# LIST DECODING BARNES-WALL LATTICES

ELENA GRIGORESCU AND CHRIS PEIKERT

**Abstract.**
The question of *list decoding* error-correcting codes over finite fields (under the Hamming metric) has been widely studied in recent years. Motivated by the similar discrete linear structure of linear codes and *point lattices* in $\mathbb{R}^N$, and their many shared applications across complexity theory, cryptography, and coding theory, we initiate the study of list decoding for lattices. Namely: for a lattice $\mathcal{L} \subseteq \mathbb{R}^N$, given a target vector $r \in \mathbb{R}^N$ and a distance parameter $d$, output the set of all lattice points $w \in \mathcal{L}$ that are within distance $d$ of $r$.
In this work we focus on combinatorial and algorithmic questions related to list decoding for the well-studied family of *Barnes-Wall* lattices. Our main contributions are twofold:

1. We give tight combinatorial bounds on the worst-case list size, showing it to be polynomial in the lattice dimension for any error radius bounded away from the lattice's minimum distance (in the Euclidean norm).
2. We use our combinatorial bounds to generalize the unique-decoding algorithm of Micciancio and Nicolosi (*IEEE International Symposium on Information Theory 2008*) to work beyond the unique decoding radius, and still run in polynomial time up to the list-decoding radius. Just like the Micciancio-Nicolosi algorithm, our algorithm is highly parallelizable, and with sufficiently many processors can run in parallel time only *poly-logarithmic* in the lattice dimension.

**Keywords.** Barnes-Wall lattices, Johnson bound, List decoding, Reed-Muller codes

**Subject classification.** 68Q25

## 1. Introduction

A linear error-correcting *code* $\mathcal{C}$ of block length $N$ and dimension $K$ over a field $\mathbb{F}$ is a $K$-dimensional subspace of $\mathbb{F}^N$, generated as all

$\mathbb{F}$-linear combinations of $K$ linearly independent vectors. The code's *minimum distance*, denoted $d(\mathcal{C})$, is the minimum Hamming distance between any two distinct codewords in $\mathcal{C}$, or equivalently the minimum Hamming weight over all nonzero codewords. It is often convenient to normalize distances by the dimension, yielding the *relative* (minimum) distance $\delta(\mathcal{C}) = d(\mathcal{C})/N$ of the code. Similarly, a point *lattice* of dimension $N$ and rank $K$ (where often $K = N$) is a discrete additive subgroup of $\mathbb{R}^N$ (or $\mathbb{C}^N$), generated as all integer linear combinations of $K$ linearly independent vectors. The lattice's minimum distance $\lambda(\mathcal{L})$ is the minimum Euclidean norm over all nonzero lattice points $x \in \mathcal{L}$. Here it can also be convenient to normalize by the dimension, and for a closer analogy between the Hamming and Euclidean distances, in what follows we work with the *relative squared* distance (abbreviated rsd) $\delta(x, y) = \delta(x - y)$ on $\mathbb{R}^N$ or $\mathbb{C}^N$, where $\delta(z) = \frac{1}{N}\|z\|^2 = \frac{1}{N}\sum_{i=1}^{N}|z_i|^2$. The *relative squared minimum distance* (abbreviated rsmd) $\delta(\mathcal{L})$ of a lattice is therefore $\delta(\mathcal{L}) = \lambda(\mathcal{L})^2/N$.

Codes and lattices are well-studied studied objects, with many applications in computational complexity, cryptography, and coding theory. In particular, both kinds of objects can be used to encode data in order to achieve reliable communication in noisy channels: while error-correcting codes are used over discrete channels, in which symbols are possibly flipped during transmission, lattices are used over Gaussian-noise channels, in which the noise is usually modeled by a normal distribution.

A central question associated with codes is *unique decoding*: given a received word $r \in \mathbb{F}^N$ within relative Hamming distance less than $\delta(\mathcal{C})/2$ of some codeword $w \in \mathcal{C}$, find $w$. Similarly, the unique (also known as bounded-distance) decoding problem on lattices is: given a received word $r \in \mathbb{R}^N$ within rsd less than $\delta(\mathcal{L})/4$ of some lattice vector $v \in \mathcal{L}$, find $v$. (Note that the $1/4$ factor arises because distances are squared in our formulation.)

It is also possible that the noise amount affecting the transmission exceeds the regime of unique decoding. To model this situation, Elias (1957) and Wozencraft (1958) proposed extending the classical unique-decoding problem for error-correcting codes to settings where the amount of error could cause ambiguous decoding. More precisely, the goal of *list decoding* is to find all codewords within a certain relative

distance (typically exceeding $d(\mathcal{C})/2$) of a received word; in many cases, the list is guaranteed to contain few codewords. The first breakthrough algorithmic list decoding results were due to Goldreich & Levin (1989) for the Hadamard code, and to Sudan (1997) and Guruswami & Sudan (1999) for Reed-Solomon codes. These results and others have had countless applications, e.g., in building hard-core predicates for one-way functions Goldreich & Levin (1989), in hardness amplification Sudan *et al.* (2001), in learning Fourier coefficients Akavia *et al.* (2003); Gilbert *et al.* (2002); Kushilevitz & Mansour (1993), and in constructing randomness extractors Guruswami *et al.* (2009); Ta-Shma & Zuckerman (2001); Trevisan (2001).

There are two central tasks associated with list decoding: combinatorially bounding the number of codewords within a given radius of a received word, and algorithmically finding these codewords. An important question in understanding list decodability is finding the *list-decoding radius* of the code, i.e., the maximum distance from a received word within which the number of codewords is guaranteed to be polynomial in the input parameters.

**The Johnson bound.** Under the Hamming metric, the *Johnson bound* gives a distance up to which list decoding is guaranteed to be combinatorially efficient. One version of the Johnson bound states that for any code $\mathcal{C}$ of relative distance $\delta$, a Hamming ball of relative radius $J(\delta) - \epsilon$ contains at most $1/\epsilon^2$ codewords, and a ball of relative radius $J(\delta)$ contains at most $\delta N^2|\mathbb{F}|$ codewords, where $J(\delta) = 1 - \sqrt{1-\delta}$. The Johnson bound is generic since it does not use any structure of the code (not even linearity), and in many cases it is not necessarily the same as the list-decoding radius. It is, however, a barrier in the current analysis of combinatorial list decoding for many well-studied families like Reed-Solomon codes, algebraic geometry codes, Chinese remainder codes, and others. The breakthrough works of Parvaresh & Vardy (2005) and Guruswami & Rudra (2006) gave families of codes which could be (efficiently) list decoded beyond the Johnson bound, and were followed by several related combinatorial and algorithmic results for other codes (e.g., Dinur *et al.* (2008); Gopalan *et al.* (2011, 2008); Kaufman *et al.* (2010)). For more detailed surveys on list decoding of codes we refer to Guruswami (2004, 2006, 2010); Sudan (2000).

**1.1. Contributions.** Motivated by the common discrete linear structure of codes and lattices, we initiate the study of *efficient* list decoding for lattices, from both a combinatorial and algorithmic perspective. The problem of finding all the lattice points within a given distance from a target is also the problem of *lattice enumeration*, a technique commonly used in classical computational problems on lattices (e.g., Kannan (1987); Pujol & Stehlé (2008)), but with exponential running time in general lattices. Conway & Sloane (1998) promoted the applicability of lattices in practice as alternatives to codes. Therefore, our study of efficient list decoding is motivated by practical applications in error-tolerant communication, but primarily by the naturalness of the list-decoding problem from a mathematical and computational perspective, and we hope that our work will find other applications in theoretical computer science.

In this work we focus on the *Barnes-Wall* (BW) Barnes & Wall (1959) family of lattices in $\mathbb{C}^N$, which have been well-studied in coding theory (see, e.g., Amir H. Banihashemi (1998); Forney (1988); Forney & Vardy (1996); Nebe *et al.* (2001); Salomon & Amrani (2005)) and share many connections to the Reed-Muller Muller (1954); Reed (1954) family of error-correcting codes (we elaborate below). Barnes-Wall lattices were first constructed in order to demonstrate dense sphere packings, a feature that makes them useful in communications settings. Specifically, Barnes-Wall lattices are particularly useful instantiations of 'Construction-D' lattices, which themselves provide a general framework for constructing lattices approaching the capacity of band-limited channels. (For further details, see e.g., Conway & Sloane (1998); Forney (1988).)

Minimum-distance decoding algorithms for BW lattices were given in Forney (1988); Ran & Snyders (1998), but they are either for fixed low dimensions or have runtimes exponential in the lattice dimension $N$. Micciancio & Nicolosi (2008) gave the first $\mathrm{poly}(N)$-time algorithms for bounded-distance (unique) decoding of any BW lattice up to $\delta/4$ relative error. In fact, Micciancio & Nicolosi (2008) give two algorithms, a sequential one with $O(N \log N)$ running time, and a parallelizable one with $O(N^2)$ circuit size. They also posed list decoding of BW lattices as an open problem.

Our main contributions are twofold:

1. We give tight (up to polynomials) combinatorial bounds on the worst-case list size for BW lattices, showing it to be polynomial in the lattice dimension $N$ for any relative squared distance (rsd) bounded away from the rsmd $\delta$ of the lattice. (See Theorem 1.2 and Theorem 1.3 below for precise statements.) We note that it was already known that the list size is super-polynomial $N^{\Theta(\log N)}$ when the rsd equals $\delta$ (see, e.g., (Conway & Sloane 1998, Chapter 1, §2.2, page 24)).

2. We give a corresponding list-decoding algorithm that, for any rsd, runs in time polynomial in the lattice dimension and worst-case list size. Our algorithm generalizes the Micciancio-Nicolosi parallel algorithm, and with sufficiently many processors it runs in only poly-logarithmic $O(\log^2 N)$ parallel time. (See Section 3 for further details.)

We note that Johnson-type bounds for lattices are known and easy to obtain (in fact, the Johnson bound for codes under the Hamming metric is typically proved by reducing it to a packing bound in $\mathbb{R}^N$ under the Euclidean norm; see, e.g., Bollobás (1986); Guruswami & Sudan (2001); Micciancio & Goldwasser (2002); Sudan (2001)). For a lattice $\mathcal{L} \subset \mathbb{C}^N$ with rsmd $\delta$, the list size for rsd $\delta \cdot (\frac{1}{2} - \epsilon)$ is at most $\frac{1}{2\epsilon}$, and for rsd $\frac{\delta}{2}$ is at most $4N$ (see Lemma 2.3). Interestingly, the latter bound is tight for BW lattices (see Corollary 2.4). Since $\delta = 1$ for every BW lattice, our combinatorial and algorithmic results for rsd up to $1$ therefore apply far beyond the Johnson bound.

To describe our results in more detail, we need to define Barnes-Wall lattices. Let $\mathbb{G} = \mathbb{Z}[i]$ be the ring of Gaussian integers, and let $\phi = 1 + i \in \mathbb{G}$.

DEFINITION 1.1 (Barnes-Wall lattice). *The $n$th Barnes-Wall lattice $BW_n \subseteq \mathbb{G}^N$ of dimension $N = 2^n$ is defined recursively as $BW_0 = \mathbb{G}$, and for positive integer $n \geq 1$ as*

$$BW_n = \{[u, u + \phi v] : u, v \in BW_{n-1}\}.$$

One can check that $\mathrm{BW}_n$ is a lattice; indeed, it is easy to verify that it is generated as the $\mathbb{G}$-linear combinations of the rows of the $n$-fold

Kronecker product

$$W = \begin{bmatrix} 1 & 1 \\ 0 & \phi \end{bmatrix}^{\otimes n} \in \mathbb{C}^{N \times N}.$$

A simple induction proves that the minimum distance of $\mathrm{BW}_n$ is $\sqrt{N}$, i.e., its rsmd is $\delta = 1$.[1] Also observe that if $[u, w = u + \phi v] \in \mathrm{BW}_n$ for $u, w \in \mathbb{C}^{N/2}$, then $[w, u] \in \mathrm{BW}_n$: indeed, we have $w, -v \in \mathrm{BW}_{n-1}$ and so $[w, u = w + \phi \cdot -v] \in \mathrm{BW}_n$. The mathematical and coding properties of Barnes-Wall lattices have been studied in numerous works, e.g., Agrawal & Vardy (2000); Conway & Sloane (1998); Forney (1988); Forney & Vardy (1996); Micciancio & Nicolosi (2008); Nebe *et al.* (2001); Salomon & Amrani (2005).

**Combinatorial bounds.** Let $\ell(\eta, n)$ denote the worst-case list size (over all received words) for $\mathrm{BW}_n$ at rsd $\eta$. We prove the following upper bound.

THEOREM 1.2. *For any integer $n \geq 0$ and $\epsilon \in (0, 1]$, we have*

$$\ell(1 - \epsilon, n) \leq 4 \cdot (1/\epsilon)^{16n} = N^{O(\log(1/\epsilon))}.$$

Moreover, we show that the dependence on $\log(1/\epsilon)$ in the exponent is necessary, and thus the above bound is tight, up to polynomials.

THEOREM 1.3. *For any integer $n \geq 0$ and $\epsilon \in [2^{-n}, 1]$, we have*

$$\ell(1 - \epsilon, n) \geq 2^{(n - \log \frac{1}{\epsilon}) \log \frac{1}{2\epsilon}}.$$

*In particular, for any constant $\epsilon > 0$ (or even any $\epsilon \geq N^{-c}$ for $c < 1$), we have $\ell(1 - \epsilon, n) = N^{\Omega(\log(1/\epsilon))}$.*

As previously mentioned, it is also known that at rsd $\eta = 1$, the maximum list size $\ell(1, n)$ is quasi-polynomial $N^{\Theta(\log N)}$ in the lattice dimension, and is achieved by letting the received word be any lattice

[1] The fundamental volume of $\mathrm{BW}_n$ in $\mathbb{C}^N$ is $\det(W) = 2^{nN/2}$, so its determinant-normalized minimum distance is $\sqrt{N}/\det(W)^{1/(2N)} = \sqrt[4]{N}$. This is better than the normalized minimum distance 1 of the integer lattice $\mathbb{G}^N$, but worse that the largest possible of $\Theta(\sqrt{N})$ for $N$-dimensional lattices.

point (Conway & Sloane 1998, Chapter 1, §2.2, page 24). Because the rsmd of $\mathrm{BW}_n$ is exactly 1, here we are just considering the number of lattice points at minimum distance from the origin, the so-called "kissing number" of the lattice.

**List-decoding algorithm.** We complement the above combinatorial bounds with an algorithmic counterpart, which builds upon the unique (bounded-distance) decoding algorithm of Micciancio & Nicolosi (2008) for rsd up to $\frac{1}{4}$.

THEOREM 1.4. *There is a deterministic algorithm that, given any received word $r \in \mathbb{C}^N$ and $\eta \geq 0$, outputs the list of all points in $BW_n$ that lie within* rsd *$\eta$ of $r$, and runs in time $O(N^2) \cdot \ell(\eta, n)^2$.*

We also remark that the algorithm can be parallelized just as in Micciancio & Nicolosi (2008), and runs in only polylogarithmic $O(\log^2 N)$ parallel time on $p \geq N^2 \cdot \ell(\eta, n)^2$ processors.

Theorem 1.2 and Theorem 1.4 immediately imply the following corollary for $\eta = 1 - \epsilon$.

COROLLARY 1.5. *There is a deterministic algorithm that, given a received word $r \in \mathbb{C}^N$ and $\epsilon > 0$, outputs the list of all lattice points in $BW_n$ that lie within* rsd *$(1 - \epsilon)$ of $r$, and runs in time $(1/\epsilon)^{O(n)} = N^{O(\log(1/\epsilon))}$.*

Given the lower bounds, our algorithm is optimal in the sense that for any constant $\epsilon > 0$, it runs in $\mathrm{poly}(N)$ time for rsd $1 - \epsilon$, and that list decoding in $\mathrm{poly}(N)$ time is impossible (in the worst case) at rsd 1.

### 1.2. Proof Overview and Techniques.

**Combinatorial bounds.** Our combinatorial results exploit a few simple observations, some of which were initially used in obtaining the algorithmic results of Micciancio & Nicolosi (2008). The first is that by the Pythagorean theorem, if $\eta = \delta(r, w)$ is the rsd between a received vector $r = [r_0, r_1] \in \mathbb{C}^N$ and a lattice vector $w = [w_0, w_1] \in \mathrm{BW}_n$ (where $r_i \in \mathbb{C}^{N/2}$ and $w_i \in \mathrm{BW}_{n-1}$), then $\delta(r_b, w_b) \leq \eta$ for some $b \in \{0, 1\}$. The second observation (proved above) is that BW lattices

are closed under the operation of swapping the two halves of their vectors, namely, $[w_0, w_1] \in \mathrm{BW}_n$ if and only if $[w_1, w_0] \in \mathrm{BW}_n$. Therefore, without loss of generality we can assume that $\delta(r_0, w_0) \leq \eta$, while incurring only an extra factor of 2 in the final list size. A final important fact is the relationship between the rsd's for the two Barnes-Wall vectors $u = w_0, v = \frac{1}{\phi}(w_1 - w_0) \in \mathrm{BW}_{n-1}$ that determine $w$; namely, we have

$$\eta = \tfrac{1}{2}\delta(r_0, u) + \delta(\tfrac{1}{\phi}(r_1 - u), v).$$

(See Lemma 2.1.) Since $\delta(r_0, u) \leq \eta$, we have must have $\delta(\frac{1}{\phi}(r_1 - u), v) = \eta - \frac{1}{2}\delta(r_0, w_0) \in [\eta/2, \eta]$.

Our critical insight in analyzing the list size is to carefully partition the lattice vectors in the list according to their distances from the respective halves of the received word. Informally, a larger distance on the left half (between $r_0$ and $u$) allows for a larger list of $u$'s, but also implies a smaller distance on the right half (between $\frac{1}{\phi}(r_1 - u)$ and $v$), which limits the number of possible corresponding $v$'s. We bound the total list size using an inductive argument for various carefully chosen ranges of the distances at lower dimensions. Remarkably, this technique along with the Johnson bound allows us to obtain tight combinatorial bounds on the list size for distances all the way up to the minimum distance.

As a warm-up example, which also serves as an important step when analyzing larger rsd's, Lemma 2.5 gives a bound of $\ell(\frac{5}{8}, n) \leq 4 \cdot 24^n = \mathrm{poly}(N)$ for rsd $\eta = \frac{5}{8}$. This bound is obtained by partitioning according to the two cases $\delta(r_0, u) \in [0, \frac{5}{12})$ and $\delta(r_0, u) \in [\frac{5}{12}, \frac{5}{8}]$, which imply that the rsd between $v$ and $\frac{1}{\phi}(r_1 - u)$ is at most $\frac{5}{8}$ and $\frac{5}{12}$, respectively. When bounding the corresponding number of $u$'s and $v$'s, the rsd's up to $\frac{5}{12} < \frac{1}{2}$ are handled by the Johnson bound, and rsd's up to $\frac{5}{8}$ are handled by induction on the dimension.

To extend the argument to rsd's up to $\eta = 1 - \epsilon$, we need to partition into three cases, including ones which involve rsd's $1 - \frac{3\epsilon}{2}$ and $\frac{3}{4}$. In turn, the bound for rsd $\frac{3}{4}$ also uses three cases, plus the above bound for rsd $\frac{5}{8}$. Interestingly, all our attempts to use fewer cases or a more direct analysis resulted in qualitatively worse list size bounds, such as $N^{O(\log^2(1/\epsilon))}$ or worse.

Lastly, our lower bounds from Theorem 1.3 are obtained by using a representation of BW lattices in terms of RM codes (see Fact 1.9), and

by adapting the lower bounds from Gopalan *et al.* (2008) for RM codes to BW lattices.

**List-decoding algorithm.** A natural approach to devising a list-decoding algorithm using the above facts (also used in the context of Reed-Muller codes Gopalan *et al.* (2008)) is to first list decode the left half $r_0$ of the received word to get a list of $u$'s, and then sequentially run through the output list to decode the right half $\frac{1}{\phi}(r_1 - u)$ and get a corresponding list of $v$'s for each value of $u$. However, because the recursion has depth $n$, the straightforward analysis reveals a super-polynomial runtime $N^{\Omega(n)}$ for $\mathsf{rsd}\ \eta \geq 1/2$, because the list size at depth $d$ can be $\geq 4N/2^d$.

Instead, our list-decoding algorithm is based on the elegant divide-and-conquer algorithm of Micciancio & Nicolosi (2008) for bounded-distance (unique) decoding, which decodes up to half the minimum distance (i.e., $\eta = \frac{1}{4}$) in quasi-linear $\tilde{O}(N)$ time, or even poly-logarithmic $O(\log^c N)$ parallel time on a sufficiently large $\mathrm{poly}(N)$ number of processors.

The main feature of the algorithm, which we exploit in our algorithm as well, is the use of a distance-preserving linear automorphism $\mathcal{T}$ of the BW lattice, i.e., $\mathcal{T}(\mathrm{BW}_n) = \mathrm{BW}_n$ (see Fact 3.1). In particular, a lattice vector $w \in \mathrm{BW}_n$ can be reconstructed from just one arbitrary half of each of $w = [w_0, w_1]$ and $\mathcal{T}(w) = [\mathcal{T}_0(w), \mathcal{T}_1(w)]$. Recall that for a received word $r = [r_0, r_1]$ (where $r_i \in \mathbb{C}^{N/2}$), we are guaranteed that $\delta(r_b, w_b) \leq \delta(r, w)$ for some $b \in \{0, 1\}$, and similarly for $\mathcal{T}(r)$ and $\mathcal{T}(w)$. These facts straightforwardly yield a divide-and-conquer, parallelizable list-decoding algorithm that recursively list decodes each of the four halves $r_0, r_1, T_0(r), T_1(r)$ and reconstructs a list of solutions by combining appropriate pairs from the sub-lists, and keeping only those that are within the distance bound. The runtime of this algorithm is only quadratic in the worst-case list size, times a $\mathrm{poly}(N)$ factor (see Section 3). We emphasize that the only difference between our algorithm and the Micciancio-Nicolosi algorithm is the simple but crucial observation that one can replace single words by lists in the recursive steps. The runtime analysis, however, is entirely different, because it depends on the combinatorial bounds on list size.

**1.3. Comparison with Reed-Muller Codes.** Here we discuss several common and distinguishing features of Barnes-Wall lattices and Reed-Muller codes.

DEFINITION 1.6 (Reed-Muller code). *For integers $d, n \geq 0$, the Reed-Muller code of degree $d$ in $n$ variables (over $\mathbb{F}_2$) is defined as*

$$\mathit{RM}_n^d = \left\{ \langle p(\alpha) \rangle_{\alpha \in \mathbb{F}_2^n} : p \in \mathbb{F}_2[x_1, \ldots, x_n], \deg(p) \leq d \right\}.$$

*An equivalent recursive definition is $\mathit{RM}_n^0 = \{\bar{0}, \bar{1}\} \subseteq \mathbb{F}_2^{2^n}$ for any integer $n \geq 0$, and*

$$\mathit{RM}_n^d = \left\{ [u, u+v] : u \in \mathit{RM}_{n-1}^d, v \in \mathit{RM}_{n-1}^{d-1} \right\}.$$

*Here if $u \in \mathit{RM}_{n-1}^d, v \in \mathit{RM}_{n-1}^{d-1}$ correspond to polynomials $p_u, p_v \in \mathbb{F}_2[x_1, \ldots x_{n-1}]$ respectively, then the codeword $[u, u+v] \in \mathit{RM}_n^d$ corresponds to the polynomial $p = p_u + x_n \cdot p_v \in \mathbb{F}_2[x_1, \ldots, x_n]$.*

The recursive definition of RM codes already hints at structural similarities between BW lattices and RM codes. Indeed, BW lattices can be equivalently defined as evaluations modulo $\phi^n$ of (Gaussian) integer multilinear polynomials in $n$ variables over the domain $\{0, \phi\}^n$. Recall that an integer multilinear polynomial $p \in \mathbb{G}[x_1, \ldots, x_n]$ is one whose monomials have degree at most one in each variable (and hence total degree at most $n$), i.e.,

$$p(x_1, \ldots, x_n) = \sum_{S \in \{0,1\}^n} a_S \cdot \prod_{i \in S} x_i$$

where each $a_S \in \mathbb{G}$. A simple inductive argument proves the following lemma.

LEMMA 1.7.

$$\mathit{BW}_n = \{ \langle p(x) \rangle_{x \in \{0,\phi\}^n} : p \in \mathbb{G}[x_1, \ldots, x_n] \text{ is multilinear} \} + \phi^n \mathbb{G}^{2^n}.$$

Thus, while $\mathrm{RM}_n^d$ codewords correspond to low-degree polynomials (when $d$ is small), BW lattice points correspond to possibly high-degree polynomials. As an immediate application, our main theorems imply the following corollary regarding the set of integer multilinear polynomials that approximate a function $f \colon \{0, \phi\}^n \to \mathbb{C}$.

COROLLARY 1.8. *Given a map $f : \{0,\phi\}^n \to \mathbb{C}$ (represented as a lookup table) and $\epsilon = \Omega(N^{-c})$ for some $c < 1$ and $N = 2^n$, there exists an algorithm that outputs in time $N^{O(\log(1/\epsilon))}$ all the integer multilinear polynomials $g\colon \{0,\phi\}^n \to \mathbb{C}$ such that $\|f - g\|^2 \leq (1-\epsilon)N$.*

Just as in our algorithmic results for BW lattices, the recursive structure of RM codes is critically used in list-decoding algorithms for these codes, but in a different way than in our algorithm. The list-decoding algorithm for $\mathrm{RM}_n^d$ given in Gopalan *et al.* (2008) recursively list decodes one of the halves of a received word, and then for each codeword in the list it recursively list decodes the other half of the received word. The recursion has depth $d$ and thus has a total running time of $\mathrm{poly}(N) \cdot \ell(\eta)^d$, where $\ell(\eta)$ is the list size at relative (Hamming) distance $\eta$. As mentioned above, a similar algorithm can work for BW lattices, but the natural analysis implies a super-polynomial $\ell(\eta)^n$ lower bound on the running time, since now the recursion has depth $n$. The reason we can overcome this potential bottleneck is the existence of the linear automorphism $\mathcal{T}$ of $\mathrm{BW}_n$, which allows us to make only a *constant* number of recursive calls (independently of each other), plus a $\mathrm{poly}(N) \cdot \ell(\eta)^2$-time combining step, which yields a runtime of the form $O(1)^n \cdot \mathrm{poly}(N) \cdot \ell(\eta)^2 = \mathrm{poly}(N) \cdot \ell(\eta)^2$.

We note that $\mathrm{RM}_n^d$ codes are efficiently list decodable up to a radius larger than the minimum distance Gopalan *et al.* (2008), and remark that while RM codes are some of the oldest and most intensively studied codes, it was not until recently that their list-decoding properties have been very well understood Gopalan *et al.* (2008); Kaufman *et al.* (2010); Pellikaan & Wu (2004).

We finally note that the connection to Reed-Muller codes can also be made more explicit in the following alternate description of BW lattices, which we use in Section 2.3.

FACT 1.9 (Forney 1988, §IV.B).

$$BW_n = \{\sum_{d=0}^{n-1} \phi^d \cdot c_d + \phi^n \cdot c_n, \text{ with } c_d \in RM_n^d, \text{ and } 0 \leq d \leq n-1, \text{ and } c_n \in \mathbb{G}^N\}$$

*where the embedding of $\mathbb{F}_2$ into $\mathbb{C}$ is given by $0 \mapsto 0$ and $1 \mapsto 1$.*

*In particular, any codeword $c_d \in RM_d^n$ gives rise to a lattice point $\phi^d \cdot c_d \in BW_n$,*

**1.4. Other Related Work.** Cohn & Heninger (2015) study a list-decoding model on polynomial lattices, under both the Hamming metric and certain 'non-Archimedian' norms. Their polynomial analogue of Coppersmith's theorem Coppersmith (2001) implies, as a special case, Guruswami and Sudan's result on list decoding Reed-Solomon codes Guruswami & Sudan (1999).

Decoding and list decoding in the Euclidean space has been also considered for embeddings into real vector spaces of codes classically defined over finite fields. These embeddings can give rise to so-called spherical codes, where the decoding problem has as input a received vector on the unit sphere, and is required to output the points in the code (also on the unit sphere) that form a small angle with the given target. Another related decoding model is soft-decision decoding, where for each position of the received word, each alphabet symbol is assigned a real-valued weight representing the confidence that the received symbol matches it. Soft decision unique decoding for RM codes was studied in Dumer & Krichevskiy (2000); Dumer & Shabunov (2006a,b), and list-decoding algorithms were shown in Dumer *et al.* (2008); Fourquet & Tavernier (2008).

Further, the question of decoding lattices is related to the well-studied *vector quantization* problem. In this problem, vectors in the ambient space need to be rounded to nearby points of a discrete lattice; for further details on this problem see, for example, Conway & Sloane (1998).

**Organization.** In Section 2 we prove our combinatorial upper and lower bounds for BW lattices. In Section 3 we present and analyze our main list-decoding algorithm. We conclude in Section 4 with several open problems.

## 2. Combinatorial Bounds

We start with a few basic definitions. For a lattice $\mathcal{L}$, a vector $r \in \mathbb{C}^m$ (often called a received word) and any $\eta \geq 0$, define $L_{\mathcal{L}}(r, \eta) = \{x \in \mathcal{L} : \delta(r, x) \leq \eta\}$ to be the list of lattice points $w \in \mathcal{L}$ such that

$\delta(r, w) \leq \eta$. We often omit the subscript $\mathcal{L}$ when the lattice is clear from context. For $\eta \geq 0$ and nonnegative integer $n$ with $N = 2^n$, we define $\ell(\eta, n) = \max_{r \in \mathbb{C}^n} |L_{\mathrm{BW}_n}(r, \eta)|$ to be the maximum list size for rsd $\eta$, for the $n$th Barnes-Wall lattice.

**2.1. Helpful Lemmas.** We start with two simple but important observations about Barnes-Wall lattices. The first relates the rsd's between the respective "left" and "right" halves of a received word and a lattice point. The second relates the list sizes for the same rsd but different dimensions.

LEMMA 2.1. *Let $r = [r_0, r_1] \in \mathbb{C}^N$ with $r_0, r_1 \in \mathbb{C}^{N/2}$, and $w = [u, u + \phi v] \in BW_n$ for $u, v \in BW_{n-1}$. Let $\eta = \delta(r, w)$, $\eta_0 = \delta(r_0, u)$ and $\eta_1 = \delta(\frac{1}{\phi}(r_1 - u), v)$. Then $\eta = \frac{\eta_0}{2} + \eta_1$.*

PROOF. We have

$$\begin{aligned} \delta(r, w) &= \frac{\delta(r_0, u) + \delta(r_1, u + \phi v)}{2} \\ &= \frac{\eta_0}{2} + \frac{|\phi|^2 \cdot \delta(\frac{1}{\phi}(r_1 - u), v)}{2} \\ &= \frac{\eta_0}{2} + \eta_1. \end{aligned}$$

□

LEMMA 2.2. *For any $\eta \geq 0$ and $n \geq 1$, we have $\ell(\eta, n-1) \leq \ell(\eta, n)$.*

PROOF. Let $r \in \mathbb{C}^{N/2}$ and $w \in L(r, \eta) \subseteq \mathrm{BW}_{n-1}$. Then $\delta([r, r], [w, w]) = \delta(r, w)$, and since $[w, w] \in \mathrm{BW}_n$ (because $w \in \mathrm{BW}_{n-1}$) it follows that $[w, w] \in L([r, r], \eta)$. □

We next state a Johnson-type bound on the list size for arbitrary lattices; see, e.g., Bollobás (1986); Guruswami & Sudan (2001); Micciancio & Goldwasser (2002); Sudan (2001) for proofs. Note that these sources work in $\mathbb{R}^N$; our form follows because the standard isomorphism between $\mathbb{C}^N$ and $\mathbb{R}^{2N}$ as real vector spaces also preserves Euclidean norm.

LEMMA 2.3 (Johnson bound). *Let $\mathcal{L} \subset \mathbb{C}^N$ be a lattice of* rsmd $\delta = \delta(\mathcal{L})$ *and let $r \in \mathbb{C}^N$. Then*

*(i)* $|L(r, \frac{\delta}{2})| \leq 4N$, *and*

*(ii)* $|L(r, \delta \cdot (\frac{1}{2} - \epsilon))| \leq \frac{1}{2\epsilon}$ *for any $\epsilon > 0$.*

(In reading these bounds, recall that $\delta(\mathcal{L})/4$, not $\delta(\mathcal{L})/2$, is the relative unique-decoding distance of $\mathcal{L}$, because $\delta(\mathcal{L})$ is the relative *squared* minimum distance of the lattice.)

COROLLARY 2.4. *For the lattice $BW_n \subseteq \mathbb{C}^N$ and any $\epsilon > 0$, we have $\ell(\frac{1}{2}, n) = 4N$ and $\ell(\frac{1}{2} - \epsilon, n) \leq \frac{1}{2\epsilon}$.*

PROOF. Since $\delta(\mathrm{BW}_n) = 1$, the upper bounds follow immediately by Lemma 2.3. For the lower bound $\ell(\frac{1}{2}, n) \geq 4N$, we give an inductive argument showing that $|L(r, \frac{1}{2})| \geq 4N$ for the received word $r = (\frac{\phi}{2}, \ldots, \frac{\phi}{2}) \in \mathbb{C}^N$. To do this, we show by induction on $n$ that $L(r, \frac{1}{2})$ contains $2N$ pairwise disjoint (unordered) pairs $\{w_i, w_i'\}$ where $w_i - w_i' \in \phi \cdot \mathrm{BW}_n$.

For the base case $n = 0$, notice that $L(\frac{\phi}{2}, \frac{1}{2}) = \{0, 1, i, 1 + i\}$, and that $(1 + i) - 0 = \phi \in \phi \cdot \mathrm{BW}_0$ and $i - 1 = \phi \cdot i \in \phi \cdot \mathrm{BW}_0$. Next, let $\{w_i, w_i'\}$ denote the pairs guaranteed by the inductive hypothesis for some $n$, and recall that $[a, b] \in \mathrm{BW}_{n+1}$ if and only if $a, b \in \mathrm{BW}_n$ and $a - b \in \phi \cdot \mathrm{BW}_n$. It is easy to verify that the pairs $\{[w_i, w_i], [w_i', w_i']\}$ and $\{[w_i, w_i'], [w_i', w_i]\}$ establish the inductive hypothesis for $n + 1$. Indeed, $[w_i, w_i] - [w_i', w_i'] = \phi \cdot [w, w] \in \phi \cdot \mathrm{BW}_{n+1}$ for some $w \in \mathrm{BW}_n$, and similarly, $[w_i, w_i'] - [w_i', w_i] = \phi \cdot [w, -w] \in \phi \cdot \mathrm{BW}_{n+1}$, because $w - (-w) = 2w \in \phi \cdot \mathrm{BW}_n$. Also, $\delta([r, r], [w_i, w_i]) = \delta(r, w_i) \leq \frac{1}{2}$ and similarly for the other vectors. □

**2.2. Beyond the Johnson Bound.** In this section we prove our main combinatorial bounds on the list size for Barnes-Wall lattices $\mathrm{BW}_n \subseteq \mathbb{G}^N$. Our main result is that the list size at rsd $(1 - \epsilon)$ is $(1/\epsilon)^{O(n)} = N^{O(\log(1/\epsilon))}$ for any $\epsilon > 0$. The proof strategy is inductive, and is based on a careful partitioning of the lattice vectors in the list according to the distances of their left and right halves from the respective halves of the received word. Intuitively, the larger the distance on one half, the smaller the distance on the other (Lemma 2.1 above makes this precise).

The total list size can therefore be bounded using list bounds for various carefully chosen distances at lower dimensions. Our analysis relies on a $\mathrm{poly}(N)$ list-size bound for $\mathsf{rsd}\ \frac{3}{4}$, which in turn relies on a $\mathrm{poly}(N)$ bound for $\mathsf{rsd}\ \frac{5}{8}$. We first prove these simpler bounds, also using a partitioning argument. (Note that the concrete constants appearing below are chosen to simplify the analysis, and are likely not optimal.)

LEMMA 2.5. *For any integer $n \geq 0$, we have $\ell(\frac{5}{8}, n) \leq 4 \cdot 24^n$.*

PROOF. For $n = 0$, it is easy to see that for all $\eta < 1$, there are at most $4$ Gaussian integers within a ball of radius $\eta$ from any $r \in \mathbb{G}$, so $\ell(\eta, 0) \leq 4$. Suppose now that $n \geq 1$ with $N = 2^n$. Let $r = [r_0, r_1] \in \mathbb{C}^N$ with $r_0, r_1 \in \mathbb{C}^{N/2}$ be an arbitrary received word, and let $w = [u, u + \phi v] \in L(r, \frac{5}{8})$ for $u, v \in \mathrm{BW}_{n-1}$. Let $\eta = \delta(r, w) \leq \frac{5}{8}$, $\eta_0 = \delta(r_0, u)$ and $\eta_1 = \delta(\frac{1}{\phi}(r_1 - u), v)$.

Note that from Lemma 2.1 we have that $\eta = \frac{\eta_0}{2} + \eta_1 = \frac{1}{2}(\delta(r_0, u) + \delta(r_1, u + \phi v)) \leq \frac{5}{8}$. Without loss of generality, we can assume that $\eta_0 = \delta(r_0, u) \leq \frac{5}{8}$. For if not, then we would have $\delta(r_1, u + \phi v) \leq \frac{5}{8}$, and since $[a, b] \in \mathrm{BW}_n$ implies $[b, a] \in \mathrm{BW}_n$ for $a, b \in \mathbb{G}^{N/2}$, we could instead work with the received word $r' = [r_1, r_0]$ and $w' = [u + \phi v, u] \in L(r', \frac{5}{8})$. This incurs a factor of at most 2 in the total list size, which we account for in the analysis below.

Assuming $\eta_0 \leq \frac{5}{8}$, we now split the analysis into two cases: $\eta_0 \in [0, \frac{5}{12})$, and $\eta_0 \in [\frac{5}{12}, \frac{5}{8}]$. By Lemma 2.1, these cases correspond to $\eta_1 \leq \frac{5}{8}$ and $\eta_1 \leq \frac{5}{12}$, respectively. Since $u \in L(r_0, \eta_0)$ and $v \in L(\frac{1}{\phi}(r_1 - u), \eta_1)$, after combining the lists we obtain at most $\ell(\frac{5}{12}, n - 1) \cdot \ell(\frac{5}{8}, n - 1) + \ell(\frac{5}{8}, n - 1) \cdot \ell(\frac{5}{12}, n - 1)$ potential vectors in the list.

Finally, after incorporating the factor of 2 from the argument above, we have (where for conciseness we write $\ell(\eta)$ for $\ell(\eta, n - 1)$):

$$\begin{aligned}
\ell(\tfrac{5}{8}, n) &\leq 2 \cdot \left(\ell(\tfrac{5}{12}) \cdot \ell(\tfrac{5}{8}) + \ell(\tfrac{5}{8}) \cdot \ell(\tfrac{5}{12})\right) \\
&= 4 \cdot \ell(\tfrac{5}{12}) \cdot \ell(\tfrac{5}{8}) \\
&\leq 4 \cdot 6 \cdot \ell(\tfrac{5}{8}) && \text{( Corollary 2.4)} \\
&\leq 24^n \cdot \ell(\tfrac{5}{8}, 0). && \text{(unwind the recurrence)}
\end{aligned}$$

□

LEMMA 2.6. *For any integer* $n \geq 0$, *we have* $\ell(\frac{3}{4}, n) \leq 4 \cdot 24^{2n}$.

PROOF. As noted in the proof of Lemma 2.5, the claim is clearly true for $n = 0$, so suppose $n \geq 1$; we proceed by induction on $n$. Define the same notation as in the proof of Lemma 2.5, using rsd bound $\frac{3}{4}$ instead of $\frac{5}{8}$.

As before, we assume that $\eta_0 \leq \frac{3}{4}$ and account for the accompanying factor of 2 in the list size. This time we split the analysis into three cases: $\eta_0 \in [0, \frac{1}{4})$, $\eta_0 \in [\frac{1}{4}, \frac{5}{8})$, and $\eta_0 \in [\frac{5}{8}, \frac{3}{4}]$. By Lemma 2.1, these correspond to $\eta_1 \leq \frac{3}{4}$, $\eta_1 \leq \frac{5}{8}$, and $\eta_1 \leq \frac{7}{16}$, respectively.

For conciseness, in the calculation below we write $\ell(\eta)$ for $\ell(\eta, n-1)$. We have

$$\begin{aligned}
\ell(\tfrac{3}{4}, n) &\leq 2 \cdot \big(\ell(\tfrac{1}{4}) \cdot \ell(\tfrac{3}{4}) + \ell(\tfrac{5}{8}) \cdot \ell(\tfrac{5}{8}) + \ell(\tfrac{3}{4}) \cdot \ell(\tfrac{7}{16})\big) \\
&\leq 2 \cdot (2+8) \cdot \ell(\tfrac{3}{4}) + 2 \cdot \ell(\tfrac{5}{8})^2 \\
&\leq 20 \cdot 4 \cdot 24^{2(n-1)} + 32 \cdot 24^{2(n-1)} \\
&\leq 4 \cdot 24^{2n},
\end{aligned}$$

where we used Corollary 2.4, the induction hypothesis and Lemma 2.5. □

We are now ready to prove our main combinatorial bound (Theorem 1.2). We restate it here for convenience.

THEOREM 1.2. *For any integer* $n \geq 0$ *and* $\epsilon \in (0, 1]$, *we have*

$$\ell(1-\epsilon, n) \leq 4 \cdot (1/\epsilon)^{16n} = N^{O(\log(1/\epsilon))}.$$

PROOF. We need to show that $\ell(1-\epsilon, n) \leq 4 \cdot (1/\epsilon)^{16n}$ for any $n \geq 0$ and $\epsilon > 0$; obviously, we can assume $\epsilon \leq 1$ as well. As noted in the proof of Lemma 2.5, the claim is clearly true for $n = 0$. We proceed by induction on $n$; namely, we assume that for all $\gamma > 0$ it is the case that $\ell(1-\gamma, n-1) \leq 4 \cdot (1/\gamma)^{16(n-1)}$. Define the same notation as in the proof of Lemma 2.5, using rsd bound $1 - \epsilon$ instead of $\frac{5}{8}$.

As in earlier proofs, we assume that $\eta_0 \leq 1 - \epsilon$ and account for the accompanying factor of 2 in the list size. We split the analysis into 3 cases: $\eta_0 \in [0, \frac{1}{2} - \epsilon)$, $\eta_0 \in [\frac{1}{2} - \epsilon, 1 - \frac{3\epsilon}{2})$, and $\eta_0 \in [1 - \frac{3\epsilon}{2}, 1 - \epsilon]$. By Lemma 2.1, these correspond to $\eta_1 \leq 1 - \epsilon$, $\eta_1 \leq \frac{3}{4} - \frac{\epsilon}{2} < \frac{3}{4}$, and $\eta_1 \leq \frac{1}{2} - \frac{\epsilon}{4}$, respectively.

For conciseness, in the calculation below we write $\ell(\eta)$ for $\ell(\eta, n-1)$. It follows that $\ell(1-\epsilon, n)$ is bounded by

$$
\begin{aligned}
&2\left(\ell(1-\epsilon)\ell(\tfrac{1}{2}-\epsilon) + \ell(1-\epsilon)\ell(\tfrac{1}{2}-\tfrac{\epsilon}{4}) + \ell(1-\tfrac{3\epsilon}{2})\ell(\tfrac{3}{4})\right) \\
&\quad \le 2\ell(1-\epsilon)(\tfrac{1}{2\epsilon}+\tfrac{2}{\epsilon}) + 2\ell(1-\tfrac{3\epsilon}{2})\cdot 4\cdot 24^{2(n-1)} \text{(Corollary 2.4, Lemma 2.6)} \\
&\quad = \tfrac{5}{\epsilon}\cdot\ell(1-\epsilon) + 8\cdot 24^{2(n-1)}\cdot\ell(1-\tfrac{3\epsilon}{2}) \\
&\quad \le \tfrac{20}{\epsilon}\cdot(\tfrac{1}{\epsilon})^{16(n-1)} + 32\cdot 24^{2(n-1)}\cdot(\tfrac{2}{3\epsilon})^{16(n-1)}, \quad \text{(induction hypothesis)} \\
&\quad = (\tfrac{1}{\epsilon})^{16(n-1)}\cdot(\tfrac{20}{\epsilon} + 32\cdot(24^2\cdot(\tfrac{2}{3})^{16})^{(n-1)}) \\
&\quad \le (\tfrac{1}{\epsilon})^{16(n-1)}\cdot(\tfrac{52}{\epsilon}) \\
&\quad \le 4\cdot(\tfrac{1}{\epsilon})^{16n}
\end{aligned}
$$

when $\epsilon \le \frac{4}{5}$. If $\epsilon \in (\frac{4}{5}, 1]$ then $\ell(1-\epsilon, n) = 1 \le 4\cdot(\frac{1}{\epsilon})^{16n}$, and the proof is complete. □

Notice that in the above proof, it is important to use an upper bound like $\eta_0 \le 1 - \frac{3\epsilon}{2}$ in one of the cases, so that the factor $(\frac{2}{3})^{16(n-1)}$ from the inductive list bound can cancel out the corresponding factor of $24^{2(n-1)}$ for the corresponding rsd bound $\eta_1 \le \frac{3}{4}$. This allows the recurrence to be dominated by the term

$$\ell(1-\epsilon)\cdot\ell(\tfrac{1}{2}-\tfrac{\epsilon}{4}) = O(\tfrac{1}{\epsilon})\cdot\ell(1-\epsilon),$$

yielding a solution of the form $(1/\epsilon)^{O(n)}$.

**2.3. Lower Bounds.** For our lower bounds we make use of the relationship between Barnes-Wall lattices and Reed-Muller codes from Fact 1.9, and then apply known lower bounds for the latter.

FACT 2.7 (MacWilliams & Sloane 1981, Chap. 13, §4).

*(i) The minimum distance of $RM_n^d$ is $2^{n-d}$. In particular, the characteristic vector $c_V \in \mathbb{F}_2^{2^n}$ of any subspace $V \subseteq \mathbb{F}_2^n$ of dimension $k \ge n-d$ is a codeword of $RM_n^d$.*

*(The characteristic vector $c_S \in \mathbb{F}_2^{2^n}$ of a set $S \subseteq \mathbb{F}_2^n$ is defined by indexing the coordinates of $\mathbb{F}_2^{2^n}$ by elements $\alpha \in \mathbb{F}_2^n$, and letting $(c_S)_\alpha = 1$ if and only if $\alpha \in S$.)*

*(ii) There are* $2^d \cdot \prod_{i=0}^{n-d-1} \frac{2^{n-i}-1}{2^{n-d-i}-1} > 2^{d(n-d)}$ *subspaces of dimension* $n-d$ *in* $\mathbb{F}_2^n$.

We now prove Theorem 1.3, restated here for convenience.

THEOREM 1.3. *For any integer* $n \geq 0$ *and* $\epsilon \in [2^{-n}, 1]$, *we have*

$$\ell(1-\epsilon, n) \geq 2^{(n-\log\frac{1}{\epsilon})\log\frac{1}{2\epsilon}}.$$

*In particular, for any constant* $\epsilon > 0$ *(or even any* $\epsilon \geq N^{-c}$ *for* $c < 1$*), we have* $\ell(1-\epsilon, n) = N^{\Omega(\log(1/\epsilon))}$.

PROOF. Let $k \geq 0$ be an integer such that $2^n\epsilon \leq 2^k \leq 2^{n+1}\epsilon$. Let the received word be $r = \phi^k \cdot [1, 0, \ldots, 0] \in \mathbb{G}^N$, where we assume that the first coordinate is indexed by $0^n \in \mathbb{F}_2^n$. By Fact 2.7 and Fact 1.9, for any subspace $H \subseteq \mathbb{F}_2^n$ of dimension $n-k$, we have $\phi^k \cdot c_H \in \mathrm{BW}_n$. Notice that

$$\begin{aligned} \|r - \phi^k \cdot c_H\|^2 &= |\phi^k|^2 \cdot \|c_H - [1, 0, \ldots, 0]\|^2 \\ &= 2^k \cdot (2^{n-k} - 1) \\ &= 2^n - 2^k \leq 2^n(1-\epsilon). \end{aligned}$$

By Fact 2.7, there are at least $2^{k(n-k)} \geq 2^{(n-\log\frac{1}{\epsilon})\log\frac{1}{2\epsilon}}$ subspaces $H \subset \mathbb{F}_2^n$ of dimension $n-k$, which completes the proof. □

## 3. List-Decoding Algorithm

In this section we give a list-decoding algorithm that runs in time polynomial in $N$ and the list size; in particular, by Theorem 1.2 it runs in time $N^{O(\log(1/\epsilon))}$ for $\mathsf{rsd}$ $(1-\epsilon)$ for any fixed $\epsilon > 0$. This runtime and error tolerance are optimal (up to polynomial overhead) in the sense that the list size can be $N^{\Omega(\log(1/\epsilon))}$ by Theorem 1.3, and can be super-polynomial in $N$ for $\mathsf{rsd}$ 1 or more.

Our list-decoding algorithm is very similar to the (parallel) Bounded Distance Decoding algorithm of Micciancio & Nicolosi (2008), which outputs the unique lattice point within $\mathsf{rsd}$ $\eta < \frac{1}{4}$ of the received word (if it exists). In particular, both algorithms work by recursively (and independently) decoding four words of dimension $N/2$ that are derived from

the received word, and then combining the results appropriately. In our case, the runtime is strongly influenced by the sizes of the lists returned by the recursive calls, and so the combinatorial bounds from Section 2 are critical to the runtime analysis.

We need the following easily-verified fact regarding the symmetries (automorphisms) of $\mathrm{BW}_n$.

FACT 3.1. *For $N = 2^n$, the linear transformation $\mathcal{T} : \mathbb{C}^N \to \mathbb{C}^N$ given by $\mathcal{T}([u, v]) = \frac{\phi}{2} \cdot [u + v, u - v]$ is a distance-preserving automorphism of $BW_n$, namely $\mathcal{T}(BW_n) = BW_n$ and $\delta(x) = \delta(\mathcal{T}(x))$ for all $x \in \mathbb{C}^N$.*

**Algorithm 1** LISTDECODEBW: List-decoding algorithm for Barnes-Wall lattices.

**Input:** $r \in \mathbb{C}^N$ (for $N = 2^n$) and $\eta \geq 0$.

**Output:** The list $L(r, \eta) \subset \mathrm{BW}_n$.

1: **if** $n = 0$ **then**
2: output $L(r, \eta) \subset \mathbb{G}$ by enumeration.
3: parse $r = [r_0, r_1]$ for $r_0, r_1 \in \mathbb{C}^{N/2}$, and let $r_+ = \frac{\phi}{2}(r_0 + r_1)$ and $r_- = \frac{\phi}{2}(r_0 - r_1)$, so $[r_+, r_-] = \mathcal{T}(r)$.
4: **for all** $j \in \{0, 1, +, -\}$ **do**
5: let $L_j = \text{LISTDECODEBW}(r_j, \eta)$.
6: for each $(b, s) \in \{0, 1\} \times \{+, -\}$ and $(w_b, w_s) \in L_b \times L_s$, compute the corresponding candidate vector $w = [w_0, w_1] \in \mathrm{BW}_n$ as the appropriate one of the following:

$$[w_0, \tfrac{2}{\phi} w_+ - w_0], \quad [w_0, w_0 - \tfrac{2}{\phi} w_-],$$
$$[\tfrac{2}{\phi} w_+ - w_1, w_1], \quad [\tfrac{2}{\phi} w_- + w_1, w_1].$$

7: remove all candidate vectors $w$ such that $\delta(r, w) > \eta$.
8: sort the remaining list of candidates lexicographically and remove all duplicates.
9: **return** the set $L$ of all the candidate vectors remaining.

The following theorem, when combined with our combinatorial upper bound (Theorem 1.2), yields Theorem 1.4 as an immediate corollary.

THEOREM 3.2. *Algorithm 1 is correct and runs in $O(N^2) \cdot \ell(\eta, n)^2$ scalar operations over $\mathbb{C}$.*

PROOF. We need to show that on input $r \in \mathbb{C}^N$ and $\eta \geq 0$, Algorithm 1 runs in time $O(N^2) \cdot \ell(\eta, n)^2$ and outputs $L = L(r, \eta)$.

We first prove correctness, by induction. The algorithm is clearly correct for $n = 0$; now suppose that $n \geq 1$ and the algorithm is correct for $n - 1$. Adopt the notation from Algorithm 1, and let $w = [w_0, w_1] \in L(r, \eta)$ for $w_0, w_1 \in \mathrm{BW}_{n-1}$ be arbitrary. Since $\delta(w, r) \leq \eta$, we have $\delta(r_0, w_0) \leq \eta$ or $\delta(r_1, w_1) \leq \eta$ or both, so $w_0 \in L(r_0, \eta)$ or $w_1 \in L(r_1, \eta)$ or both. The same is true about the corresponding vectors after applying the automorphism $\mathcal{T}$. Namely, letting $[w_+, w_-] = \mathcal{T}(w) \in \mathrm{BW}_n$ for $w_+, w_- \in \mathrm{BW}_{n-1}$, we have $[w_+, w_-] \in L([r_+, r_-], \eta)$ and so $w_+ \in L(r_+, \eta)$ or $w_- \in L(r_-, \eta)$ or both.

By the inductive hypothesis and the above observations, we will have $(w_b, w_s) \in L_b \times L_s$ for at least one choice of $(b, s) \in \{0, 1\} \times \{+, -\}$. The algorithm calculates the vector $w = [w_0, w_1]$ as a candidate, simply by solving for $w_0, w_1$ using $w_b, w_s$ and the definition of $\mathcal{T}$. Therefore, $w$ will appear in the output list $L$. And because $L \subseteq L(r, \eta)$ by Step 7, the claim follows.

We now analyze $T(N)$, the number of operations over $\mathbb{C}$ for an input of dimension $N = 2^n$. We first observe that after filtering (Step 7), each remaining vector can appear at most four times in the list. Indeed, by induction $L_0, L_1, L_+$ and $L_-$ themselves do not contain any duplicates, and no two distinct elements from one of these lists can give rise to the same lattice point in $\mathrm{BW}_n$. Therefore, sorting and de-duplicating (Step 8) takes $O(N) \cdot \ell(n, \eta)^2$ operations, which implies that $T(N)$ satisfies

$$\begin{aligned} T(N) &= 4 \cdot T(N/2) + 4 \cdot O(N) \cdot \ell(\eta, n-1)^2 + O(N) \cdot \ell(\eta, n)^2 \\ &= 4 \cdot T(N/2) + O(N) \cdot \ell(\eta, n)^2 \\ &= O(N^2) \cdot \ell(\eta, n)^2 \end{aligned}$$

by the Master Theorem for recurrences (since $\ell(\eta, n - i) \leq \ell(\eta, n)$ for all $i \geq 0$). □

REMARK 3.3. *We note that the above algorithm, like the unique decoder of Micciancio & Nicolosi (2008), can be easily parallelized. On $p$ processors, the parallel runtime (measured in number of operations over*

$\mathbb{C}$) *satisfies the recurrence*

$$T(N,p) = \begin{cases} T(N), \text{ if } n = 0 \text{ or } p < 4 \\ T(N/2, p/4) + O(N \cdot \ell(\eta, n-1)^2/p + \log N), \text{ otherwise,} \end{cases}$$

*where $T(N)$ is as in the proof of Theorem 3.2. This is because it takes $O(N \cdot \ell(\eta, n-1)^2/p)$ operations per processor to combine the lists in Step 6, and computing each of the $\ell(\eta, n-1)^2$ distances in Step 9 requires computing a sum of $N$ real numbers, for a total of $O(N \cdot \ell(\eta, n-1)^2/p + \log N)$ parallel runtime. Notice that when $p \geq N^2 \cdot \ell(\eta, n-1)^2$, the algorithm runs in only polylogarithmic $O(\log^2 N)$ parallel time. Note also that when the list size $\ell(\eta, n-1) = 1$, this analysis specializes exactly to that of Micciancio & Nicolosi (2008).*

## 4. Discussion and Open Problems

Some immediate open questions arise from comparison to the results of Micciancio & Nicolosi (2008). Motivated by the sequential unique decoder proposed in Micciancio & Nicolosi (2008), is there a (possibly sequential) list decoder that runs in time *quasilinear* in $N$ and the list size, rather than quadratic? Also, as asked in Micciancio & Nicolosi (2008), is there an efficient algorithm for solving the Closest Vector Problem (i.e., minimum-distance decoding) on Barnes-Wall lattices? Note that our combinatorial lower bounds do not rule out the existence of such an algorithm, since for the Closest Vector Problem the algorithm only needs to output a single vector, not the list of all closest vectors.

An important variant of the list-decoding problem for codes is *local* list decoding. In this model, the algorithm is required to run in time polylogarithmic in the block length, and output succinct representations of all the codewords within a given radius. Defining a meaningful notion of local decoding for lattices (and BW lattices in particular) would require additional constraints, since lattice points do not in general admit succinct representations (since one needs to specify an integer coefficient for each basis vector). While by the Johnson bound we have a $\mathrm{poly}(n)$ list size for $\mathsf{rsd}$ up to $1/2 - \mathrm{poly}(1/n)$, achieving a meaningful notion of local decoding in this context would be interesting.

Another interesting direction is to find (or construct) more asymptotic families of lattices with nice list-decoding properties. In particular,

are there generic operations, which, when applied to lattices, guarantee good list-decoding properties? For codes, list decodability has been shown to behave well under the tensoring and interleaving operations, as demonstrated in Gopalan *et al.* (2011). Tensoring is also well-defined for lattices, but it does not behave so well as in codes. For example, tensoring a code with itself results in a code whose minimum distance is squared, while tensoring a lattice with itself does not square the distance. This issue has appeared in deciding NP-hardness of the Shortest Vector Problem Haviv & Regev (2012); Micciancio (2012) where the tensoring technique turned out to be much trickier than a tensoring approach for deciding the NP-hardness of the analogous minimum distance problem in codes Dumer *et al.* (2003) [2]. Understanding how list decoding behaves in the context of tensoring could bring up novel aspects of lattice list decoding, and it remains an intriguing further direction.

Finally, it would be also interesting and potentially useful to consider list decoding for norms other than the Euclidean norm, such as the $\ell_\infty$ or $\ell_0$ norms.

## Acknowledgements

A preliminary version appeared in the proceedings of Conference on Computational Complexity, Porto, Portugal, 2012 Grigorescu & Peikert (2012). E.G.'s work was supported by the NSF under Grant #1019343 to the Computing Research Association for the CI Fellows Project, and C.P.'s work was supported by the NSF under CAREER Award CCF-1054495 and the Alfred P. Sloan Foundation. The views expressed are those of the authors and do not necessarily reflect the official policy or position of the National Science Foundation of the Sloan Foundation. We thank Eli Ben-Sasson, Daniele Micciancio, Madhu Sudan, and Santosh Vempala for helpful discussions and comments.

## References

DAKSHI AGRAWAL & ALEXANDER VARDY (2000). Generalized minimum distance decoding in Euclidean space: Performance analysis. *IEEE Transaction on Information Theory* **46**(1), 60–83.

[2] We thank an anonymous reviewer for pointing this out to us.

ADI AKAVIA, SHAFI GOLDWASSER & SCHMUEL SAFRA (2003). Proving Hard-Core Predicates Using List Decoding. In *IEEE Symposium on Foundations of Computer Science*, 146–157.

IAN F. BLAKE AMIR H. BANIHASHEMI (1998). Trellis Complexity and Minimal Trellis Diagrams of Lattices. *IEEE Transaction on Information Theory* **44**(5), 1829–1847.

E. S. BARNES & G. E. WALL (1959). Some extreme forms defined in terms of Abelian groups. *Journal of the Australian Mathematical Society* **1**(01), 47–63.

BELA BOLLOBÁS (1986). *Combinatorics*. Cambridge University Press, Cambridge, U.K.

HENRY COHN & NADIA HENINGER (2015). Ideal forms of Coppersmith's theorem and Guruswami-Sudan list decoding. *Advances in Mathematics of Communications* **9**(3), 311–339.

JOHN H. CONWAY & NEIL J. A. SLOANE (1998). *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York.

DON COPPERSMITH (2001). Finding Small Solutions to Small Degree Polynomials. In *Cryptography and Lattices, International Conference*, 20–31.

IRIT DINUR, ELENA GRIGORESCU, SWASTIK KOPPARTY & MADHU SUDAN (2008). Decodability of group homomorphisms beyond the Johnson bound. In *ACM Symposium on Theory of Computing*, 275–284.

ILYA DUMER, GREGORY A. KABATIANSKY & CÉDRIC TAVERNIER (2008). List Decoding of Biorthogonal Codes and the Hadamard Transform With Linear Complexity. *IEEE Transactions on Information Theory* **54**(10), 4488–4492.

ILYA DUMER & RAFAIL E. KRICHEVSKIY (2000). Soft-decision majority decoding of Reed-Muller codes. *IEEE Transactions on Information Theory* **46**(1), 258–264.

ILYA DUMER, DANIELE MICCIANCIO & MADHU SUDAN (2003). Hardness of approximating the minimum distance of a linear code. *IEEE Transaction on Information Theory* **49**(1), 22–37.

ILYA DUMER & KIRILL SHABUNOV (2006a). Recursive error correction for general Reed-Muller codes. *Discrete Applied Mathematics* **154**(2), 253–269.

ILYA DUMER & KIRILL SHABUNOV (2006b). Soft-decision decoding of Reed-Muller codes: recursive lists. *IEEE Transactions on Information Theory* **52**(3), 1260–1266.

PETER ELIAS (1957). List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT* .

G. DAVID FORNEY (1988). Coset codes-II: Binary lattices and related codes. *IEEE Transactions on Information Theory* **34**(5), 1152–1187.

G. DAVID FORNEY & ALEXANDER VARDY (1996). Generalized minimum-distance decoding of Euclidean-space codes and lattices. *IEEE Transactions on Information Theory* **42**(6), 1992–2026.

RAFAËL FOURQUET & CÉDRIC TAVERNIER (2008). An improved list decoding algorithm for the second order Reed-Muller codes and its applications. *Designs, Codes and Cryptography* **49**(1-3), 323–340.

ANNA C. GILBERT, SUDIPTO GUHA, PIOTR INDYK, S. MUTHUKRISHNAN & MARTIN STRAUSS (2002). Near-optimal sparse Fourier representations via sampling. In *ACM Symposium on the Theory of Computing*, 152–161.

ODED GOLDREICH & LEONID A. LEVIN (1989). A hard-core predicate for all one-way functions. In *In Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, 25–32.

PARIKSHIT GOPALAN, VENKATESAN GURUSWAMI & PRASAD RAGHAVENDRA (2011). List Decoding Tensor Products and Interleaved Codes. *SIAM Journal of Computing* **40**(5), 1432–1462.

PARIKSHIT GOPALAN, ADAM R. KLIVANS & DAVID ZUCKERMAN (2008). List-decoding Reed-Muller codes over small fields. In *ACM Symposium on the Theory of Computing*, 265–274.

ELENA GRIGORESCU & CHRIS PEIKERT (2012). List Decoding Barnes-Wall Lattices. In *IEEE Conference on Computational Complexity*, 316–325.

VENKATESAN GURUSWAMI (2004). *List Decoding of Error-Correcting Codes (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition)*, volume 3282 of *Lecture Notes in Computer Science*. Springer.

VENKATESAN GURUSWAMI (2006). Algorithmic Results in List Decoding. *Foundations and Trends in Theoretical Computer Science* **2**(2).

VENKATESAN GURUSWAMI (2010). Bridging Shannon and Hamming: List Error-Correction with Optimal Rate. ICM Invited Survey.

VENKATESAN GURUSWAMI & ATRI RUDRA (2006). Explicit capacity-achieving list-decodable codes. In *ACM Symposium on the Theory of Computing*, 1–10.

VENKATESAN GURUSWAMI & MADHU SUDAN (1999). Improved decoding of Reed-Solomon and Algebraic-geometric codes. *IEEE Transactions on Information Theory* **45**, 1757–1767.

VENKATESAN GURUSWAMI & MADHU SUDAN (2001). Extensions to the Johnson Bound. Manuscript. Available from `http://people.csail.mit.edu/madhu/papers/johnson.ps`.

VENKATESAN GURUSWAMI, CHRISTOPHER UMANS & SALIL P. VADHAN (2009). Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM* **56**(4).

ISHAY HAVIV & ODED REGEV (2012). Tensor-based Hardness of the Shortest Vector Problem to within Almost Polynomial Factors. *Theory of Computing* **8**(1), 513–531.

RAVI KANNAN (1987). Minkowski's Convex Body Theorem and Integer Programming. *Mathematics of Operations Research* **12**(3), 415–440.

TALI KAUFMAN, SHACHAR LOVETT & ELY PORAT (2010). Weight Distribution and List-Decoding Size of Reed- Muller Codes. In *Innovations in Computer Science*, 422–433.

EYAL KUSHILEVITZ & YISHAY MANSOUR (1993). Learning Decision Trees Using the Fourier Spectrum. *SICOMP: SIAM Journal on Computing* **22**(6), 1331–1348.

FLORENCE J. MACWILLIAMS & NEIL J. A. SLOANE (1981). *The Theory of Error-Correcting Codes*. Elsevier/North-Holland, Amsterdam.

DANIELE MICCIANCIO (2012). Inapproximability of the Shortest Vector Problem: Toward a Deterministic Reduction. *Theory of Computing* **8**(1), 487–512.

DANIELE MICCIANCIO & SHAFI GOLDWASSER (2002). *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts.

DANIELE MICCIANCIO & ANTONIO NICOLOSI (2008). Efficient Bounded Distance Decoder for Barnes-Wall Lattices. In *IEEE International Symposium on Information Theory*, 2484–2488.

D. E. MULLER (1954). Application of Boolean algebra to switching circuit design and to error detection. *IEEE Transactions on Computers* **3**, 6–12.

GABRIELE NEBE, ERIC M. RAINS & NEIL J. A. SLOANE (2001). The Invariants of the Clifford Groups. *Designs, Codes and Cryptography* **24**(1), 99–122.

FARZAD PARVARESH & ALEXANDER VARDY (2005). Correcting Errors Beyond the Guruswami-Sudan Radius in Polynomial Time. In *Symposium on Foundations of Computer Science*, 285–294. Symposium on Foundations of Computer Science.

RUUD PELLIKAAN & XIN-WEN WU (2004). List decoding of $q$-ary Reed-Muller codes. *IEEE Transactions on Information Theory* **50**(4), 679–682.

XAVIER PUJOL & DAMIEN STEHLÉ (2008). Rigorous and Efficient Short Lattice Vectors Enumeration. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security*, 390–405.

M. RAN & J. SNYDERS (1998). Efficient decoding of the Gosset, Coxeter-Todd and the Barnes-Wall lattices. In *IEEE International Symposium on Information Theory*, 92.

IRVING S. REED (1954). A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory* **4**, 38–49.

AMIR J. SALOMON & OFER AMRANI (2005). Augmented product codes and lattices: Reed-Muller codes and Barnes-Wall lattices. *IEEE Transactions on Information Theory* **51**(11), 3918–3930.

MADHU SUDAN (1997). Decoding of Reed Solomon Codes beyond the Error-Correction Bound. *Journal of Complexity* **13**(1), 180–193.

MADHU SUDAN (2000). List decoding: algorithms and applications. *SIGACT News* **31**(1), 16–27.

MADHU SUDAN (2001). Algorithmic Introduction to Coding Theory, Lecture Notes. Available from `http://people.csail.mit.edu/madhu/FT01/`.

MADHU SUDAN, LUCA TREVISAN & SALIL P. VADHAN (2001). Pseudorandom Generators without the XOR Lemma. *Journal of Computer and System Sciences* **62**(2), 236–266.

AMNON TA-SHMA & DAVID ZUCKERMAN (2001). Extractor codes. In *ACM Symposium on Theory of Computing*, 193–199.

LUCA TREVISAN (2001). Extractors and pseudorandom generators. *Journal of the ACM* **48**(4), 860–879.

JOHN M. WOZENCRAFT (1958). List Decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT* **48**, 90–95.

Manuscript received 23 March 2015

ELENA GRIGORESCU
Purdue University,
West Lafayette, IN
elena-g@purdue.edu.

CHRIS PEIKERT
University of Michigan,
Ann Arbor, MI
cpeikert@umich.edu