 & 5 & 1\end{bmatrix} \ .$$

The matrix $\mathtt{S}$ acts as an indexing tool in j to construct $\mathtt{X}$ from $\mathbf{x}$. Since $X_{00} = 0$, we set $X_{00} = x_0$. Similarly, $S_{01} = 2$, so $X_{01} = x_2$. Continuing to set $X_{mn} = x_{S_{mn}}$ produces the matrix and its DFT,

$$\mathtt{X} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \end{bmatrix} \quad , \quad \tilde{\mathtt{X}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2-2\eta_3 & 2-2\eta_3^* \end{bmatrix} .$$

One can see that $\tilde{\mathtt{X}}$ only has nonzero DFT coefficients at frequencies $(1,1)$ and $(1,2) \sim (1,1)$.

Example 3.2. Let $N_1 = 4, N_2 = 6$ (so $N_1' = 2, N_2' = 3$, and $N = 6$) and let $k = l = 2$. We compute $d_1 = \gcd(k, N_1) = 2$ and $d_2 = \gcd(l, N_2) = 2$, so $D_1 = N_1/d_1 = 2$, $D_2 = N_2/d_2 = 3$, $D = \operatorname{lcm}(D_1, D_2) = 6$, and $d = N/D = \operatorname{lcm}(N_1, N_2)/6 = 2$. Since $D = 6$, the signal from Example 2.1, namely $\mathbf{x} = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 1 \end{bmatrix}$, is used again. The indexing matrix $S_{mn} = (kmN_2' + lnN_1')/d \bmod N = 3m + 2n \bmod 6$ is given by

$$\mathtt{S} = \begin{bmatrix} 0 & 2 & 4 & 0 & 2 & 4 \\ 3 & 5 & 1 & 3 & 5 & 1 \\ 0 & 2 & 4 & 0 & 2 & 4 \\ 3 & 5 & 1 & 3 & 5 & 1 \end{bmatrix} .$$

This gives the signal which only has nonzero DFT coefficients at frequencies $(2,2)$ and $(2,4) \sim (2,2)$,

$$\mathtt{X} = \begin{bmatrix} 1 & -1 & 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & -1 & 1 & 0 \\ 1 & -1 & 0 & 1 & -1 & 0 \\ -1 & 1 & 0 & -1 & 1 & 0 \end{bmatrix} \quad , \quad \tilde{\mathtt{X}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8-8\eta_3 & 0 & 8-8\eta_3^* & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} .$$

The construction given by Lemma 3.2 in fact proves that the upper bound given by Theorem 3.1 is tight for integer matrices as the integer matrix $\mathtt{X}$ from Eq. (3.7) requires frequency (k,l) to distinguish it from the matrix of all zeros. The following theorem summarizes the main result and further establishes that the bound remains tight when considering only binary matrices.

Theorem 3.2 (Uniqueness Lower Bound). *A representative of each coefficient class is required to guarantee recovery of any $N_1 \times N_2$ integer matrix, and this requirement remains necessary even when restricting to the set of all $N_1 \times N_2$ binary matrices.*

Proof. For any fixed (k,l), Lemma 3.2 gives a matrix $\mathtt{X}$ with entries in $\{-1, 0, 1\}$ which satisfies $\tilde{\mathtt{X}}_{k',l'} \neq 0$ if and only if $(k',l') \sim (k,l)$. To distinguish $\mathtt{X}$ from the $N_1 \times N_2$ zero matrix, we must measure a nonzero DFT coefficient of $\mathtt{X}$. This demonstrates that a frequency (k',l') equivalent to (k,l) must be sampled to recover the integer matrix $\mathtt{X}$.

For the stronger statement about binary inversion, define the matrices $\mathtt{X}^{(1)}$ and $\mathtt{X}^{(2)}$ by

$$X_{mn}^{(1)} = \begin{cases} 1 & \text{if } X_{mn} = 1 \\ 0 & \text{if } X_{mn} \neq 1 \end{cases} \quad , \quad X_{mn}^{(2)} = \begin{cases} 1 & \text{if } X_{mn} = -1 \\ 0 & \text{if } X_{mn} \neq -1 \end{cases} .$$

Note that these matrices satisfy $\mathtt{X} = \mathtt{X}^{(1)} - \mathtt{X}^{(2)}$. In particular, this implies that for all $(k',l') \not\sim (k,l)$, as $\tilde{X}_{k'l'} = 0$, we have $\tilde{X}_{k'l'}^{(1)} = \tilde{X}_{k'l'}^{(2)}$. Therefore, the binary matrices $\mathtt{X}^{(1)}$ and $\mathtt{X}^{(2)}$ may only be distinguished by a coefficient in the equivalence class with representative (k,l).

To demonstrate Theorem 3.2, we build upon Example 3.1 to generate a pair of binary matrices $\mathtt{X}^{(1)}$ and $\mathtt{X}^{(2)}$ that are indistinguishable except by a representative of the given equivalence class.

Example 3.3. Let $N_1 = 2, N_2 = 3$ and $k = l = 1$. From Example 3.1, we have the matrix

$$\mathtt{X} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \end{bmatrix} \quad , \quad \tilde{\mathtt{X}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2-2\eta_3 & 2-2\eta_3^* \end{bmatrix} ,$$

The construction in Theorem 3.2 thus yields,

$$\mathtt{X}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad , \quad \tilde{\mathtt{X}}^{(1)} = \begin{bmatrix} 2 & \eta_6 & \eta_6^* \\ 0 & 1-\eta_3 & 1-\eta_3^* \end{bmatrix}$$
$$\mathtt{X}^{(2)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad , \quad \tilde{\mathtt{X}}^{(2)} = \begin{bmatrix} 2 & \eta_6 & \eta_6^* \\ 0 & \eta_3-1 & \eta_3^*-1 \end{bmatrix} ,$$

where these two matrices only disagree at DFT frequencies $(1,1)$ and $(1,2) \sim (1,1)$.

3.3. Enumeration and Algebraic Structure of Coefficient Classes

Our work in the previous section demonstrated that Theorem 3.1 gives a tight upper bound on the number of DFT measurements required for uniqueness. This section investigates the algebraic structure of the equivalence classes, which may be used to count the number of elements of $\tilde{\mathtt{X}}/\sim$.

The DFT coefficient equivalence classes of an $N_1 \times N_2$ integer matrix may be identified with cyclic subgroups of $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$. First, if (k,l) and (k',l') satisfy $(k,l) \sim (k',l')$, then there exists $0 < \lambda < D$ such that $k' = \lambda k \pmod{N_1}$, $l' = \lambda l \pmod{N_2}$. This is equivalent to saying $(k',l') = \lambda(k,l)$ in $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$, which shows that $\langle (k',l') \rangle \subseteq \langle (k,l) \rangle$ in $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$. The symmetry of the equivalence relation $\sim$ implies $\langle (k,l) \rangle = \langle (k',l') \rangle$.

For the reverse direction, now suppose that $\langle (k,l) \rangle = \langle (k',l') \rangle$. The order of $\langle (k,l) \rangle$ in $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$ is known to be

$$|\langle (k,l) \rangle| = \operatorname{lcm}\left(\frac{N_1}{\gcd(k,N_1)}, \frac{N_2}{\gcd(l,N_2)} \right) = D \ .$$

Since $(k',l') \in \langle (k,l) \rangle$, and D is the order of (k,l) and (k',l'), there exists $0 \leq \lambda < D$ such that $(k',l') = \lambda(k,l)$. Properties of cyclic groups thus imply that the order of (k',l') in $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$ is $D/\gcd(\lambda, D)$. As the order of (k',l') is D, we must have $\gcd(\lambda, D) = 1$ which shows that $(k,l) \sim (k',l')$.

This correspondence yields an approach for computing the theoretical number of DFT coefficients required for inversion, which is stated as the following corollary to Theorems 3.1 and 3.2.

Corollary 3.1. *The minimal number of DFT coefficients required to uniquely recover an integer $N_1 \times N_2$ matrix is the number of cyclic subgroups of $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$.*

Generally, the number of cyclic subgroups of $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$ may be computed from

$$c(N_1, N_2) = \sum_{a \,|\, N_1, b \,|\, N_2} \phi(\gcd(a,b)), \tag{3.10}$$

where the sum ranges over pairs of divisors of N_1 and N_2, and ϕ is the totient function [48]. The following examples show a few cases where this expression can be simplified.

Example 3.4 (Coprime Dimensions). If $\gcd(N_1, N_2) = 1$, then

$$\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2} \cong \mathbb{Z}_{N_1 N_2},$$

which is cyclic. In any cyclic group, there is exactly one subgroup of order d for each divisor d of its order, so we have $\tau(N_1 N_2) = \tau(N_1)\tau(N_2)$ coefficient classes.

Example 3.5 (Prime Power Square). When $N_1 = N_2 = N$, Eq. (3.10) becomes,

$$c(N) = \sum_{d \cdot e = N} d 2^{\omega(e)},$$

where ω counts the number of prime factors [48]. If $N = p^\alpha$, for some prime p and integer α, the number of coefficient classes is then given by

$$\sum_{d \cdot e = p^\alpha} d 2^{\omega(e)} = \sum_{j=0}^{\alpha} p^j 2^{\omega(p^{\alpha-j})} = p^\alpha 2^0 + \sum_{j=0}^{\alpha-1} p^j 2^1 = 2\frac{p^\alpha - 1}{p-1} + p^\alpha = \frac{p^{\alpha+1} + p^\alpha - 2}{p-1} \ .$$

4. Inversion Algorithms

We now begin to address Problem 2 regarding recovery of integer signals and matrices. Let $\mathbf{x}$ be an integer signal of length N. Without loss of generality, consider the minimal set of DFT coefficients, $\{\tilde{x}_d : d \,|\, N\}$. Lemmas 2.1 and 2.2 imply that $\mathbf{x}$ may be uniquely recovered from,

$$\begin{bmatrix} \eta_N^{0\cdot d_1} & \cdots & \eta_N^{(N-1)d_1} \\ \vdots & \ddots & \vdots \\ \eta_N^{0\cdot d_\tau} & \cdots & \eta_N^{(N-1)d_\tau} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \tilde{x}_{d_1} \\ \vdots \\ \tilde{x}_{d_\tau} \end{bmatrix}, \qquad \mathbf{x} \in \mathbb{Z}^N \,, \tag{4.1}$$

where $d_1, \ldots, d_\tau$ are the divisors of N. Although the solution to the inverse problem is known to be unique, finding the solution is a highly nontrivial task. One clear difficulty arises from the lack of bounds on the range of integer values that $\mathbf{x}$ can take. Many reconstruction methods, including branch-and-bound integer linear programming (ILP), require such bounds to be efficient [46]. Even in the simplest case, when $\mathbf{x}$ is binary with each entry equal to 0 or 1, ILP is known to be NP-hard [18]. Furthermore, since this is a feasibility problem with no objective function to optimize, the effectiveness of the method depends strongly on the rank of the linear system, which can be highly underdetermined [43]. Defining an arbitrary objective function may improve or worsen the performance of branch-and-bound, although it is difficult to know what objective is best [46]. Furthermore, solving Eq. (4.1) directly with ILP can be unstable, as there may exist integer linear combinations of the roots of unity that come arbitrarily close, but do not exactly equal, the given DFT coefficients [49, 50].

Our reconstruction methods build on the preceding theoretical results. In particular, instead of solving the entire linear system in Eq. (4.1) at once, we decompose the problem into more manageable subproblems based on the decimation and subsignal concepts in Definition 2.1 and Definition 3.1. We demonstrate that this decomposition significantly decreases the overall search space.

This section begins with a description of the general divide-and-conquer strategy. We then use this framework to develop an algorithm to reconstruct a one-dimensional integer signal from its minimal set of DFT coefficients. This algorithm generalizes the approach in [31], which addresses the case where N is a power of 2. The one-dimensional inversion algorithm also provides the foundation for the more general two-dimensional algorithm described next. Finally, we discuss a lattice-based method alternative to ILP for solving the resulting NP-hard subproblems. These lattice methods are relatively agnostic to bounds on the integer values and, by leveraging approximation algorithms, are considerably faster.

4.1. Inversion Strategy

Recall from Definition 2.1 that, for $d \,|\, N$, $\mathbf{x}^{(N/d)}$ is the signal $\mathbf{x}$ decimated in frequency space to length N/d,

$$x_m^{(N/d)} = \sum_{n=0}^{d-1} x_{m+nN/d}, \qquad \text{for } 0 \le m < N/d \,,$$

and, by Lemma 2.3, satisfies $\tilde{x}_k^{(N/d)} = \tilde{x}_{dk}$ for all $0 \le k < N/d$. Therefore, knowledge of $\mathbf{x}^{(N/d)}$ for some divisor d is equivalent to knowing $\tilde{x}_{dk}$ for all $0 \le k < N/d$. We can then replace the corresponding row of Eq. (4.1) with the equivalent set of N/d equations given by the matrix form of frequency decimation in Eq. (2.6), namely

$$\begin{bmatrix} \mathtt{I}_{N/d} & \cdots & \mathtt{I}_{N/d} \end{bmatrix} \mathbf{x} = \mathbf{x}^{(N/d)} \,. \tag{4.2}$$

This substitution is advantageous because it mitigates the stability and rank issues of Eq. (4.1). Replacing the complex irrational coefficients of the linear system with integer coefficients improves numerical stability. Indeed, while integer linear combination of Nth roots of unity are dense in the complex numbers [49–51], any integer vector $\mathbf{x}$ that does not satisfy Eq. (4.2) will have an error of at least 1 (in any p-norm). Furthermore, when $d \neq N$, the substitution replaces a single equation with N/d linearly independent equations, which increases the rank of the system. Note that the theoretical equivalence between the single row for $\tilde{x}_d$ in the original equation and the the N/d substituted equations in Eq. (4.2) only holds when $\mathbf{x}$ is an integer signal.

For each divisor d of N, if the decimated signal $\mathbf{x}^{(N/d)}$ is known, then performing the substitution in Eq. (4.2) can improve the ILP formulation. However, note that if $d \,|\, d' \,|\, N$, then

$$\begin{bmatrix} \mathtt{I}_{N/d} & \cdots & \mathtt{I}_{N/d} \end{bmatrix} \mathbf{x} = \mathbf{x}^{(N/d)} \text{ implies that } \begin{bmatrix} \mathtt{I}_{N/d'} & \cdots & \mathtt{I}_{N/d'} \end{bmatrix} \mathbf{x} = \mathbf{x}^{(N/d')}$$

by Lemma 2.3, since $\{\tilde{x}_k : d' \,|\, k\} \subseteq \{\tilde{x}_k : d \,|\, k\}$. Therefore, it is sufficient to only consider the prime divisors of N when performing the replacements of Eq. (4.2). This work shows that the ILP in Eq. (4.1) may be written as the equivalent ILP,

$$\left[\begin{array}{ccc} \mathtt{I}_{N/p_1} & \cdots & \mathtt{I}_{N/p_1} \\ \vdots & \ddots & \vdots \\ \mathtt{I}_{N/p_\omega} & \cdots & \mathtt{I}_{N/p_\omega} \\ \hline \eta_N^0 & \cdots & \eta_N^{(N-1)} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathbf{x}^{(N/p_1)} \\ \vdots \\ \mathbf{x}^{(N/p_\omega)} \\ \hline \tilde{x}_1 \end{array}\right], \qquad \mathbf{x} \in \mathbb{Z}^{\mathbf{N}}\,, \tag{4.3}$$

where $p_1, \ldots, p_\omega$ are the prime factors of N. If both $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^N$ satisfy the linear constraints in Eq. (4.3), then Lemma 2.3 implies that $\tilde{x}_k = \tilde{y}_k$ for all k such that some prime factor p of N divides k. Since this is equivalent to the condition $\gcd(k, N) \neq 1$, the top ω blocks of the matrix in Eq. (4.3) fix $N - \phi(N)$ DFT coefficients of $\mathbf{y}$. The remaining last row implies that $\tilde{y}_1 = \tilde{x}_1$. Note that as both $\mathbf{x}$ and $\mathbf{y}$ are real-valued, this last row also implies that $\tilde{y}_{-1} = \tilde{x}_{-1}$, as these DFT coefficients are the complex conjugates of $\tilde{y}_1$ and $\tilde{x}_1$. Therefore, $N - \phi(N) + 2$ DFT coefficients of $\mathbf{x}$ are fixed by linear constraints in Eq. (4.3).

In general, as $\mathbf{x}$ is real-valued, knowledge of the DFT coefficient $\tilde{x}_k$ also gives $\tilde{x}_{-k} = \tilde{x}_k^*$. We can thus always rewrite the naive ILP in Eq. (4.1) as,

$$\left[\begin{array}{ccc} \eta_N^{0 \cdot d_1} & \cdots & \eta_N^{(N-1)d_1} \\ \eta_N^{-0 \cdot d_1} & \cdots & \eta_N^{-(N-1)d_1} \\ \hline \vdots & \ddots & \vdots \\ \hline \eta_N^{0 \cdot d_\tau} & \cdots & \eta_N^{(N-1)d_\tau} \\ \eta_N^{-0 \cdot d_\tau} & \cdots & \eta_N^{-(N-1)d_\tau} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \tilde{x}_{d_1} \\ \tilde{x}_{d_1}^* \\ \hline \vdots \\ \hline \tilde{x}_{d_\tau} \\ \tilde{x}_{d_\tau}^* \end{array}\right], \qquad \mathbf{x} \in \mathbb{Z}^N\,, \tag{4.4}$$

Since the DFT is bijective, the rank of this matrix is the number of unique rows. For each divisor d, the two rows $\eta_N^{\pm nd}$ of Eq. (4.4) are the same if and only if $d = N$ or $d = N/2$. Therefore, the rank of the matrix in Eq. (4.4) is $2\tau(N) - 2$ if N is even, and $2\tau(N) - 1$ if N is odd.

In either Eq. (4.3) or (4.4), the rank of the matrix describes the size of the ILP search space. An exhaustive search may solve the ILP by iterating through integer combinations of the free variables and checking whether the corresponding pivot variable values are integers. As such, we assume that the dimension of the search space of any ILP we construct is the number of free variables of the linear system over $\mathbb{R}$, or equivalently the nullity of its matrix. Note that while conjugate rows are always included in implementations of the ILPs, we will often omit them in the matrix formulations for clarity. The corresponding conjugate linear equations can be assumed to be implicitly present.

The ILP in Eq. (4.4) will serve as a baseline for evaluating the performance of our algorithms. As the rank of this matrix is $2\tau(N) - 1$ if N is odd and $2\tau(N) - 2$ if N is even, the dimension of its search space is given by its nullity $N - 2\tau(N) + 1$ or $N - 2\tau(N) + 2$. On the other hand, the DFT is bijective, so the rank of the matrix in Eq. (4.3) is $N - \phi(N) + 2$ (where we have included the implicit conjugate last row). Thus, the dimension of the search space of Eq. (4.3) is $N - (N - \phi(N) + 2) = \phi(N) - 2$. If N is prime (and odd), $N - 2\tau(N) + 1 = N - 3 = \phi(N) - 2$, showing that the ILP formulations of Eqs. (4.1) and (4.3) are the same. However, when N is composite, this construction causes a significant reduction in the dimension of the search space. For example, if $N = 30$, the search space associated with Eq. (4.1) has dimenension $30 - 2\tau(30) + 2 = 16$. For Eq. (4.3), the dimension of the search space is reduced to $\phi(N) - 2 = 6$. We now develop an algorithm which uses this approach iteratively to solve the inverse problem.

4.2. 1D Inversion

For each divisor d of N, the vector $\mathbf{x}^{(N/d)}$ is an integer signal of length N/d, so it may be recovered recursively by the same method used for $\mathbf{x}$. Algorithm 1 outlines this recursive approach to invert a one-dimensional integer signal from its minimal set of DFT coefficients. Lines 2-3 give the base case as recovery is trivial if $N = 1$. Line 4 iterates through the prime factors of N, recursively computing the integer signal decimated by each prime factor p in frequency space. Finally, Line 6 uses the collected subproblem solutions to set up the ILP in Eq. (4.3), returning its unique solution as the recovered integer signal.

Algorithm 1 Recursive 1D Inversion

Require: DFT coefficients $\tilde{x}_d$ for each divisor d of N

1: **procedure** INVERT1D($\{\tilde{x}_d : d \in \text{DIVISORS}(N)\}$)
2: **if** $N = 1$ **then**
3: **return** $\left[\tilde{x}_0\right]$
4: **for all** $p \in \text{PRIMEFACTORS}(N)$ **do**
5: $\mathbf{x}^{(N/p)} \leftarrow \text{INVERT1D}(\{\tilde{y}_d = \tilde{x}_{dp} : d \in \text{DIVISORS}(N/p)\})$
6: **return** the solution to the ILP,

$$\left[\begin{array}{ccc} \mathtt{I}_{N/p_1} & \cdots & \mathtt{I}_{N/p_1} \\ \vdots & \ddots & \vdots \\ \mathtt{I}_{N/p_\omega} & \cdots & \mathtt{I}_{N/p_\omega} \\ \hline \eta_N^0 & \cdots & \eta_N^{(N-1)} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathbf{x}^{(N/p_1)} \\ \vdots \\ \mathbf{x}^{(N/p_\omega)} \\ \hline \tilde{x}_1 \end{array}\right], \qquad \mathbf{x} \in \mathbb{Z}^N \ .$$

While Algorithm 1 offers several improvements over a naive ILP or brute-force search solution, it suffers from computational redundancy due to repeated subproblems. Figure 1 displays the subproblem recursion tree of Algorithm 1 when $N = 30$. Observe that $\mathbf{x}^{(2)}$ appears as a subproblem of both $\mathbf{x}^{(6)}$ and $\mathbf{x}^{(10)}$. Thus, Algorithm 1 solves the subproblem for $\mathbf{x}^{(2)}$ twice. The same is true for $\mathbf{x}^{(3)}$ and $\mathbf{x}^{(5)}$. Note that the subproblem of size 1 is solved 6 times by Algorithm 1, but is not computationally relevant as this is the trivial base case.

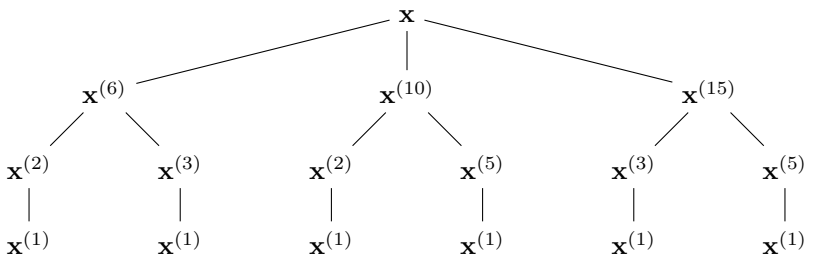

Figure 1: Recursion tree showing subproblem structure when Algorithm 1 is applied to a length $N = 30$ signal.

In general, this computational redundancy may be understood by considering the subgroup lattice of $\mathbb{Z}_N$. The subgroup lattice uses set inclusion to partially order the subgroups of $\mathbb{Z}_N$. Identifying the subgroups of $\mathbb{Z}_N$ with the divisors of N gives a correspondence between the subgroup $\langle d \rangle$ of size N/d and the subproblem $\mathbf{x}^{(N/d)}$. The lattice structures in Figure 2 illustrate the redundancy of the size 2, size 3, and size 5 subproblems of a length 30 signal. For example, both the subgroup $\langle 5 \rangle$ of order 6 and the subgroup $\langle 3 \rangle$ of order 10 on the third lattice level connect to the subgroup $\langle 15 \rangle$ on the second level. This parallels the dependence of both $\mathbf{x}^{(6)}$ and $\mathbf{x}^{(10)}$ on $\mathbf{x}^{(2)}$ in the subproblem lattice.

Using the correspondence between divisors of N and subgroups of $\mathbb{Z}_N$, it is apparent that two subgroups are connected in the lattice if and only if their orders differ by a prime factor. Applying this fact to the recursive step of Algorithm 1, we see that each subproblem explicitly depends only on subproblems of the

previous lattice layer. In particular, this implies that a dynamic programming implementation of Algorithm 1 can remove computational redundancy by first solving each of the subproblems on the bottom level, then the second level, and so on. Since subgroups on the same lattice level are not related by inclusion, subproblems which share a lattice layer cannot depend on each other. Thus, the order in which to solve the subproblems on each level is arbitrary.

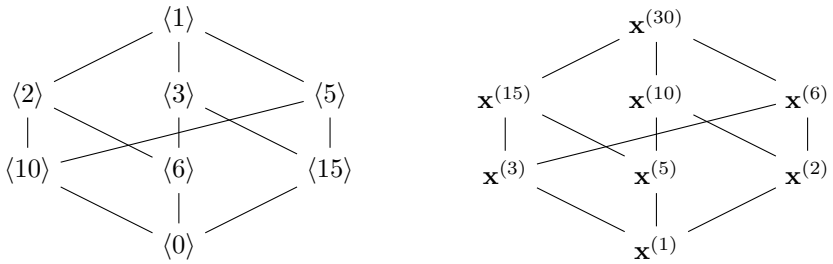

Figure 2: **Left**: The Hasse diagram for the subgroup lattice of $\mathbb{Z}_{30}$. Two subgroups $H \subsetneq K$ are connected if they are related by inclusion and there is no subgroup H' satisfying $H \subsetneq H' \subsetneq K$. The lattice is arranged into levels where each subgroup on any given level is only connected to subgroups on the levels immediately above and below it. **Right**: In contrast with Figure 1, the subproblem lattice of recovering a length 30 integer signal. The identical lattice structure between the two diagrams suggest how the correspondence between subgroups and subproblems makes a dynamic programming algorithm efficient.

Algorithm 2 implements one-dimensional inversion with dynamic programming. The iteration order through the subproblems in Line 1 sorts the divisors N' of N in increasing order. Since $d \,|\, N'$ only if $d \leq N'$, this ensures that the subproblems for all proper divisors of N' have been solved prior to subproblem N' itself. Therefore, each step has all decimated signals required to set up the ILP in Eq. (4.3). Line 3 solves the ILP, storing the result as the signal decimated to length N' in $\mathbf{x}^{(N')}$ for future use. After solving the subproblem for each divisor of N, the inverted signal is given as $\mathbf{x}^{(N)}$, which is returned in Line 4.

While Algorithm 2 decreases the search space of each subproblem relative to an exhaustive search over the entire problem in Eq. (4.1), this comes at the potential cost of solving a larger number of ILPs. To see that introducing the additional subproblems generally improves the overall complexity, consider the example of a length $N = 30$ binary signal $\mathbf{x}$. Recovering $\mathbf{x}^{(2)}$, $\mathbf{x}^{(3)}$, and $\mathbf{x}^{(6)}$ is trivial, since every DFT coefficient is required to ensure uniqueness of length 2, 3, and 6 signals. The binary constraint on $\mathbf{x}$ implies the entries of $\mathbf{x}^{(5)}$ are between 0 and 6. The nullity of the ILP matrix for this subproblem is $\phi(5) - 2 = 2$. The size of the search space for this subproblem is thus 7^2. Similarly, the entries of $\mathbf{x}^{(10)}$ are between 0 and 3 and the entries of $\mathbf{x}^{(15)}$ are between 0 and 2. Since $\phi(10) - 2 = 2$ and $\phi(15) - 2 = 6$, these subproblems have size 4^2 and 3^6, respectively. Finally, since $\phi(30) - 2 = 6$, the recovery of the binary signal $\mathbf{x}^{(30)}$ has a search space of size 2^6. The sum of the search space sizes is $7^2 + 4^2 + 3^6 + 2^6 = 858$. On the other hand, as $30 - 2\tau(3) + 2 = 16$, an exhaustive search of Eq. (4.1) would have a much larger search space of size $2^{16} = 65,536$.

Algorithm 2 Memoized 1D Inversion

Require: DFT coefficients $\tilde{x}_d$ for each divisor d of N

1: **for** $N' \in$ Divisors(N) **do** ▷ *Includes N, iterate in increasing order.*

2: $\quad \{p_1, \ldots, p_\omega\} \leftarrow$ PrimeFactors(N')

3: $\quad \mathbf{x}^{(N')} \leftarrow$ the solution to the ILP,

$$\left[\begin{array}{ccc} \mathtt{I}_{N'/p_1} & \cdots & \mathtt{I}_{N'/p_1} \\ \vdots & \ddots & \vdots \\ \mathtt{I}_{N'/p_\omega} & \cdots & \mathtt{I}_{N'/p_\omega} \\ \hline \eta_{N'}^{0} & \cdots & \eta_{N'}^{(N'-1)} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathbf{x}^{(N'/p_1)} \\ \vdots \\ \mathbf{x}^{(N'/p_\omega)} \\ \hline \tilde{x}_{N/N'} \end{array}\right], \qquad \mathbf{x} \in \mathbb{Z}^{N'} .$$

4: **return** $\mathbf{x}^{(N)}$

4.3. 2D Inversion

We next develop a two-dimensional inversion algorithm by using the subsignal structure as a reduction to the one-dimensional problem. Algorithm 3 inverts an integer matrix $\mathtt{X}$ by individually inverting each of its subsignals. Line 1 iterates through representative frequencies (k, l) of each DFT coefficient equivalence class. Lines 2-4 compute the length of the (k, l)-subsignal as defined in Eq. (3.3). Line 5 inverts the (k, l)-subsignal by passing a minimal set of DFT coefficients to the one-dimensional inversion procedure. Once the (k, l)-subsignal has been inverted, the corresponding DFT coefficients of $\mathtt{X}$ are updated. Once all subsignals have been inverted, Line 8 returns the inverted matrix $\mathtt{X}$. Note that in Lines 7 and 8, we convert between real space and Fourier space. In Line 7, $\tilde{\mathbf{x}}^{(k,l)}$ is computed from the reconstructed subsignal $\mathbf{x}^{(k,l)}$. To return $\mathtt{X}$ in Line 8, one must first take an inverse DFT of $\tilde{\mathtt{X}}$ which was compiled through the iterations of the algorithm. In a practical implementation, these steps are performed efficiently using fast Fourier transforms. This takes $O(D\log(D))$ time in Line 7 and $O(N_1N_2\log(N_1+N_2))$ time in Line 8 [52]. As the fast Fourier transforms are much quicker than solving the NP-hard ILP at each iteration, conversion between real space and Fourier space makes a negligible impact on the runtime of Algorithm 4.

Algorithm 3 High Level 2D Inversion

Require: a representative DFT coefficient $\tilde{X}_{k,l}$ for each coefficient class

1: **for all** $(k, l) \in \textsc{RepCoeffs}(N_1, N_2)$ **do**
2: $\quad D_1 \leftarrow N_1/\gcd(k, N_1)$
3: $\quad D_2 \leftarrow N_2/\gcd(l, N_2)$
4: $\quad D \leftarrow \operatorname{lcm}(D_1, D_2)$
5: $\quad \mathbf{x}^{(k,l)} \leftarrow \textsc{Invert1D}(\{\tilde{X}_{\lambda k,\lambda l} : \lambda \,|\, D\})$
6: $\quad$ **for** $0 \le \lambda < D$ **do**
7: $\quad\quad \tilde{X}_{\lambda k,\lambda l} \leftarrow \tilde{x}^{(k,l)}_{\lambda}$
8: **return** $\mathtt{X}$

In one dimension, the correspondence between (cyclic) subgroups of $\mathbb{Z}_N$ and sets of equivalent DFT coefficients revealed the lattice structure of subproblems. We can generalize this subproblem structure to two dimensions by considering the correspondence in Corollary 3.1 between cyclic subgroups of $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$ and DFT coefficient classes of an $N_1 \times N_2$ integer matrix. Figure 3 shows the lattice of cyclic subgroups of $\mathbb{Z}_4 \times \mathbb{Z}_6$. As in the one-dimensional case, for $(k', l') \in \langle (k, l)\rangle$, the subgroups $\langle (k, l)\rangle$ and $\langle (k', l')\rangle$ are connected in the lattice if and only if $(k', l') = p(k, l)$ for some prime p dividing the order of (k, l). Thus, solutions to connected subproblems lower in the lattice are required to set up Eq. (4.3). The subproblem lattice reveals the computational redundancy arising from repeated subproblems in the naive divide-and-conquer implementation in Algorithm 3.

Algorithm 3 admits two types of redundancies. The first comes from explicitly solving subproblems that are also solved implicitly by the one-dimensional inversion. To invert a 4×6 integer matrix, the outer for loop of Algorithm 3 solves a one-dimensional inversion problem for each vertex of the lattice in Figure 3. Consider the iteration where the $(k, l) = (0, 1)$ subsignal is recovered. Within the one-dimensional inversion problem of this subsignal $\mathbf{x}^{(0,1)}$ of length 6, we will recover $\left(\mathbf{x}^{(0,1)}\right)^{(3)}$, which is this signal decimated by 2. This decimated signal is exactly to the subsignal $\mathbf{x}^{(0,3)}$. However, Algorithm 3 also explicitly recovers the $(0, 3)$-subsignal at a different iteration of the loop in Line 1 when $(k, l) = (0, 3)$.

The second type of redundancy comes from overlapping subproblems in the one-dimensional recovery of the (k, l)- and (k', l')-subsignals of $\mathtt{X}$. Consider $(k, l) = (1, 1)$ and $(k', l') = (1, 2)$. The cyclic subgroup lattice shows that both $\langle (k, l)\rangle$ and $\langle (k', l')\rangle$ contain $\langle (2, 2)\rangle = \{(2, 2), (2, 4)\}$. Thus, recovering both the $(1, 1)$-subsignal and the $(1, 2)$-subsignal implicitly reconstructs the length 6 $(2, 2)$-subsignal. As in the one-dimensional case, the lattice structure of the subproblems implies that a dynamic programming implementation can remove these redundancies.

Such an implementation is described in Algorithm 4. In Lines 1 and 2, the algorithm loops over pairs of divisors D_1 and D_2 of N_1 and N_2, respectively. This orders the frequencies so that solutions to previous subproblems are available as needed. Line 5 iterates through the representative frequencies (k, l) corresponding

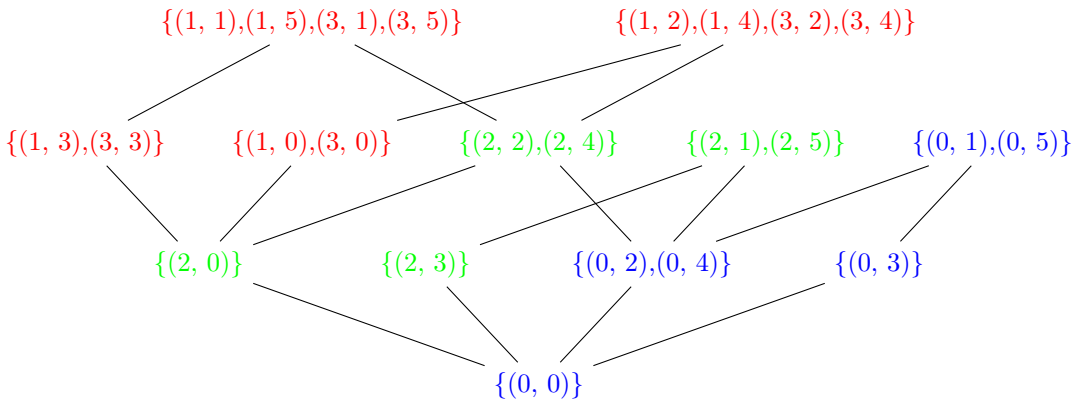

Figure 3: The Hasse diagram of the cyclic subgroup lattice of $\mathbb{Z}_4 \times \mathbb{Z}_6$. Only the generators of each subgroup are shown explicitly, as all other elements are contained in a proper subgroup lower in the lattice. Color of vertices of $\langle (k,l) \rangle$ indicates iteration of loop in Line 1 of Algorithm 4 where the (k,l)-subsignal is recovered ($D_1 = 1$ is blue, $D_1 = 2$ is green, and $D_1 = 4$ is red).

to the divisors D_1 and D_2. The order of this iteration is arbitrary. In Line 6, each (k,l)-subsignal is recovered by solving the ILP in Eq. (4.3). Using the ILP solution, Line 8 updates the DFT coefficients of `X` from the coefficient class represented by (k,l). Finally, Line 9 returns the inverted matrix `X`. Note that Algorithm 4 exhibits neither type of redundancy observed in Algorithm 3. Algorithm 4 does not use a one-dimensional inversion subroutine, which eliminates the first redundancy type. The second type is handled by solving exactly one ILP for each coefficient class.

Next, we verify the correctness of the dynamic programming iteration order in Algorithm 4. Fix a representative frequency (k,l) and consider the iteration of the loops in Lines 1 and 2 that recovers the (k,l)-subsignal. The condition in Line 5 implies that this iteration satisfies $D_1 = N_1/\gcd(k,N_1)$ and $D_2 = N_2/\gcd(l,N_2)$. For each prime factor p of D, the (k,l)-subsignal decimated by p in frequency space is given by $(\mathbf{x}^{(k,l)})^{(D/p)} = \mathbf{x}^{(pk,pl)}$. To check that the (pk,pl)-subsignal was recovered prior to the current iteration, we consider two cases for the prime factor p. If $p \,|\, N_1$, then $\gcd(pk,N_1) = p\gcd(k,N_1) = N_1/(D_1/p)$. Since $D_1/p < D_1$, the (pk,pl)-subsignal was recovered at a previous iteration of the for loop in Line 1. If $p \nmid N_1$, then $\gcd(pk,N_1) = \gcd(k,N_1)$. Additionally, p must divide N_2, so $\gcd(pl,N_2) = p\gcd(l,N_2) = N_2/(D_2/p)$. Since $D_2/p < D_2$, the (pk,pl)-subsignal was recovered at the current iteration of the for loop in Line 1 and a previous iteration of the for loop in Line 2. Note that since $D \,|\, \mathrm{lcm}(N_1,N_2)$, its prime factor p must divide N_1 or N_2, so this argument shows that each required decimated subsignal is available.

Unlike the one-dimensional dynamic programming solution in Algorithm 2, Algorithm 4 does not completely explore each lattice layer before moving onto the next. The double for loop through pairs of divisors performs more of a depth-first traversal of the subproblem lattice. The cyclic subgroup lattice in Figure 3 illustrates the subproblem order for inverting a 4×6 integer matrix. The lattice vertices are colored according to which iteration of the outer for loop in Line 1 solves the subproblem. For example, the blue sublattice is recovered in the first iteration of the outer for loop, where $D_1 = 1$ (so the condition $\gcd(k,N_1) = N_1/D_1$ in Line 5 implies $k = 0$). The divisors of 6 are $D_1 = 1,2,3,6$, so the subproblems in this sublattice are solved in the order,

$$\{(0,0)\}, \{(0,3)\}, \{(0,2),(0,4)\}, \{(0,1),(0,5)\}.$$

At the next iteration of the outer for loop, $D_1 = 2$ and the green sublattice is recovered. The red sublattice is recovered in the final iteration of the outer for loop, where $D_1 = 4$.

Note that Algorithm 4 is not significantly more efficient than ILP for the 4×6 example considered above. Since there are 12 cyclic subgroups in Figure 3, there are 12 classes of DFT coefficients. In particular, the two-dimensional analog of the ILP in Eq. (4.1) has rank $2 \cdot 12 - 4 = 20$. Since there are only four free variables in this system, a brute-force search is tractable. To demonstrate the computational speedup of Algorithm 4 in general, consider a 30×30 binary matrix. A 30×30 integer matrix has 140 coefficient equivalence

Algorithm 4 Memoized 2D Inversion

Require: a representative DFT coefficient $\tilde{X}_{k,l}$ for each coefficient class

for $D_1 \in \textsc{Divisors}(N_1)$ **do**
  for $D_2 \in \textsc{Divisors}(N_2)$ **do**
    $D \leftarrow \operatorname{lcm}(D_1, D_2)$
    $\{p_1, \ldots, p_\omega\} \leftarrow \textsc{PrimeFactors}(D)$
    for all $(k,l) \in \textsc{RepCoeffs}(N_1, N_2) : \gcd(k, N_1) = N_1/D_1, \gcd(l, N_2) = N_2/D_2$ **do**
      $\mathbf{x}^{(k,l)} \leftarrow$ the solution to the ILP,

$$\left[\begin{array}{ccc} \mathtt{I}_{D/p_1} & \cdots & \mathtt{I}_{D/p_1} \\ \vdots & \ddots & \vdots \\ \mathtt{I}_{D/p_\omega} & \cdots & \mathtt{I}_{D/p_\omega} \\ \hline \eta_D^0 & \cdots & \eta_D^{(D-1)} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathbf{x}^{(p_1 k, p_1 l)} \\ \vdots \\ \mathbf{x}^{(p_\omega k, p_\omega)} \\ \hline \tilde{X}_{kl} \end{array}\right], \qquad \mathbf{x} \in \mathbb{Z}^D \ .$$

      for $0 \le \lambda < D : \gcd(\lambda, D) = 1$ **do**
        $\tilde{X}_{\lambda k, \lambda l} \leftarrow \tilde{x}_\lambda^{(k,l)}$
return $\mathtt{X}$

classes, so we will not show the size of every subproblem. As one example, consider recovery of the $(2,6)$-subsignal. For the $(k,l) = (2,6)$ frequency, we have $D_1 = N_1/\gcd(k, N_1) = 15$, $D_2 = N_2/\gcd(l, N_2) = 5$, and $D = \operatorname{lcm}(5, 15) = 15$. Each entry of the $(2,6)$-subsignal is thus a sum of $N_1 N_2/D = 60$ entries of $\mathtt{x}$, and thus has 61 possible values. As the nullity of the relevant matrix is $\phi(15) - 2 = 6$, we have a total search space size of $61^6 = 5.15\text{e}10$ for this subproblem. Repeating this computation for each of the 140 subproblems yields a total search space size of 4.88e11. The two-dimensional analogue of the ILP in Eq. (4.1) has a search space size of $2^{900-140*2+4} = 2^{624} = 6.96\text{e}187$.

4.4. Lattice Methods

As discussed in the previous two subsections, inversion methods Algorithm 2 and Algorithm 4 offer substantial improvements over a naive ILP approach. However, each iteration of the algorithms still involves solving a smaller ILP, and in this section we describe a lattice-based method for doing so. Lattice methods are preferable to ILP in this problem for several reasons, including their ability to handle a wider range of integer values, to naturally incorporate a guess, and to take advantage of available approximation algorithms.

When using ILP techniques to solve this feasibility problem, as there is a unique solution but no objective function to minimize, branch-and-bound methods do not offer significant improvement over exhaustive search. As such, the performance of ILP is dominated by the size of the search space. By defining an arbitrary objective function to minimize, branch and bound might converge faster or slower, although it is typically difficult to tell *a priori* which is the case [46]. The only major computational speedup from ILP comes from preprocessing steps, such as Gomory cutting planes, which reduce the size of the search space [53].

Tight bounds, such as binarity constraints, on the signal entries may yield a significant reduction in the size of the ILP search space. However, even when the signal is known to be binary, many of the associated subproblems considered by Algorithms 2 and 4 will not have the binary constraint. In one dimension, if $d \,|\, N$, then for $N' = N/d$, the subproblem $\mathbf{x}^{(N')}$ satisfies $0 \le x_n^{(N')} \le d$. In two dimensions, Lemma 3.1 implies that the (k,l)-subsignal satisfies $0 \le x_j^{(k,l)} \le N_1 N_2/D$ where D is defined as in Eq. (3.3). Thus, if $\gcd(N_1, N_2) \neq 1$, none of the subproblems are binary. On the other hand, the performance of lattice methods appears relatively independent of the bounds on the singal entries [33].

Moreover, the lattice formulation benefits from the incorporation of a (non-integer) guess for the inverted signal, as described below. In contrast, we are not aware of how such a guess could improve the performance of an ILP method, beyond prescribing an initial point for a breadth-first exhaustive search. Lastly, as with ILP, the proposed lattice-based reformulation is NP-hard. However, polynomial-time approximation algorithms are available. We discuss this implementation of Algorithms 2 and 4 in Section 6.1, which employs

a polynomial-time approximation algorithm for significant speed improvements. In this section, we describe the lattice-based approach in general, which converts the ILP in Eq. (4.3) into a lattice instance of the shortest vector problem.

Consider a linearly independent set of vectors $B = \{\mathbf{b}_0, \ldots, \mathbf{b}_{d-1}\} \subseteq \mathbb{R}^n$. The set of all integer linear combinations of vectors in B defines a d-dimensional lattice in $\mathbb{R}^n$,

$$\mathcal{L} = \left\{ \sum_{n=0}^{d-1} \alpha_n \mathbf{b}_n : \boldsymbol{\alpha} \in \mathbb{Z}^d \right\}.$$

Since only integer combinations of basis vectors are allowed, the lattice always has a shortest nonzero vector. Given a lattice basis, the shortest vector problem (SVP) finds the shortest nonzero vector in the associated lattice. We reduce the ILP in Eq. (4.3) to SVP by constructing the basis,

$$\mathtt{B} = \left[\mathbf{b}_0 \quad \cdots \quad \mathbf{b}_{N-1} \;\middle|\; \mathbf{b}_N\right] = \left[\begin{array}{ccc|c} \multicolumn{3}{c|}{\mathtt{I}_N} & -\overline{\mathbf{x}} \\ \hline 0 & \cdots & 0 & \beta_0 \\ \hline \beta_1 \mathtt{I}_{N/p_1} & \cdots & \beta_1 \mathtt{I}_{N/p_1} & -\beta_1 \mathbf{x}^{N/p_1} \\ \vdots & \ddots & \vdots & \vdots \\ \beta_1 \mathtt{I}_{N/p_\omega} & \cdots & \beta_1 \mathtt{I}_{N/p_\omega} & -\beta_1 \mathbf{x}^{N/p_\omega} \\ \hline \beta_2 \eta_N^{0 \cdot k_1} & \cdots & \beta_2 \eta_N^{(N-1)k_1} & -\beta_2 \tilde{x}_{k_1} \\ \vdots & \ddots & \vdots & \vdots \\ \beta_2 \eta_N^{0 \cdot k_M} & \cdots & \beta_2 \eta_N^{(N-1)k_M} & -\beta_2 \tilde{x}_{k_M} \end{array}\right] = \left[\begin{array}{c} \mathtt{A} \\ \hline \beta_0 \mathtt{B}_0 \\ \hline \\ \beta_1 \mathtt{B}_1 \\ \\ \hline \\ \beta_2 \mathtt{B}_2 \\ \\ \end{array}\right] \tag{4.5}$$

where $\overline{\mathbf{x}}$ is a guess and β_0, β_1, and β_2 are parameters [43, 46, 54]. In the last block $\mathtt{B}_2$ of $\mathtt{B}$, we have allowed for M DFT coefficients k_m with $\gcd(k_m, N) = 1$ for each $1 \leq m \leq M$, instead of just the single DFT coefficient required for uniqueness. This formulation allows for the flexibility of choosing $M > 1$. Providing additional DFT measurements improves stability and practicality of reconstruction for larger dimensions, as demonstrated in the following sections. Recall that each row k_m of $\mathtt{B}_2$ actually represents two rows for k_m and $-k_m$. To be consistent with real-valued basis vectors in the lattice definition, in practice we use the equivalent final basis block, $\beta_2 \begin{bmatrix} \Re[\mathtt{B}_2] \\ \Im[\mathtt{B}_2] \end{bmatrix}$, which stacks the real and imaginary parts of $\beta_2 \mathtt{B}_2$.

Any vector in the lattice may be written as $\mathtt{B}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$ for a vector $\boldsymbol{\alpha} \in \mathbb{Z}^N$ and $\gamma \in \mathbb{Z}$. The choice of β parameters ideally guarantees that the shortest vector in the lattice $\mathtt{B}$ has coefficients $\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T = \pm \begin{bmatrix}\mathbf{x} & 1\end{bmatrix}^T$. As $\mathtt{A}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T = \boldsymbol{\alpha} - \gamma\overline{\mathbf{x}}$ and $\mathtt{B}_0\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T = \begin{bmatrix}\gamma\beta_0\end{bmatrix}$, the recovered vector $\boldsymbol{\alpha}$ (which is ideally $\mathbf{x}$) can easily be obtained from the top two blocks of the shortest vector $\mathtt{B}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$.

The parameters β_1 and β_2 act as penalties for the recovered vector $\boldsymbol{\alpha}$ not satisfying Eq. (4.3) exactly. Suppose that $\gamma = \pm 1$. Then $\begin{bmatrix}\mathtt{B}_1 \\ \mathtt{B}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$ is exactly the difference between the two sides of Eq. (4.3). By Lemma 2.1, $\boldsymbol{\alpha} = \pm\mathbf{x}$ if and only if this difference is zero. Scaling the system by β_1 and β_2 also scales this difference. Thus, if $|\gamma| = 1$, setting β_1 and β_2 sufficiently large should ensure that we recover the correct signal. The last parameter β_0 thus acts as a penalty on the size of $|\gamma|$. A large value of β_0 prevents $|\gamma|$ from being too large in the shortest vector $\mathtt{B}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$. However, we want $|\gamma| = 1$, so β_0 cannot be set too large as this would force $\gamma = 0$. The following section analyzes the values of β_0, β_1, and β_2 required for this reduction to hold.

The basis $\mathtt{B}$ also incorporates a guess $\overline{\mathbf{x}}$ for the signal. The top block of any vector in the lattice is $\mathtt{A}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T = \boldsymbol{\alpha} - \gamma\overline{\mathbf{x}}$. If $\gamma = \pm 1$ as desired, this block will have a small norm when the coefficients $\boldsymbol{\alpha}$ are close to the guess, $\pm\overline{\mathbf{x}}$. An accurate guess thus favors coefficients which are close to the true signal. The

guess $\overline{\mathbf{x}}$ is constructed from the limited set of known DFT coefficients. As discussed in relation to Eq. (4.3), the divide-and-conquer strategy gives the final iteration of Algorithm 2 access to all DFT coefficients $\tilde{\mathbf{x}}_k$ with $\gcd(k, N) \neq 1$. These $N - \phi(N)$ DFT coefficients from the decimated signals combine with the $2M$ frequencies $\pm k_1, \ldots, \pm k_M$ for a set of $N - \phi(N) + 2M$ known frequencies,

$$F = \{\pm k_1, \ldots, \pm k_M\} \cup \{k : \gcd(k, N) > 1\}.$$

The guess $\overline{\mathbf{x}}$ is then defined in frequency space to align with the known DFT data,

$$\tilde{\overline{x}}_k = \begin{cases} \tilde{x}_k & \text{if } k \in F \\ 0 & \text{if } k \notin F \ , \end{cases}$$

Note that $\overline{\mathbf{x}}$ cannot be an integer signal, as this would violate Lemma 2.1. Without considering integer constraints, it is unclear how to further refine the guess. The following section analyzing the lattice method estimates the error of the guess and quantifies the benefit of incorporating a close guess for $\overline{\mathbf{x}}$.

5. Lattice Analysis

In this section, we analyze the lattice with basis $\mathtt{B}$ from Eq. (4.5). In general, the parameters β_1 and β_2 should be chosen sufficiently large to ensure that the shortest vector has lattice coefficients $\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T = \pm \begin{bmatrix}\mathbf{x} & 1\end{bmatrix}^T$. In practice however, limited precision in the DFT measurements and algorithm runtime considerations prohibit arbitrarily large parameter values. With the estimate in this section, we intend to approximate conditions for the parameters $\beta_0, \beta_1, \beta_2$ which ensure the correctness of the reduction from ILP to SVP. We emphasize that this analysis is an estimate rather than a strict upper bound. It provides guidance for parameter selection, which often requires additional tuning in practice.

The desired shortest vector in the lattice, denoted by $\mathbf{x}^*$, is related to the original signal $\mathbf{x}$ by,

$$\mathbf{x}^* = \mathtt{B}\begin{bmatrix}\mathbf{x} \\ \hline 1\end{bmatrix} = \left[\begin{array}{c} \mathtt{A} \\ \hline \beta_0 \mathtt{B}_0 \\ \hline \beta_1 \mathtt{B}_1 \\ \hline \beta_2 \mathtt{B}_2 \end{array}\right] \left[\begin{array}{c} \mathbf{x} \\ \hline 1 \end{array}\right] = \left[\begin{array}{c} x_0 - \overline{x}_0 \\ \vdots \\ x_{N-1} - \overline{x}_{N-1} \\ \hline \gamma\beta_0 \\ \hline 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{array}\right] \tag{5.1}$$

where we recall that any vector in the lattice can be written in the form $\mathtt{B}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$ for a vector $\boldsymbol{\alpha} \in \mathbb{Z}^N$ and $\gamma \in \mathbb{Z}$. Let $\boldsymbol{\ell} = \mathtt{B}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$ be the true shortest vector in this lattice, which necessarily satisfies

$$\|\boldsymbol{\ell}\|_2 \leq \|\mathbf{x}^*\|_2 = \sqrt{\sum_{n=0}^{N-1} |x_n - \overline{x}_n|^2 + \beta_0^2} = \sqrt{K^2 + \beta_0^2} \ , \tag{5.2}$$

where $K = \|\mathbf{x} - \overline{\mathbf{x}}\|_2$ measures the error of the guess $\overline{\mathbf{x}}$. Recall from Section 4.4 that the lattice basis $\mathtt{B}$ is actually real valued with each complex row of $\mathtt{B}$ representing two rows consisting of its real and imaginary parts. However, the 2-norm of a lattice vector is invariant with respect to these representation choices,

$$|\ell_n|^2 = |\Re\ell_n|^2 + |\Im\ell_n|^2,$$

so we will perform the lattice analysis using the more compact complex formulation in Eq. (4.5).

Analogous to the block structure of the basis $\mathtt{B}$, we express the shortest vector in block structure as $\boldsymbol{\ell} = \begin{bmatrix}\boldsymbol{\ell}^{(\mathtt{A})} & \boldsymbol{\ell}^{(\mathtt{B}_0)} & \boldsymbol{\ell}^{(\mathtt{B}_1)} & \boldsymbol{\ell}^{(\mathtt{B}_2)}\end{bmatrix}$, where $\boldsymbol{\ell}^{(\mathtt{A})} = \mathtt{A}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$ and $\boldsymbol{\ell}^{(\mathtt{B}_j)} = \beta_j \mathtt{B}_j \begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$ for $j = 0, 1, 2$. The blocks $\boldsymbol{\ell}^{(\mathtt{A})}$ and $\boldsymbol{\ell}^{(\mathtt{B}_0)}$ may be computed from Eq. (4.5),

$$\boldsymbol{\ell}^{(\mathtt{A})} = \boldsymbol{\alpha} - \gamma\overline{\mathbf{x}} \qquad , \qquad \boldsymbol{\ell}^{(\mathtt{B}_0)} = \begin{bmatrix}\gamma\beta_0\end{bmatrix} \ .$$

The vector $\boldsymbol{\ell}^{(\mathtt{B}_1)}$ itself has a natural block structure based on the prime factors $p_1, \ldots, p_\omega$ of N. We write $\boldsymbol{\ell}^{(\mathtt{B}_1)} = \begin{bmatrix}\boldsymbol{\ell}^{(p_1)} & \cdots & \boldsymbol{\ell}^{(p_\omega)}\end{bmatrix}$, where each vector $\boldsymbol{\ell}^{(p_j)}$ is given by

$$\boldsymbol{\ell}^{(p_j)} = \beta_1 \left(\boldsymbol{\alpha}^{(N/p_j)} - \gamma \mathbf{x}^{(N/p_j)}\right) \ .$$

Lastly, $\boldsymbol{\ell}^{(\mathtt{B}_2)}$ is defined entrywise by

$$\ell_m^{(\mathtt{B}_2)} = \beta_2 \left(\tilde{\alpha}_{k_m} - \gamma \tilde{x}_{k_m}\right), \qquad \text{for } 1 \le m \le M \ .$$

While $\boldsymbol{\ell} = \mathtt{B}\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$ is the shortest vector in the lattice, we have no *a priori* knowledge about the magnitude of its basis coefficients $\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$. Our analysis begins with a bound on the coefficients, which will guide the choice of β_0. Because the size of β_0 affects the length of $\boldsymbol{\ell}$, the estimated values of β_1 and β_2 depend on the chosen β_0 value.

5.1. The Role of β_0 and Estimating β_1

We first bound γ in terms of the parameter β_0. Restricting to the subvector $\boldsymbol{\ell}^{(\mathtt{B}_0)}$ of $\boldsymbol{\ell}$ gives the trivial bound $\|\boldsymbol{\ell}\|_2^2 \ge \|\boldsymbol{\ell}^{(\mathtt{B}_0)}\|_2^2 = (\gamma\beta_0)^2$, so Eq. (5.2) implies that

$$(\gamma\beta_0)^2 \le \|\boldsymbol{\ell}\|_2^2 \le K^2 + \beta_0^2 \ .$$

Rearranging this equation yields the bound,

$$\gamma^2 \le \frac{K^2}{\beta_0^2} + 1 \iff |\gamma| \le \left\lfloor \sqrt{\frac{K^2}{\beta_0^2} + 1} \right\rfloor \ , \tag{5.3}$$

where the floor function comes from the fact that $\gamma \in \mathbb{Z}$. Eq. (5.3) shows how the choice of β_0 bounds the size of γ. In particular, choosing $\beta_0 \ge K$ implies that γ necessarily satisfies $|\gamma| \le 1$. This is ideal as the desired shortest vector $\mathbf{x}^*$ has $|\gamma| = 1$. However, our subsequent analysis shows that setting β_0 large causes a corresponding increase in the magnitudes of β_1 and β_2. The optimal choice of β_0 forces γ to be close to 1 without requiring excessively large β_1 and β_2 values.

To obtain useful bounds for the coefficients in $\boldsymbol{\alpha}$, we similarly bound the magnitude of $\boldsymbol{\ell}$ below by its components in $\boldsymbol{\ell}^{(\mathtt{A})}$ and $\boldsymbol{\ell}^{(\mathtt{B}_0)}$. Therefore, we have

$$\sum_{n=0}^{N-1} |\alpha_n - \gamma\overline{x}_n|^2 + |\gamma\beta_0|^2 = \|[\boldsymbol{\ell}^{(A)} \ \boldsymbol{\ell}^{(\mathtt{B}_0)}]\|_2^2 \le \|\boldsymbol{\ell}\|_2^2 \le K^2 + \beta_0^2 \ . \tag{5.4}$$

Consider two cases. If $\gamma = 0$, the left hand side of the above equation reduces to $\|\boldsymbol{\alpha}\|_2^2$, so Eq. (5.4) gives the bound,

$$\|\boldsymbol{\alpha}\|_2 \le \sqrt{K^2 + \beta_0^2} \ . \tag{5.5}$$

In the case when $\gamma \neq 0$, we have $1 - |\gamma|^2 \le 0$, since γ is an integer. Therefore, Eq. (5.4) yields

$$\sum_{n=0}^{N-1} |\alpha_n - \gamma\overline{x}_n|^2 \le K^2 + (1 - |\gamma|^2)\beta_0^2 \le K^2 \ .$$

Thus we obtain,

$$\|\boldsymbol{\alpha} - \gamma\overline{\mathbf{x}}\|_2 = \sqrt{\sum_{n=0}^{N-1} |\alpha_n - \gamma\overline{x}_n|^2} \leq K \ , \tag{5.6}$$

which, bounds the distance from the coefficients α_n to the scaled guess $\gamma\overline{\mathbf{x}}$. The bounds on $\boldsymbol{\alpha}$ in each case are important for estimating the value of β_2 below.

We now estimate the parameter β_1 which scales the $\mathtt{B}_1$ block of the lattice basis in Eq. (4.5). Fix any prime divisor p_j of N. Considering only the components $\boldsymbol{\ell}^{(p_j)}$ in $\boldsymbol{\ell}^{(\mathtt{B}_1)}$ and $\boldsymbol{\ell}^{(\mathtt{B}_0)}$ gives a trivial lower bound on the length $\|\boldsymbol{\ell}\|_2$. Thus,

$$|\gamma\beta_0|^2 + \beta_1^2\|\boldsymbol{\alpha}^{(N/p_j)} - \gamma\mathbf{x}^{(N/p_j)}\|_2^2 = \|\boldsymbol{\ell}^{(\mathtt{B}_0)}\|_2^2 + \|\boldsymbol{\ell}^{(p_j)}\|_2^2 = \|\boldsymbol{\ell}^{(\mathtt{B}_0)} \ \boldsymbol{\ell}^{(p_j)}\|_2^2 \leq \|\boldsymbol{\ell}\|_2^2 \leq K^2 + \beta_0^2 \ , \tag{5.7}$$

where the final inequality comes from Eq. (5.2). Rearranging Eq. (5.7) yields,

$$\beta_1^2\|\boldsymbol{\alpha}^{(N/p_j)} - \gamma\mathbf{x}^{(N/p_j)}\|_2^2 \leq K^2 + (1 - |\gamma|^2)\beta_0^2 \leq K^2 + \beta_0^2 \ . \tag{5.8}$$

Since all entries of $\boldsymbol{\alpha}^{(N/p_j)} - \gamma\mathbf{x}^{(N/p_j)}$ are integers, Eq. (5.8) implies that choosing $\beta_1 > \sqrt{K^2 + \beta_0^2}$ guarantees $\|\boldsymbol{\alpha}^{(N/p_j)} - \gamma\mathbf{x}^{(N/p_j)}\|_2^2 = 0$. This choice of β_1 thus implies that $\boldsymbol{\alpha}^{(N/p_j)} = \gamma\mathbf{x}^{(N/p_j)}$. In particular, when $|\gamma| = 1$ as desired, the recovered vector $\boldsymbol{\alpha}$ will match the DFT data $\tilde{x}_k$ for all k such that $p_j \,|\, k$. Since this analysis holds for any choice of j, the recovered vector satisfies $\tilde{\alpha}_d = \tilde{x}_d$ for all $d > 1$ such that $d \,|\, N$. Lemma 2.3 implies that if the last required DFT coefficient satisfies $\tilde{\alpha}_1 = \tilde{x}_1$, we will necessarily have $\boldsymbol{\alpha} = \boldsymbol{x}$. The next section analyzes the value of β_2 required for this to hold. We note that even if $|\gamma| \neq 1$, the condition $\boldsymbol{\ell}^{(\mathtt{B}_1)} = \mathbf{0}$ reduces the space of candidate shortest vectors.

5.2. Estimating Parameter β_2

Considering only the $\boldsymbol{\ell}^{(\mathtt{B}_2)}$ and $\boldsymbol{\ell}^{(\mathtt{B}_0)}$ components of $\boldsymbol{\ell}$ gives a trivial lower bound on its length,

$$\beta_2^2 \sum_{m=1}^{M} |\tilde{\alpha}_{k_m} - \gamma\tilde{x}_m|^2 + |\gamma\beta_0|^2 = \|[\boldsymbol{\ell}^{(\mathtt{B}_0)} \ \boldsymbol{\ell}^{(\mathtt{B}_2)}]\|_2^2 \leq \|\boldsymbol{\ell}\|_2^2 \leq K^2 + \beta_0^2 \ . \tag{5.9}$$

Ideally, we would choose a value of β_2 to exactly enforce the equations,

$$\tilde{\alpha}_{k_m} - \gamma\tilde{x}_{k_m} = 0, \qquad \text{for } 1 \leq m \leq M \ . \tag{5.10}$$

In particular, when $|\gamma| = 1$, Eq. (5.10) ensures that $\tilde{\boldsymbol{\alpha}}$ matches the known DFT coefficients $\tilde{x}_{k_1}, \ldots, \tilde{x}_{k_M}$. Unlike Eq. (5.7) however, the left-hand sides of Eqs. (5.9) and (5.10) do not have integer constraints. Moreover, as integer combinations of roots of unity are dense in $\mathbb{C}$, Eq. (5.10) may come arbitrarily close to 0 [49–51]. To analyze Eq. (5.10) despite this instability, we model the sum of roots of unity as a sum of random points on the unit circle. Note that the probabilistic model must still obey the previously derived constraints on the coefficients $\begin{bmatrix}\boldsymbol{\alpha} & \gamma\end{bmatrix}^T$. In particular, the entries of $\boldsymbol{\alpha}$ should obey the bound in Eq. (5.6) (or Eq. (5.4) if $\gamma = 0$) and $|\gamma|$ is bounded by Eq. (5.3). As the estimate for β_1 is independent of β_2, we also assume that β_1 was chosen sufficiently large to guarantee that $\boldsymbol{\ell}^{(\mathtt{B}_1)} = \mathbf{0}$.

Fix values of β_0 and $\beta_1 > \sqrt{K^2 + \beta_0^2}$. Under the probabilistic model, define $\rho(\beta_2, \gamma)$ as the expected number of vectors $\boldsymbol{\alpha} \in \mathbb{Z}^N$ that the inequality in Eq. (5.9) holds for the given values of β_2 and γ, with the aforementioned constraints on $\boldsymbol{\alpha}$. We expect $\rho(\beta_2, \gamma)$ to decrease as β_2 increases, as fewer choices of $\boldsymbol{\alpha}$ make the sum in Eq. (5.9) sufficiently small. We also expect $\rho(\beta_2, \gamma)$ to decrease as $|\gamma|$ increases, as increases in $|\gamma|$ magnify any inaccuracy of the guess $\overline{\mathbf{x}}$. Since Eq. (5.6) constrains the coefficients $\boldsymbol{\alpha}$ to be near the scaled guess $\gamma\overline{\mathbf{x}}$, it is less likely that the DFT coefficients $\tilde{\alpha}_k$ are close to $\gamma\tilde{x}_k$ for larger values of γ. Note that $\rho(\beta_2, 1) \geq 1$ and $\rho(\beta_2, -1) \geq 1$, as coefficient vectors $\boldsymbol{\alpha} = \pm\mathbf{x}$ solve Eq. (5.10) exactly. The behavior of Eq. (5.9) changes if $\gamma = 0$, so we consider two cases.

Case (1): Fix $\gamma = 0$.
Substituting $\gamma = 0$ into Eq. (5.9) yields the condition,

$$\beta_2^2 \sum_{m=1}^{M} \left| \sum_{n=0}^{N-1} \alpha_n \eta_N^{nk_m} \right|^2 = \beta_2^2 \sum_{m=1}^{M} |\tilde{\alpha}_{k_m}|^2 \leq K^2 + \beta_0^2 \ , \tag{5.11}$$

where we have expanded the DFT coefficients of $\boldsymbol{\alpha}$. To analyze this equation, we model the sum of roots of unity, $\tilde{\alpha}_{k_m} = \sum \alpha_n \eta_N^{nk_m}$, as the sum of $\sum |\alpha_n| = \|\boldsymbol{\alpha}\|_1$ random points on the unit circle [33]. The probability that the sum of n random points on the unit circle has modulus at most R goes to

$$1 - \exp(-R^2/n), \tag{5.12}$$

as n goes to infinity, which makes this model a useful approximation [55].

As every term in the outer sum in Eq. (5.11) is positive, the sum is bounded by,

$$\sum_{m=1}^{M} |\tilde{\alpha}_{k_m}|^2 \geq \max_{m=1}^{M} |\tilde{\alpha}_{k_m}|^2 \ . \tag{5.13}$$

This gives the immediate probabilistic upper bound,

$$\mathbb{P}\left[\sum_{m=1}^{M} |\tilde{\alpha}_{k_m}|^2 < \frac{K^2 + \beta_0^2}{\beta_2^2} \right] \leq \mathbb{P}\left[\max_{m=1}^{M} |\tilde{\alpha}_{k_m}| < \frac{\sqrt{K^2 + \beta_0^2}}{\beta_2} \right] \ .$$

This last probability involves the M events, $|\tilde{\alpha}_{k_m}| < \frac{\sqrt{K^2+\beta_0^2}}{\beta_2}$, for $1 \leq m \leq M$. Approximating these events as independent and identically distributed gives,

$$\mathbb{P}\left[\max_{m=1}^{M} |\tilde{\alpha}_{k_m}| < \frac{\sqrt{K^2 + \beta_0^2}}{\beta_2} \right] \approx \mathbb{P}\left[|\tilde{\alpha}_k| < \frac{\sqrt{K^2 + \beta_0^2}}{\beta_2} \right]^M \ ,$$

for a generic index k. We now apply our model of the sum of roots of unity as a sum of $\|\boldsymbol{\alpha}\|_1$ random points on the unit circle. Eq. (5.12) gives the approximation,

$$\mathbb{P}\left[\left| \sum_{n=0}^{N-1} \alpha_n \eta_N^{nk} \right| < \frac{\sqrt{K^2 + \beta_0^2}}{\beta_2} \right]^M \approx \left[1 - \exp\left(- \frac{(\sqrt{K^2 + \beta_0^2}/\beta_2)^2}{\|\boldsymbol{\alpha}\|_1} \right) \right]^M \approx \left(\frac{(K^2 + \beta_0)^2/\beta_2^2}{\|\boldsymbol{\alpha}\|_1} \right)^M , \tag{5.14}$$

where the Taylor series approximation for the exponential function $e^x \approx 1 + x$ was used in the second step. This first-order Taylor approximation is valid for sufficiently large values of β_2. Note that the vector $\boldsymbol{\alpha} = \mathbf{0}$ is excluded, since we are looking for the shortest nonzero vector in the lattice. Therefore, this expression on the right hand side of Eq. (5.14) is well-defined with $\|\boldsymbol{\alpha}\|_1 \geq 1$ in the denominator.

We now compute $\rho(\beta_2, 0)$, the expected number of valid coefficient vectors $\boldsymbol{\alpha}$ that satisfy Eq. (5.11). Since $\gamma = 0$, Eq. (5.5) requires the coefficients $\boldsymbol{\alpha}$ to satisfy $\|\boldsymbol{\alpha}\|_2 \leq \sqrt{K^2 + \beta_0^2}$. Define D_0 as the intersection between this radius $\sqrt{K^2 + \beta_0^2}$ ball and the linear constraints on $\boldsymbol{\alpha}$ given by $\boldsymbol{\ell}^{(\mathtt{B}_1)} = \mathbf{0}$,

$$D_0 = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^N : \|\boldsymbol{\alpha}\|_2 \leq \sqrt{K^2 + \beta_0^2},\ \boldsymbol{\alpha}^{(N/p)} = \mathbf{0} \text{ for each prime factor } p \right\} \ . \tag{5.15}$$

As $\boldsymbol{\alpha} \in \mathbb{Z}^N$, the collection $D_0^{\mathbb{Z}} = D_0 \cap \mathbb{Z}^N$ of integer valued vectors in D_0 contains the set of valid coefficient vectors $\boldsymbol{\alpha}$. We can thus approximate $\rho(\beta_2, 0)$ by,

$$\rho(\beta_2, 0) = \mathbb{E}\left| \left\{ \boldsymbol{\alpha} \in D_0^{\mathbb{Z}} : \beta_2^2 \sum_{m=1}^{M} |\tilde{\alpha}_{k_m}|^2 \leq K^2 + \beta_0^2 \right\} \right| = \sum_{\boldsymbol{\alpha} \in D_0^{\mathbb{Z}}} \mathbb{P}\left[\sum_{m=1}^{M} |\tilde{\alpha}_{k_m}|^2 < \frac{K^2 + \beta_0^2}{\beta_2^2} \right]$$

$$\lesssim \sum_{\boldsymbol{\alpha}\in D_0^{\mathbb{Z}}} \left(\frac{(K^2+\beta_0^2)/\beta_2^2}{\|\boldsymbol{\alpha}\|_1}\right)^M \approx \left|D_0^{\mathbb{Z}}\right| \left(\frac{(K^2+\beta_0^2)/\beta_2^2}{\mathbb{E}\left[\|\boldsymbol{\alpha}\|_1 : \boldsymbol{\alpha}\in D_0^{\mathbb{Z}}\right]}\right)^M . \tag{5.16}$$

We note that the last approximation in Eq. (5.16) is an underestimate by Jensen's inequality. However, numerical testing supported the approximation as the underestimate is very slight for moderately sized N.

It remains to approximate the size of the set $D_0^{\mathbb{Z}}$ and the expected value of $\|\boldsymbol{\alpha}\|_1$ in Eq. (5.16). The volume of D_0 approximates the number of integer points it contains. To compute the volume, note that D_0 is the intersection of an N-dimensional hyperball with a set of linear hyperplane constraints. This intersection results in another hyperball of smaller dimension. Since each hyperplane passes through the hyperball center $\mathbf{0}$, Section Appendix A shows that the intersection preserves the radius $\sqrt{K^2+\beta_0^2}$. Furthermore, the nullity of the constraint system determines the dimension of the intersection hyperball. The $\mathtt{B}_1$ basis block enforces the top block of the linear system in Eq. (4.3), and Section 4.1 showed that the totient $\Phi = \phi(N)$ gives the nullity of this system. In particular, we see that D_0 is a Φ-dimensional hyperball of radius $\sqrt{K^2+\beta_0^2}$. The number of integer points in D_0 is thus approximated by,

$$\left|D_0^{\mathbb{Z}}\right| \approx \mathrm{vol}(D_0) = V_\Phi (K^2+\beta_0^2)^{\Phi/2} , \tag{5.17}$$

where V_n is the volume of the n-dimensional unit hyperball,

$$V_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)} .$$

The other term to approximate in Eq. (5.16) is the expected value of $\|\boldsymbol{\alpha}\|_1$ for $\boldsymbol{\alpha}\in D_0^{\mathbb{Z}}$. As before, the approximation uses the continuous ball D_0. To begin, consider a vector $\boldsymbol{\alpha}$ distributed uniformly in a hyperball of radius R in $\mathbb{R}^n$. A single coordinate α_1 of $\boldsymbol{\alpha}$ has probability density,

$$\frac{V_{n-1}}{R^n V_n}\left(\sqrt{R^2-\alpha_1^2}\right)^{n-1} ,$$

where V_n again denotes the volume of the n-dimensional unit ball [56]. The expected absolute value of α_1 can be computed as

$$\mathbb{E}[|\alpha_1|] = \frac{V_{n-1}}{R^n V_n}\int_{-R}^{R} |\alpha_1| \left(\sqrt{R^2-\alpha_1^2}\right)^{n-1} d\alpha_1 = \frac{2RV_{n-1}}{(n+1)V_n} . \tag{5.18}$$

Therefore, the expected value of $\|\boldsymbol{\alpha}\|_1$ in an n-dimensional ball of radius R centered at the origin is

$$\mathbb{E}\left[\|\boldsymbol{\alpha}\|_1\right] = \mathbb{E}\left[\sum_{j=0}^{n-1}|\alpha_j|\right] = \sum_{j=0}^{n-1}\mathbb{E}\left[|\alpha_j|\right] = \frac{2nRV_{n-1}}{(n+1)V_n} . \tag{5.19}$$

While Eq. (5.19) computed the expected 1-norm over all points in an n-dimensional hyperball in $\mathbb{R}^n$, D_0 is a Φ-dimensional hyperball embedded in $\mathbb{R}^N$. In Section Appendix A.1, we show that Eq. (5.19) actually yields a lower bound on the expected 1-norm of points in D_0. This result is rather subtle, and it relies on the way 1-norms are increased by coordinate systems not aligned with the axes. By substituting the appropriate dimensions into the result of Eq. (5.19) and using the expected 1-norm in D_0 to approximate the expected 1-norm of its integer points, we obtain

$$\mathbb{E}\left[\|\boldsymbol{\alpha}\|_1 : \boldsymbol{\alpha}\in D_0^{\mathbb{Z}}\right] \approx \mathbb{E}\left[\|\boldsymbol{\alpha}\|_1 : \boldsymbol{\alpha}\in D_0\right] \geq \frac{2\Phi V_{\Phi-1}\sqrt{K^2+\beta_0^2}}{(\Phi+1)V_\Phi} . \tag{5.20}$$

After substituting our estimates in Eqs. (5.17) and (5.20) into Eq. (5.16), some simplication yields the estimated bound,

$$\rho(\beta_2, 0) \lesssim V_\Phi \left(K^2+\beta_0^2\right)^{(\Phi+M)/2}\left(\frac{(\Phi+1)V_\Phi}{2\Phi V_{\Phi-1}\,\beta_2^2}\right)^M . \tag{5.21}$$

Case (2): Fix $\gamma \neq 0$.
Without loss of generality, assume $\gamma > 0$. Continuing the inequality from Eq. (5.9) yields,

$$\beta_2^2 \sum_{m=1}^{M} |\tilde{\alpha}_{k_m} - \gamma \tilde{x}_{k_m}|^2 \leq K^2 + (1-\gamma^2)\beta_0^2 \leq K^2 \ , \tag{5.22}$$

where the last inequality holds as $\gamma \geq 1$. Thus, this bound will be tighter for smaller values of γ. Applying the definition of the DFT coefficients of $\boldsymbol{\alpha}, \mathbf{x}$ to Eq. (5.22) yields,

$$\beta_2^2 \sum_{m=1}^{M} \left| \sum_{n=0}^{N-1} (\alpha_n - \gamma x_n) \eta_N^{nk_m} \right|^2 \leq K^2 \ . \tag{5.23}$$

To compute $\rho(\beta_2, \gamma)$ for $\gamma \neq 0$, recall that Eq. (5.6) constrains the distance from the coefficient vector $\boldsymbol{\alpha}$ to the scaled guess $\gamma\overline{\mathbf{x}}$. Additionally incorporating the linear constraints from $\boldsymbol{\ell}^{(\mathtt{B}_1)} = \mathbf{0}$ yields the set of interest,

$$D_\gamma = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^N : \|\boldsymbol{\alpha} - \gamma\overline{\mathbf{x}}\|_2 < K,\ \boldsymbol{\alpha}^{(N/p)} = \gamma \mathbf{x}^{(N/p)} \text{ for each prime factor } p \right\} \ . \tag{5.24}$$

We again define the integer points in D_γ as $D_\gamma^{\mathbb{Z}} = D_\gamma \cap \mathbb{Z}^N$, which contains the set of valid coefficient vectors $\boldsymbol{\alpha}$ for the given γ. Note that Eq. (5.23) is almost identical to Eq. (5.11) from the $\gamma = 0$ case. Therefore, repeating the previous analysis with the substitutions $\alpha_n \to \alpha_n - \gamma x_n$, $\sqrt{K^2 + \beta_0^2} \to K$, and $D_0^{\mathbb{Z}} \to D_\gamma^{\mathbb{Z}}$ gives the approximation,

$$\begin{aligned} \rho(\beta_2, \gamma) &= \mathbb{E}\left| \left\{ \boldsymbol{\alpha} \in D_\gamma^{\mathbb{Z}} : \beta_2^2 \sum_{m=1}^{M} |\tilde{\alpha}_{k_m} - \gamma \tilde{x}_m|^2 \leq K^2 + (1-\gamma^2)\beta_0^2 \right\} \right| \\ &\lesssim \sum_{\boldsymbol{\alpha} \in D_\gamma^{\mathbb{Z}}} \left(\frac{(K/\beta_2)^2}{\|\boldsymbol{\alpha} - \gamma\mathbf{x}\|_1} \right)^M \approx \left| D_\gamma^{\mathbb{Z}} \right| \left(\frac{(K/\beta_2)^2}{\mathbb{E}\left[\|\boldsymbol{\alpha} - \gamma\mathbf{x}\|_1 : \boldsymbol{\alpha} \in D_\gamma^{\mathbb{Z}} \right]} \right)^M \ . \end{aligned} \tag{5.25}$$

Note that Lemma 2.3 implies $\mathbf{x}^{(N/p)} = \overline{\mathbf{x}}^{(N/p)}$, since $\tilde{x}_k = \tilde{\overline{x}}_k$ for all $p \,|\, k$. In particular, each hyperplane constraint of D_γ in Eq. (5.24) passes through the center $\gamma\overline{\mathbf{x}}$ of the ball $\|\boldsymbol{\alpha} - \gamma\overline{\mathbf{x}}\|_2 \leq K$. Since the constraint system still has nullity Φ, Section Appendix A implies that D_γ is a Φ-dimensional hyperball of radius K. Approximating the number of integer points in D_γ by its volume, we obtain the estimate,

$$\left| D_\gamma^{\mathbb{Z}} \right| \approx V_\Phi K^\Phi \ . \tag{5.26}$$

We now bound the expected value of $\|\boldsymbol{\alpha} - \gamma\mathbf{x}\|_1$ in Eq. (5.25). Again, we approximate with the expected value over $\boldsymbol{\alpha} \in D_\gamma$. Applying linearity of the expected value yields,

$$\mathbb{E}[\|\boldsymbol{\alpha} - \gamma\mathbf{x}\|_1 : \boldsymbol{\alpha} \in D_\gamma] = \mathbb{E}\left[\sum_{n=0}^{N-1} |\alpha_n - \gamma x_n| \right] = \sum_{n=0}^{N-1} \mathbb{E}[|\alpha_n - \gamma x_n|] \geq \sum_{n=0}^{N-1} |\mathbb{E}[\alpha_n - \gamma x_n]| = \sum_{n=0}^{N-1} |\gamma\overline{x}_n - \gamma x_n| \ ,$$

where the last equality used the fact that the expected location of $\boldsymbol{\alpha} \in D_\gamma$ is the center of the sphere at $\gamma\overline{\mathbf{x}}$. We further bound this result in terms of K by,

$$\mathbb{E}[\|\boldsymbol{\alpha} - \gamma\mathbf{x}\|_1] \geq \gamma \|\mathbf{x} - \overline{\mathbf{x}}\|_1 \geq \gamma \|\mathbf{x} - \overline{\mathbf{x}}\|_2 = \gamma K \ . \tag{5.27}$$

Finally, substituting Eqs. (5.26) and (5.27) into Eq. (5.25) yields the bound,

$$\rho(\beta_2, \gamma) \lesssim V_\Phi K^\Phi \left(\frac{K^2}{\gamma K \beta_2^2} \right)^M = V_\Phi K^\Phi \left(\frac{K}{\gamma \beta_2^2} \right)^M \ . \tag{5.28}$$

We can now combine the results of the two cases in Eqs. (5.21) and (5.28) to estimate $\rho(\beta_2)$, the expected number of valid vectors $\boldsymbol{\alpha} \in \mathbb{Z}^N$ which satisfy Eq. (5.9) for fixed β_2 and any valid γ. Since Eq. (5.3) implies $|\gamma| \leq \gamma_{\max} = \lfloor \sqrt{(K/\beta_0)^2 + 1} \rfloor$, $\rho(\beta_2)$ is given by,

$$\begin{aligned} \rho(\beta_2) = \sum_{\gamma=-\gamma_{\max}}^{\gamma_{\max}} \rho(\beta_2, \gamma) &= \rho(\beta_2, 0) + 2 \sum_{\gamma=1}^{\gamma_{\max}} \rho(\beta_2, \gamma) \\ &\lesssim V_\Phi \left(K^2 + \beta_0^2\right)^{(\Phi+M)/2} \left(\frac{(\Phi+1)V_\Phi}{2\Phi V_{\Phi-1}\, \beta_2^2} \right)^M + 2 \sum_{\gamma=1}^{\gamma_{\max}} V_\Phi K^\Phi \left(\frac{K}{\gamma \beta_2^2} \right)^M . \end{aligned} \tag{5.29}$$

We ideally choose β_2 large enough that only $\mathbf{x}^*$ and $-\mathbf{x}^*$ satisfy Eq. (5.9). We thus set $\rho(\beta_2) = 2$ in Eq. (5.29), giving,

$$2\beta_2^{2M} = V_\Phi \left(K^2 + \beta_0^2\right)^{(\Phi+M)/2} \left(\frac{(\Phi+1)V_\Phi}{2\Phi V_{\Phi-1}} \right)^M + 2V_\Phi K^{\Phi+M} \sum_{\gamma=1}^{\gamma_{\max}} \left(\frac{1}{\gamma} \right)^M .$$

Finally, solving for β_2 gives the estimate,

$$\beta_2 = \left[\frac{1}{2} V_\Phi \left(K^2 + \beta_0^2\right)^{(\Phi+M)/2} \left(\frac{(\Phi+1)V_\Phi}{2\Phi V_{\Phi-1}} \right)^M + V_\Phi K^{\Phi+M} \sum_{\gamma=1}^{\gamma_{\max}} \left(\frac{1}{\gamma} \right)^M \right]^{1/2M} . \tag{5.30}$$

5.3. Estimating K and Application to Matrix Reconstruction

The estimates for β_2 in the previous sections depend on the parameter $K = \|\mathbf{x} - \overline{\mathbf{x}}\|_2$ which measures the error between the true signal $\mathbf{x}$ and the guess $\overline{\mathbf{x}}$. Eq. (5.30) demonstrates a significant dependence of β_2 on K. In practical applications however, K will not be known *a priori* as it depends on the true signal $\mathbf{x}$. In this section, we provide an approximation of K for random signals, so that β_2 may be computed in practice.

As an example, let $\mathtt{X}$ be an $N_1 \times N_2$ matrix where each entry takes a binomial distribution,

$$X_{mn} \sim \text{binom}(L, p) .$$

The binomial model is a natural choice to study, and in particular includes the binary image case when $L = 1$. Similar calculations could, in principle, be carried out for other distributions, including a uniform distribution [57]. However, these computations are generally less direct, as they require analyzing the sum of such distributions.

Fix divisors D_1 and D_2 of N_1 and N_2, respectively. If k and l are integers such that $\gcd(k, N_1) = N_1/D_1$ and $\gcd(l, N_2) = N_2/D_2$, then the (k, l)-subsignal has length $D = \text{lcm}(D_1, D_2)$. By Lemma 3.1, each entry of this subsignal is a sum of $N_1 N_2/D$ entries of $\mathtt{X}$. Since the sum of binomial random variables is another binomial random variable, the subsignal entries have distribution,

$$x_j^{(k,l)} \sim \text{binom}\left(\frac{N_1 N_2}{D} L, p \right) . \tag{5.31}$$

For simplicity, we estimate K for a one-dimensional length N signal $\mathbf{x}$ with entries distributed as

$$x_n \sim \text{binom}(L, p) .$$

The substitutions $N \to D$ and $N_1 N_2 L/D \to L$ will then give an error estimate for the subsignals of a binomial matrix $\mathtt{X}$.

The second non-central moment of the binomial random variable x_n is given by,

$$\mathbb{E}\left[|x_n|^2\right] = Lp(1-p) + (Lp)^2 .$$

The expected value of 2-norm of $\|\mathbf{x}\|_2^2$ can then be computed by

$$\mathbb{E}\left[\|\mathbf{x}\|_2^2\right] = \mathbb{E}\left[\sum_{n=0}^{N-1} |x_n|^2 \right] = \sum_{n=0}^{N-1} \mathbb{E}\left[|x_n|^2\right] = N\, \mathbb{E}\left[|x_0|^2\right] = NLp(1-p) + N(Lp)^2, \tag{5.32}$$

where we have used linearity of expectation and that the coordinates of $\mathbf{x}$ are identically distributed. As a sum of binomial variables, the zeroth DFT coefficient has distribution,

$$\tilde{x}_0 = \sum_{n=0}^{N-1} x_n \sim \text{binom}(NL, p) \ ,$$

so the second non-central moment of $\tilde{x}_0$ is given by

$$\mathbb{E}\left[|\tilde{x}_0|^2\right] = NLp(1-p) + (NLp)^2 \ . \tag{5.33}$$

We make the reasonable assumption that Eq. (5.31) implies all DFT coefficients (except for $\tilde{x}_0$) have roughly the same distribution. Thus, let $\nu = \mathbb{E}\left[|\tilde{x}_k|^2\right]$ for each $0 < k < N$. The discrete Parseval relation implies,

$$N\,\mathbb{E}\left[\|\mathbf{x}\|_2^2\right] = \mathbb{E}\left[\|\tilde{\mathbf{x}}\|_2^2\right] = \mathbb{E}\left[\tilde{x}_0^2\right] + \sum_{k=1}^{N-1} \mathbb{E}\left[|\tilde{x}_k|^2\right] = \mathbb{E}\left[\tilde{x}_0^2\right] + (N-1)\nu. \tag{5.34}$$

Applying Eqs. (5.32) and (5.33) to Eq. (5.34) yields

$$N^2Lp(1-p) + N^2(Lp)^2 = NLp(1-p) + (NLp)^2 + (N-1)\nu \implies \nu = NLp(1-p) \ .$$

As discussed in Section 4.4, the guess $\overline{\mathbf{x}}$ is defined with $N - \phi(N) + 2M$ DFT coefficients of $\mathbf{x}$, including $\tilde{x}_0$. We can thus use ν to approximate the frequency space error of $\overline{\mathbf{x}}$, so one final application of the discrete Parseval relation gives,

$$\mathbb{E}\left[K^2\right] = \mathbb{E}\left[\|\mathbf{x} - \overline{\mathbf{x}}\|_2^2\right] = \frac{1}{N}\,\mathbb{E}\left[\|\tilde{\mathbf{x}} - \widetilde{(\overline{\mathbf{x}})}\|_2^2\right] = \frac{1}{N}(\phi(N) - 2M)\nu = (\phi(N) - 2M)Lp(1-p).$$

Finally, Jensen's inequality implies,

$$\mathbb{E}[K] = \mathbb{E}\left[\|\mathbf{x} - \overline{\mathbf{x}}\|_2\right] \leq \sqrt{\mathbb{E}\left[\|\mathbf{x} - \overline{\mathbf{x}}\|_2^2\right]} = \sqrt{(\phi(N) - 2M)Lp(1-p)} \ . \tag{5.35}$$

For the (k,l)-subsignal of the matrix $\mathtt{X}$, Eq. (5.35) becomes,

$$\mathbb{E}[K] = \mathbb{E}\left[\|\mathbf{x}^{(k,l)} - \overline{\mathbf{x}}^{(k,l)}\|_2\right] \leq \sqrt{(\phi(D) - 2M)\frac{N_1N_2}{D}Lp(1-p)}. \tag{5.36}$$

Substituting these bounds for K into Eq. (5.30) yields an approximate value of β_2 for practical usage.

5.4. Lattice Analysis Discussion

We emphasize that some of the parameter estimates in Section 5 are based on approximations and probabilistic assumptions. Nevertheless, these estimates highlight how short vectors in the lattice depend on key parameters. In this section, we examine these trends and comment on the reliability of the overall estimates. The following section provides numerical results that justify the accuracy of the approximations.

Among the parameters, β_0 is chosen first, which influences the value of β_1 and β_2 (both of which also depend on K). Section 5.1 gives a modest condition on β_1. It suffices to take $\beta_1 > \sqrt{K^2 + \beta_0^2}$, a value substantially smaller than the typical size of β_2 from Eq. (5.30). Since β_2 can be many orders of magnitude larger, the value of β_2 relates closely to the precision required in the DFT coefficient data. Indeed, the parameter β_2 scales the DFT coefficients in the $\mathtt{B}_2$ block of the lattice in Eq. (4.5). This scaling amplifies any imprecision in the DFT data, and making β_2 significantly larger than the DFT coefficient precision introduces additional noise.

The blocks $\mathtt{B}_1$ and $\mathtt{B}_2$ also have an important qualitative difference which distinguishes the conditions for β_1 and β_2. The requirement $\beta_1 > \sqrt{K^2 + \beta_0^2}$ exactly enforces the integer constraints of the $\mathtt{B}_1$ block. In contrast, spurious short lattice vectors may come arbitrarily close to satisfying the $\mathtt{B}_2$ block constraints,

which we modeled with a probabilistic estimate for β_2. It is therefore natural to focus on the accuracy of the estimate for β_2. Because this estimate depends on both K and β_0, we begin by considering the accuracy of the approximation for K in Eq. (5.35) before turning to the choice of β_0 and finally β_2.

We tested the approximation for K numerically through a Monte Carlo simulation. The quantity $K = \|\mathbf{x}-\overline{\mathbf{x}}\|_2$ was computed for 1e5 randomly generated integer vectors of length $N = 30$ when different numbers M of DFT coefficients are known. Each vector $\mathbf{x}$ was randomly generated according to a binomial distribution with $L = N$ and $p = 0.5$. For $M = 1, 2$, and 3, Figure 4 plots the smoothed density of the distribution of $\|\mathbf{x}-\overline{\mathbf{x}}\|_2$ from the simulation. The upper bound in Eq. (5.35) (red dashed line) appears to be a reasonably tight approximation of the experimental average, $\mathbb{E}[\|\mathbf{x}-\overline{\mathbf{x}}\|_2]$ (blue dashed line). However, Figure 4 shows that the expected value bound in Eq. (5.35) is clearly not an upper bound on possible values of $\|\mathbf{x}-\overline{\mathbf{x}}\|_2$. Nevertheless, Eq. (5.35) gives a reasonable approximation, and unless otherwise noted, all future computations of β_2 will use the estimate for K in Eq. (5.35) with the binomial success probability parameter set to $p = 0.5$. Figure 4 also demonstrates that the distributions of K shift left as M increases, as predicted by Eq. (5.35). This consequence of the Parseval relation reflects that knowledge of additional DFT coefficients generally improves the accuracy of the guess.

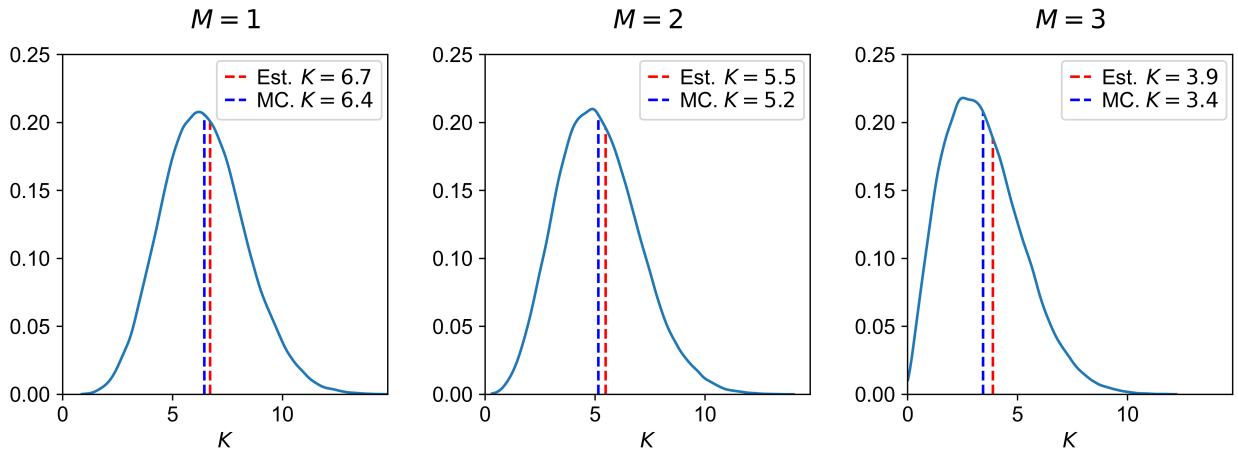

Figure 4: For $M = 1, 2, 3$, the smoothed Monte Carlo distribution of guess errors $\|\mathbf{x} - \overline{\mathbf{x}}\|_2$ (solid blue line), the experimental average error (blue dashed line, value in legend), and the theoretical bound (red dashed line, value in legend). The test set used 1e5 signals of length $N = 30$ with entries distributed as binom$(N, 0.5)$.

We turn now to the choice of β_0. Recall from Section 5.1 that, although setting $\beta_0 > K$ ensures $|\gamma| \leq 1$, larger β_0 values lengthen the shortest lattice vector. An intermediate value of β_0 smaller than K will balance these effects. Figure 5 plots the expression for β_2 in Eq. (5.30) as a function of β_0 for $N = 30$ and $M = 1, 2$, and 3. Note that the sharp jumps in the curves come from the floor function in the definition of $\gamma_{\max}$. Ignoring these jumps, the curves are generally convex, consistent with the expectation that intermediate values of β_0 are optimal. Overall however, the sensitivity of β_2 to β_0 is fairly weak. We remark that the β_2 analysis required a value of $\beta_0 > 0$ to bound $|\gamma|$, as otherwise the sum in Eq. (5.30) diverges when $M = 1$. In numerical experiments, however, smaller values of β_0 seem to work better, and the main benefit to including this

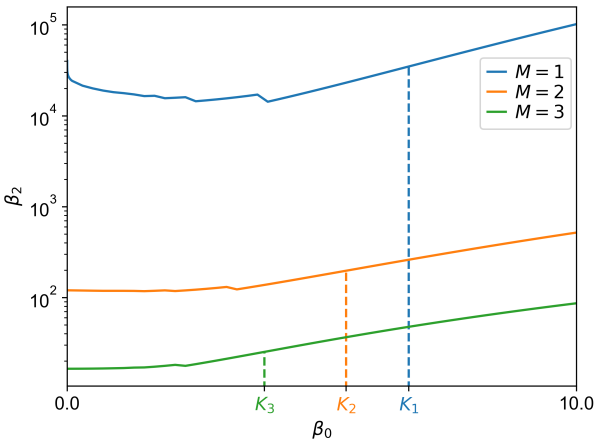

Figure 5: For $M = 1, 2, 3$, the estimated β_2 from Eq. (5.30) plotted against β_0 when $N = 30$ and K is computed from the bound in Eq. (5.35).

parameter comes from our ability to determine whether $|\gamma| = 1$ from the $\boldsymbol{\ell}^{(\mathbb{B}_0)}$ block of the SVP result. Since Figure 5 suggests that β_0 does not significantly impact the estimate of β_2, and numerical tests suggest that smaller choices of β_0 perform better in practice, we fix $\beta_0 = 0.1$ for all subsequent computations.

With β_0 and K fixed, we can analyze the estimate for β_2. The estimate used several approximations and assumptions with varying degrees of accuracy. In particular, modeling the DFT coefficients as random sums of points on the unit circle is more accurate for large N. The bound in Eq. (5.13) is tight for $M = 1$, but becomes coarser as M increases. Various estimates in Section 5.2 use continuous analogues to approximate properties of sets of integer vectors. We will see in Section 6 that despite these approximations, the estimate for β_2 has strong numerical evidence for practical usage.

We first investigate the dependence of β_2 on the signal length N. In estimate Eq. (5.30), N only appears through $\Phi = \phi(N)$. Therefore, it is not simply the size of N that matters, but also both the number and the specific primes dividing N. For example, compare random binary signals of length $N = 59$ and $N = 60$. For $M = 1$ (with β_0 and K chosen as described above), the corresponding values of β_2 are 9.12e08 and 1.93e02, respectively. It may seem paradoxical that the prime $N = 59$ requires a significantly larger value of β_2 (and thus much greater DFT coefficient precision) than the larger (but composite) $N = 60$. However, this setup uses the minimal amount of DFT data for unique inversion, which corresponds to the set of all divisors for one-dimensional signals. For $N = 59$, we sample $\tau(59) = 2$ DFT coefficients, whereas $N = 60$ requires the measurement of $\tau(60) = 12$ DFT coefficients. Given this large disparity in data size, it is natural that the prime $N = 59$ requires much higher precision for inversion.

As shown in Figure 6, the plots of β_2 versus N closely resemble the plot of the totient function $\phi(N)$. The order of magnitude of β_2 is roughly proportional to $\phi(N)$, consistent with the exponent of $\phi(N)$ in Eq. (5.30). Several different values of M and L display this general shape (where L is the binomial trial parameter and affects the approximation for K). As expected, the graph is shifted down and rescaled for larger values of M or smaller values of L, since the associated inverse problems have more data and smaller solution spaces, respectively.

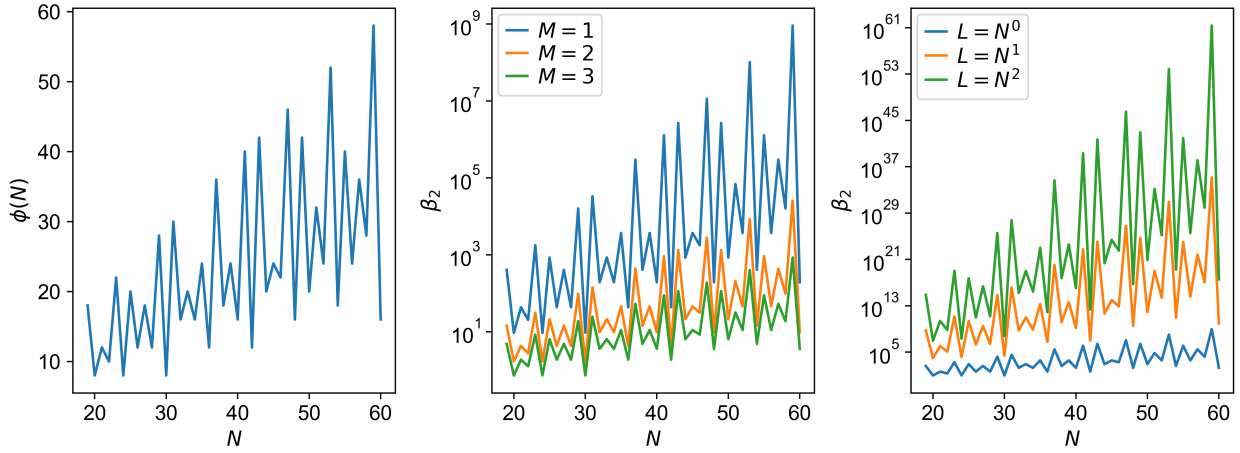

Figure 6: **Left**: plot of the totient function $\phi(N)$ vs N. **Center**: For number of DFT coefficients $M = 1, 2, 3$, plots of the estimated β_2 from Eq. (5.30) vs N with $L = 1$. **Right**: For binomial trial parameter $L = N^0, N^1, N^2$, plots of the estimated β_2 from Eq. (5.30) vs N with $M = 1$. In the estimate for K from Eq. (5.35), the binomial probability parameter is fixed at $p = 0.5$.

Finally, Figure 7 displays two additional plots regarding the dependence of β_2 on M. For any fixed N, the required value of β_2 decreases as M increases, since providing more data improves reconstruction stability. The second plot in Figure 7 plots $\rho(\beta_2, \gamma)$ from Eq. (5.29) as a function of $|\gamma|$ for $N = 11$. For each value of M, β_2 is fixed as the heuristic in Eq. (5.30), which guarantees that $\rho(\beta_2) = 2$. Recall that the desired shortest vector corresponds to $\gamma = 1$. For small M, ρ exhibits a heavier tail, implying a higher likelihood of spurious short vectors that have $|\gamma| > 1$. As M increases, the density of sufficiently short vectors increasingly concentrates at $\gamma = 1$.

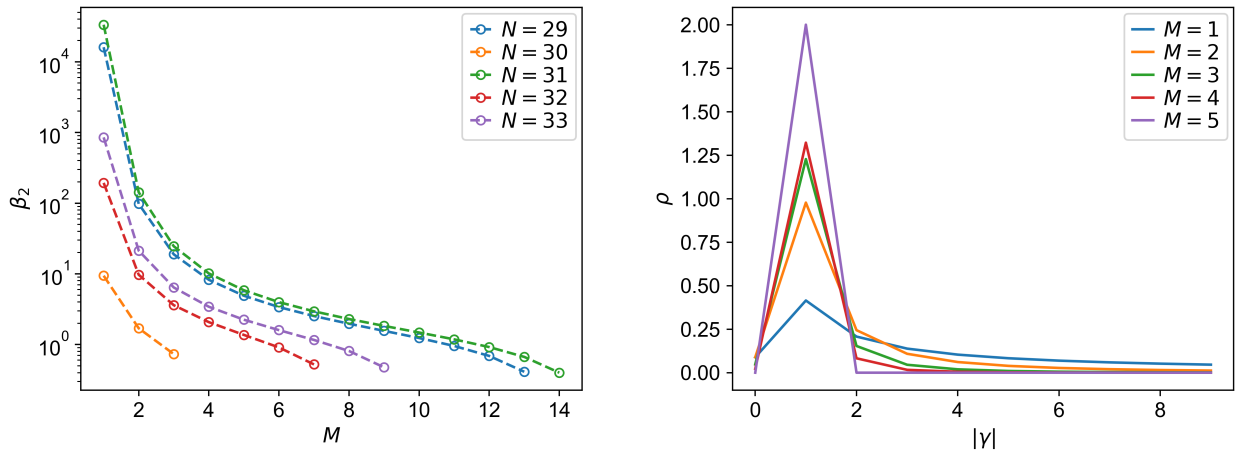

Figure 7: **Left**: For $N = 29$ through 33, plots of the estimated β_2 from Eq. (5.30) vs M for $1 \leq M < \phi(N)/2$. **Right**: For $N = 11$ and $M = 1$ through 5, plots of $\rho(\beta_2, \gamma)$ from Eq. (5.29) vs $|\gamma|$ where β_2 is given by Eq. (5.30).

6. Numerical Reconstructions

6.1. Floating Point LLL for Integer Lattices

As mentioned in Section 4.4, our implementations of Algorithms 2 and 4 use an approximation algorithm to address the NP-hard shortest vector problem (SVP) for the lattice defined in Eq. (4.5). The Lenstra-Lenstra-Lovász (LLL) algorithm gives a polynomial-time method for approximating the SVP. While finding an exact shortest basis is NP-hard, LLL outputs a reduced basis guaranteed to be within an approximation factor of the true shortest basis. In practice, the LLL reduced basis vectors are nearly orthogonal and much shorter than the original input basis, making the structure of the lattice more tractable [41].

Although the algorithm implementations use LLL, the parameter analysis in Section 5 assumed that the SVP was solved exactly. We estimated sufficient values of β_0, β_1, and β_2 to guarantee that the desired $\mathbf{x}^*$ is the shortest vector in the lattice. In reality, when applied to the SVP in Eq. (4.5), LLL offers the approximation guarantee,

$$\|\hat{\boldsymbol{\ell}}\|_2 \leq \left(\frac{4}{4\delta - 1}\right)^{N/2} \|\boldsymbol{\ell}\|_2, \tag{6.1}$$

where $\boldsymbol{\ell}$ denotes the shortest lattice vector as in Section 5, $\hat{\boldsymbol{\ell}}$ denotes the shortest vector in the LLL reduced basis, and δ is a parameter that can be tuned to trade off between speed and accuracy [58, Theorem 5.2]. The lattice parameter analysis could be adapted to this approximate SVP formulation by replacing the condition $\|\boldsymbol{\ell}\|_2 \leq \sqrt{K^2 + \beta_0^2}$ in Eq. (5.2) with,

$$\|\hat{\boldsymbol{\ell}}\|_2 \leq \left(\frac{4}{4\delta - 1}\right)^{N/2} \sqrt{K^2 + \beta_0^2}\ . \tag{6.2}$$

However, conducting the analysis with Eq. (6.2) would drastically increase the estimated value of β_2. Moreover, the approximation bound in Eq. (6.1) is not tight in general [46]. The numerical results in the next section support the use of Eq. (5.30) as an effective estimate, and discuss when the approximation factor becomes significant.

On the other hand, using the bound from Eq. (6.2) in the β_1 analysis of Section 5.1 yields the condition,

$$\beta_1 \geq \left(\frac{4}{4\delta - 1}\right)^{N/2} \sqrt{K^2 + \beta_0^2}\ . \tag{6.3}$$

Choosing β_1 to satisfy Eq. (6.3) ensures that the shortest LLL vector satisfies the $\mathtt{B}_1$ block of the lattice in Eq. (4.5) exactly. Since the LLL approximation factor is not particularly tight, the smaller computed value

of β_1 in Section 5.1 would likely suffice in practice. However, the expression in Eq. (6.3) is still significantly smaller than the β_2 estimate in Eq. (5.30). As such, we may use a larger β_1 value which satisfies Eq. (6.3) without any repercussions.

We now discuss the implementation used in the numerical experiments. Algorithms 2 and 4 were implemented in pure Python leveraging the fast vectorization provided by NumPy [59]. To solve the SVP in Eq. (4.5), we use the floating point LLL algorithm (FPLLL) [60, 61] included in the fplll library [62] with Python interface fpylll [63]. The FPLLL algorithm replaces exact rational arithmetic with floating-point computations, accelerating lattice basis reduction. This significantly improves speed, but introduces the challenge of accumulated rounding errors. FPLLL balances efficiency and stability with adaptive precision and error monitoring, achieving a reduced basis quality comparable to exact LLL while being much faster in practice. This efficiency allowed us to recover larger signals than were feasible in our previous work [33].

The FPLLL libraries operate on integer bases, whereas the $\mathtt{B}_2$ block of the lattice in Eq. (4.5) contains irrational real entries. To handle this discrepancy, our algorithm implementation introduces an additional parameter β_3. We multiply the entire basis $\mathtt{B}$ by β_3 and then truncate the scaled basis, before passing $\lfloor \beta_3 \mathtt{B} \rfloor$ to the FPLLL algorithm. Truncation limits the effective precision in the data by discarding trailing digits of the DFT coefficients. As both β_2 and β_3 scale $\mathtt{B}_2$ before truncation, the number of preserved digits past the decimal point is approximately $\lfloor \log_{10}(\beta_2 \beta_3) \rfloor$. After sufficient testing, we set $\beta_3 =$1e2 in our numerical simulations.

This interpretation of precision assumes exact knowledge of the DFT coefficients. In practice however, the DFT coefficient data has finite precision. While our theoretical results guarantee that two distinct integer signals cannot have identical sampled spectra, the spectra may be indistinguishable at certain precision level. Thus, standard single or even double precision data may not suffice for unique recovery in some cases. To explore the limits of the lattice method and enable recovery of larger signals, we conduct some numerical experiments with mixed precision floating point arithmetic using the Python module gmpy2 module. Setting the precision much higher (sometimes up to 50 digits) lets us evaluate the algorithm's performance in a more theoretical regime.

Algorithm 5 presents our implementation of the lattice-based solution to Eq. (4.3). The algorithm uses a tolerance parameter ϵ to determine whether a candidate integer signal matches the DFT data within the specified tolerance. Lines 1-3 first check whether the guess $\overline{\mathbf{x}}$ is sufficiently close to the DFT data. For certain values of N, including $N = 1, 2, 3, 4, 6$, unique inversion requires all DFT coefficients. These cases require no lattice reduction, since the guess is immediately correct with $\overline{\mathbf{x}} = \mathbf{x}$.

The remainder of Algorithm 5 recovers $\mathbf{x}$ using FPLLL. Lines 4 and 5 construct the integer lattice basis and reduce it with FPLLL, yielding a δ-reduced basis with parameter $\delta = 0.9972$. For each vector in the reduced basis, we round its entries in Line 7 before checking against the DFT data. This rounding handles the errors introduced by rescaling and truncating $\mathtt{B}$ to $\lfloor \beta_3 \mathtt{B} \rfloor$. Since the correct $\mathbf{b}$ lies within $1/\beta_3$ of $\mathbf{x}^*$, rounding succeeds for sufficiently large β_3 values. Finally, the check step in Line 8 includes the additional condition $b_N = \pm\beta_0$, as the vector $\mathbf{b} = \mathtt{B} \begin{bmatrix} \boldsymbol{\alpha} & \gamma \end{bmatrix}^T$ satisfies $b_N = \pm\beta_0$ if and only if $\gamma = \pm 1$. The full implementation of minimal DFT sampling and the one-dimensional and two-dimensional inversion algorithms is available at [64].

The FPLLL algorithm implementation used in this work has the runtime bound, $O(d^4 n \log^2 B)$, for a d-dimensional lattice in $\mathbb{R}^n$, where B bounds the 2-norms of the input basis vectors [60]. To apply this runtime bound to the lattice basis in Eq. (4.5), note that the longest basis vector is expected to be $\mathbf{b}_N$, with

$$B^2 = \|\mathbf{b}_N\|_2^2 = K^2 + \beta_0^2 + \beta_1^2 \sum_{j=1}^{\omega} \|\mathbf{x}^{(N/p_j)}\|_2^2 + \beta_2^2 \sum_{m=1}^{M} |\tilde{x}_{k_m}|^2 \ .$$

Assuming that $\beta_2 \gg K, \beta_0, \beta_1$ and the entries of $\mathbf{x}$ are bounded by $0 \le x_n \le L$, we have

$$B^2 = O\left(\beta_2^2 \sum_{m=1}^{M} |\tilde{x}_{k_m}|^2 \right) = O(M(\beta_2 N L)^2), \tag{6.4}$$

Algorithm 5 Shortest Vector Approximation with LLL

Require: Error tolerance ϵ, scale β_3

$\mathbf{y} \leftarrow \overline{\mathbf{x}}$ rounded entrywise to integers
if $\max_k |\tilde{y}_k - \tilde{x}_k| \leq \epsilon$ **then** ▷ max *over known DFT frequencies.*
    return $\mathbf{y}$
Construct the lattice basis $\mathtt{B}$ in Eq. (4.5)
$\mathtt{B}^* \leftarrow$ the LLL reduction of $\lfloor \beta_3 \mathtt{B} \rfloor$
for $\mathbf{b} \in \mathtt{B}^*/\beta_3$ **do**
    $\mathbf{y} \leftarrow \pm\mathbf{b} + \overline{\mathbf{x}}$ rounded entrywise to integers
    if $b_N = \pm\beta_0$ and $\max_k |\tilde{y}_k - \tilde{x}_k| \leq \epsilon$ **then**
        return $\mathbf{y}$

where we have used the triangle inequality to bound $|\tilde{x}_k| \leq \sum |x_n| \leq NL$. Substituting the estimated β_2 value from Eq. (5.30) into Eq. (6.4), we obtain

$$\begin{aligned} \log B &= O(\log MNL\beta_2) = O(\log MNLK^{\Phi}) \\ &= O(\Phi \log MNLK) = O(N \log MNLK) = O(N \log MNL) \ , \end{aligned} \tag{6.5}$$

where the final step uses the bound for K in Eq. (5.35). Since $\mathtt{B}$ is an $(N+1)$-dimensional lattice in $\mathbb{R}^{N+\phi(N)+2M}$, and $N + \phi(N) + 2M = O(N)$, the runtime bound becomes, $O(N^7 \log^2 MNL)$. This gives a pseudo-polynomial upper bound on the runtime of any iteration of Algorithm 2 or Algorithm 4 in terms of the current subproblem size N, number of coefficients M, and integer bounds L.

6.2. Lattice Analysis Verification

We now present numerical results supporting the analysis in Section 5. To evaluate the theoretical results, we examine how the parameter β_2 influences the recovery of random signals. In general, the value of β_2 required to invert a fixed length integer signal $\mathbf{x}$ depends on the particular choice of $\mathbf{x}$. Each value of β_2 can recover a certain proportion of signals of length N, and increasing β_2 generally allows a larger proportion of signals to be recovered. Figure 8 displays the cumulative distribution of required β_2 values for $N = 23$, $L = N$, and $M = 1$, 2, and 3. To simulate the distribution, 1000 random test signals with entries sampled from binom$(N, 0.5)$ were generated. For a range of β_2 values near the heuristic estimate from Eq. (5.30), we inverted each test signal and plotted the fraction of successful recoveries. Recovery improves monotonically with β_2, and the theoretical value from Eq. (5.30) (vertical dashed red line) serves as an accurate estimate on the β_2 value needed to recover all signals.

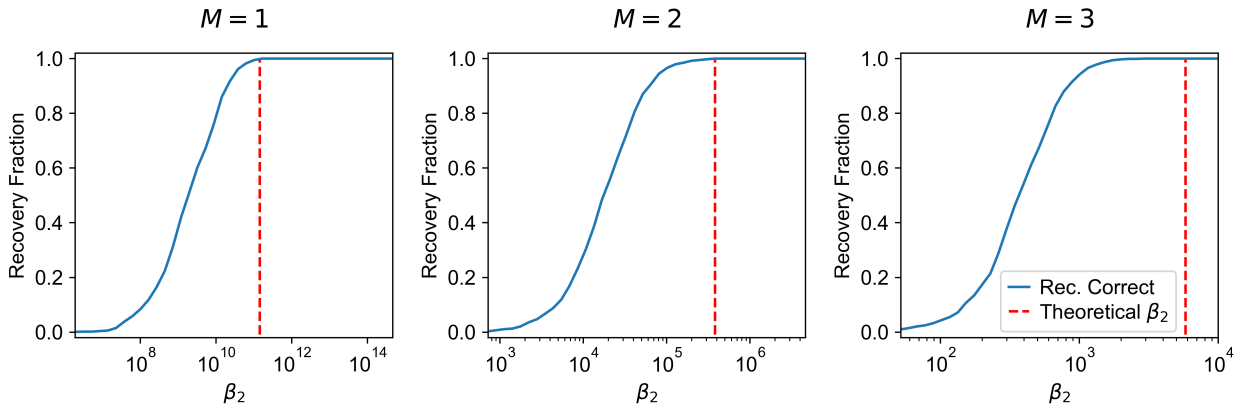

Figure 8: For $M = 1$, 2, and 3: Fraction of test signals correctly recovered over a range of β_2 values (blue solid line) compared to heuristic β_2 value (red dashed line). The test set used 1000 random signals of length $N = 23$ with entries distributed as binom$(N, 0.5)$.

Table 1 compares the theoretical estimate for β_2 in Eq. (5.30) to experimental values for several combinations of N and M. For each test, we set the binomial parameter to $L = N$. For each signal length N, 100 test signals were randomly generated according to $\text{binom}(N, 0.5)$. Instead of solving the full inversion problem, we performed only the final iteration in Algorithm 2, which we call a "top-level inversion". The decimated signals from the intermediate subproblems were computed directly from the test signal and given as input to the algorithm. For each pair of N and M, the β_2 values required to recover 50%, 90%, and 100% of the test signals were computed numerically with a 30 iteration exponential search. For example, the search for the 50th percentile β_2 started with the estimated value as a guess, and then iteratively rescaled the guess by 1e2 to obtain a lower and upper bound for the value of β_2 that recovered exactly 50% of the test signals. These bounds were given to a bisection search using the geometric mean, which continued for the remaining iterations.

The experimental results in Table 1 broadly support the theoretical β_2 estimate, showing similar dependence on N and M. For small values of N and M, the heuristic β_2 is very close to the value required to recover all test signals. The first instance where the heuristic β_2 fails to bound this 100th percentile value occurs at $N = 31$ with $M = 1$. Although the estimated β_2 is not always a tight upper bound, it consistently provides a reasonable baseline, and, importantly, captures the relative changes with increases in M.

The accuracy of the β_2 estimate, relative to the empirical 100th percentile value, depends primarily on the number of known DFT coefficients. Prior to its final iteration, Algorithm 2 has recovered all $N - \phi(N)$ DFT frequencies which are not relatively prime with N, so a top-level inversion has access to a total of $N - \phi(N) + 2M$ DFT coefficients. The coefficients are distinguished by their role in the lattice: the $N - \phi(N)$ coefficients from proper subproblems contribute to the $\mathtt{B}_1$ block of the lattice and constrain the recovered signal exactly, while the remaining $2M$ DFT coefficient constraints form the $\mathtt{B}_2$ block. Increasing M by 1 contributes two conjugate DFT coefficients to $\mathtt{B}_2$ but leaves $\mathtt{B}_1$ unchanged. In general, when N has many divisors the estimated value of β_2 tends to overestimate the empirical 100th percentile (for example, $N = 48$, 60, 90), while it tends to underestimate when N has few divisors (for example, $N = 31$, 47, 49).

We rationalize these empirical trends by considering the lattice structure and the LLL approximation factor. The key parameter $\phi(N)$ gives the nullity of the $\mathtt{B}_1$ block constraints in Eq. (4.5). As we chose β_1 sufficiently large to guarantee the shortest lattice vector satisfies the $\mathtt{B}_1$ block constraints exactly, a larger $\phi(N)$ puts fewer constraints on the shortest vector found by LLL. A large $\phi(N)$ thus creates more opportunities for vectors with small $\mathtt{A}$, $\mathtt{B}_0$, and $\mathtt{B}_2$ components to simultaneously satisfy the limited set of $\mathtt{B}_1$ block constraints. Furthermore, Eq. (5.35) implies that a larger value of $\phi(N)$ directly increases the error of the guess $\overline{\mathbf{x}}$, lengthening the shortest lattice vector $\mathbf{x}^*$ in Eq. (5.1). This amplifies the impact of the LLL approximation factor, raising the chance that LLL selects a spurious short vector. As indicated by Eq. (6.2), a larger K makes other lattice vectors more likely to fall within the approximation factor of $\mathbf{x}^*$. Together, these effects provide some justification for why a large $\phi(N)$ necessitates values of β_2 beyond the theoretical estimate.

To test the role of the LLL approximation factor in the empirical β_2 values, we inverted 300 test signals across a range of β_2 values near the theoretical estimate from Eq. (5.30). For each β_2 value, we counted the number of test signals inverted successfully. For each failure, we further compared the shortest vector $\hat{\boldsymbol{\ell}}$ in the LLL reduced basis to the desired shortest vector $\mathbf{x}^*$. If $\|\hat{\boldsymbol{\ell}}\|_2 > \|\mathbf{x}^*\|_2$, the failure can be attributed to the approximation factor, since LLL certainly did not find the shortest vector. If $\|\hat{\boldsymbol{\ell}}\|_2 \leq \|\mathbf{x}^*\|_2$, then β_2 was set too small. Of course, it remains possible that even shorter vectors than $\hat{\boldsymbol{\ell}}$ and $\mathbf{x}^*$ exist in either case.

The plots in Figure 9 illustrate the data from this test for $N = 47$ and 48, with $M = 1$. Each plot shows the fraction of test signals recovered successfully (blue), the fraction of tests where inversion failed with a returned lattice vector $\widehat{\boldsymbol{\ell}}$ shorter than $\mathbf{x}^*$ (orange), and the sum of these two cases (green). In an exact solution of SVP, this green curve would remain identically 1. Deviations from 1 therefore indicate inexact SVP solutions, as we know some failures did not return the shortest vector.

For $N = 48$ with $M = 1$, the top-level inversion has 34 available DFT coefficients (about 71% of the total), giving both a strong initial guess and many constraints in the $\mathtt{B}_1$ block. As expected, the approximation factor is insignificant, with the blue curve essentially staying at 1. In line with the previous trends and Figure 8, the inversion success rate increases as β_2 increases, and the estimated β_2 value proves overly

conservative. For the case of $N = 47$ however, only 3 DFT coefficients are available with $M = 1$. The green curve dips as low as 0.15, showing that many inversion failures arise from the approximation factor. For $\beta_2 >$ 1e24, most inversion failures involved an approximate shortest vector which was longer than $\mathbf{x}^*$. While the theoretical β_2 drastically underestimates the value required for reliable recovery with LLL, it accurately bounds the value required for $\mathbf{x}^*$ to be the shortest lattice vector.

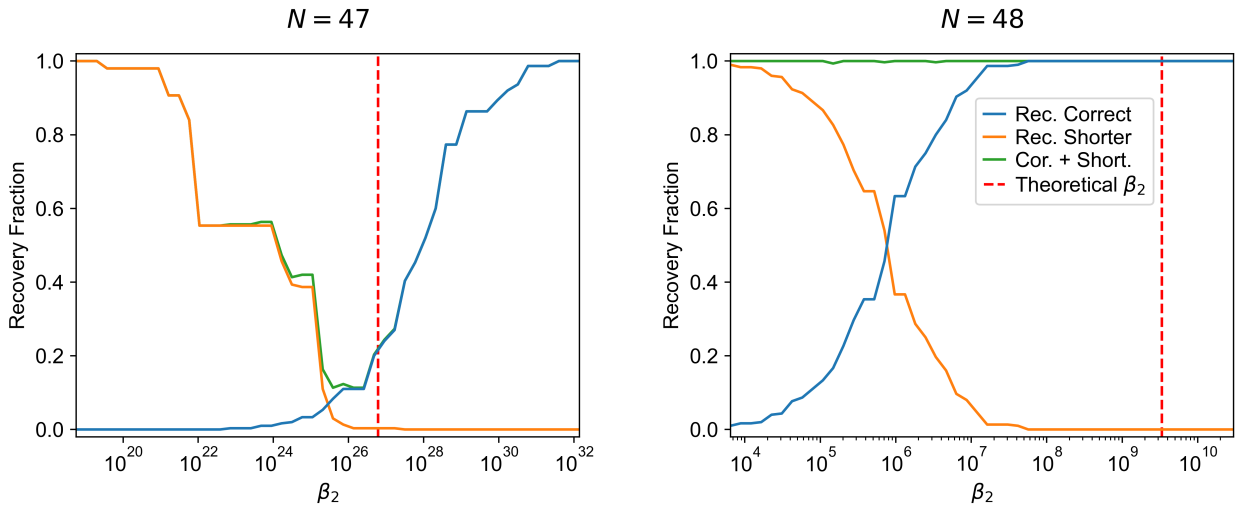

Figure 9: For $N = 47$ and 48 with $M = 1$: fraction of test signals recovered correctly (green), fraction where LLL found shorter basis vector than desired (orange), and fraction where either LLL succeeded or found a shorter vector (blue) compared to theoretical estimated β_2 value (red dashed line). For each N, the test set used 300 random signal of length N with entries distributed as $\text{binom}(N, 0.5)$.

The verification tests above focus on the values of β_2 required for the top-level inversion to succeed. However, a full application of Algorithm 2 or Algorithm 4 involves solving SVPs for several lattice bases. While the final subproblem addressed by the top-level inversion often requires the largest β_2 value, this will not always be the case. Proper subproblems recover shorter signals, but also involve larger integer bounds. Consider a signal of size N with an immediate subproblem of size $N' = N/p$, for some prime factor p of N. To understand the relationship between $\phi(N)$ and $\phi(N')$, consider the identity

$$\phi(n) = n \prod_{q \mid n} \left(1 - \frac{1}{q}\right), \tag{6.6}$$

where the product ranges over the prime factors q of n. If $p \mid N'$, then N' has the same prime factors as N, so Eq. (6.6) implies that

$$\phi(N') = N' \prod_{q \mid N'} \left(1 - \frac{1}{q}\right) = \frac{N}{p} \prod_{q \mid N} \left(1 - \frac{1}{q}\right) = \frac{\phi(N)}{p} .$$

On the other hand, if $p \nmid N'$, then the products in Eq. (6.6) for N and N' differ by a factor of $1 - 1/p$, giving

$$\phi(N') = N' \prod_{q \mid N'} \left(1 - \frac{1}{q}\right) = \frac{N}{p} \cdot \frac{\prod_{q \mid N} (1 - 1/q)}{(1 - 1/p)} = \frac{\phi(N)}{p(1 - 1/p)} = \frac{\phi(N)}{p - 1} .$$

To apply the error bound K on the guess from Section 5.3, assume the entries of the signal $\mathbf{x}^{(N)}$ are distributed as $\text{binom}(L, 0.5)$. Then, the decimated signal $\mathbf{x}^{(N')}$ has entries distributed as $\text{binom}(L', 0.5)$, where $L' = N/N' \cdot L = pL$. Applying Eq. (5.35) as an approximation for $K' = \|\mathbf{x}^{(N')} - \overline{\mathbf{x}}^{(N')}\|_2$ with these

N	M	Coeffs, #	Theory β_2	50th perc. β_2	90th perc. β_2	100th perc. β_2
19	1	3	5.69e08	1.43e07	1.03e08	5.12e08
	2	5	2.28e04	1.64e03	5.24e03	1.07e04
	3	7	8.35e02	7.18e01	1.65e02	2.64e02
24	1	18	1.45e04	8.56e01	6.38e02	4.47e03
	2	20	9.09e01	3.11e00	9.08e00	2.16e01
	3	22	1.34e01	1.00e00	1.80e00	4.26e00
30	1	24	2.42e04	7.15e01	5.05e02	2.50e03
	2	26	1.20e02	2.16e00	6.73e00	3.56e01
	3	28	1.65e01	1.00e00	1.64e00	3.40e00
31	1	3	1.44e16	1.37e15	2.35e16	1.39e17
	2	5	1.32e08	1.19e07	6.64e07	1.34e08
	3	7	3.10e05	2.88e04	7.57e04	1.55e05
32	1	18	5.85e08	1.60e06	1.19e07	9.74e07
	2	20	2.38e04	4.76e02	1.16e03	3.31e03
	3	22	8.70e02	2.89e01	5.69e01	1.06e02
39	1	17	3.95e13	1.39e10	9.64e10	7.20e11
	2	19	6.82e06	4.30e04	1.35e05	3.57e05
	3	21	4.22e04	6.09e02	1.12e03	2.11e03
45	1	23	9.72e13	1.27e10	1.18e11	4.02e11
	2	25	1.09e07	3.25e04	1.33e05	3.28e05
	3	27	5.82e04	4.80e02	1.18e03	2.23e03
47	1	3	6.17e26	2.40e28	4.20e29	5.69e30
	2	5	3.02e13	4.88e13	4.19e14	4.04e15
	3	7	1.27e09	5.23e08	1.64e09	5.23e09
48	1	34	3.34e09	7.15e05	4.26e06	2.36e07
	2	36	5.92e04	3.03e02	7.76e02	1.47e03
	3	38	1.65e03	1.84e01	3.76e01	9.33e01
49	1	9	4.79e24	1.21e24	4.05e25	5.34e26
	2	11	2.65e12	3.55e11	1.89e12	1.12e13
	3	13	2.49e08	1.64e07	9.91e07	2.00e08
50	1	32	8.70e11	1.08e08	9.25e08	1.97e09
	2	34	1.00e06	3.64e03	1.00e04	2.06e04
	3	36	1.16e04	1.04e02	1.85e02	3.46e02
60	1	46	8.70e09	3.42e05	2.73e06	9.73e06
	2	48	9.79e04	2.03e02	5.12e02	9.19e02
	3	50	2.35e03	1.27e01	2.83e01	4.60e01
90	1	68	7.61e15	9.71e09	7.71e10	7.53e11
	2	70	1.03e08	2.90e04	8.28e04	2.24e05
	3	72	2.77e05	3.81e02	7.79e02	1.63e03

Table 1: For selected values of N and $M = 1, 2, 3$, number of DFT coefficients available at the top level inversion, $N - \phi(N) + 2M$, the heuristic value of β_2 from Eq. (5.30), and the empirical 50th, 90th, and 100th percentile β_2 values computed from an exponential search with 30 iterations. For each N, the test set consisted of 100 random signals of length N with entries distributed as $\text{binom}(N, 0.5)$.

values of $\phi(N')$ and L' yields,

$$\mathbb{E}[K'] \approx \frac{1}{2}\sqrt{(\phi(N')-2M)L'} = \frac{1}{2}\sqrt{(\phi(N')-2M)pL} = \begin{cases} \frac{1}{2}\sqrt{(\phi(N)-2pM)L} & \text{if } p \,|\, N' \\ \frac{1}{2}\sqrt{(\phi(N)\frac{p}{p-1}-2pM)L} & \text{if } p \nmid N' \end{cases}.$$

Let β_2 and β_2' be the estimate from Eq. (5.30) for N and N', respectively. Note that if $p \,|\, N'$, then both $\phi(N') < \phi(N)$ and $\mathbb{E}[K'] < \mathbb{E}[K]$, so we must have $\beta_2' < \beta_2$. However, when $p \nmid N'$, smaller values of p make both $\mathbb{E}[K']$ and $\phi(N') = \phi(N)/(p-1)$ larger, in turn increasing β_2'/β_2. In particular, when $p = 2$ and $\phi(N) > 2M$, $\mathbb{E}[K'] > \mathbb{E}[K]$ and $\phi(N') = \phi(N)$, so $\beta_2' > \beta_2$.

We confirm this analysis numerically for $N = 38$ and $N' = 38/2 = 19$, showing that the size N subproblem does not always require the largest value of β_2. We generated 100 test signals of length $N = 38$ with entries distributed as binom$(38, 0.5)$. Each test signal $\mathbf{x}^{(38)}$ was also decimated by 2 in frequency space to produce a length $N = 19$ signal $\mathbf{x}^{(19)}$ with $L = 76$. Using the same exponential search procedure as in Table 1, we computed the 50th, 90th, and 100th percentile β_2 values given $M = 1$ coefficient. Note that only the top level inversion was performed here as well, so the values in Table 2 describe the subproblems $\mathbf{x}^{(19)}$ and $\mathbf{x}^{(38)}$. As predicted, all β_2 values for $N = 38$, $L = 38$ are smaller than the corresponding values for the $N = 19, L = 76$ case. The experimental values exhibit an even more extreme gap than the heuristic estimates. Since 3 DFT coefficients were available for the $N = 19$ subproblem while 22 were available for the $N = 38$ subproblem, this discrepancy may also be attributed to the LLL approximation factor.

N	M	L	Coeffs, #	Theoretical β_2	50th percentile β_2	90th percentile β_2	100th percentile β_2
19	1	76	3	4.38e11	1.38e10	1.23e11	4.97e11
38	1	38	22	1.58e10	1.88e07	1.15e08	1.11e09

Table 2: For $N = 19$ with $L = 76$ and for $N = 38$ with $L = 38$ (both with $M = 1$): the number of DFT coefficients available at the top level inversion, $N - \phi(N) + 2M$, the theoretical value of β_2 from Eq. (5.30), and the empirical 50th, 90th, and 100th percentile β_2 values computed from an exponential search with 30 iterations.

The previous experiments fixed the binomial parameter $L = N$, so the one-dimensional test signals simulated subsignals of an $N \times N$ binary matrix. Here, we instead fix the signal length at $N = 31$ and investigate how varying L affects the required value of β_2. For each L, we generated 100 test signals with entries distributed as binom$(L, 0.5)$. Each test collected β_2 data using the same process as in Tables 1 and 2. For $M = 1$, 2, and 3, Figure 10 plots the theoretical estimate of β_2 and the empirical percentile values for varying L. The theoretical estimate captures the overall dependence on L, serving as a consistent lower bound at small M and becoming an upper bound on the 100th percentile value as M increases.

Each of the previous numerical experiments essentially used "infinite" precision. Since the lattice analysis did not account for limited numerical precision, we removed this parameter for verification tests. For practical applications, however, we are also interested in how the inversion algorithms perform with realistic precision in the DFT coefficient data. The following numerical tests therefore use standard single- or double-precision floating point arithmetic for all computations. Under the IEEE 754 standard for binary floating point, single-precision floating point numbers have 24 bits of precision, which corresponds to just over 7 decimal digits. Likewise, double precision floating point numbers have 53 bits of precision, which is equivalent to almost 16 decimal digits.

With fixed precision, large values of β_2 may no longer be advantageous to inversion. When the order of magnitude of β_2 exceeds the number of digits of precision, β_2 amplifies numerical errors in the DFT coefficient data. This effect is illustrated in Figure 11, which shows the recovery rate for signals of length $N = 30$ in single precision across a range of β_2 values for $M = 1$ and $L = N$. For $\beta_2 <$ 1e7, the plot demonstrates the behavior of Figure 8, where the fraction of correctly recovered test signals gradually increases with β_2. However, the recovery rate begins to decrease for $\beta_2 >$ 1e7, demonstrating how single precision data prohibits excessively large β_2 values. Based on these results and additional tests, we set $\beta_2 = 1e7$ for single precision

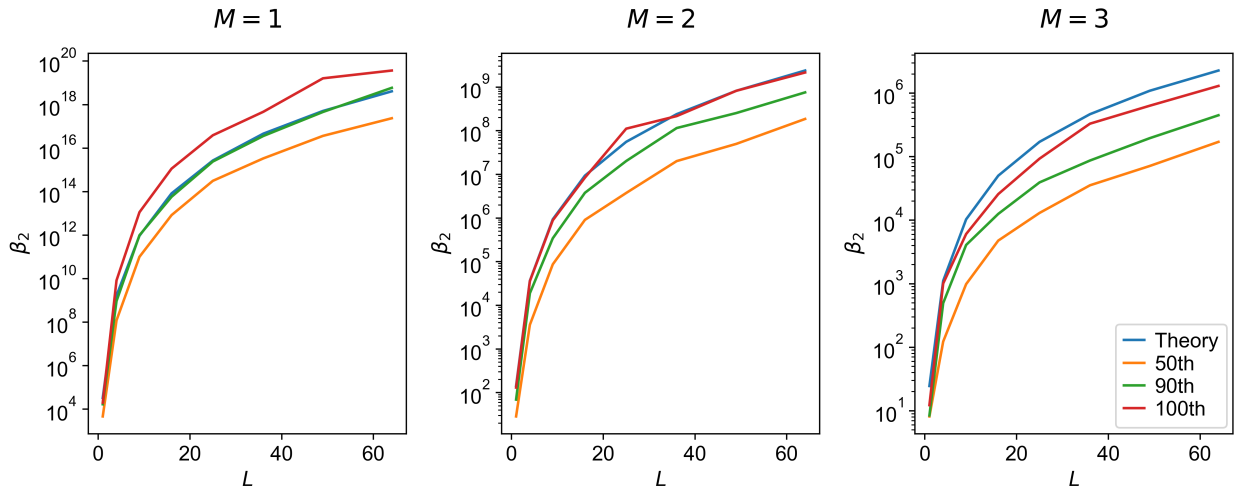

Figure 10: For $M = 1, 2$, and 3, plots of β_2 values dependence on L. For several values of L, the associated test set consisted of 100 random signals of length $N = 31$ with entries distributed as binom$(L, 0.5)$. The theoretical estimate of β_2 (blue line), 50th percentile (orange line), 90th percentile (green line), and 100th percentile (red line) are all plotted.

inversion and $\beta_2 = 1e14$ for double precision inversion. We generally expect the success of inversion to depend on whether these chosen values are sufficiently large for the given setup.

Table 3 reports the percentage of successfully recovered test signals for varying N and M. For each signal length N, we generated 100 test signals with entries distributed as binom$(N, 0.5)$. We attempted recovery with increasing values of M (up to $M = 10$), until every test signal was correctly inverted, again performing only the top-level inversion. This experiment was performed in both single and double precision. Note that the top-level inversion has access to every DFT coefficient when $2M \geq \phi(N)$, making recovery trivial. In Table 3, the recovery percentage is replaced by "–" for such M.

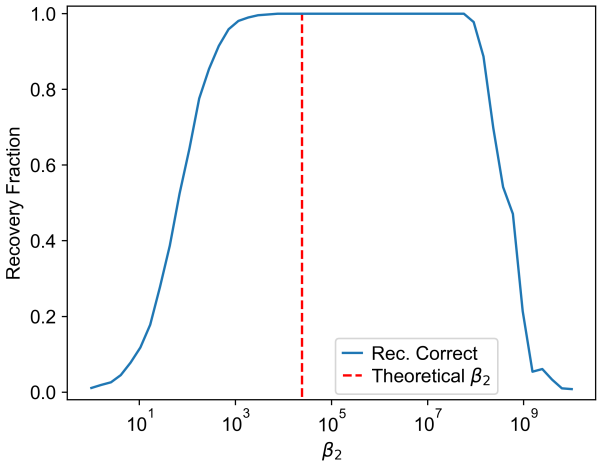

Figure 11: For $N = 30$ and $M = 1$ with single precision: Fraction of test signals correctly recovered over a range of β_2 values (blue solid line) compared to heuristic β_2 value (red dashed line). The test set used 1000 random signals of length N with entries distributed as binom$(N, 0.5)$.

The results in Table 3 show that, with single precision data, large integer signals require more than the minimal set of DFT coefficients for reliably recovery. For tested values of $N > 30$, recovery usually failed at $M = 1$. The only exceptions were $N = 32$, 48, and 60, which all have relatively small $\phi(N)$. These findings are consistent with the β_2 values in Table 1. For example, for $N = 50$ with $M = 1$, the theoretical value of β_2 is 8.70e11, which is well beyond the 1e7 threshold of single precision. However, when $M = 2$, this value drops to 1.00e6, and all test signals were successfully recovered. In general, double precision greatly improves recovery, as nearly every tested N succeeded with $M = 1$. The only exceptions were $N = 31$, 47, and 49, each with relatively large $\phi(N)$. Even in these cases with fewer supplied coefficients, reliable recovery required at most $M = 3$. For $N = 47$, this corresponds to successful inversion from only 4 DFT coefficient measurements, or roughly 8.5% of the spectrum.

Precision	N \ M	1	2	3	4	5	6	7	8	9	10
Single	19	14	100	100	100	100	100	100	100	–	–
	24	100	100	100	–	–	–	–	–	–	–
	30	100	100	100	–	–	–	–	–	–	–
	31	0	0	2	100	100	100	100	100	100	100
	32	63	100	100	100	100	100	100	–	–	–
	39	0	0	100	100	100	100	100	100	100	100
	45	0	99	100	100	100	100	100	100	100	100
	47	0	0	0	0	0	78	100	100	100	100
	48	51	100	100	100	100	100	100	–	–	–
	49	0	0	0	0	0	97	100	100	100	100
	50	5	100	100	100	100	100	100	100	100	–
	60	93	100	100	100	100	100	100	–	–	–
	90	0	99	100	100	100	100	100	100	100	100
Double	19	100	100	100	100	100	100	100	100	–	–
	24	100	100	100	–	–	–	–	–	–	–
	30	100	100	100	–	–	–	–	–	–	–
	31	11	100	100	100	100	100	100	100	100	100
	32	100	100	100	100	100	100	100	–	–	–
	39	100	100	100	100	100	100	100	100	100	100
	45	100	100	100	100	100	100	100	100	100	100
	47	0	55	100	100	100	100	100	100	100	100
	48	100	100	100	100	100	100	100	–	–	–
	49	0	100	100	100	100	100	100	100	100	100
	50	100	100	100	100	100	100	100	100	100	–
	60	100	100	100	100	100	100	100	–	–	–
	90	100	100	100	100	100	100	100	100	100	100

Table 3: For single and double precision: For various N and $M = 1$–10, the percentage of test signals recovered. For each N, the test set used 100 random signals of length N with entries distributed as $\text{binom}(N, 0.5)$. Trivial recoveries are indicated by "–".

6.3. Image Reconstructions

While the previous section focused on numerical tests supporting the lattice analysis, here we present results for the full two-dimensional inversion method of Algorithm 4. Table 4 displays results for binary matrix inversion using double precision. As in the previous double precision test, we set $\beta_2 = 1\text{e}14$ for all signal sizes. Unlike the partial top-level experiments of Table 3, these tests solve the full inversion problem. For each matrix size $N_1 \times N_2$, we generated 100 random $N_1 \times N_2$ binary matrices. Numerical tests started with $M = 1$ DFT coefficient for each equivalence class. However, if recovery of any test matrices failed, M was increased for the longest subsignals of length $N = \text{lcm}(N_1, N_2)$. Even for $M > 1$, we used only 1 DFT coefficient for lower-order frequencies. The 90×90 case when $M = 2$ was the sole exception, where the larger simulation necessitated multiple coefficients for some lower-order frequencies. For simplicity, we supplied two DFT coefficients from every sufficiently large equivalence class in this case.

For each matrix size and value of M, Table 4 lists the maximum subsignal length $N = \text{lcm}(N_1, N_2)$, the number of DFT coefficients provided to the algorithm, the largest theoretical estimate of β_2 from Eq. (5.30) across all subproblems (computed using each subproblem's M value), the recovery percentage, and the average runtimes of both the successful recoveries and failures. These runtimes were measured on a standard workstation. Note that when a matrix recovery failed with a given M, the tolerance condition of Algorithm 5 in Line 8 was often violated. Since an early subproblem of Algorithm 4 may violate the tolerance condition,

the algorithm often terminated quickly when M was too small. Additionally, note that both the length of the basis vectors in Eq. (4.5) and the value of B in Eq. (6.5) scale with M. Applying these observations to the LLL runtime bound explains why the average time of successful recoveries increases with M.

$N_1 \times N_2$	N	M	Coeffs, #	Theory β_2	Rec, %	succ. t, sec	fail. t, sec
9×11	99	1	6	1.89e09	90	0.25	0.00
		2	10	2.91e05	100	0.59	–
11×13	143	1	4	5.64e18	0	–	0.13
		2	7	2.18e09	0	–	0.36
		3	10	1.78e06	0	–	0.66
		4	13	1.22e05	25	1.95	0.00
		5	16	1.22e05	100	3.01	–
12×18	36	1	48	1.64e04	100	0.07	–
23×23	23	1	25	1.42e11	100	0.06	–
30×40	120	1	112	1.40e13	100	7.00	–
31×31	31	1	33	1.44e16	0	–	0.02
		2	65	1.32e08	100	0.27	–
36×45	180	1	102	1.31e19	6	22.25	2.03
		2	193	2.39e13	100	46.14	–
40×40	40	1	154	1.53e09	100	0.98	–
41×41	41	1	43	5.28e22	0	–	0.09
		2	85	2.71e11	100	0.95	–
43×43	43	1	45	1.17e24	0	–	0.13
		2	89	1.29e12	96	1.15	0.30
		3	133	1.53e08	100	1.66	–
45×45	45	1	119	9.72e13	100	1.57	–
49×49	49	1	65	4.79e24	0	–	0.38
		2	129	2.65e12	100	2.26	–
60×60	60	1	350	8.70e09	100	6.40	–
90×90	90	1	476	5.95e17	0	–	0.61
		2*	932	9.83e08	100	69.51	–

Table 4: For various matrix dimensions $N_1 \times N_2$ and values of M up to what was required for recovery at double precision: the number of DFT coefficients supplied, the largest value of the theoretical β_2 over all subproblems, the percentage of test signals recovered, the average time of successful recoveries, and the average time of unsuccessful recoveries. For each matrix size, the test set consisted of 100 random binary matrices.

As in the previous section, the theoretical β_2 value roughly indicates whether inversion will succeed at double precision. Empirically, the largest usable value of β_2 is about 1e14, so we expect successful recovery only if this threshold exceeds the theoretical estimate in Eq. (5.30). Take the 36×45 case as an example. For $M = 1$, the estimated β_2 is 1.31e19, far greater than 1e14, and inversion failed for most test signals. Increasing to $M = 2$ reduces the theoretical β_2 below the threshold, and recovers all test matrices. A similar pattern appears for 90×90 matrices. Inversion fails for $M = 1$ with the theoretical β_2 at 5.95e17. For $M = 2$, the estimated value reduces to 5.83e8, justifying the successful inversions. Note that the estimated β_2 for inverting the length $N = 45$ subsignals with $L = 180$ is 5.95e17 when $M = 1$, which justifies the use

of $M = 2$ at this level as well.

While the β_2 analysis correctly differentiates between successful and failed inversion in some cases, it is by no means an exact threshold. In the 43×43 case, $M = 2$ yields a theoretical β_2 value of 1.29e12, yet recovery failed for four test matrices. This is plausible given that the estimated value is close to the precision limit. More strikingly, the 11×13 case gives an example where the β_2 estimates fail. For $M = 2$, 3, and 4, the predicted β_2 falls well below 1e14, yet inversion failed until M was increased to $M = 5$. As in the previous section, we attribute this discrepancy to the LLL approximation factor, which becomes more significant in higher-dimensional lattices. Here, the $N + 1 = 144$-dimensional lattice makes solving an exact SVP difficult.

The results in Table 4 also demonstrate the strong dependence on the relative dimensions of the matrices. A comparison between the 11×13 and the 60×60 cases exemplifies this dependence. While the 60×60 matrix has far more unknowns, Algorithm 4 performs better from a minimal set of DFT coefficients, since unique inversion requires more data. In contrast, the relatively small 11×13 matrix requires $M = 5$ for reliable inversion. Despite this difference in M, the two cases are comparable. The 60×60 matrix required $350/3600 \approx 9.7\%$ of DFT coefficients, whereas the 11×13 matrix required $8/143 \approx 5.6\%$ of DFT coefficients for reliable inversion. Thus, even though the 11×13 matrices could not be inverted from a minimal set of DFT coefficients at double precision (only 4 coefficients), they were still inverted from a relatively smaller sampled spectrum than the 60×60 matrices.

A key distinction lies in the role of divisors. Since 11 and 13 are distinct primes, the 11×13 matrix has a single length D subproblem for each $D = 11, 13, 143$. On the other hand, the shared divisors of 60×60 create many intermediate subproblems, each involving a subsignal of maximum length 60. Note that the length 143 subsignal of the 11×13 matrix is binary, while the length 60 subproblems of the 60×60 matrix have integer bound $L = 60$. Regardless, the exponential dependence of the LLL approximation factor on the lattice size causes the reduction in subsignal length to provide a much greater advantage to the latter case.

We conclude this section with a few structured, nonrandom image reconstructions. As an initial binary example, Figure 12 shows a 45×45 QR code, which is the Version 7 standard [65]. In line with the results of Table 4, we set $M = 1$, which corresponds to sampling 119 out of 2025 total DFT coefficients (5.9%). The left image in Figure 12 shows the zero-filled reconstruction in real space, where all unsampled DFT coefficients are set to 0. Using double precision in the data, Algorithm 4 quickly recovered the QR code exactly. Note that this reconstruction did not take advantage of the fixed structure and built-in error correction present in the QR code.

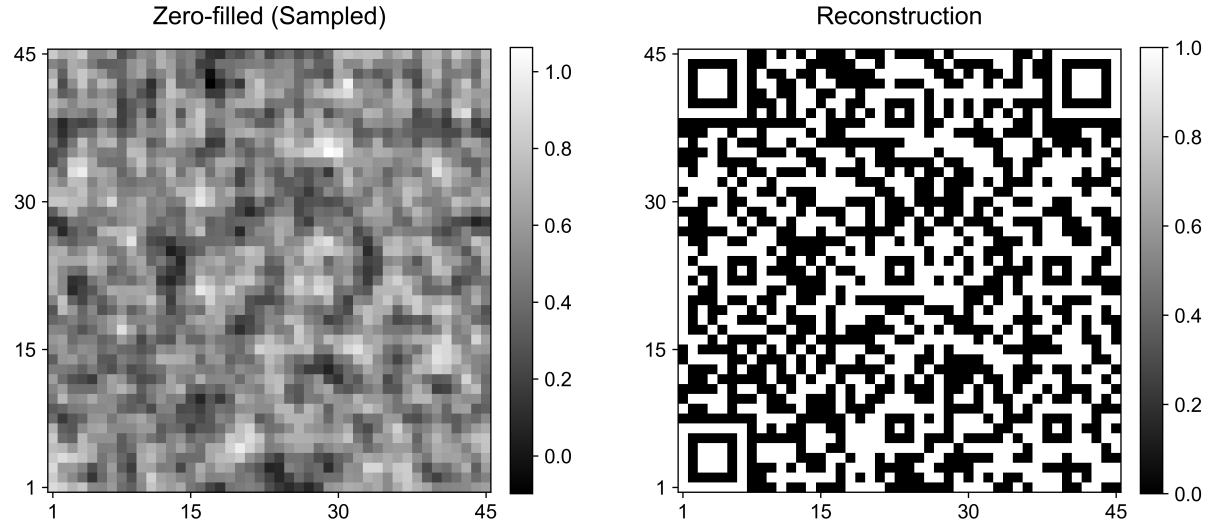

Figure 12: Using a 45×45 QR code containing the message "Hello World!" as a model image, **Left**: the zero-filled reconstruction from a minimal set of sampled DFT coefficients; **Right**: the reconstruction by Algorithm 4 from the minimal spectrum (coincides exactly with model).

Next, we consider a non-binary image with a larger integer range. We begin with the classic 256×256 'camera man' image, where the integer values range from 0 to 255 [dataset][66]. This resolution was too large for our algorithm to handle directly, so for practicality (and taking into consideration the number of divisors), we rescaled the image to 60×60. To ensure feasibility at double precision, we also rescaled the intensity values to be between 0 and $L = 13$. The top right image in Figure 13 shows this rescaled model image. The top left image shows the corresponding zero-filled reconstruction, obtained with $M = 1$ and 350 sampled DFT coefficients. With these parameters, Algorithm 4 recovered the model image exactly in 6.5 seconds.

The previous reconstruction already pushes the limit of double precision. Moving to higher precision lets us expand the integer range L and work with larger images. Our final two examples consider both directions. In the middle row of Figure 13, we consider the original range of integer values by setting $L = 255$. This reconstruction failed in double precision, but succeeds exactly in 13.3 seconds with infinite precision. Finally, we increase the image size to 100×100. With $M = 1$, this corresponds to 370 sampled DFT coefficients. To make this reconstruction feasible (even without precision limits), we rescaled the model image to be binary with $L = 1$. The reconstruction was again exact, with no reconstruction error and took 135.2 seconds.

7. Discussion

The results of this work demonstrate the strength of integer-valued priors, as they enable perfect recovery from a limited set of DFT coefficients. Although reconstruction chances generally improve when more coefficients are available, our algorithms typically succeeded with fewer than 15% of the DFT coefficients, often with less than 10%.

The reconstruction methods demonstrate reasonable speed and accuracy, but often demand high precision DFT data. While elevated precision permits recovery of larger images, moving to single precision reduces algorithm reliability, especially when given the minimal set of data. Furthermore, the presence of noise in the data can make the methods unreliable. For example, reconstruction of random 30×30 binary images was reliable with 5-6 decimal digits of precision, but frequently failed with fewer than 5 digits. This behavior coincides with the stability prediction from the theoretical β_2 estimate. For 30×30 with $M = 1$, the maximum β_2 value is 1.09e5, which suggests that roughly five digits of precision are required. For $M = 2$ however, the β_2 value becomes 2.86e2, and reliable recovery required 3-4 digits of precision. While our lattice analysis provides some insight into stability, the numerical experiments in this paper focused primarily on reconstruction from a minimal set of coefficients. The remaining experimental and theoretical stability questions warrant further investigation to obtain tighter results.

Incorporating algorithm elements from related works could further improve the stability and speed of our methods. Combining integer linear programming with lattice methods shows promising results, especially when tight integer bounds are available [33]. With an appropriate sampling strategy, an FFT-like adaptation of the algorithms benefits from the inclusion of many redundant DFT coefficients [31]. Additionally, the partial order on the subproblems of Algorithms 2 and 4 prescribes efficient ways to parallelize the algorithm implementations. While our results demonstrate the effectiveness of integer-valued priors, a natural direction for future work is to combine these priors with other common ones, such as sparsity or connectivity. Finally, we note that the reduction techniques developed in this work for theoretical results in Theorems 3.1 and 3.2 and Algorithm 4 extend naturally to higher-dimensional integer signals.

8. Funding Sources

This work was supported by the National Science Foundation [grant number 2513653];

Appendix A. Intersecting Hyperspheres and Hyperplanes

This section considers the general intersection of an N-dimensional hyperball with a collection of hyperplanes that contain its center. The set D_0 in Eq. (5.15) and D_γ in Eq. (5.24) are examples of this. Without

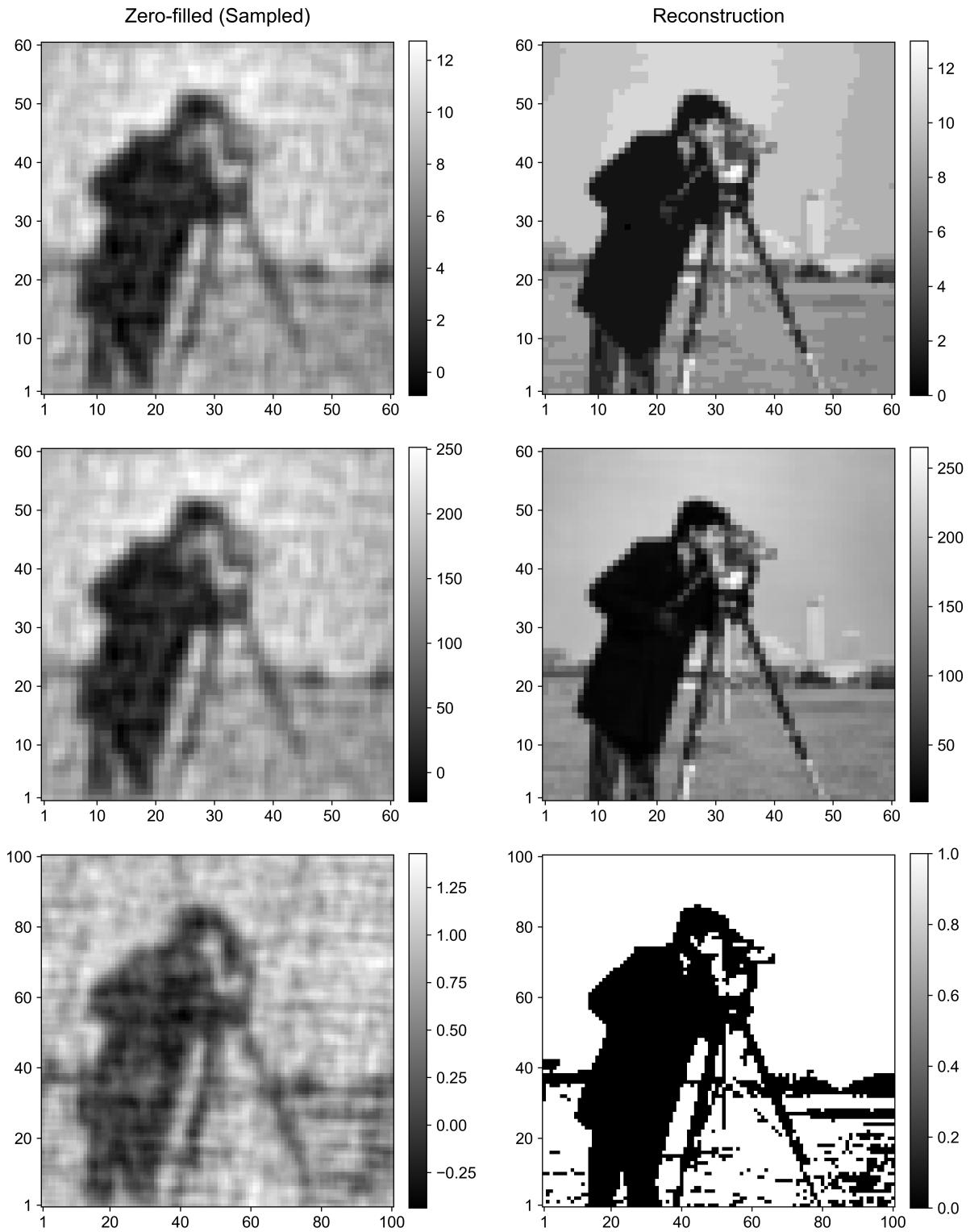

Figure 13: For three models, the left column has the zero-filled reconstruction from a minimal set of sampled DFT coefficients; the right column is the reconstruction by Algorithm 4 from the minimal spectrum (which coincides exactly with model). **Top**: A 60×60 model with $L = 13$ using double precision in the data. **Middle**: A 60×60 model with $L = 255$ using extra precision in the data. **Bottom**: A 100×100 model with $L = 1$ using extra precision in the data.

loss of generality, assume the hyperball is centered at $\mathbf{0}$, so the intersection is written as

$$B = \{\mathbf{x} \in \mathbb{R}^N : \|\mathbf{x}\|_2 \leq R\} \cap \ker A = \{\mathbf{x} \in \mathbb{R}^N : \|\mathbf{x}\|_2 \leq R : A\mathbf{x} = \mathbf{0}\} \ ,$$

where A is the matrix of hyperplane coefficients. We show that B is an N'-dimensional hyperball of radius R, where N' is the nullity of A.

Recall that an n-dimensional subspace of $\mathbb{R}^N$ is isometric to $\mathbb{R}^n$. In particular, since $N' = \dim(\ker A)$, let $T : \mathbb{R}^{N'} \to \ker A$ be an isometry. If $B^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 \leq R\}$ is the n-ball of radius R, we show that T maps $B^{N'}$ to B,

$$TB^{N'} = B \ . \tag{A.1}$$

Suppose $\mathbf{x} \in TB$. Since T maps $\mathbb{R}^{N'}$ to $\ker A$, $\mathbf{x} \in \ker A$. Since T is an isometry, $\|T\mathbf{x}\|_2 = \|\mathbf{x}\|_2 \leq R$, so $\mathbf{x} \in B$. The other containment in Eq. (A.1) follows from a similar argument with the isometry T^{-1}. The existence of the isometry between $B^{N'}$ and B implies that B is an N'-dimensional hyperball of radius R embedded in $\mathbb{R}^N$. It follows that,

$$\mathrm{vol}(B) = V_{N'} R^{N'} \ .$$

Appendix A.1. Lower Bound on Average 1-Norm in the Intersection

The lattice parameter analysis requires the average 1-norm in the intersection B. Section 5.2 computes the average 1-norm in a hyperball $B^{N'}$, which generally differs from the average 1-norm in B, since T preserves 2-norms instead of 1-norms. The current section uses the former result for $B^{N'}$ and the framework above to compute the average 1-norm in B. We then provide a concise lower bound for use in the final estimate of β_2.

Let $\mathbf{e}_n$ be the nth standard basis vector in $\mathbb{R}^{N'}$. Note that each coordinate of $\mathbf{x}$ satisfies, $x_n = \langle \mathbf{x}, \mathbf{e}_n \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the scalar product. For any unit vector $\mathbf{u} \in B^{N'}$, there exists a rotation about the origin taking $\mathbf{u}$ to $\mathbf{e}_1$. The rotational symmetry of $B^{N'}$ implies that this rotation maps $B^{N'}$ isometrically onto itself. Therefore, $\langle \mathbf{x}, \mathbf{u} \rangle$ takes the same distribution as x_1 when $\mathbf{x}$ is drawn uniformly at random from $B^{N'}$. In particular, Eq. (5.18) implies,

$$\mathbb{E}[|\langle \mathbf{x}, \mathbf{u} \rangle| : \mathbf{x} \in B^{N'}] = \frac{2RV_{N'-1}}{(N'+1)V_{N'}} \ . \tag{A.2}$$

For any unit vector $\mathbf{u} \in B$,

$$\mathbb{E}[|\langle \mathbf{y}, \mathbf{u} \rangle| : \mathbf{y} \in B] = \mathbb{E}[|\langle T^{-1}\mathbf{y}, T^{-1}\mathbf{u} \rangle| : \mathbf{y} \in B] = \mathbb{E}[|\langle \mathbf{x}, T^{-1}\mathbf{u} \rangle| : \mathbf{x} \in B^{N'}] = \frac{2RV_{N'-1}}{(N'+1)V_{N'}} \ , \tag{A.3}$$

where the first equality comes from the isometry T^{-1} preserving inner products, and the last equality comes from Eq. (A.2) and $\|T^{-1}\mathbf{u}\|_2 = \|\mathbf{u}\|_2 = 1$.

Now, let P be the orthogonal projection of $\mathbb{R}^N$ onto $\ker A$. Note that the 1-norm of any vector $\mathbf{x} \in \mathbb{R}^N$ is given by

$$\|\mathbf{x}\|_1 = \sum_{n=0}^{N-1} |x_n| = \sum_{n=0}^{N-1} |\langle \mathbf{e}_n, \mathbf{x} \rangle| \ , \tag{A.4}$$

where $\mathbf{e}_n$ now denotes the nth standard basis vector in $\mathbb{R}^N$. If $\mathbf{x} \in \ker A$, $P\mathbf{x} = \mathbf{x}$, so

$$|\langle \mathbf{e}_n, \mathbf{x} \rangle| = |\langle \mathbf{e}_n, P\mathbf{x} \rangle| = |\langle P\mathbf{e}_n, \mathbf{x} \rangle| = \|P\mathbf{e}_n\|_2 \left| \langle \frac{P\mathbf{e}_n}{\|P\mathbf{e}_n\|_2}, \mathbf{x} \rangle \right| \ ,$$

where the second equality uses the fact that orthogonal projections are self-adjoint. Note that $\frac{P\mathbf{e}_n}{\|P\mathbf{e}_n\|_2}$ is a unit vector in $\ker A$, so Eq. (A.3) implies,

$$\mathbb{E}\left[\left| \langle \frac{P\mathbf{e}_n}{\|P\mathbf{e}_n\|_2}, \mathbf{x} \rangle \right| : \mathbf{x} \in B \right] = \frac{2V_{N'-1}}{(N'+1)V_{N'}} \ .$$

Combining this expression with Eq. (A.4), we see that the expected 1-norm of points in the N'-dimensional hyperball embedded in $\mathbb{R}^N$ is,

$$\mathbb{E}[\|\mathbf{x}\|_1 : \mathbf{x} \in B] = \frac{2RV_{N'-1}}{(N'+1)V_{N'}} \sum_{n=0}^{N-1} \|P\mathbf{e}_n\|_2 \ . \tag{A.5}$$

Lastly, we appeal to an explicit formula for P to lower bound the sum in Eq. (A.5). Let $\mathbf{u}_0, \ldots, \mathbf{u}_{N'-1}$ be an orthonormal basis for $\ker A$. Then,

$$P\mathbf{e}_n = \sum_{i=0}^{N'-1} \langle \mathbf{u}_i, \mathbf{e}_n \rangle \mathbf{u}_i = \sum_{i=0}^{N'-1} (\mathbf{u}_i)_n \mathbf{u}_i \ ,$$

and since the $\mathbf{u}_i$'s are pairwise orthogonal, the Pythagorean theorem implies

$$\|P\mathbf{e}_n\|_2^2 = \sum_{i=0}^{N'-1} (\mathbf{u}_i)_n^2 \ .$$

Therefore,

$$\sum_{n=0}^{N-1} \|P\mathbf{e}_n\|_2^2 = \sum_{n=0}^{N-1} \sum_{i=0}^{N'-1} (\mathbf{u}_i)_n^2 = \sum_{i=0}^{N'-1} \sum_{n=0}^{N-1} (\mathbf{u}_i)_n^2 = \sum_{i=0}^{N'-1} \|\mathbf{u}_i\|_2^2 = N' \ ,$$

where the last equality follows from the $\mathbf{u}_i$'s being normalized. As P is orthogonal projection, it is also a contraction, so the images of the standard basis vectors satisfy,

$$0 \leq \|P\mathbf{e}_n\|_2 \leq \|\mathbf{e}_n\|_2 = 1 \ .$$

Therefore, the convexity of $(\cdot)^2$ implies

$$\|P\mathbf{e}_n\|_2^2 \leq \|P\mathbf{e}_n\|_2 \ ,$$

so we can bound the sum in Eq. (A.5) by,

$$\sum_{n=0}^{N-1} \|P\mathbf{e}_n\|_2 \geq \sum_{n=0}^{N-1} \|P\mathbf{e}_n\|_2^2 = N' \ .$$

Finally, substituting this bound into Eq. (A.5) yields the desired bound,

$$\mathbb{E}[\|\mathbf{x}\|_1 : \mathbf{x} \in B] \geq \frac{2N'RV_{N'-1}}{(N'+1)V_{N'}} \ .$$

References

[1] E. J. Candes, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences 59 (2006) 1207–1223.

[2] D. L. Donoho, Compressed sensing, IEEE Transactions on information theory 52 (2006) 1289–1306.

[3] M. F. Duarte, Y. C. Eldar, Structured compressed sensing: From theory to applications, IEEE Transactions on signal processing 59 (2011) 4053–4085.

[4] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, M. Strauss, Near-optimal sparse fourier representations via sampling, in: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, 2002, pp. 152–161.

[5] A. C. Gilbert, S. Muthukrishnan, M. Strauss, Improved time bounds for near-optimal sparse fourier representations, in: Wavelets xi, volume 5914, SPIE, 2005, pp. 398–412.

[6] M. Rudelson, R. Vershynin, On sparse reconstruction from fourier and gaussian measurements, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences 61 (2008) 1025–1045.

[7] M. Rudelson, 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.

[8] E. Candes, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Transactions on Information Theory 52 (2006) 489–509. doi:10.1109/TIT.2005.862083.

[9] G. Plonka, K. Wannenwetsch, A deterministic sparse fft algorithm for vectors with small support, Numerical Algorithms 71 (2016) 889–905.

[10] E. Rajaby, S. M. Sayedi, A structured review of sparse fast fourier transform algorithms, Digital Signal Processing 123 (2022) 103403.

[11] S. Bittens, R. Zhang, M. A. Iwen, A deterministic sparse fft for functions with structured fourier sparsity, Advances in Computational Mathematics 45 (2019) 519–561.

[12] S.-H. Hsieh, C.-S. Lu, S.-C. Pei, Sparse fast fourier transform by downsampling, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 5637–5641.

[13] A. C. Gilbert, P. Indyk, M. Iwen, L. Schmidt, Recent developments in the sparse fourier transform: A compressed fourier transform for big data, IEEE Signal Processing Magazine 31 (2014) 91–100.

[14] G. T. Herman, A. Kuba, Discrete tomography: Foundations, algorithms, and applications, Springer Science & Business Media, 2012.

[15] P. T. Boufounos, R. G. Baraniuk, 1-bit compressive sensing, in: 2008 42nd Annual Conference on Information Sciences and Systems, IEEE, 2008, pp. 16–21.

[16] R. G. Parker, R. L. Rardin, Discrete optimization, Elsevier, 2014.

[17] A. Alpers, P. Gritzmann, L. Thorens, Stability and instability in discrete tomography, in: Digital and Image Geometry: Advanced Lectures, Springer, 2002, pp. 175–186.

[18] R. M. Karp, On the computational complexity of combinatorial problems, Networks 5 (1975) 45–68.

[19] P. Q. Nguyen, B. Vallée, The LLL algorithm, Springer, 2010.

[20] G. Joseph, V. Gandikota, A. Bhandari, J. Choi, I.-s. Kim, G. Lee, M. Matthaiou, C. R. Murthy, H. Q. Ngo, P. K. Varshney, et al., Low-resolution compressed sensing and beyond for communications and sensing: Trends and opportunities, Signal Processing 235 (2025) 110020.

[21] J. Zhan, B. Nazer, U. Erez, M. Gastpar, Integer-forcing linear receivers, IEEE Transactions on Information Theory 60 (2014) 7661–7685.

[22] G. T. Herman, A. Kuba, Discrete tomography in medical imaging, Proceedings of the IEEE 91 (2003) 1612–1626.

[23] A. Kadu, T. van Leeuwen, A convex formulation for binary tomography, IEEE Transactions on Computational Imaging 6 (2019) 1–11.

[24] K. J. Batenburg, J. Sijbers, Dart: a practical reconstruction algorithm for discrete tomography, IEEE Transactions on Image Processing 20 (2011) 2542–2553.

[25] K. J. Batenburg, S. Bals, J. Sijbers, C. Kübel, P. A. Midgley, J. Hernandez, U. Kaiser, E. R. Encina, E. A. Coronado, G. Van Tendeloo, 3d imaging of nanomaterials by discrete tomography, Ultramicroscopy 109 (2009) 730–740.

[26] A. E. Yagle, A convergent composite mapping fourier domain iterative algorithm for 3-d discrete tomography, Linear Algebra and its Applications 339 (2001) 91–109. URL: https://www.sciencedirect.com/science/article/pii/S002437950100458X. doi:https://doi.org/10.1016/S0024-3795(01)00458-X.

[27] S. Esedoglu, Blind deconvolution of bar code signals, Inverse Problems 20 (2003) 121.

[28] S. Di Zenzo, L. Cinque, S. Levialdi, Run-based algorithms for binary image analysis and processing, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 83–89. doi:10.1109/34.476016.

[29] M. Ren, J. Yang, H. Sun, Tracing boundary contours in a binary image, Image and Vision Computing 20 (2002) 125–131. URL: https://www.sciencedirect.com/science/article/pii/S0262885601000919. doi:https://doi.org/10.1016/S0262-8856(01)00091-9.

[30] S. Marchand-Maillet, Y. M. Sharaiha, Binary Digital Image Processing, Elsevier, 2000. URL: http://dx.doi.org/10.1016/B978-0-12-470505-0.X5000-X. doi:10.1016/b978-0-12-470505-0.x5000-x.

[31] S.-C. Pei, K.-W. Chang, Binary signal perfect recovery from partial dft coefficients, IEEE Transactions on Signal Processing 70 (2022) 3848–3861. doi:10.1109/TSP.2022.3190615.

[32] H. W. Levinson, V. A. Markel, Binary discrete fourier transform and its inversion, IEEE Transactions on Signal Processing 69 (2021) 3484–3499. doi:10.1109/TSP.2021.3088215.

[33] H. W. Levinson, V. Markel, N. Triantafillou, Inversion of band-limited discrete fourier transforms of binary images: Uniqueness and algorithms, SIAM Journal on Imaging Sciences 16 (2023) 1338–1369.

[34] K. M. Nashold, B. E. Saleh, Synthesis of two-dimensional binary images through band-limited systems: a slicing method, IEEE Transactions on Acoustics, Speech, and Signal Processing 37 (2002) 1271–1279.

[35] K. M. Nashold, J. A. Bucklew, W. Rudin, B. E. Saleh, Synthesis of binary images from band-limited functions, Journal of the Optical Society of America A 6 (1989) 852–858.

[36] Y. Mao, Reconstruction of binary functions and shapes from incomplete frequency information, IEEE Transactions on Information Theory 58 (2012) 3642–3653.

[37] M. Vetterli, P. Marziliano, T. Blu, Sampling signals with finite rate of innovation, IEEE transactions on Signal Processing 50 (2002) 1417–1428.

[38] A. Stolk, K. J. Batenburg, An algebraic framework for discrete tomography: Revealing the structure of dependencies, SIAM Journal on Discrete Mathematics 24 (2010) 1056–1079.

[39] J. A. Tropp, On the linear independence of spikes and sines, Journal of Fourier Analysis and Applications 14 (2008) 838–858.

[40] T. Tao, An uncertainty principle for cyclic groups of prime order, Math. Res. Lett. 12 (2005) 121–127.

[41] A. K. Lenstra, H. W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (1982) 515–534. URL: http://dx.doi.org/10.1007/BF01457454. doi:10.1007/bf01457454.

[42] D. Coppersmith, Finding a small root of a univariate modular equation, in: U. Maurer (Ed.), Advances in Cryptology — EUROCRYPT '96, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996, pp. 155–165.

[43] H. W. Lenstra, Integer programming with a fixed number of variables, Mathematics of Operations Research 8 (1983) 538–548. URL: http://www.jstor.org/stable/3689168.

[44] J. Hastad, B. Just, J. C. Lagarias, C. P. Schnorr, Polynomial time algorithms for finding integer relations among real numbers, SIAM Journal on Computing 18 (1989) 859–881. URL: https://doi.org/10.1137/0218059. doi:10.1137/0218059. arXiv:https://doi.org/10.1137/0218059.

[45] D. H. Bailey, J. M. Borwein, PSLQ: An Algorithm to Discover Integer Relations, Technical Report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2009. URL: https://www.osti.gov/biblio/963658. doi:10.2172/963658.

[46] K. Aardal, C. A. J. Hurkens, A. K. Lenstra, Solving a system of linear diophantine equations with lower and upper bounds on the variables, Mathematics of Operations Research 25 (2000) 427–442. URL: http://www.jstor.org/stable/3690477.

[47] S.-C. Pei, K.-W. Chang, Binary image fast perfect recovery from sparse 2d-dft coefficients, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2023, pp. 1–5.

[48] M. Hampejs, N. Holighaus, L. Tóth, C. Wiesmeyr, Representing and counting the subgroups of the group $z_m \times z_n$, 2014. URL: https://arxiv.org/abs/1211.1797. arXiv:1211.1797.

[49] G. Myerson, How small can a sum of roots of unity be?, The American Mathematical Monthly 93 (1986) 457–459.

[50] B. Barber, Small sums of five roots of unity, Bulletin of the London Mathematical Society 55 (2023) 1890–1906.

[51] J. Buhler, M. Shokrollahi, V. Stemann, Fast and precise fourier transforms, IEEE Transactions on Information Theory 46 (2000) 213–228. doi:10.1109/18.817519.

[52] J. W. Cooley, J. W. Tukey, An algorithm for the machine calculation of complex fourier series, Mathematics of Computation 19 (1965) 297–301. URL: http://www.jstor.org/stable/2003354.

[53] H. Marchand, A. Martin, R. Weismantel, L. Wolsey, Cutting planes in integer and mixed integer programming, Discrete Applied Mathematics 123 (2002) 397–446. URL: https://www.sciencedirect.com/science/article/pii/S0166218X01003481. doi:https://doi.org/10.1016/S0166-218X(01)00348-1.

[54] J. C. Lagarias, A. M. Odlyzko, Solving low-density subset sum problems, J. ACM 32 (1985) 229–246. URL: https://doi.org/10.1145/2455.2461. doi:10.1145/2455.2461.

[55] J. A. Greenwood, D. Durand, The distribution of length and components of the sum of n random unit vectors, The Annals of Mathematical Statistics (1955) 233–246.

[56] A. Kuketayev, Probability density function of the cartesian x-coordinate of the random point inside the hypersphere, arXiv preprint arXiv:1306.0290 (2013).

[57] C. Caiado, P. Rathie, Polynomial coefficients and distribution of the sum of discrete uniform variables, in: M. A. M., P. M. A., J. K. K., J. Joy (Eds.), Eighth Annual Conference of the Society of Special Functions and their Applications, 2007. URL: https://durham-repository.worktribe.com/output/1158118, ePrint Processing Status: DRO Team waiting for permission from publisher to deposit full text.

[58] A. Kalbach, T. Chinburg, Lll algorithm for lattice basis reduction, 2024. URL: https://arxiv.org/abs/2410.22196. arXiv:2410.22196.

[59] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with NumPy, Nature 585 (2020) 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2. doi:10.1038/s41586-020-2649-2.

[60] P. Q. Nguyen, D. Stehlé, An lll algorithm with quadratic complexity, SIAM Journal on Computing 39 (2009) 874–903.

[61] D. Stehlé, Floating-point lll: theoretical and practical aspects, in: The LLL Algorithm: survey and applications, Springer, 2009, pp. 179–213.

[62] T. F. development team, fplll, a lattice reduction library, Version: 5.5.0, 2023. URL: https://github.com/fplll/fplll, available at https://github.com/fplll/fplll.

[63] T. F. development team, fpylll, a Python wrapper for the fplll lattice reduction library, Version: 0.6.4, 2025. URL: https://github.com/fplll/fpylll, available at https://github.com/fplll/fpylll.

[64] I. Viviano, intvert, a Python package for inversion of integer arrays from partial DFT samples, 2025. URL: https://pypi.org/project/intvert/, available at https://pypi.org/project/intvert/.

[65] S. Tiwari, An introduction to qr code technology, in: 2016 International Conference on Information Technology (ICIT), 2016, pp. 39–44. doi:10.1109/ICIT.2016.021.

[66] W. F. Schreiber, Image processing for quality improvement, Proceedings of the IEEE 66 (2005) 1640–1651.