Lattice (List) Decoding Near Minkowski's Inequality

Ethan Mook* Chris Peikert†

September 9, 2021

Abstract

Minkowski proved that any n-dimensional lattice of unit determinant has a nonzero vector of Euclidean norm at most $\sqrt{n}$; in fact, there are $2^{\Omega(n)}$ such lattice vectors. Lattices whose minimum distances come close to Minkowski's bound provide excellent sphere packings and error-correcting codes in $\mathbb{R}^n$.

The focus of this work is a certain family of efficiently constructible n-dimensional lattices due to Barnes and Sloane, whose minimum distances are within an $O(\sqrt{\log n})$ factor of Minkowski's bound. Our primary contribution is a polynomial-time algorithm that *list decodes* this family to distances approaching $1/\sqrt{2}$ of the minimum distance. The main technique is to decode Reed-Solomon codes under error measured in the Euclidean norm, using the Koetter-Vardy "soft decision" variant of the Guruswami-Sudan list-decoding algorithm.

1 Introduction

A linear (error-correcting) *code* $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$, and its *minimum distance* $d = d(\mathcal{C})$ is the minimum Hamming weight of its nonzero code words. Such a code is uniquely decodable under error having weight up to $\frac{d-1}{2}$, and for many codes it is known how to perform such decoding efficiently. See, e.g., [GRS19] for extensive background on the combinatorial and algorithmic aspects of codes.

Elias [Eli88] and Wozencraft [Woz58] put forward the idea of decoding under error having weight $d/2$ or more, which can potentially lead to ambiguity. The goal of *list decoding* is to find *all* codewords within a certain distance of a received word, and it is often possible to guarantee that there are not too many. Breakthrough list-decoding algorithms were obtained by Goldreich and Levin [GL89] for the Hadamard code, and by Sudan [Sud97] and Guruswami and Sudan [GS98] for Reed-Solomon codes. These results and others have had countless applications across computer science.

A (full-rank) *lattice* $\Lambda \subset \mathbb{R}^n$ is a discrete additive subgroup whose linear span is $\mathbb{R}^n$, and its minimum distance $\lambda_1 = \lambda_1(\Lambda)$ is the minimum Euclidean norm of its nonzero lattice points. Naturally, a lattice is uniquely decodable under error having norm less than $\lambda_1/2$, and efficient algorithms are known for decoding some lattices up to that distance. For example, for the integer lattice $\mathbb{Z}^n$, which has unit minimum distance, we can simply round each coordinate to the nearest integer.

A main measure of a lattice's "quality," e.g., as a sphere packing or error-correcting code, is its normalized minimum distance $\sqrt{\gamma(\Lambda)} = \lambda_1(\Lambda)/\det(\Lambda)^{1/n}$ (the square root is present for historical reasons), where the

*University of Michigan, emook@umich.edu.

†Computer Science and Engineering, University of Michigan. Email: cpeikert@umich.edu. This material is based upon work supported by the National Science Foundation under Award CCF-2006857. The views expressed are those of the authors and do not necessarily reflect the official policy or position of the National Science Foundation. Part of this work was done while the author was visiting the Simons Institute Spring 2020 program "Lattices: Algorithms, Complexity, and Cryptography."

determinant $\det(\Lambda) = \mathrm{vol}(\mathbb{R}^n/\Lambda)$ is the covolume of the lattice, i.e., the absolute value of the determinant of any $\mathbb{Z}$-basis of Λ. A theorem of Minkowski bounds this quality by $\sqrt{\gamma(\Lambda)} \leq \sqrt{n}$ for any lattice $\Lambda \subseteq \mathbb{R}^n$; in fact, a refined version says that there are an exponential $2^{\Omega(n)}$ number of lattice vectors of norm at most $\sqrt{n} \cdot \det(\Lambda)^{1/n}$. There exist infinite families of lattices whose minimum distances are asymptotically tight with Minkowski's bound (up to a small constant factor), and there are efficiently constructible families that are nearly tight with it (see [CS99] for extensive background, and below for further details).

Lattice (list) decoding. Motivated by the many similarities between codes and lattices, Grigorescu and Peikert [GP12] initiated the study of (efficient) list decoding for lattices, and, building on the unique-decoding algorithm of Micciancio and Nicolosi [MN08], gave an algorithm for the well known Barnes-Wall family of lattices $\mathrm{BW}_k \subset \mathbb{R}^n$, where $n = 2^k$. Their algorithm's running time, and hence list size, is polynomial in n for decoding distances approaching the minimum distance $\lambda_1(\mathrm{BW}_k) = \Theta(\sqrt{n})$ (but at this threshold and beyond, the list size can be super-polynomial). However, Barnes-Wall lattices have only moderately good quality: compared to Minkowski's inequality, they satisfy the much tighter bound $\sqrt{\gamma(\mathrm{BW}_k)} = O(n^{1/4})$. So, the results of [GP12] are quite far from optimal in terms of (determinant-normalized) decoding distance. By contrast, a recent work of Ducas and Pierrot [DP19] gave, for a certain family of lattices, a simple and efficient decoding algorithm for normalized distance $\Theta(\sqrt{n}/\log n)$, which is tight with Minkowski's bound up to an $O(\log n)$ factor. However, their algorithm only performs *unique* (not list) decoding, below half the minimum distance.

In this work we focus on an infinite family of lattices $\Lambda_n \subset \mathbb{R}^n$, originally constructed by Barnes and Sloane [BS83], having normalized minimum distance $\sqrt{\gamma(\Lambda_n)} = \Omega(\sqrt{n/\log n})$. Our main contribution is a polynomial-time list-decoding algorithm for this family, which decodes to distance nearly $\lambda_1(\Lambda_n)/\sqrt{2}$. An immediate corollary is the first (to our knowledge) polynomial-time *unique* decoder for this family, for up to half the minimum distance. In summary, we obtain efficient (list-)decoding algorithms for distances within an $O(\sqrt{\log n})$ factor of the universal barrier implied by Minkowski's theorem (since for any lattice there can be exponentially many lattice points within normalized distance $\sqrt{n}$ of a target point).

Theorem 1.1. *There is an efficiently constructible family of lattices $\Lambda = \Lambda_n \subset \mathbb{R}^n$ having normalized minimum distance $\lambda_1(\Lambda)/\det(\Lambda)^{1/n} = \Omega(\sqrt{n/\log n})$, which for any constant $\varepsilon > 0$ are list decodable to within distance $(1/\sqrt{2} - \varepsilon) \cdot \lambda_1(\Lambda)$ in some $\mathrm{poly}(n)$ time.*

We remark that $\lambda_1(\Lambda)/\sqrt{2}$ is a natural generic barrier for (combinatorially) efficient list decoding on lattices. On the one hand, Rankin's bound implies that for any lattice $\Lambda \subset \mathbb{R}^n$ and target point, there are at most $2n$ lattice points within distance $\lambda_1(\Lambda)/\sqrt{2}$ of the target (this is analogous to Johnson's bound for codes). On the other hand, for any constant $\varepsilon > 0$ there exists a family of "locally dense" lattices $\Lambda_n \subset \mathbb{R}^n$ and target points $\mathbf{t}_n \in \mathbb{R}^n$ for which there are $\exp(n^{\Omega(1)})$ points in Λ_n within distance $(1/\sqrt{2} + \varepsilon) \cdot \lambda_1(\Lambda_n)$ of $\mathbf{t}_n$ [Mic98]. In fact, as shown in [Mic12], the family of lattices from Theorem 1.1 has this property, but with $1/\sqrt{2}$ replaced by $\sqrt{2/3}$. It is an interesting question whether constant factors smaller than $\sqrt{2/3}$ for local density, or larger than $1/\sqrt{2}$ for efficient list decoding, can be obtained for this family.

Finally, we point out that the lattices from Theorem 1.1 also have efficiently constructible *short* bases, consisting of vectors whose norms are within a $\sqrt{n}$ factor of the minimum distance (see Remark 5.3). This can be useful for *encoding*, where one typically wants to map a message to a relatively short lattice point: its norm corresponds to the power required to send it, and one wants to minimize the power-to-noise ratio. By contrast, for the uniquely decodable lattices studied in [DP19], relatively short nonzero lattice vectors are not known, and may even be infeasible to compute, so it is unclear whether power-efficient encoding is possible.

Techniques and organization. The family of lattices from Theorem 1.1 is obtained by applying "Construction D" [CS99] to a certain tower of BCH codes, which are subfield subcodes of certain Reed-Solomon codes. It was shown in [BS83, Mic12] that this family of lattices yields an excellent packing, with normalized minimum distance $\Omega(\sqrt{n/\log n})$. We obtain an efficient list-decoding algorithm for this family as follows.

First, in Section 3 we give an efficient list-decoding algorithm for any prime-subfield subcode $\mathcal{C} = \mathcal{R} \cap \mathbb{F}_p^n$ of a Reed-Solomon code $\mathcal{R} \subseteq \mathbb{F}_{p^r}^n$ (which in particular includes the BCH codes mentioned above), where error is measured in the *Euclidean* (rather than Hamming) norm. More specifically, embedding the code $\mathcal{C} \subseteq \mathbb{F}_p^n$ into $(\mathbb{R}/p\mathbb{Z})^n$ in the natural way, the algorithm recovers all codewords of $\mathcal{C}$ that are within squared Euclidean norm nearly $d/2$ of an arbitrary received word in $(\mathbb{R}/p\mathbb{Z})^n$, where d is the minimum *Hamming* distance of $\mathcal{R}$ (and hence is a lower bound on the minimum distance of $\mathcal{C}$). The core of our algorithm is the Koetter-Vardy [KV03] "soft-decision" list decoder for Reed-Solomon codes, which takes as one of its inputs a *reliability vector* representing the probability of each field element at each position of the received word. However, in our setting there are no probabilities, just a fixed received word in $(\mathbb{R}/p\mathbb{Z})^n$. We use it to define an *implicit* reliability vector, and show that with this vector the Koetter-Vardy algorithm recovers all codewords within the desired Euclidean norm. In fact, using the framework of [KV01] we show that our choice of reliability vector yields an optimal tradeoff between code dimension and Euclidean decoding distance for the analysis of the Koetter-Vardy soft-decision decoder, for squared distances at most $n/4$.

Next, in Section 4 we give a list-decoding algorithm for any Construction D lattice defined by a tower of codes over a prime field $\mathbb{F}_p$, using subroutines that list decode the component codes to appropriate distances (in any ℓ_q norm, including the Euclidean norm). The algorithm naturally arises from the iterative definition of Construction D, and works by recursively recovering each nearby lattice vector from its least- to most-significant digit in base p. (A similar algorithm for *unique* decoding of Construction D lattices was studied in [MKO18], with a focus on simulation in moderate dimensions for BCH codes, but no theorems about its behavior were given.) The depth of the recursion is the number of codes in the tower, and the branching factor at each level of the recursion is the size of the list output by the decoding subroutine at that level.

Finally, in Section 5 we recall the lattice family obtained by instantiating Construction D with a certain tower of BCH codes, and instantiate the subroutines in the list-decoding algorithm from Section 4 with the algorithm from Section 3. Because a tower of $\Theta(\log n)$ codes is needed to obtain the desired density, to obtain a polynomial-time algorithm we need to ensure constant $O(1)$ list sizes for each code in the tower. This allows us list decode the lattice to within a $1/\sqrt{2} - \varepsilon$ factor of its minimum distance in the Euclidean norm, for any constant $\varepsilon > 0$.

Acknowledgments. We thank Daniele Micciancio and the anonymous reviewers for helpful comments on the presentation.

2 Preliminaries

For real vectors $\mathbf{v} = (v_1, \ldots, v_n), \mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n$, define their inner product as $\langle \mathbf{v}, \mathbf{w} \rangle := \sum_{i=1}^n v_i w_i$. For any positive integer p, let $\mathbb{Z}_p := \mathbb{Z}/p\mathbb{Z}$ denote the quotient group of the integers modulo p. When p is prime, we identify $\mathbb{Z}_p$ with the finite field $\mathbb{F}_p$ in the obvious way. Fixing some arbitrary set of representatives of $\mathbb{Z}_p$ (e.g., $\{0, 1, \ldots, p-1\}$), for $v \in \mathbb{Z}_p$ let $\overline{v} \in \mathbb{Z}$ denote its representative, and extend this notation coordinate-wise to vectors over $\mathbb{Z}_p$. Similarly, let $\mathbb{R}_p := \mathbb{R}/p\mathbb{Z}$ be the quotient group of the real numbers modulo the integer multiples of p; then $\mathbb{Z}_p$ is a subgroup of $\mathbb{R}_p$.

For $y \in \mathbb{R}_p$, we define $|y| := \min\{|z| : z \in \mathbb{R} \cap (y + p\mathbb{Z})\}$ to be the minimal absolute value over all real numbers congruent to y (modulo $p\mathbb{Z}$). Equivalently, it is the absolute value of the single element in

$[-p/2, p/2) \cap (y + p\mathbb{Z})$. For $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{R}_p^n$, we define the Euclidean norm

$$\|\mathbf{y}\| := \Big(\sum_{i=1}^{n} |y_i|^2\Big)^{1/2} = \min\{\|\mathbf{z}\| : \mathbf{z} \in \mathbb{R}^n \cap (\mathbf{y} + p\mathbb{Z}^n)\}.$$

Error-correcting codes. For a prime power q and nonnegative integers $k \leq n \leq q$, a *Reed-Solomon code* [RS60] of length n and dimension k over $\mathbb{F}_q$ is the set

$$\mathcal{R}_{\mathbb{F}_q}[n, k] := \{(p(\alpha_1), p(\alpha_2), \ldots, p(\alpha_n)) \in \mathbb{F}_q^n : p(X) \in \mathbb{F}_q[X], \deg(p) < k\}$$

for some fixed distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$. It is easy to see that $\mathcal{R}$ is a linear code (i.e., a linear subspace), and that its minimum Hamming distance $d := \min_{\mathbf{c} \in \mathcal{R}} \mathrm{wt}(\mathbf{c}) = n - k + 1$, because a polynomial $p(X) \in \mathbb{F}_q[X]$ can have at most $\deg(p)$ zeros (and this is attainable when $\deg(p) \leq q$). It is also clear that for any $k \leq k'$ we have $\mathcal{R}[n, k] \subseteq \mathcal{R}[n, k']$. For our purposes it is convenient to define the adjusted rate $R^* = (k-1)/n = 1 - d/n$.

If q is a power of a prime p, making $\mathbb{F}_p$ a subfield of $\mathbb{F}_q$, *BCH codes* [BR60, Hoc59] can be obtained as $\mathbb{F}_p$-subfield subcodes of certain Reed-Solomon codes. More specifically, letting $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$ be the set of evaluation points, the (primitive, narrow-sense) BCH code of designed distance $1 \leq d \leq n$ is defined as

$$\mathcal{C}_{\mathbb{F}_q}[n = q - 1, d] := \mathcal{R}_{\mathbb{F}_q}[n, k = n - d + 1] \cap \mathbb{F}_p^n.$$

Clearly, $\mathcal{C}_{\mathbb{F}_q}[n, d]$ has minimum distance at least d (because it is a subset of a distance-d code). It follows from the nesting property of Reed-Solomon codes that for any $d \leq d'$ we have $\mathcal{C}_{\mathbb{F}_q}[n, d] \supseteq \mathcal{C}_{\mathbb{F}_q}[n, d']$. It is also known that $\mathcal{C}_{\mathbb{F}_q}[n, d]$ is efficiently constructible, in the sense that an $\mathbb{F}_p$-basis for it can be produced in time $\mathrm{poly}(n)$. Finally, its dimension k satisfies the following well known bound (see, e.g., [GRS19, Exercise 5.10]).

Lemma 2.1. *For $1 \leq d \leq n = q-1$, the BCH code $\mathcal{C}_{\mathbb{F}_q}[n, d]$ has codimension $n - k \leq \lceil \frac{p-1}{p}(d-1) \rceil \log_p q$.*

Lattices. A *lattice* Λ is a discrete additive subgroup of $\mathbb{R}^n$. If its linear span is $\mathbb{R}^n$, the lattice is said to be *full rank*; from now on, we limit our attention to such lattices. Any lattice is generated (non-uniquely) as the integer linear combinations of the vectors in a *basis* $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$, as $\Lambda = \{\sum_{i=1}^n z_i \mathbf{b}_i : z_i \in \mathbb{Z}\}$. The *minimum distance* $\lambda_1(\Lambda) := \min_{\mathbf{v} \in \Lambda \setminus \{\mathbf{0}\}} \|\mathbf{v}\|$ is the length of any shortest nonzero lattice vector. The *determinant* $\det(\Lambda) := \mathrm{vol}(\mathbb{R}^n/\Lambda) = |\det(\mathbf{B})|$ is the absolute value of the determinant of any basis of the lattice.

A standard way of measuring the "density" of a lattice is to normalize its minimum distance by its (dimension-adjusted) determinant. More specifically, for a (full-rank) lattice $\Lambda \subset \mathbb{R}^n$, its normalized minimum distance is $\lambda_1(\Lambda)/\det(\Lambda)^{1/n}$. A simple application of a theorem of Minkowski yields the inequality $\lambda_1(\Lambda) \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}$, which is often called Minkowski's bound. This bound is known to be tight up to a constant factor, i.e., there exists an infinite family of n-dimensional lattices Λ_n of unit determinant for which $\lambda_1(\Lambda) = \Omega(\sqrt{n})$.

3 Decoding Reed-Solomon Subfield Subcodes in the Euclidean Norm

In this section we give a list decoder, for error measured in the *Euclidean norm*, for the $\mathbb{F}_p$-subfield subcode $\mathcal{C} = \mathcal{R} \cap \mathbb{F}_p^n$ (for a prime p) of any Reed-Solomon code $\mathcal{R} = \mathcal{R}_{\mathbb{F}_q}[n, k > 1]$, where $q = p^r$ for some $r \geq 1$. (In particular, this includes BCH codes.) More specifically, given a received word in $\mathbb{R}_p^n$, the decoder outputs all codewords in $\mathcal{C}$ that are within Euclidean norm nearly $\sqrt{d/2}$ of the received word, where $d = n - k + 1$ is the minimum Hamming distance of the Reed-Solomon code (and hence a lower bound on the minimum Hamming distance of $\mathcal{C}$).

The heart of our algorithm is the "soft-decision" list-decoding algorithm of Koetter and Vardy [KV03] for Reed-Solomon codes, which uses data about the *likelihood* of each alphabet symbol in each position. More precisely, to decode a length-n Reed-Solomon code over $\mathbb{F}_q$, their algorithm takes as one of its inputs a so-called *reliability vector* $\Pi \in [0,1]^{qn}$. Such a vector consists of n length-q blocks, where the jth entry of the ith block represents the probability that the transmitted codeword had the jth element of $\mathbb{F}_q$ in its ith coordinate. That is, each of the length-q blocks in Π is a probability mass function, and in particular has unit ℓ_1 norm.

In our setting, we have no explicit probabilities of transmitted symbols, only a received word $\mathbf{y} \in \mathbb{R}_p^n$. In Section 3.1 we define a mapping that converts the received word to an *implicit* reliability vector, which we provide to the Koetter-Vardy soft decoder. As we show in Section 3.2, this choice of reliability vector allows the decoder to find all codewords within a desired Euclidean distance of the received vector.[1] Moreover, in Section 3.3 we show that our choice of reliability vector is essentially optimal for the range of decoding distances that are relevant to this work (and even somewhat beyond).

3.1 Reliability Vectors

We now define the mapping from received words to reliability vectors. For $c \in \mathbb{F}_q \setminus \mathbb{F}_p$ define $[c] = \mathbf{0} \in [0,1]^p$, and for $c \in \mathbb{F}_p$ define $[c] \in \{0,1\}^p$ to be the indicator vector of c, i.e., the entry indexed by c is 1 and all other entries are 0. We extend this definition to $\mathbb{R}_p$ by mapping each interval $(c, c+1) \subseteq \mathbb{R}_p$ to the open line segment $\{(1-t)[c] + t[c+1] : t \in (0,1)\} \subseteq [0,1]^p$ in the natural way. In other words, for $y \in [c, c+1]$ we define

$$[y] := [c] + ([c+1] - [c]) \cdot |y - c|. \tag{3.1}$$

We extend the notation $[\cdot]$ to vectors by applying it entry-wise, mapping n-dimensional vectors to $[0,1]^{pn}$.[2] Following the terminology of [KV03], we call $[\mathbf{y}] \in [0,1]^{pn}$ the *reliability vector* of a received word $\mathbf{y} \in \mathbb{R}_p^n$.

Lemma 3.1. *For any $\mathbf{y} \in \mathbb{R}_p^n$ and $\mathbf{c} \in \mathbb{Z}_p^n$, we have*

$$\|[\mathbf{y}] - [\mathbf{c}]\|^2 \leq 2\|\mathbf{y} - \mathbf{c}\|^2. \tag{3.2}$$

Proof. It suffices to prove the lemma for $n = 1$ because both sides split as sums over their components. Let $y \in \mathbb{R}_p$ and $c \in \mathbb{Z}_p$. If $|y - c| \geq 1$, then the nonzero entries of $[y], [c] \in [0,1]^p$ are in distinct positions, hence $\|[y] - [c]\|^2 = \|[y]\|^2 + \|[c]\|^2 \leq \|[y]\|^2 + 1 \leq 2$. Otherwise $|y - c| < 1$; assume that $y \in [c, c+1]$. Then by Equation (3.1),

$$\|[y] - [c]\|^2 = \|[c+1] - [c]\|^2 \cdot |y - c|^2 = 2|y - c|^2. \tag{3.3}$$

The case $y \in [c-1, c]$ proceeds symmetrically. □

[1] It seems likely that this approach generalizes somewhat to Reed-Solomon codes themselves (over prime-power fields). We restrict our attention to prime-subfield subcodes because they admit a natural Euclidean norm, and they are required for defining Construction D lattices (see Section 4).

[2] This can equivalently be viewed as outputting a matrix in $[0,1]^{p \times n}$, which is the perspective used more in [KV03], but the vector view will be more natural for us.

3.2 Decoding Algorithm

Here we define and analyze our decoding algorithm, presented in Algorithm 1. There and in what follows, we let $\mathrm{KV}(\Pi, S)$ denote the Koetter-Vardy soft-decision decoder for $\mathcal{R}$, run on input reliability vector $\Pi \in [0,1]^{pn}$ with output list size limited to S.[3]

Although we use the Koetter-Vardy algorithm essentially as a 'black box,' we include the following high-level description of its operation for completeness. First, it defines a 'multiplicity vector' $M = \lfloor \lambda \Pi \rfloor \in \mathbb{Z}^{pn}$ for some suitably large scaling factor $\lambda \in \mathbb{R}^+$, which is determined based on the desired list size bound S. It then proceeds by a generalization of the list-decoding algorithm of Guruswami and Sudan [GS98]. More specifically, it uses M to set up a system of linear equations, which it solves to compute the minimal bivariate polynomial $\mathcal{Q}_M(X, Y)$ having zeroes with multiplicities given by M at specified points. Finally, it (partially) factors $\mathcal{Q}_M$ to identify all the factors of the form $Y - f(X)$, which directly correspond to the output list of codewords. The running time of the algorithm is primarily determined by the number of equations in the linear system, which is asymptotically the sum of the squares of the entries of M.

Algorithm 1 List-decoding algorithm for code $\mathcal{C}$, for the Euclidean norm

---

Input: Received word $\mathbf{y} \in \mathbb{R}_p^n$ and $\varepsilon > 0$.
Output: A list of the codewords $\mathbf{c} \in \mathcal{C}$ for which $\|\mathbf{y} - \mathbf{c}\|^2 \leq (1-\varepsilon)d/2$.

1. Let $L = \mathrm{KV}([\mathbf{y}], S) \subseteq \mathcal{R}$ be the output list of the soft-decision decoder of $\mathcal{R}$ on reliability vector $[\mathbf{y}]$, with list size limited to
$$S := \frac{\frac{1}{R^*} + \frac{1}{\sqrt{2R^*}}}{1 - \sqrt{\frac{R^*}{\varepsilon + (1-\varepsilon)R^*}}}. \tag{3.4}$$
2. Output $\{\mathbf{c} \in L \cap \mathbb{F}_p^n : \|\mathbf{y} - \mathbf{c}\|^2 \leq (1-\varepsilon)d/2\}$.

Theorem 3.2 (Adapted from [KV03, Theorem 17].). *Let $\mathbf{y} \in \mathbb{R}_p^n$ and $S > 0$. The soft-decision decoder $\mathrm{KV}([\mathbf{y}], S)$ for $\mathcal{R}$ outputs a list of all the at most S codewords $\mathbf{c} \in \mathcal{R}$ such that*

$$\frac{\langle [\mathbf{y}], [\mathbf{c}] \rangle}{\|[\mathbf{y}]\|} \geq \frac{\sqrt{k-1}}{1 - \frac{1}{S}\left(\frac{1}{R^*} + \frac{1}{\sqrt{2R^*}}\right)}. \tag{3.5}$$

Additionally, the algorithm runs in time polynomial in n, $\log q$ and S.

Proof. The proof is identical to that of [KV03, Theorem 17], but using the fact that both the numerator and denominator of $\langle [\mathbf{y}], [\mathbf{c}] \rangle / \|[\mathbf{y}]\|$ are unchanged when using $[\mathbf{c}] \in [0,1]^{qn}$ (instead of $[0,1]^{pn}$) as defined in [KV03] with the appropriate (zero-padded) $[\mathbf{y}] \in [0,1]^{qn}$, and replacing the inequality $\|[\mathbf{y}]\|^2 \geq n/q$ with $\|[\mathbf{y}]\|^2 \geq n/2$. The latter inequality holds because each block of $[\mathbf{y}]$ has unit ℓ_1 norm, but has at most two nonzero entries.

The claim on the running time follows from the fact that the algorithm runs in time polynomial in n, $\log q$, and the "cost" of the employed multiplicity matrix, which is shown in [KV03, Lemma 15] to be polynomial in n and S. $\square$

[3] To be completely accurate, the KV algorithm as defined in [KV03] takes a reliability vector in $[0,1]^{qn}$; we can either appropriately pad the $[0,1]^{pn}$-vector with zeros, or, to be more efficient, modify the algorithm to work directly with $[0,1]^{pn}$-vectors in the obvious way.

The following immediate corollary gives a more geometric sufficient condition for a subfield subcode word to be recovered.

Corollary 3.3. *Under the same setup as in Theorem 3.2, a subfield subcode word* $\mathbf{c} \in \mathcal{C} = \mathcal{R} \cap \mathbb{F}_p^n$ *will be in the list output by* $\mathrm{KV}([\mathbf{y}], S)$ *if the angle* β *between the vectors* $[\mathbf{y}], [\mathbf{c}] \in [0,1]^{pn}$ *satisfies*

$$\cos\beta \geq \frac{\sqrt{R^*}}{1 - \frac{1}{S}\left(\frac{1}{R^*} + \frac{1}{\sqrt{2R^*}}\right)}. \tag{3.6}$$

Proof. Follows from Theorem 3.2 and the identity $\langle [\mathbf{y}], [\mathbf{c}] \rangle = \|[\mathbf{y}]\| \cdot \|[\mathbf{c}]\| \cos\beta = \|[\mathbf{y}]\| \cdot \sqrt{n}\cos\beta$, where $\|[\mathbf{c}]\| = \sqrt{n}$ because $\mathbf{c} \in \mathbb{F}_p^n$. □

Theorem 3.4. *Algorithm 1 is correct. More specifically, given input* $\mathbf{y} \in \mathbb{R}_p^n$ *and* $\varepsilon > 0$*, it outputs a list of exactly those codewords* $\mathbf{c} \in \mathcal{C}$ *for which* $\|\mathbf{y} - \mathbf{c}\|^2 \leq (1-\varepsilon)d/2$*, in time polynomial in* n*,* $\log q$*, and* $1/\varepsilon$*.*

Proof. The running time follows from the fact that S is polynomial in n and $1/\varepsilon$ by Equation (3.4), and by Theorem 3.2.

We now prove correctness. By the final step, the algorithm outputs only codewords $\mathbf{c} \in \mathcal{C}$ for which $\|\mathbf{y} - \mathbf{c}\|^2 \leq (1-\varepsilon)d/2$. Now letting $\mathbf{c}$ be any such a codeword, we show that it appears in $\mathrm{KV}([\mathbf{y}], S)$, and hence in the output of the overall algorithm. From Lemma 3.1 and the fact that $d = (1-R^*)n$, we have

$$\|[\mathbf{y}] - [\mathbf{c}]\|^2 \leq 2\|\mathbf{y} - \mathbf{c}\|^2 \leq (1-\varepsilon)(1-R^*)n. \tag{3.7}$$

Let β be the angle between $[\mathbf{y}]$ and $[\mathbf{c}]$, which has non-negative cosine because both vectors have only non-negative entries. Since $\|[\mathbf{c}]\|^2 = n$ (because $\mathbf{c} \in \mathbb{F}_p^n$), the squared distance from $[\mathbf{c}]$ to the line passing through the origin and $[\mathbf{y}]$ is $n\sin^2\beta$, so

$$n\sin^2\beta \leq \|[\mathbf{y}] - [\mathbf{c}]\|^2 \tag{3.8}$$

and hence $\sin^2\beta \leq (1-\varepsilon)(1-R^*)$. Therefore,

$$\cos\beta = \sqrt{1 - \sin^2\beta} \geq \sqrt{\varepsilon + (1-\varepsilon)R^*}. \tag{3.9}$$

Finally, based on our choice of S, a straightforward algebraic manipulation yields

$$\sqrt{\varepsilon + (1-\varepsilon)R^*} = \frac{\sqrt{R^*}}{1 - \frac{1}{S}\left(\frac{1}{R^*} + \frac{1}{\sqrt{2R^*}}\right)}, \tag{3.10}$$

so $\cos\beta$ satisfies Equation (3.6), and invoking Corollary 3.3 completes the proof. □

We conclude this subsection by noting that (list) decoding a linear code $\mathcal{C} \subseteq \mathbb{F}_p^n$ with error measured in the Euclidean norm is syntactically very similar, but not quite identical, to (list) decoding the Construction A lattice $\Lambda = \overline{\mathcal{C}} + p\mathbb{Z}^n \subseteq \mathbb{R}^n$. Indeed, when p exceeds twice the decoding distance, these tasks are equivalent (and hence Algorithm 1 can efficiently solve the latter) because by the triangle inequality, any given coset of $p\mathbb{Z}^n$ can have at most one lattice point that is within the decoding distance of a given received word. However, when p is significantly smaller than d (as it is in Section 5), there may be a huge number of different lattice vectors in a given coset of $p\mathbb{Z}^n$ that are within the decoding distance of the received word. So even if it is possible to decode the code efficiently, it may be (combinatorially) infeasible to decode its Construction A lattice.

3.3 Optimality of Our Reliability Vector

Here we show that our choice of reliability vector $[\mathbf{y}] \in [0,1]^{pn}$ as a function of the received word $\mathbf{y} \in \mathbb{R}_p^n$ (as used in Algorithm 1) yields, for the Koetter-Vardy polynomial-time soft-decision decoder, an essentially optimal tradeoff between the squared Euclidean decoding distance (up to $n/4$) and the adjusted rate R^* of the Reed-Solomon code. (The material in this section is not needed for anything else in the paper.)

As shown in [KV01, Section IV] (see Equation (12)), the Koetter-Vardy algorithm is guaranteed to decode a received word $\mathbf{y} \in \mathbb{R}_p^n$ to some desired distance if it is given a reliability vector $W \in [0,1]^{pn}$ such that

$$\min_{\mathbf{c}} \frac{\langle [\mathbf{c}], W \rangle}{\|W\|} > \sqrt{k-1} = \sqrt{nR^*}, \tag{3.11}$$

where the minimum is taken over all $\mathbf{c} \in \mathbb{Z}_p^n$ within the desired distance of $\mathbf{y}$. So, for provable decoding with this algorithm, the code's adjusted rate R^* is bounded by the minimum (over the choice of $\mathbf{y}$) of the maximum (over the choice of W) left-hand side of Equation (3.11).

In what follows we will show that for any squared Euclidean decoding distance δn with $\delta \leq 1/4$, there is a particular received word $\mathbf{y} \in \mathbb{R}_p^n$ for which our choice of reliability vector $W = [\mathbf{y}]$ maximizes the left-hand side of Equation (3.11), and bounds the adjusted rate by $R^* < 1 - 2\delta$. Therefore, according to the best available analysis, the Koetter-Vardy algorithm is limited to decoding to within squared distance $\delta n < n(1-R^*)/2 = d/2$. Because our reliability vector allows for decoding to squared distance $(1-\varepsilon)d/2$ for any positive constant (or even inverse polynomial) ε, it is therefore an essentially optimal choice.

The analysis. Fix some $\delta \in (0, 1/4]$ and let $\beta \in (0, 1/2]$ be the unique solution to $\beta(1-\beta) = \delta$. Then let $\mathbf{y} = (\beta, \ldots, \beta) \bmod p$ be the received word in $\mathbb{R}_p^n$ where each entry is $\beta \bmod p$. Also let $\Delta \in \mathbb{R}^p$ be defined by $\Delta_\alpha = |\alpha - \beta|^2$ for each $\alpha \in \mathbb{Z}_p$, i.e., the squared Euclidean distance between α and β (modulo p). We proceed by showing that, for this received word $\mathbf{y}$, taking $W = [\mathbf{y}] \in [0,1]^{pn}$ maximizes the left-hand side of Equation (3.11) and makes it equal $\sqrt{n(1-2\delta)}$, hence $R^* < 1 - 2\delta$.

Because $\mathbf{y}$ is the all-βs vector, to maximize the left-hand size of Equation (3.11), without loss of generality we can take W to be made up of n identical blocks $w \in [0,1]^p$. For any word $\mathbf{c} \in \mathbb{Z}_p^n$, the distance $\|\mathbf{y} - \mathbf{c}\|$ is entirely determined by the frequencies with which the various $\alpha \in \mathbb{Z}_p$ appear in $\mathbf{c}$. More specifically, define a vector $T \in [0,1]^p$ by $T_\alpha = \frac{1}{n}\sum_{i=1}^n [\mathbf{c}_i = \alpha]$, the fraction of entries that are α in $\mathbf{c}$. Then we have $\frac{1}{n}\|\mathbf{y} - \mathbf{c}\|^2 = \langle T, \Delta \rangle$. Next, define the set

$$B(\delta) = \{T \in [0,1]^p : \langle T, \Delta \rangle \leq \delta, \sum_\alpha T_\alpha = 1, T_\alpha \geq 0 \ \forall \alpha\}. \tag{3.12}$$

In [KV01] it is shown (in a more general form) that taking any $w \in \arg\min_{T \in B(\delta)} \langle T, T \rangle$ maximizes the left-hand side of Equation (3.11).

Lemma 3.5. *For any $\delta \in (0, 1/4]$ and the unique $\beta \in (0, 1/2]$ satisfying $\beta(1-\beta) = \delta$, the reliability vector $[\beta] \in [0,1]^p$ as defined in Section 3.1 is the unique element of* $\arg\min_{T \in B(\delta)} \langle T, T \rangle$.

Proof. Because we are dealing with a convex optimization problem with strictly convex objective function $\langle T, T \rangle$, any local minimizer is the unique global minimizer, so it suffices to show that $[\beta]$ is the former. Additionally, because the constraints defining $B(\delta)$ are affine, a vector $T \in B(\delta)$ is a local minimizer if and only if it satisfies the Karush-Kuhn-Tucker conditions

$$\forall \alpha,\ 2T_\alpha + \mu\Delta_\alpha - \mu_\alpha + \lambda = 0, \tag{3.13}$$

$$\mu(\langle T, \Delta \rangle - \delta) = 0, \tag{3.14}$$

$$\forall \alpha,\ \mu_\alpha T_\alpha = 0. \tag{3.15}$$

for some real constants $\mu, \mu_\alpha \geq 0$ and (unrestricted) λ.

We begin by showing that $[\beta] \in B(\delta)$. Recall from Section 3.1 that $[\beta]$ has two nonzero entries, $[\beta]_0 = 1 - \beta$ and $[\beta]_1 = \beta$, so it is clear that $[\beta]$ is non-negative with entries that sum to one, and it remains to check that $\langle [\beta], \Delta \rangle \leq \delta$. Because $\beta \in (0, 1/2]$, we have $\Delta_0 = \beta^2$ and $\Delta_1 = (1-\beta)^2$, and thus

$$\langle [\beta], \Delta \rangle = (1-\beta) \cdot \beta^2 + \beta \cdot (1-\beta)^2 = \beta(1-\beta) = \delta.$$

Moreover, this shows that Equation (3.14) is satisfied for $T = [\beta]$ and any $\mu \geq 0$.

Finally, we find $\mu, \mu_\alpha \geq 0$ and $\lambda \in \mathbb{R}$ for which $T = [\beta]$ satisfies Equations (3.13) and (3.15). Because $[\beta]_0$ and $[\beta]_1$ are the only nonzero entries of $[\beta]$, Equation (3.15) says that we must take $\mu_0 = \mu_1 = 0$, but $\mu_\alpha \geq 0$ is unrestricted for $\alpha \notin \{0, 1\}$. We solve the system given by Equation (3.13) to get $\mu = 2 \geq 0$, $\lambda = -2(\beta^2 - \beta + 1)$, and for all $\alpha \notin \{0, 1\}$,

$$\mu_\alpha = \mu \Delta_\alpha + \lambda = 2(\Delta_\alpha - (\beta^2 - \beta + 1)),$$

which is non-negative because $\Delta_\alpha = |\alpha - \beta|^2 \geq 1$ and $\beta^2 - \beta \leq 0$ for $\beta \in [0, 1/2]$. □

4 Decoding Construction D Lattices

In this section we give a list-decoding algorithm for Construction D lattices. First we recall the definition of Construction D. The definition we use here is very similar to the one from [CS99, Section 8.8.1], with the only differences being that we scale so that the lattice is integral (see Remark 4.4), and we generalize in the obvious way to codes over $\mathbb{F}_p$ for any prime p.

Definition 4.1 (Construction D). Let $\mathbb{F}_p^n = \mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots \supseteq \mathcal{C}_\ell$ be a tower of length-n linear codes where $\mathcal{C}_i$ has dimension k_i for $i = 0, \ldots, \ell$. Choose a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of $\mathbb{F}_p^n$ such that (i) $\mathbf{b}_1, \ldots, \mathbf{b}_{k_i}$ form a basis of $\mathcal{C}_i$ for $i = 0, \ldots, \ell$, and (ii) some permutation of the (row) vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ forms a upper-triangular matrix.[4] Define a set of distinguished $\mathbb{Z}^n$-representatives for $\mathcal{C}_i$ as follows: for any $\mathbf{c} \in \mathcal{C}_i$, write it uniquely as $\mathbf{c} = \sum_{j=1}^{k_i} a_j \mathbf{b}_j$ for some $a_j \in \mathbb{F}_p$, and define its representative $\tilde{\mathbf{c}} := \sum_{j=1}^{k_i} \overline{a}_j \overline{\mathbf{b}}_j \in \mathbb{Z}^n$.

Define $\Lambda_0 = \mathbb{Z}^n$, and for each $i = 1, \ldots, \ell$ define the integer lattice

$$\Lambda_i := \tilde{\mathcal{C}}_i + p\Lambda_{i-1}. \tag{4.1}$$

The Construction D lattice for the full tower $\{\mathcal{C}_i\}$ is $\Lambda = \Lambda_\ell$.

In Theorem 5.1 below (see also Remark 5.2) we recall various important properties of Construction D lattices, which we use there to obtain our main results.

Remark 4.2. Observe that any vector $\mathbf{v}_i \in \Lambda_i$ can be written uniquely as $\mathbf{v}_i = \tilde{\mathbf{c}} + p\mathbf{v}_{i-1}$ for some $\mathbf{c} \in \mathcal{C}_i$ and $\mathbf{v}_{i-1} \in \Lambda_{i-1}$. This is simply because if $\tilde{\mathbf{c}} + p\mathbf{v}_{i-1} = \tilde{\mathbf{c}}' + p\mathbf{v}'_{i-1}$ for some $\mathbf{c}, \mathbf{c}' \in \mathcal{C}_i$ and $\mathbf{v}_{i-1}, \mathbf{v}'_{i-1} \in \Lambda_{i-1}$, then by reducing modulo $p\mathbb{Z}^n$, we have $\mathbf{c} = \mathbf{c}'$ and hence $\mathbf{v}_{i-1} = \mathbf{v}'_{i-1}$ as well.

Remark 4.3. Because the set of representatives $\tilde{\mathcal{C}}_i$ depends on the choice of basis, so do the above lattices Λ_i. For $\mathbf{c} = \sum_{j=1}^{k_j} a_j \mathbf{b}_j \in \mathcal{C}_i$, $\overline{\mathbf{c}}$ may differ from $\tilde{\mathbf{c}}$ because the addition in $\mathbb{Z}^n$ does not "wrap around" as it does in $\mathbb{F}_p^n$, and their difference may not be a (scaled) lattice vector in $p\Lambda_{i-1}$.

[4] Without loss of generality, such a basis can be obtained by starting with an arbitrary basis of $\mathcal{C}_\ell$, extending it to a basis of $\mathcal{C}_{\ell-1}$, then extending that basis to one of $\mathcal{C}_{\ell-2}$, and so on through $\mathcal{C}_0 = \mathbb{F}_p^n$. Finally, perform Gaussian elimination on the resulting basis of $\mathbb{F}_p^n$ so that some permutation of the vectors forms an upper-triangular matrix. See [Mic12, Section 4] for full details.

As a concrete example, let $\mathcal{C}_0 = \mathbb{F}_3^2$ and let $\mathcal{C}_1 = \mathcal{C}_2$ be the code generated by the vector $\mathbf{b}_1 = (1, 2) \in \mathbb{F}_3^2$.[5] Now let $\Lambda_0, \Lambda_1, \Lambda_2$ be the lattices obtained via Construction D for this tower using basis $\mathbf{b}_1, \mathbf{b}_2 = (0, 1) \in \mathbb{F}_3^2$, and let $\Lambda'_0, \Lambda'_1, \Lambda'_2$ be obtained instead using the basis $\mathbf{b}'_1 = (2, 1), \mathbf{b}_2 \in \mathbb{F}_3^2$. Then we have $\overline{\mathbf{b}}'_1 \in \Lambda'_2 \subseteq \mathbb{Z}^2$ by construction (Equation (4.1)). However, $\overline{\mathbf{b}}'_1 = (2, 1) \notin \Lambda_2$ because $\tilde{\mathbf{b}}'_1 = 2 \cdot \overline{\mathbf{b}}_1 = 2 \cdot (1, 2) = (2, 4) \in \Lambda_2$ (where we have used the representative $\overline{\mathbf{b}}_1 \in \mathbb{Z}^2$ to define $\tilde{\mathbf{b}}'_1$, as required by the construction of Λ_2), but $\tilde{\mathbf{b}}'_1 - \overline{\mathbf{b}}'_1 = (0, 3) \notin 3\Lambda_1$ because $(0, 1) \notin \mathcal{C}_1$.

Remark 4.4. Let $\Lambda = \Lambda_\ell$ be the lattice obtained via Definition 4.1 and let Λ' be the lattice obtained via [CS99, Section 8.8.1], for the same tower of (binary) codes. Then $\Lambda = 2^{\ell-1}\Lambda'$. To see this, unwind Definition 4.1 to see that Λ consists of all vectors of the form

$$\mathbf{z} + \sum_{i=1}^{\ell} \sum_{j=1}^{k_i} 2^{\ell-i} \overline{a}_j^{(i)} \overline{\mathbf{b}}_j$$

where $\mathbf{z} \in 2^\ell \mathbb{Z}^n$ and $a_j^{(i)} \in \mathbb{F}_2$, which is clearly equivalent to $2^{\ell-1}\Lambda'$.

Remark 4.5. Using any basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of $\mathbb{F}_p^n$ meeting the conditions from Definition 4.1, we can efficiently construct a basis consisting of relatively short vectors for the associated Construction D lattice Λ. Defining the representatives $\overline{\mathbf{b}}_j$ to have small entries (e.g., in $\{0, \ldots, p-1\}$), there is a basis of Λ consisting of vectors $p^{i_j} \overline{\mathbf{b}}_j$ for various $i_j \in \{0, \ldots, \ell\}$. (See, e.g., the proof of [Mic12, Theorem 4.2].) Therefore, the vectors in this basis have Euclidean norm at most $(p-1)p^\ell \sqrt{n}$. For suitable towers of codes $\mathcal{C}_i$, this bound is not much more than the minimum distance of Λ; see Remark 5.3.

The recursive form of Definition 4.1 naturally leads to a recursive (list) decoder, given in Algorithm 2 below, which relies on (list-)decoding subroutines that find *all* $\mathbf{c}_i \in \mathcal{C}_i$ that are sufficiently close to an (appropriately updated) received word at each stage. More precisely, for each $i = 0, \ldots, \ell$ let $\mathcal{D}_i$ be a (list) decoder for the code $\mathcal{C}_i$ to distance $e_i := p^i e_0$ (for some $e_0 > 0$) in some desired ℓ_q norm $\|\cdot\|$, e.g., the Euclidean norm.

Algorithm 2 List-decoding algorithm $\mathcal{L}(\mathbf{y}, i)$ for the lattices Λ_i

---

Input: Received word $\mathbf{y} \in \mathbb{R}^n$ and integer $i \in \{0, \ldots, \ell\}$.
Output: A list of the lattice vectors $\mathbf{v} \in \Lambda_i$ for which $\|\mathbf{v} - \mathbf{y}\| \le e_i$.

1. Let $L = \mathcal{D}_i(\mathbf{w}) \subseteq \mathcal{C}_i$ where $\mathbf{w} = \mathbf{y} \bmod p\mathbb{Z}^n$.
2. For each $\mathbf{c} \in L$:
   (a) If $i = 0$, let $R_{\mathbf{c}} = \{\tilde{\mathbf{c}} + p\lfloor(\mathbf{y} - \tilde{\mathbf{c}})/p\rceil\} \subseteq \mathbb{Z}^n$.
   (Alternatively, let $R_{\mathbf{c}}$ be the list of all elements of $\tilde{\mathbf{c}} + p\mathbb{Z}^n$ that are sufficiently close to $\mathbf{y}$.)
   (b) Otherwise, let $R_{\mathbf{c}} = \{\tilde{\mathbf{c}} + p\mathbf{v} : \mathbf{v} \in \mathcal{L}((\mathbf{y} - \tilde{\mathbf{c}})/p, i-1)\} \subseteq \mathbb{Z}^n$.
3. Output $\bigcup_{\mathbf{c} \in L} R_{\mathbf{c}}$.

---

[5] One can also construct a similar example over $\mathbb{F}_2$, but in higher dimension.

Theorem 4.6. *For any $e_0 < p/2$, Algorithm 2 is correct: given any $\mathbf{y} \in \mathbb{R}^n$ and $i \in \{0, \dots, \ell\}$, it outputs a list of exactly those $\mathbf{v} \in \Lambda_i$ for which $\|\mathbf{y} - \mathbf{v}\| \le e_i = p^i e_0$.*

Proof. We proceed by induction on i. Starting with the base case $i = 0$, by assumption, $D_0(\mathbf{w})$ outputs a list L of exactly those $\mathbf{c} \in \mathbb{F}_p^n = \mathbb{Z}_p^n$ for which $\|\mathbf{w} - \mathbf{c}\| \le e_0$. For each $\mathbf{c} \in L$, there is a unique element $\mathbf{v} \in \tilde{\mathbf{c}} + p\mathbb{Z}^n$ for which $\|\mathbf{y} - \mathbf{v}\| \le e_0$, because $e_0 < p/2$ and by the triangle inequality. Indeed, $\mathbf{v} = \tilde{\mathbf{c}} + p\lfloor(\mathbf{y} - \tilde{\mathbf{c}})/p\rceil$ is that unique element, because it minimizes the magnitude of each coordinate of $\mathbf{y} - \mathbf{v}$. Therefore, $\mathcal{L}(\mathbf{y}, 0)$ outputs a list of exactly those $\mathbf{v} \in \Lambda_0$ for which $\|\mathbf{y} - \mathbf{v}\| \le e_0$, as claimed.

Next, assume by induction that the algorithm correctly list decodes Λ_{i-1} to distance e_{i-1} for some $i \ge 1$. First, it is clear that $\mathcal{L}(\mathbf{y}, i)$ outputs *only* vectors within distance e_i of $\mathbf{y}$: since $\mathcal{L}((\mathbf{y} - \tilde{\mathbf{c}})/p, i-1)$ outputs only vectors $\mathbf{v}$ for which $\|(\mathbf{y} - \tilde{\mathbf{c}})/p - \mathbf{v}\| \le e_{i-1}$ by assumption, we have $\|\mathbf{y} - (\tilde{\mathbf{c}} + p\mathbf{v})\| \le pe_{i-1} = e_i$ for all the vectors $\tilde{\mathbf{c}} + p\mathbf{v} \in R_{\mathbf{c}}$, as needed.

Finally, we show that $\mathcal{L}(\mathbf{y}, i)$ outputs a list containing *all* $\mathbf{v} \in \Lambda_i$ for which $\|\mathbf{y} - \mathbf{v}\| \le e_i$. Let $\mathbf{v}$ be such a vector. By Remark 4.2 we can uniquely write $\mathbf{v} = \tilde{\mathbf{c}} + p\mathbf{v}_{i-1}$ for some $\mathbf{c} \in \mathcal{C}_i$ and $\mathbf{v}_{i-1} \in \Lambda_{i-1}$. Then, because $\|\mathbf{y} - \mathbf{v}\| \le e_i$, we also have $\|\mathbf{w} - \mathbf{c}\| \le e_i$, so $\mathbf{c}$ appears in the list output by $\mathcal{D}_i(\mathbf{w})$. Finally, we have that

$$\|(\mathbf{y} - \tilde{\mathbf{c}})/p - \mathbf{v}_{i-1}\| = \|(\mathbf{y} - (\tilde{\mathbf{c}} + p\mathbf{v}_{i-1}))/p\| \le e_i/p = e_{i-1}, \tag{4.2}$$

so $\mathbf{v}_{i-1}$ appears in the list output by $\mathcal{L}((\mathbf{y} - \tilde{\mathbf{c}})/p, i-1)$. Thus, $\mathbf{v}$ appears in the list output by $\mathcal{L}(\mathbf{y}, i)$, as needed. $\square$

Theorem 4.7. *Let $S_i, T_i \ge 1$ be upper bounds on the output list size and running time, respectively, of the decoder $\mathcal{D}_i$, and let $S = \max_i S_i$ and $T = \max_i T_i$. Then the running time $R(i)$ of $\mathcal{L}(\cdot, i)$ satisfies*

$$R(i) \le (i+1)(T+K) \prod_{j=1}^{i} S_j \le (i+1)(T+K) \cdot S^i \tag{4.3}$$

for some $K = \mathrm{poly}(S, n, \log p)$.

Proof. The execution of $\mathcal{L}(\mathbf{y}, i)$ consists of one call to $\mathcal{D}_i$, at most S_i calls to $\mathcal{L}(\cdot, i-1)$, and some $K = \mathrm{poly}(S, n, \log p)$ additional work. So $R(i)$ satisfies the recurrence

$$R(i) \le S_i \cdot R(i-1) + (T_i + K) \tag{4.4}$$

with the initial value $R(0) = T_0 + K$. The bound from Equation (4.3) follows immediately by unwinding this recurrence. $\square$

5 Decoding Near Minkowski's Bound

In this section we recall a certain family of n-dimensional Construction D lattices whose (normalized) minimum distances are within an $O(\sqrt{\log n})$ factor of optimal, and instantiate Algorithm 2 to efficiently list decode those lattices to within a $1/\sqrt{2} - \varepsilon$ factor of the minimum distance, for any constant $\varepsilon > 0$.

Throughout this section, for simplicity we restrict our focus to codes over characteristic-two fields. We can get similar results for larger characteristic $p > 2$ by generalizing Theorem 5.1 in the natural way. However, because BCH codes for larger characteristic have a weaker codimension bound (see Lemma 2.1), the corresponding family of lattices admit a weaker bound on their normalized minimum distances.

Theorem 5.1 ([CS99, Chapter 8, Theorem 13, rescaled]). *Let $\mathbb{F}_2^n = \mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots \supseteq \mathcal{C}_\ell$ be a tower of length-n binary linear codes where $\mathcal{C}_i$ has dimension k_i and minimum Hamming distance $d_i \geq 4^i$ for $i = 0, \ldots, \ell$. The Construction D lattice $\Lambda = \Lambda_\ell$ for the tower $\{\mathcal{C}_i\}$ has Euclidean minimum distance*[6] *$\lambda_1(\Lambda) = 2^\ell$ and determinant*

$$\det(\Lambda) = 2^{n\ell - \sum_{i=1}^{\ell} k_i} = 2^{\sum_{i=1}^{\ell}(n-k_i)}. \tag{5.1}$$

Remark 5.2. The determinant of the Construction D lattice Λ follows directly from the fact that the (row) vectors $\overline{\mathbf{b}}_i$ can be permuted to form an upper-triangular matrix, and scaling them by appropriate powers of two yields a basis of Λ.

As intuition (but not a proof) for the minimum distance of Λ, this arises from the fact that the minimum Hamming distance of the codes increases by a factor of four at each level. This means that the minimum Euclidean norm of the vectors in the representative sets $\tilde{\mathcal{C}}_i$ increases by a factor of two at each level, because for integer vectors, the Euclidean norm is at least the square root of the Hamming weight. These varying Euclidean minimum distances are then equalized by scaling the layers Λ_i by corresponding powers of two.

Remark 5.3. Using Remark 4.5, we can obtain a basis of Λ whose vectors have Euclidean norm at most $2^\ell\sqrt{n}$. For a tower of codes $\mathcal{C}_i$ meeting the conditions from Theorem 5.1, the minimum distance of Λ is 2^ℓ. Therefore, the lengths of the basis vectors are within a $\sqrt{n}$ factor of optimal. We can use this basis to generate many more relatively short lattice vectors, by taking small integer linear combinations. This can be used for encoding messages as short lattice vectors, as mentioned in the introduction.

Construction 5.4 (BCH lattice family). Let q be a power of two, let $n = q-1$, and let $\ell \leq \log_4 n$ be a positive integer. For each $i = 0, \ldots, \ell$ let $\mathcal{C}_i = \mathcal{C}_{\mathbb{F}_q}[n, 4^i]$ be the BCH code of length n with designed distance $d_i = 4^i \leq n$. Define the lattice $\Lambda_{q,\ell}$ to be the Construction D lattice for the tower $\mathbb{F}_2^n = \mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots \supseteq \mathcal{C}_\ell$.[7]

This construction and the following lemma are essentially the same as the ones that appear in [BS83, Mic12]. The construction could also be performed with towers of *extended* BCH codes of length $n = q$, in which a parity-check bit is appended to every BCH codeword. That alternative construction achieves the same asymptotic bound on the normalized minimum distance, as shown in [Mic12].

Lemma 5.5. *For any $q = 2^\kappa$ with $n = q-1$ and $\ell = \log_4 \Theta(n/\log n)$, the n-dimensional lattice $\Lambda = \Lambda_{q,\ell}$ satisfies $\lambda_1(\Lambda)/\det(\Lambda)^{1/n} = \Omega(\sqrt{n/\log n})$.*

Proof. Let $h := 2^\ell$. By Theorem 5.1, Λ has minimum distance $\lambda_1(\Lambda) \geq h$ and determinant $\det(\Lambda) = 2^{\sum_{i=1}^{\ell}(n-k_i)}$. Using the codimension bound from Lemma 2.1, for all $i \geq 1$ we have $n - k_i \leq \kappa(4^i/2)$. Substituting this bound and expanding the resulting geometric series, we get

$$\sum_{i=1}^{\ell}(n-k_i) \leq \sum_{i=1}^{\ell} \kappa(4^i/2) = \kappa(4^{\ell+1}/6) = 2\kappa h^2/3.$$

Thus we have $\det(\Lambda) \leq 2^{2\kappa h^2/3} = q^{2h^2/3}$. Finally, using $h = 2^\ell = \Theta(\sqrt{n/\log n})$, this yields

$$\frac{\lambda_1(\Lambda)}{\det(\Lambda)^{1/n}} \geq \frac{h}{q^{2h^2/(3n)}} = \Omega(h) = \Omega(\sqrt{n/\log n}). \qquad \square$$

[6]In [CS99] only a lower bound on the minimum distance is claimed, but it is easy to see that this is an equality, since Λ_ℓ has $2^\ell\mathbb{Z}^n$ as a sublattice.

[7]Recall from Section 2 that $\mathcal{C}_{i-1} \supseteq \mathcal{C}_i$ because $d_{i-1} \leq d_i$, so the codes form a tower, as required.

Theorem 5.6. *Let q be a power of two with $n = q - 1$, and let $\ell \leq \log_4(n-1)$. Algorithm 2, using Algorithm 1 as the decoder $\mathcal{D}_i$ for code $\mathcal{C}_i$, is a list-decoding algorithm for the lattice $\Lambda = \Lambda_{q,\ell}$ defined in Construction 5.4. Specifically, on input a received word $\mathbf{y} \in \mathbb{R}^n$ and $\varepsilon > 0$, the algorithm outputs a list of exactly those $\mathbf{v} \in \Lambda$ such that $\|\mathbf{y} - \mathbf{v}\| \leq \lambda_1(\Lambda) \cdot \sqrt{(1-\varepsilon)/2}$, in time $\mathrm{poly}(n, 1/\varepsilon)^\ell$. Moreover, for any $\ell \leq \log_4((1-\Omega(1))n)$ and constant $\varepsilon > 0$, the algorithm runs in time $\mathrm{poly}(n)$.*

Proof. Because $d_i = 4^i$, by Theorem 3.4, $\mathcal{D}_i$ decodes $\mathcal{C}_i$ to Euclidean distance $2^i\sqrt{(1-\varepsilon)/2}$. So, the decoders $\mathcal{D}_i$ satisfy the requirements of Algorithm 2, with $e_0 = \sqrt{(1-\varepsilon)/2} < 1$. Therefore, by Theorem 4.6 and Theorem 5.1, Algorithm 2 list decodes Λ to Euclidean distance $2^\ell e_0 = \lambda_1(\Lambda) \cdot \sqrt{(1-\varepsilon)/2}$.

By Theorem 3.4 and Theorem 4.7, the running time is $\mathrm{poly}(n, 1/\varepsilon) \cdot S^\ell$, where $S = \mathrm{poly}(n, 1/\varepsilon)$ is as in Equation (3.4), for $1/R^* \leq n/(n - d_\ell) \leq n$. In particular, when $d_\ell = 4^\ell \leq (1 - \Omega(1))n$, we have $1/R^* = O(1)$, and when $\varepsilon > 0$ is a positive constant, we have $S = O(1)$ by Equation (3.4). Therefore, because $\ell = O(\log n)$, the running time $\mathrm{poly}(n, 1/\varepsilon) \cdot S^\ell = \mathrm{poly}(n)$, as claimed. □

Finally, taking any $\ell = \log_4 \Theta(n/\log n)$ and combining Lemma 5.5 with Theorem 5.6 yields our main result, Theorem 1.1.

References

[BR60] R. C. Bose and D. K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, 1960.

[BS83] E. S. Barnes and N. J. A. Sloane. New lattices packings of spheres. *Can. J. Math*, XXXV(1):117–130, 1983.

[CS99] J. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag New York, third edition, 1999.

[DP19] L. Ducas and C. Pierrot. Polynomial time bounded distance decoding near Minkowski's bound in discrete logarithm lattices. *Des. Codes Cryptogr.*, 87(8):1737–1748, 2019.

[Eli88] P. Elias. Zero error capacity under list decoding. *IEEE Transactions on Information Theory*, 34(5):1070–1074, September 1988. Originally appeared as *Quarterly Progress Report*, vol. 48, pp. 88-90, Research Laboratory of Electronics, MIT, January 1958.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32. 1989.

[GP12] E. Grigorescu and C. Peikert. List decoding Barnes-Wall lattices. In *IEEE Conference on Computational Complexity*, pages 316–325. 2012.

[GRS19] V. Guruswami, A. Rudra, and M. Sudan. Essential coding theory. March 2019. `https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/web-coding-book.pdf`.

[GS98] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inf. Theory*, 45(6):1757–1767, 1999. Preliminary version in FOCS 1998.

[Hoc59] A. Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2:147–156, 1959.

[KV01] R. Koetter and A. Vardy. Optimal weight assignment for algebraic soft decoding. In *Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, pages 280–284. 2001.

[KV03] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Trans. Inf. Theory*, 49(11):2809–2825, 2003.

[Mic98] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2000. Preliminary version in FOCS 1998.

[Mic12] D. Micciancio. Inapproximability of the Shortest Vector Problem: Toward a deterministic reduction. *Theory Comput.*, 8(1):487–512, 2012.

[MKO18] T. Matsumine, B. M. Kurkoski, and H. Ochiai. Construction D lattice decoding and its application to BCH code lattices. In *IEEE Global Communications Conference*, pages 1–6. 2018.

[MN08] D. Micciancio and A. Nicolosi. Efficient bounded distance decoders for Barnes-Wall lattices. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2484–2488. 2008.

[RS60] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[Sud97] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complex.*, 13(1):180–193, 1997.

[Woz58] J. M. Wozencraft. List decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.