On the decoding of lattices constructed via a single parity check

Vincent Corlay, Joseph J Boutros, Philippe Ciblat

On the decoding of lattices constructed via a single parity check

Vincent Corlay, *Member, IEEE*, Joseph J. Boutros, *Senior Member, IEEE*, Philippe Ciblat, *Senior Member, IEEE*, and Loïc Brunel, *Senior Member, IEEE*

***Abstract*—This paper investigates the decoding of a remarkable set of lattices: We treat in a unified framework the Leech lattice in dimension 24, the Nebe lattice in dimension 72, and the Barnes-Wall lattices. A new interesting lattice is constructed as a simple application of single parity-check principle on the Leech lattice. The common aspect of these lattices is that they can be obtained via a single parity check or via the k-ing construction. We exploit these constructions to introduce a new efficient paradigm for decoding. This leads to efficient list decoders and quasi-optimal decoders on the Gaussian channel. Both theoretical and practical performance (point error probability and complexity) of the new decoders are provided.**

***Index Terms*—Single parity check, Leech lattice, Nebe lattice, Barnes-Wall lattices, bounded-distance decoding, list decoding.**

I. Introduction

The Leech lattice was discovered at the dawn of the communications era [26]. Recently, it was proved that the Leech lattice is the densest packing of congruent spheres in 24 dimensions [6]. Between these two major events, it has been subject to countless studies. This 24-dimensional lattice is exceptionally dense for its dimension and has a remarkable structure. For instance, it contains the densest known lattices in many lower dimensions and it can be obtained in different ways from these lower dimensional lattices. In fact, finding the simplest structure for efficient decoding of the Leech lattice has become a challenge among engineers. Forney even refers to the performance of the best algorithm as a world record [17]. Of course, decoding the Leech lattice is not just an amusing game between engineers as it has many practical interests: Its high fundamental coding gain of 6 dB makes it a good candidate for high spectral efficiency short block length channel coding and its spherical-like Voronoi region of 16969680 facets [10] enables to get state-of-the-art performance for operations such as vector quantization or lattice shaping.

Recently, Nebe solved a long standing open problem when she found an extremal even unimodular lattice in dimension 72 [35]. The construction she used to obtain this new lattice involves the Leech lattice and Turyn's construction [43] [29, Chap. 18, Sec 7.4]. This 72-dimensional extremal lattice (referred to as the Nebe lattice) is likely to have better property than the Leech lattice for the operations mentioned above. However, unlike the Leech lattice, its decoding aspect has not been studied much and, to the best of our knowledge, no efficient decoding algorithm is known in the literature. Moreover, none of the existing decoding algorithms for the Leech lattice seems to scale to the Nebe lattice. The primary motivation of this work was to propose a new decoder for this lattice[1].

In this paper, the Leech lattice and the Nebe lattice are presented as special instances of general constructions: the k-ing construction $\Gamma(V, \alpha, \beta, k)$ and the single parity-check k-lattices $\Gamma(V, \beta, k)_{\mathcal{P}}$, where $\Gamma(V, \beta, k)_{\mathcal{P}} \subseteq \Gamma(V, \alpha, \beta, k)$. These constructions consist in using lattices in smaller dimensions, e.g. n/k, along with a single parity check (and a repetition code for the k-ing construction) to obtain a new lattice in dimension n. Definitions of these constructions are provided in Section III-A. As examples, the set of lattices obtained as $\Gamma(V, \alpha, \beta, k)$ for $k = 3$ (known as Turyn's construction [29, Chap. 18, Sec 7.4]) include the Leech lattice and the Nebe lattice. Regarding the single parity-check k-lattices, Barnes-Wall lattices are part of the case $k = 2$.

This framework enables to jointly investigate the construction of several lattices and to present a new decoding paradigm for all of them. The paradigm can either be used for bounded-distance decoding (BDD), for list decoding, or for (quasi or exact)-maximum likelihood decoding (MLD) on the additive white Gaussian channel. For regular list decoding (i.e. enumerating all the lattice points in a sphere whose radius is greater than half the minimum distance[2] of the lattice), the paradigm can be combined with a technique called the splitting strategy which enables to reduce the complexity. Regarding quasi-optimal decoding on the Gaussian channel, our analysis reveals that regular list decoding is not the best choice with our decoding paradigm from a complexity point of view. A modified version of the regular list decoder is therefore presented. Formulas to predict the performance of these algorithms on the Gaussian channel are provided.

The paper is organized as follows. Section II gives preliminaries. The k-ing construction and the single parity-check

V. Corlay is with Mitsubishi Electric R&D Centre Europe, Rennes, France, and Telecom Paris, Palaiseau, France (v.corlay@fr.merce.mee.com). J. J. Boutros is with the Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar (boutros@tamu.edu). P. Ciblat is with Telecom Paris, Palaiseau, France (philippe.ciblat@telecom-paris.fr). L. Brunel is with Mitsubishi Electric R&D Centre Europe, Rennes, France (l.brunel@fr.merce.mee.com).

Part of the section on Barnes-Wall lattices was presented at the IEEE International Symposium on Information Theory, Los Angeles, USA, July 2020.

[1] An efficient decoder was found, see Section VI.

[2] In this paper, we consider squared distances. Therefore, for consistency we should have stated: Greater than a *quarter* of the minimum distance.

k-lattices are introduced in Section III. It is then shown how famous lattices are obtained from these constructions, as well as the parity lattices. The decoding paradigms are presented in Section IV. Section V is dedicated to the study of parity lattices. Formulas to assess the performance of these algorithms on the Gaussian channel are then provided. In Section V-D, we further investigate the recursive list-decoding algorithms for the parity lattices with $k = 2$ (Barnes-Wall lattices). Section VI focus on the decoding of the Leech and Nebe lattices as special cases of the k-ing construction. Section VII presents additional numerical results: A benchmark of the performance of existing schemes is provided. Finally, we draw the conclusions in Section VIII and the appendices are located in Section IX.

The main contributions of this paper are:

- A new decoding paradigm to decode $\Gamma(V, \beta, k)_{\mathcal{P}}$ is summarized within Algorithm 1. Two list decoding versions of this first algorithm (without and with a technique called the splitting strategy) are then presented. Moreover, Algorithm 5 is a direct application of Algorithm 1 to decode $\Gamma(V, \alpha, \beta, k)$. See Section IV.
- A recursive version of the algorithm of Section IV is presented to decode the parity lattices recursively built as $\Gamma(V, \beta, k)_{\mathcal{P}}$. A modified list-decoding algorithm is proposed for the Gaussian channel. Analytic expressions to assess the performance are provided, along with examples. See Section V.
- We show that the parity lattice $L_{3\cdot 24} = \Gamma(V, \beta, 3)_{\mathcal{P}}$, as sublattice of $\mathscr{N}_{72}$, has performance only 0.2 dB apart from the Nebe lattice $\mathscr{N}_{72}$ on the Gaussian channel. Moreover, the decoding complexity of $L_{3\cdot 24}$ is significantly reduced. See Section V-C. This is a remarkable result in finding a complexity-performance trade-off.
- The case $\Gamma(V, \beta, 2)_{\mathcal{P}}$, which includes Barnes-Wall (BW) lattices, is also investigated. We achieve a lower decoding complexity than the one of existing list decoders for BW lattices. The modified list-decoding algorithm yields quasi-optimal decoding performance of BW lattices over the Gaussian channel, at a reasonable complexity, up to dimension 128. See Section V.
- New decoding algorithms for Λ_{24} and $\mathscr{N}_{72}$ are developed as an application of our decoding paradigm. See Section VI.
- These new decoding algorithms uncover the performance of several lattices on the Gaussian channel. For instance, Barnes-Wall lattices, the Nebe lattice, and the 3-parity-Leech lattice $L_{3\cdot 24}$ are very competitive in their respective dimension: We observe that they have performance similar to known lattices whose dimension is an order of magnitude larger. See Section VII.

II. Preliminaries

Lattice. We define a lattice as a free J-module, where the possible rings J considered in this paper are $\mathbb{Z}$, $\mathbb{Z}[i]$, and $\mathbb{Z}[\lambda]$, $\lambda = \frac{1+i\sqrt{7}}{2}$. If J is the regular ring of integers $\mathbb{Z}$, the lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$. If J is a complex ring of integers, Λ is a discrete additive subgroup of $\mathbb{C}^n$ and we say that the lattice is complex. Given a lattice Λ of rank-n in $\mathbb{R}^n$, the rows of a $n \times n$ generator matrix G constitute a basis of the lattice and any lattice point x is obtained via $x = z \cdot G$, where $z \in \mathbb{Z}^n$. If it is of rank-n in $\mathbb{C}^n$, a generator matrix for the corresponding real lattice in $\mathbb{R}^{2n}$ can be obtained as follows. Map each component $a+ib$ of the complex generator matrix to

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} \text{ or } \begin{bmatrix} a & b \\ (a-\sqrt{7}b)/2 & (b+\sqrt{7}a)/2 \end{bmatrix}, \quad (1)$$

if J is respectively $\mathbb{Z}[i]$ and $\mathbb{Z}[\lambda]$.
Given a complex lattice $\Lambda^{\mathbb{C}}$ with generator matrix $G^{\mathbb{C}}$, the lattice generated by

$$\theta \cdot G^{\mathbb{C}} \quad (2)$$

is denoted $\theta\Lambda^{\mathbb{C}}$. Let Λ, with generator matrix G, be the real lattice obtained via (1) from the complex lattice $\Lambda^{\mathbb{C}}$. The real version of $\theta\Lambda^{\mathbb{C}}$, denoted by $\theta\Lambda$, can be either obtained using (1) on (2) or from G as follows. Let $R(2, \theta)$ be the 2×2 matrix obtained from θ via (1), e.g.

$$R(2, \lambda) = \begin{bmatrix} 1/2 & \sqrt{7}/2 \\ -\sqrt{7}/2 & 1/2 \end{bmatrix} \text{ and } R(2, \phi) = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}, \quad (3)$$

where $\phi = 1 + i$. The scaling-rotation operator $R(n, \theta)$ in dimension n is defined by the application of $R(2, \theta)$ on each pair of components. I.e. the scaling-rotation operator is $R(n, \theta) = I_{n/2} \otimes R(2, \theta)$, where I_n is the $n \times n$ identity matrix and $\otimes$ is the Kronecker product. Then, the real version of $\theta\Lambda^{\mathbb{C}}$ is generated by $G \cdot R(n, \theta)$.
For a $\mathbb{Z}$-lattice Λ, the Gram matrix is $G \cdot G^T$. The Voronoi cell of $x \in \Lambda$ is:

$$\mathcal{V}(x) = \{y \in \mathbb{R}^n : \|y - x\| \leq \|y - x'\|, \forall x' \in \Lambda\}. \quad (4)$$

The fundamental volume of Λ, i.e. the volume of its Voronoi cell and its fundamental parallelotope, is denoted by $\mathrm{vol}(\Lambda)$. The minimal distance (or minimal squared norm) of Λ is denoted $d(\Lambda)$ and the packing radius is $\rho(\Lambda) = \sqrt{d(\Lambda)}/2$. We also use $R(\Lambda)$ for the covering radius of Λ, defined as

$$R(\Lambda) = \max_{y \in \mathbb{R}^n} \min_{x \in \Lambda} \sqrt{d(y, x)}, \quad (5)$$

where $d(x, y)$ represents the squared Euclidean norm between two elements $x, y \in \mathbb{R}^n$. The number of lattice points located at a distance $d(\Lambda)$ from the origin is the kissing number $\tau(\Lambda)$. The fundamental coding gain γ of a lattice Λ is given by the following ratio

$$\gamma(\Lambda) = \frac{d(\Lambda)}{\mathrm{vol}(\Lambda)^{\frac{2}{n}}}. \quad (6)$$

We say that an integral lattice (i.e. the Gram matrix has integer entries) is even if $\|x\|^2$ is even for any x in Λ. Moreover, an integral lattice with $\mathrm{vol}(\Lambda) = 1$ is called a unimodular or a self-dual lattice. Two lattices are equivalent $\Lambda' \cong \Lambda$ if their generator matrices, respectively G' and G, are related by $G' = cUGB$, where c is a non zero constant, U a unimodular matrix, and B an orthogonal matrix. If the constant c should be explicit, we write $\Lambda' \cong c\Lambda$.

Let Λ and Λ' be lattices where $\Lambda' \subseteq \Lambda$. If the order of the quotient group Λ/Λ' is q, then Λ can be expressed as the

union of q cosets of Λ'. We denote by $[\Lambda/\Lambda']$ a system of coset representatives for this quotient group. It follows that

$$\Lambda = \bigcup_{x_i \in [\Lambda/\Lambda']} \Lambda' + x_i = \Lambda' + [\Lambda/\Lambda'].$$

It is simple to prove that [15, Lem. 1]

$$|\Lambda/\Lambda'| = \frac{\text{vol}(\Lambda')}{\text{vol}(\Lambda)}. \tag{7}$$

Let $B_r(y)$ be a ball of radius r centered at $y \in \mathbb{R}^n$. The set $\Lambda \cap B_r(y)$, $\Lambda \subset \mathbb{R}^n$, represents the elements $x \in \Lambda$ where $d(x,y) \leq r$. Let $L(\Lambda, r, y) = |\Lambda \cap B_r(y)|$ be the number of elements in the set $\Lambda \cap B_r(y)$. The quantity

$$L(\Lambda, r) = \max_{y \in \mathbb{R}^n} |\Lambda \cap B_r(y)| \tag{8}$$

denotes the maximum number of elements in the set $\Lambda \cap B_r(y)$, for any $y \in \mathbb{R}^n$. In most situations it will be convenient to consider the relative radius $\delta = r/d(\Lambda)$, which enables to define $l(\Lambda, \delta, y) = L(\Lambda, r, y)$ and $l(\Lambda, \delta) = L(\Lambda, r)$. By abuse of notations, we set $B_r(y) = B_\delta(y)$; it should be clear from the context whether the radius or relative radius is used. We also define the relative distance: $\delta(x,y) = \frac{d(x,y)}{d(\Lambda)}$.

The following Johnson-type bound on the list size for arbitrary lattices is proved in [32, Chapter 5].

Theorem 1. *Let Λ be a lattice in $\mathbb{R}^n$. The list size $L(\Lambda, r)$, defined by* (8)*, is bounded as:*

- $L(\Lambda, r) \leq \frac{1}{2\epsilon}$ *if* $r \leq d(\Lambda)(1/2 - \epsilon)$, $0 < \epsilon \leq 1/4$.
- $L(\Lambda, r) \leq 2n$ *if* $r \leq d(\Lambda)/2$.

Let $\Lambda_n \in \mathbb{R}^n$ be part of a family of lattices with instances in several dimensions n. If we want to specify the list size for the lattice in a given dimension n, we simplify the notations as follows. We let $L(n, r) = L(\Lambda_n, r)$ and $l(n, \delta) = l(\Lambda_n, \delta)$.

BDD, list decoding, optimal and quasi-optimal decoding (with Gaussian noise). Given a lattice Λ, a radius $r > 0$, and any point $y \in \mathbb{R}^n$, the task of a list decoder is to determine all points $x \in \Lambda$ satisfying $d(x,y) \leq r$: i.e. compute the set $\Lambda \cap B_r(y)$. If $r < \rho^2(\Lambda)$, there is either no point or a unique point found and the decoder is known as BDD. In this paper, BDD means that we consider a decoding radius $r = \rho^2(\Lambda)$ where in case of a tie between several lattice points, one of them is randomly chosen by the decoder. When $d(x,y) < \rho^2(\Lambda)$, we say that y is within the guaranteed (or unique) error-correction radius of the lattice. If $r \geq \rho^2(\Lambda)$, there may be more than one point in the sphere. In this case, the process is called list decoding rather than BDD.
Note that a modified list decoder may output a set of lattice points $\mathcal{T} \neq \Lambda \cap B_r(y)$. Therefore, we may refer to list decoders where $\mathcal{T} = \Lambda \cap B_r(y)$ as "regular" list decoders.
Optimal decoding simply refers to finding the closest lattice point in Λ to any point $y \in \mathbb{R}^n$. In the literature, it is usually said that an optimal decoder solves the closest vector problem (CVP). If regular list decoding is used, it is equivalent to choosing a decoding radius equal to $R(\Lambda)$ and keeping the closest point to y in the list outputted by the list decoder.

Let $x \in \Lambda$ and w be a Gaussian vector where each component is i.i.d with distribution $\mathcal{N}(0, \sigma^2)$. Consider the point y obtained as

$$y = x + w. \tag{9}$$

Since this model is often used in digital communications, x is referred to as the transmitted point, y the received point, and the model described by (9) is called a Gaussian channel. The point error probability under optimal decoding is $P_e(opt, \sigma^2) = P(y \notin \mathcal{V}(x))$. On the Gaussian channel, given equiprobable symbols, optimal decoding is also referred to as maximum likelihood decoding (MLD). Moreover, at a fixed dimension n, we say that a decoder is quasi-MLD (QMLD) if there exists $\sigma_0^2 > 0$ and $\epsilon \in (0,1)$ such that $P_e(dec, \sigma_0^2) \leq P_e(opt, \sigma_0^2) \cdot (1 + \epsilon)$.
In the scope of (infinite) lattices, the transmitted information rate and the signal-to-noise ratio based on the second-order moment are meaningless. Poltyrev introduced the generalized capacity [36], the analog of Shannon capacity for lattices. The Poltyrev limit corresponds to a noise variance of $\sigma^2_{max} = \text{vol}(\Lambda)^{\frac{2}{n}}/(2\pi e)$. The point error rate on the Gaussian channel is therefore evaluated with respect to the distance to Poltyrev limit, also called the volume-to-noise ratio (VNR), i.e. $\Delta = \sigma^2_{max}/\sigma^2$.
The performance of the considered lattices with Gaussian noise, along with their decoders, are compared with the sphere lower bound (see Section V-D4). For $\mathscr{N}_{72}$, we also plot an approximation of the MLD performance. If the MLD performance is far enough from the Poltyrev limit, the approximation can be based on a truncated *union bound estimate*, which considers only the first lattice shell[3]. However, as explained in [19], this approximation is not accurate if the MLD performance approaches the Poltyrev capacity[4]. Therefore, our truncated union bound estimate for $\mathscr{N}_{72}$ is improved by considering two shells of the lattice:

$$\tau \cdot Q\left(\sqrt{\frac{d(\Lambda)}{4\sigma^2}}\right) + \tau' \cdot Q\left(\sqrt{\frac{d(\Lambda)'}{4\sigma^2}}\right), \tag{10}$$

where τ' and $d(\Lambda)'$ are respectively the population and the squared norm of the second lattice shell, and $Q(\cdot)$ is the Gaussian tail function. This union bound takes into account that not all facets of the Voronoi region are generated by the first shell. The dropped terms in the theta series of a lattice [10] are small o of the first two terms for small σ^2, so (10) is tight at high signal-to-noise ratio.

Complexity analysis. The complexity of the algorithms is denoted by $\mathfrak{C}$ or $\mathfrak{C}_{A.i}$, where i represents the index of the algorithm. The decoding complexity of a lattice Λ is expressed as $\mathfrak{C}(\Lambda)$, where the decoding technique considered is clear from the context. In general, $\mathfrak{C}$ denotes the worst-case running time. By abuse of notation, we use equalities (e.g. $\mathfrak{C} = X$) even though we only provide upper-bounds on the worst-case running time. We adopt this approach to characterize the

[3]A lattice shell denotes the set of lattice points at a given distance from the origin.

[4]More precisely, since only finite-power constellations are discussed in [19], they state that the union bound estimate is not accurate beyond the cutoff rate.

complexity of the proposed algorithms, which does not take into account the position of the point to decode y. However, to assess the complexity of the algorithms on the Gaussian channel, we take advantage of the distribution of the point y to decode and assess the average complexity $E_y[\mathfrak{C}]$ (warning: $E_y[\mathfrak{C}]$ does not denote the average worst-case complexity but the average complexity).

The complexity of decoding in a lattice Λ with a specific decoder is denoted by $\mathfrak{C}^{\Lambda}_{dec}$, where "dec" should be replaced by the name of the decoder: E.g. the complexity of BDD, optimal decoding, MLD, and quasi-MLD are $\mathfrak{C}^{\Lambda}_{BDD}$, $\mathfrak{C}^{\Lambda}_{opt}$, $\mathfrak{C}^{\Lambda}_{MLD}$, $\mathfrak{C}^{\Lambda}_{QMLD}$, respectively. Moreover, we denote by $\mathfrak{C}_{\Lambda\cap B_\delta(y)}$, $\mathfrak{C}^{\Lambda}_{stor.}$, and $\mathfrak{C}_{clos.}(n)$, the complexity of computing the set $\Lambda\cap B_\delta(y)$, storing an element belonging to Λ, and finding the closest element to y among n elements, respectively. If not specified, the set $\Lambda\cap B_\delta(y)$ can be computed via the sphere decoding algorithm [46]. In this case $\mathfrak{C}_{\Lambda\cap B_\delta(y)}=\mathfrak{C}^{\Lambda}_{Sph.dec.,\delta}$. In general, we assume that $\mathfrak{C}^{\Lambda}_{dec} >> \mathfrak{C}^{\Lambda}_{stor.}$ and that $k\mathfrak{C}^{\Lambda}_{dec} >> \mathfrak{C}_{clos.}(k)$. Hence, we have

$$k\mathfrak{C}^{\Lambda}_{dec}+k\mathfrak{C}^{\Lambda}_{stor.}+\mathfrak{C}_{clos.}(k)\approx k\mathfrak{C}^{\Lambda}_{dec}. \tag{11}$$

Similarly, we also have:

$$\mathfrak{C}_{\Lambda\cap B_\delta(y)}+l(\Lambda,\delta)\mathfrak{C}^{\Lambda}_{stor.}+\mathfrak{C}_{clos.}(l(\Lambda,\delta))\approx \mathfrak{C}_{\Lambda\cap B_\delta(y)}. \tag{12}$$

By abuse of notations, we may write $k\mathfrak{C}^{\Lambda}_{dec}+k\mathfrak{C}^{\Lambda}_{stor.}+\mathfrak{C}_{clos.}(k)=k\mathfrak{C}^{\Lambda}_{dec}$ (e.g. if $\Lambda\in\mathbb{R}^{\frac{n}{k}}$, we sometimes write $k\mathfrak{C}^{\Lambda}_{dec}+O(n)=k\mathfrak{C}^{\Lambda}_{dec}$ if the $O(n)$ is not relevant in the context). When recursively decoding a lattice $\Lambda_n\subset\mathbb{R}^n$, we simplify the notation $\mathfrak{C}(\Lambda_n,\delta)$ by $\mathfrak{C}(n,\delta)$.

The $\widetilde{O}$ notations is used to ignore the logarithmic factors. The notation $f(n)=\widetilde{O}(h(n))$ is equivalent to $\exists k$ such that $f(n)=O(h(n)\log^k(h(n)))$ (since $\log^k(n)$ is always $o(n^\epsilon)$ for any $\epsilon>0$).

Extremal lattice. The fundamental coding gain of an even unimodular lattice of dimension n is at most $2\lfloor\frac{n}{24}\rfloor+2$. Lattices achieving this coding gain are called extremal.

III. Lattice construction

A. The k-ing construction and the single parity-check lattice

Consider lattices S,T,V, where $V\subset T\subset S$. Let us denote $\alpha=[S/T]$ and $\beta=[T/V]$, two groups of coset representatives. The k-ing construction of a lattice Γ is defined as

$$\begin{aligned}\Gamma(V,\alpha,\beta,k)=&\{(m+t_1,m+t_2,...,m+t_k),\\ &m\in\underbrace{V+\alpha}_{T^*},t_i\in\underbrace{V+\beta}_{T},\sum_{i=1}^{k}t_i\in V\}\subseteq S^k,\end{aligned} \tag{13}$$

since $V+\beta$ is the lattice T. We denote $V+\alpha$ by T^*. T^* is a lattice. Indeed, after noticing that $T^*=S-\beta$, one can prove that $0\in T^*$, $-x\in T^*$, and $x+y\in T^*$, for any $x,y\in T^*$. $\Gamma(V,\alpha,\beta,k)$ being a lattice follows immediately, because T and T^* are lattices. $\Gamma(V,\alpha,\beta,k)$ can alternatively be denoted by $\Gamma(V,T^*,T,k)$.

An obvious sublattice of $\Gamma(V,\alpha,\beta,k)$ is the single parity-check lattice in T^k:

$$\begin{aligned}\Gamma(V,\beta,k)_{\mathcal{P}}=\Gamma(V,T,k)_{\mathcal{P}}&=\{(t_1,t_2,...,t_k)\in T^k|\sum_{i=1}^{k}t_i\in V\},\\ &=\{(t_1,t_2,...,v_k-\sum_{i\neq k}t_i),v_k\in V,\ t_i\in T\},\end{aligned} \tag{14}$$

where the last expression is the most useful in practice.

Using $\Gamma(V,\beta,k)_{\mathcal{P}}$, the lattice $\Gamma(V,\alpha,\beta,k)$ can be represented as follows, an expression useful for decoding:

$$\Gamma(V,\alpha,\beta,k)=\bigcup_{m\in\alpha}\{\Gamma(V,\beta,k)_{\mathcal{P}}+m^k\}, \tag{15}$$

where $m^k=(m,...,m)$ (repeated k times). The set $V+\alpha$ for m in (13) becomes α in (15) after moving the V components into the $t_i's$.

From (14), we easily see that $d(\Gamma(V,\beta,k)_{\mathcal{P}})=\min\{d(V),2d(T)\}$. The minimum distance of $\Gamma(V,\alpha,\beta,k)$ is provided by the next theorem, proved in [16] [12].

Theorem 2. *The minimum distance of $\Gamma(V,\alpha,\beta,k)$ satisfies*

$$\min\{d(V),2d(T)\}\geq d(\Gamma(V,\alpha,\beta,k))\geq\min\{d(V),2d(T),kd(S)\}. \tag{16}$$

Families of single parity-check lattices can be built by recursively applying the single parity-check construction (Equation (14)). For instance, a new family of lattices is obtained as follows. First, since $d(\Gamma(V,\beta,k)_{\mathcal{P}})=\min\{2d(T),d(V)\}$, we shall consider only lattices where $d(V)=2d(T)$. In order to find two lattices having this property, with $V\subset T$, we consider a lattice T over a complex ring J, and rotate it by an element $\theta\in J$, with $|\theta|=\sqrt{2}$, to get V: i.e. $V=\theta\cdot T$. This yields $V\cong\sqrt{2}T$ and $d(\Gamma(\theta T,\beta,k)_{\mathcal{P}})=2d(T)$. The ring J can for instance be $\mathbb{Z}[i]$ or $\mathbb{Z}[\lambda]$. More formally, let $\Lambda^{\mathbb{C}}_c\in\mathbb{C}^{c/2}$ be a lattice over a complex ring J, where J is either $\mathbb{Z}[i]$ or $\mathbb{Z}[\lambda]$. We denote by $\Lambda_c\in\mathbb{R}^c$ the corresponding real lattice, with real dimension c. Let L_n be the real lattice obtained via (1) from a complex lattice $L^{\mathbb{C}}_n$. In the sequel, θL_n is the notation for the real lattice obtained from $\theta\cdot L^{\mathbb{C}}_n$. Also, $\beta=[L_n/\theta L_n]$.

Definition 1. *Let $n=c\cdot k^t$, $t\geq 0$. The parity lattices in dimension kn are defined by the following recursion:*

$$\begin{aligned}L_{kn}&=\Gamma(\theta L_n,\beta,k)_{\mathcal{P}},\\ &=\{(v_1+n_1,v_2+n_2,...,v_k-\sum_{i\neq k}n_i),v_i\in\theta L_n,n_i\in\beta\},\\ &=\{(t_1,t_2,...,v'_k-\sum_{i\neq k}t_i),v'_k\in\theta L_n,t_i\in L_n\},\end{aligned} \tag{17}$$

with initial condition $L_c=\Lambda_c$. The number of recursive steps t should not be confused with $t_i\in L_n$.

As we shall see in the sequel, several famous lattices can be obtained via the k-ing construction or as parity lattices, including the Leech lattice and the Nebe lattice as well as the Barnes-Wall lattices.

Most papers study the k-ing construction for a fixed k. As a result, various names exist for this construction given a fixed k:

If $k = 3$, it is Turyn's construction [29, Chap. 18, sec 7.4] (also known as the cubing construction [16]). If $k = 4$, it is the two-level squaring construction [16]. If $k = 5$, it is the quinting construction [28]. If $k = 7$, it is the septing construction [28].
Example. Take $V = 2\mathbb{Z}$ and $T = \mathbb{Z}$. Then, $\beta = [\mathbb{Z}/2\mathbb{Z}]$. The checkerboard lattice is $D_n = \Gamma(V, \beta, n)_{\mathcal{P}}$.

B. Parity lattices with $k = 2$ (BW lattices)

We consider the parity lattices obtained with $k = 2$, $L_c = \mathbb{Z}^2$ and $\theta = \phi = 1 + i$. They are called Barnes-Wall lattices in the literature [10]. These lattices are recursively expressed as

$$BW_{2n} = \Gamma(\phi BW_n, \beta, 2)_{\mathcal{P}} \text{ where } BW_2 = \mathbb{Z}^2. \tag{18}$$

In general, the lattice ϕBW_n is denoted by RBW_n [16]. We adopt this latter notation for the rest of the paper. These BW lattices were one of the first series discovered with an infinitely increasing fundamental coding gain [3]: It increases as $\gamma(BW_n) = \sqrt{2} \cdot \gamma(BW_{n/2}) = \sqrt{n/2}$.

C. Leech lattice and Nebe lattice

Both Leech and Nebe lattices can be obtained via Turyn's construction, i.e. as $\Gamma(V, \alpha, \beta, 3)$.

Among the three lattices S, T, V used in the construction $\Gamma(V, \alpha, \beta, 3)$, let us take $V = 2S$. To build the Leech lattice, we have $S, T \cong E_8$ and to build the Nebe lattice we have $S, T \cong \Lambda_{24}$. Moreover, to obtain these two lattices via the k-ing construction, the set of coset representatives α should be chosen such that $d(\Gamma(V, \alpha, \beta, 3)) > 3d(S)$ (instead of $\geq$ as in Theorem 2). We already established via (13) that choosing α is equivalent to choosing T^*. In the next section, we explain how to get T^* via lattice polarisation [35].

1) The polarisation of lattices: Assume that the lattices $S, T, T^*, V = 2S$ are of rank n. Here, T^* is a rotation of T by an angle of 2ω. Therefore, it is denoted $T_{2\omega}$.

Definition 2. *Given a lattice S, we call $(T, T_{2\omega})$ a polarisation of S [35] if*

$$S \cong T \cong T_{2\omega}, S = T_{2\omega} + T, \quad \textit{and} \quad T_{2\omega} \bigcap T = 2S. \tag{19}$$

Let G_S be a generator matrix of S. Finding a polarisation of the lattice S (if it exists) is equivalent to finding a scaling-rotation matrix R, $R \cdot R^T = 2I$, with $G_T = G_S \cdot R$ and $G_{T_{2\omega}} = G_S \cdot R^T$, such that the basis vectors g_T^i and $g_{T_{2\omega}}^i$, $1 \leq i \leq n$, are versions of the vectors g_S^i scaled by a factor of $\sqrt{2}$ and rotated by an angle of $\pm\omega = \arctan\sqrt{7}$. Indeed, consider two vectors g_T^i and $g_{T_{2\omega}}^i$ of the same size and having an angle of 2ω. Summing these two vectors yields a vector g_S^i having half the size of g_T^i: $||g_S^i||^2 = ||g_T^i + g_{T_{2\omega}}^i||^2 = 0.5 \times ||g_T^i||^2$, since $\cos(2\omega) = -3/4$. One would thus get $G_S = G_T + G_{T_{2\omega}}$. The rotation matrix R can be found via a $\mathbb{Z}[\lambda]$-structure of S; Let $G_S^{\mathbb{C}}$ be a (complex) generator matrix of S over the ring of integers $\mathbb{Z}[\lambda]$, $\lambda = \sqrt{2}e^{i\omega} = \frac{1+i\sqrt{7}}{2}$. Multiplying $G_S^{\mathbb{C}}$ by λ yields a matrix whose rows are new vectors belonging to the lattice, scaled by $\sqrt{2}$, and having the desired angle with the basis vectors. Hence, $G_T^{\mathbb{C}}$ can be obtained as $\lambda G_S^{\mathbb{C}}$ and $G_{T_{2\omega}}^{\mathbb{C}}$ as $\psi G_S^{\mathbb{C}}$, where $\psi = \bar{\lambda}$ is the conjugate of λ. Therefore, if we let G_S be the real generator matrix obtained from $G_S^{\mathbb{C}}$ (via (1)), the real rotation matrix R for polarisation is $R(n, \lambda) = I_{n/2} \otimes R(2, \lambda)$.

2) Leech lattice and Nebe lattice: Given three lattices S, T and $T_{2\omega}$, respecting properties (19), we consider the lattice $\Gamma(2S, T_{2\omega}, T, 3)$. A generator matrix of $\Gamma(2S, T_{2\omega}, T, 3)$ is

$$G_{\Gamma(2S,T_{2\omega},T,3)} = \begin{bmatrix} G_{(3,1)} \otimes G_{T_{2\omega}} \\ G_{(3,2)} \otimes G_T \end{bmatrix}, \tag{20}$$

where $G_{(3,1)}$ and $G_{(3,2)}$ are generator matrices for the $(3, 1)$ binary repetition code and the $(3, 2)$ binary single parity-check code, respectively. Obviously, $\mathbb{F}_2$ is naturally embedded into $\mathbb{Z}$ for the two binary codes. From (20), a generator matrix of $\Gamma(2S, T_{2\omega}, T, 3)$ over $\mathbb{Z}[\lambda]$ can be expressed as

$$G_{\Gamma(2S,T_{2\omega},T,3)}^{\mathbb{C}} = \underbrace{\begin{bmatrix} \lambda & \lambda & \lambda \\ \psi & \psi & 0 \\ 0 & \psi & \psi \end{bmatrix}}_{=Pb} \otimes G_S^{\mathbb{C}}, \tag{21}$$

where $G_S^{\mathbb{C}}$ is a generator matrix of S over $\mathbb{Z}[\lambda]$, $\lambda G_S^{\mathbb{C}}$ a generator matrix of T, and $\psi G_S^{\mathbb{C}}$ a generator matrix of $T_{2\omega}$.

Theorem 3. *Let $S \cong E_8$ and T,$T_{2\omega}$ be two lattices respecting properties* (19)*. Then, $\Gamma(2S, T_{2\omega}, T, 3)$ is the Leech lattice [42][27][37].*

See Appendix IX-A for a proof. A similar proof can also be used to show that when $S \cong \Lambda_{24}$ and T,$T_{2\omega}$ are two lattices respecting properties (19), then the lattice $\Gamma(2S, T_{2\omega}, T, 3)$ has a fundamental coding gain equal to 6 or 8 [21]. In this case, the polarisation does not ensure $\Gamma(2S, T_{2\omega}, T, 3) > 3d(S) = 6$. Additional work to choose $T_{2\omega}$ is needed. Nebe considers in [35] the following construction. Let S be the $\mathbb{Z}[\lambda]$-structure Λ_{24} with automorphism group $\mathrm{SL}_2(25)$. Set $T_{2\omega} = \lambda S$ and $T = \psi S$. The resulting lattice $\Gamma(2S, T_{2\omega}, T, 3)$ is named the Nebe lattice $\mathscr{N}_{72}$. It is possible to check that $\mathscr{N}_{72}$ has no vector of length 6, which leads to the following theorem, obtained in [35].

Theorem 4. *$\mathscr{N}_{72}$ has a fundamental coding gain equal to 8 [35].*

IV. Decoding paradigm for the single parity-check lattice and the k-ing lattice

A. The existing decoding algorithm for $\Gamma(V, \alpha, \beta, k)$ (and $\Gamma(V, \beta, k)_{\mathcal{P}}$)

To the best of the authors' knowledge, there exists only one "efficient" optimal algorithm for the k-ing construction called trellis decoding [16]. This decoding algorithm uses a graph-based representation to efficiently explore all the cosets of V^k in $\Gamma(V, \alpha, \beta, k)$. As an example, the trellis for $\Gamma(V, \alpha, \beta, 3)$ is illustrated on Figure 1 with $|\alpha| = 3$ and $|\beta| = 2$. Each path in this three sections trellis corresponds to a coset of V^3 in $\Gamma(V, \alpha, \beta, 3)$. Each edge is associated with a coset of V in S: E.g. given $\alpha = \{m_1, m_2, m_3\}$ and $\beta = \{n_1, n_2\}$, the two upper edges on the left should be labeled $m_1 + n_1$ and $m_1 + n_2$, respectively. All the edges in the upper part of the trellis correspond to the same m_1 and the sub-trellis formed by these edges is a standard single parity-check trellis. This sub-trellis is repeated three times for m_1,m_2, and m_3. Standard trellis algorithms, such as the Viterbi algorithm, can then be used to decode.

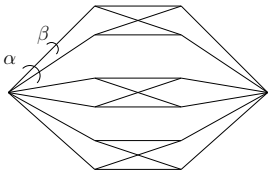

Fig. 1: Trellis representing a $\Gamma(V,\alpha,\beta,3)$ with $|\alpha|=3$ and $|\beta|=2$. The edges labeled with the same α are associated with the same $m_1 \in \alpha$.

Trellis decoding of $\Gamma(V,\alpha,\beta,k)$ involves decoding in V for each edge in the trellis. The number of edges in the trellis is $2|\alpha||\beta|+(k-2)|\alpha||\beta|^2$. Therefore, the complexity is dominated by the quantity $|\alpha||\beta|^2\mathfrak{C}_{dec}^V$. For more information on trellis decoding, the reader should refer to [16] or [13].

Of course, trellis decoding can also be used to decode the single parity-check k-lattices $\Gamma(V,\beta,k)_{\mathcal{P}}$. The number of edges in the standard single parity-check trellis is $2|\beta|+(k-2)|\beta|^2$.

B. Decoding paradigm for $\Gamma(V,\beta,k)_{\mathcal{P}}$

Set $y \in \mathbb{R}^{kn}$ and let $x=(t_1,t_2,...,t_k) \in \Gamma(V,\beta,k)_{\mathcal{P}}$ be the closest element to y. The minimum distance of the lattice V is (in general) larger than the one of T. Consequently, decoding y_j in the coset of V to which the element t_j belongs is safer than decoding in T. Moreover, any set of $k-1$ t_j's is enough to know in which coset of V^k in $\Gamma(V,\beta,k)_{\mathcal{P}}$ the point x is located. Hence, given $t_1,t_2,...t_{k-1}$, the element t_k can be recovered by decoding $y_k-(-\sum_{j=1}^{k-1}t_j)$ in V (and adding back $-\sum_{j=1}^{k-1}t_j$ on the decoded element), as shown by Algorithm 1. It is easily seen that the complexity of Algorithm 1 is

$$\mathfrak{C}_{A.1}=k\mathfrak{C}_{dec}^T+k\mathfrak{C}_{dec}^V, \tag{22}$$

where we used the simplification of Equation (11).

Algorithm 1 Decoder for $\Gamma(V,\beta,k)_{\mathcal{P}}$

Input: $y=(y_1,y_2,...,y_k)\in\mathbb{R}^{kn}$.

1: Decode $y_1,y_2,...,y_k$ in T as $t_1,t_2,...,t_k$.
2: **for** $1\le i\le k$ **do**
3: Decode $y_i-(-\sum_{j\neq i}t_j)$ in V as v_i. Add $(t_1,...,t_{i-1},v_i+(-\sum_{j\neq i}t_j),t_{i+1},...,t_k)$ to the list $\mathcal{T}$.
4: **end for**
5: **Return** the closest element of $\mathcal{T}$ to y.

Algorithm 1 can be adapted to perform list decoding as follows. For the sake of simplicity, we assume that $V\cong\sqrt{2}T$. We recall that regular list decoding consists in computing the set $\Gamma(V,\beta,k)_{\mathcal{P}}\cap B_\delta(y)$, i.e. finding all lattice points $x\in\Gamma(V,\beta,k)_{\mathcal{P}}$ where $d(y,x)\le r$, $y\in\mathbb{R}^{kn}$. The parameter $\delta=r/d(\Gamma(V,\beta,k)_{\mathcal{P}})$ is the relative decoding radius. Remember that $d(\Gamma(V,\beta,k)_{\mathcal{P}})=d(V)=2d(T)$. The list decoding of $\Gamma(V,\beta,k)_{\mathcal{P}}$ with a radius r consists in list decoding each y_j (Step 1 of Algorithm 1) in T with a radius $r/2$ and each $y_i-(-\sum_{j\neq i}t_j)$ (Step 3) in V with a radius r (see the proof of Lemma 1 for explanations on this choice). In both cases the relative radius is $\delta=\frac{r}{2d(T)}=\frac{r}{d(V)}$ and the maximum number of elements in each list is $l(T,\delta)=l(V,\delta)=L(T,\frac{r}{2})=L(V,r)$ (see Section II for the definitions of $L(\cdot,\cdot)$ and $l(\cdot,\cdot)$). As a result, Step 3 (of Algorithm 1), for a given i, should be executed for any of the combinations of candidates (for each $t_{j\neq i}$) in the $k-1$ lists: i.e. $l(T,\delta)^{k-1}$ times. The resulting maximum number of stored elements (for this given i) is $l(T,\delta)^{k-1}\cdot l(V,\delta)$. Consequently, the number of elements in $\mathcal{T}$ is bounded from above by

$$k\cdot l(T,\delta)^{k-1}\cdot l(V,\delta)=k\cdot l(T,\delta)^k. \tag{23}$$

The list-decoding version of Algorithm 1 is presented in Algorithm 2.

Lemma 1. *Algorithm 2 outputs the set* $\Gamma(V,\beta,k)_{\mathcal{P}}\cap B_\delta(y)$ *in worst-case time*

$$\mathfrak{C}_{A.2}=k\mathfrak{C}_{T\cap B_\delta(y)}+k\cdot l(T,\delta)^{k-1}\mathfrak{C}_{V\cap B_\delta(y)}. \tag{24}$$

Proof. We first prove that all points $x=(x_1,x_2,...,x_k)\in\Gamma(V,\beta,k)_{\mathcal{P}}\cap B_\delta(y)$ are outputted by Algorithm 2. If $d(y_i,x_i)>r/2$ then $d(y_j,x_j)<r/2$ for all $j\neq i$. Hence, among the k lists $\mathcal{T}_1,\mathcal{T}_2,...,\mathcal{T}_k$ computed at Step 1 of the algorithm, at least $k-1$ of them contain the correct $t_j^*=x_j$. Assume (without loss of generality) that all $\mathcal{T}_j$, $1\le j\neq i\le n$, contain t_j^*. Since $d(y,x)\le r$, one has $d(y_i,x_i)\le r$. Therefore, $\mathcal{V}_i=V\cap B_\delta(y_i-(-\sum_{j\neq i}t_j^*))$ contains $v_i^*=x_i-(-\sum_{j\neq i}t_j^*)$.
As a result, all $x\in\Gamma(V,\beta,k)_{\mathcal{P}}\cap B_\delta(y)$ are outputted by the algorithm. The complexity is obtained by reading Algorithm 2, with the simplification of Equation (11). □

Note that if $\delta<\frac{1}{4}$ (the relative packing radius), there is only one element in each of the set $\mathcal{T}_j$ computed in Algorithm 2. Algorithm 2 in this case is equivalent to Algorithm 1.

Algorithm 2 List dec. for $\Gamma(V,\beta,k)_{\mathcal{P}}\subset\mathbb{R}^{kn}$ (without the splitting strategy)

Input: $y=(y_1,y_2,...,y_k)\in\mathbb{R}^{kn}$, $\delta\ge 0$.

1: Compute the sets $\mathcal{T}_1=T\cap B_\delta(y_1),\mathcal{T}_2=T\cap B_\delta(y_2),...,\mathcal{T}_k=T\cap B_\delta(y_k)$.
2: **for** $1\le i\le k$ **do**
3: Set $j_1<j_2<...<j_{k-1}$, where $\{j_1,j_2,...,j_{k-1}\}=\{1,2,...,k\}\backslash\{i\}$.
4: **for** each $(t_{j_1},...,t_{j_{k-1}})\in\mathcal{T}_{j_1}\times\mathcal{T}_{j_2}\times...\times\mathcal{T}_{j_{k-1}}$ **do**
5: Compute the set $\mathcal{V}_i=V\cap B_\delta(y_i-(-\sum_{j'=1}^{k-1}t_{j_{j'}}))$.
6: **for** $v_i\in\mathcal{V}_i$ **do**
7: Add $(t_{j_1},...,t_{j_{i-1}},v_i+(-\sum_{j'=1}^{k-1}t_{j_{j'}}),t_{j_i},...,t_{j_{k-1}})$ to the list $\mathcal{T}$.
8: **end for**
9: **end for**
10: **end for**
11: **Return** $\mathcal{T}$.

In some cases, the complexity of Algorithm 2 can be reduced via a technique we call the splitting strategy. It exploits the following observation: Let $x=(x_1,x_2,...,x_k)\in\Gamma(V,\beta,k)_{\mathcal{P}}$. Assume (without loss of generality) that

$d(y_i, x_i) > \frac{r}{2}$ (and thus $\sum_{j\neq i} d(x_j, y_j) \leq \frac{r}{2}$). This case can be split into two sub-cases. Let $0 \leq a' \leq \frac{r}{2}$.

- If $a' \leq \sum_{j\neq i} d(x_j, y_j) \leq \frac{r}{2}$ then $\frac{r}{2} < d(x_i, y_i) \leq r - a'$: Then, each y_j should be list decoded in T with a radius $\frac{r}{2}$ and $y_i - (-\sum_{j\neq i} t_j)$ should be list decoded, for all resulting combinations of t_j, in V with a radius $r - a'$.
- Else $0 \leq \sum_{j\neq i} d(x_j, y_j) < a'$ and $\frac{r}{2} < d(x_i, y_i) \leq r$: Then, each y_j should be list decoded in T with a radius a', and $y_i - (-\sum_{j\neq i} t_j)$ should be list decoded, for all resulting combinations of t_j, in V with a radius r.

The number of stored elements (when computing each sub-case) is bounded by

- $l(T, \delta)^{k-1} \cdot l(V, a_1 = \frac{r-a'}{d(V)})$, for the first sub-case,
- $l(T, a_2 = \frac{a'}{d(T)})^{k-1} \cdot l(V, \delta)$ for the second sub-case.

Consequently, if we choose $a_1 = a_2 = \frac{2}{3}\delta$, the number of elements in the list $\mathcal{T}$, outputted by a list decoder with this splitting strategy, is bounded from above by

$$k\big[l(T, \delta)^{k-1} l(V, \frac{2}{3}\delta) + l(T, \frac{2}{3}\delta)^{k-1} l(V, \delta)\big], \tag{25}$$

which is likely to be smaller than $k \cdot l(T, \delta)^k$, the bound obtained without the splitting strategy.

Similarly, we can also split the case $0 \leq d(x_j, y_j) \leq \frac{r}{2}$, $j \neq i$, into several sub-cases. Let $0 \leq a' \leq \frac{r}{2}$. We recall that we have $\sum_{j\neq i} d(x_j, y_j) \leq \frac{r}{2}$.

- If $a' \leq d(x_j, y_j) \leq \frac{r}{2}$ then $0 \leq d(x_l, y_l) \leq \frac{r}{2} - a'$, $\forall l$ where $1 \leq l \neq j \neq i \leq k$: Then, y_j should be list decoded in T with a radius $\frac{r}{2}$ and each y_l list decoded in T with a radius $\frac{r}{2} - a'$.
- Else $0 \leq d(x_j, y_j) < a'$ and for one l, $1 \leq l \neq j \neq i \leq k$, one may have $a' \leq d(x_l, y_l) \leq \frac{r}{2}$: Then, y_j should be list decoded in T with a radius a', y_l list decoded in T with a radius $\frac{r}{2}$, and all the remaining $y's$ list decoded in T with a radius a'.

Of course, since it is not possible to know the index l where[5] $a' \leq d(x_l, y_l) \leq \frac{r}{2}$, all $k-2$ possibilities should be computed (which yields $k-1$ possibilities if we include the first sub-case $a' \leq d(x_j, y_j) \leq \frac{r}{2}$). If we choose $a' = \frac{r}{2} - a' = \frac{r}{4}$, the product of the maximum list size of each $k-1$ case is $l(T, \delta) l(T, \frac{\delta}{2})^{k-2}$. As a result, the maximum number of possibilities to consider for $\sum_{j\neq i} t_j$ is

$$(k-1) l(T, \delta) l(T, \frac{\delta}{2})^{k-2}, \tag{26}$$

instead of $l(T, \delta)^{k-1}$ without this strategy.

Substituting (26) in (25), the number of element in a list $\mathcal{T}$, outtputed by a list decoder with these two splitting strategies, is bounded from above by

$$\begin{aligned} &k(k-1)\big[l(T, \delta) l(T, \frac{\delta}{2})^{k-2} l(V, \frac{2}{3}\delta) + l(T, \frac{2}{3}\delta) l(T, \frac{\delta}{3})^{k-2} l(V, \delta)\big], \\ &= k(k-1) l(T, \delta) l(T, \frac{2}{3}\delta)\big[l(T, \frac{\delta}{2})^{k-2} + l(T, \frac{\delta}{3})^{k-2}\big], \end{aligned} \tag{27}$$

where we used $l(T, \delta) = l(V, \delta)$.

We shall refer to these two splitting strategies as the first and second splitting strategy, respectively. The first splitting strategy is listed in Algorithm 3 and can be used without or with the second splitting strategy. The function $SubR_1$ or $SubR_2$, listed in Algorithm 4, is used accordingly.

[5]One may not have $a' \leq d(x_l, y_l)$, $\forall l$, $1 \leq l \neq i \leq k$. It is not an issue as we would then simply decode with a radius greater than necessary.

Algorithm 3 List dec. for $\Gamma(V, \beta, k)_{\mathcal{P}} \subset \mathbb{R}^{kn}$ with the splitting strategy

Input: $y = (y_1, y_2, ..., y_k) \in \mathbb{R}^{kn}$, $\delta \geq 0$.

// The sets $\mathcal{T}_i^{\frac{\delta}{2}}$ and $\mathcal{T}_i^{\frac{\delta}{3}}$ are computed only if used by the subroutine.

1: **for** $\eta \in \{\delta, \frac{2}{3}\delta, \frac{\delta}{2}, \frac{\delta}{3}\}$ **do**
2: Set $\mathcal{T}_1^\eta, \mathcal{T}_2^\eta, ..., \mathcal{T}_k^\eta$ as global variables.
3: Compute the sets $\mathcal{T}_1^\eta = T \cap B_\eta(y_1), \mathcal{T}_2^\eta = T \cap B_\eta(y_2), ..., \mathcal{T}_k^\eta = T \cap B_\eta(y_k)$.
4: **end for**
5: **for** $1 \leq i \leq k$ **do**
6: $\mathcal{T}_1 \leftarrow SubR(y_1, y_2, ..., y_k, \delta, \frac{2}{3}\delta, i)$.
7: $\mathcal{T}_2 \leftarrow SubR(y_1, y_2, ..., y_k, \frac{2}{3}\delta, \delta, i)$.
// Use $SubR_1$ or $SubR_2$ (listed in Algorithm 4) if the (second) splitting strategy is not used or used, respectively.
8: **end for**
9: **Return** $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2\}$.

Algorithm 4 Subroutines of Algorithm 3

Input: $y = (y_1, y_2, ..., y_k) \in \mathbb{R}^{kn}$, $t \geq 1$, $\delta_1, \delta_2 \geq 0$, $1 \leq i \leq k$.

Function $SubR_1(y_1, y_2, ..., y_k, \delta_1, \delta_2, i)$ // no second splitting strategy

1: Set $j_1 < j_2 < ... < j_{k-1}$, where $\{j_1, j_2, ..., j_{k-1}\} = \{1, 2, ..., k\}\backslash\{i\}$.
2: **for** each $(t_{j_1}, ..., t_{j_{k-1}}) \in \mathcal{T}_{j_1}^{\delta_1} \times \mathcal{T}_{j_2}^{\delta_1} ... \times \mathcal{T}_{j_{k-1}}^{\delta_1}$ **do**
3: Compute the sets $\mathcal{V}_i = V \cap B_{\delta_2}\big(y_i - (-\sum_{j'} t_{j_{j'}})\big)$
4: **for** $v_i \in \mathcal{V}_i$ **do**
5: Add $(t_{j_1}, ..., t_{j_{i-1}}, v_i + (-\sum_{j'} t_{j_{j'}}), t_{j_i}, ..., t_{j_{k-1}})$ to the list $\mathcal{T}$.
6: **end for**
7: **end for**
8: **Return** $\mathcal{T}$.

Function $SubR_2(y_1, y_2, ..., y_k, \delta_1, \delta_2, i)$ // with the second splitting strategy

1: **for** $1 \leq l \neq i \leq k$ **do**
2: Set $j_1 < j_2 < ... < j_{k-2}$, where $\{j_1, j_2, ..., j_{k-2}\} = \{1, 2, ..., k\}\backslash\{i, l\}$.
3: **for** each $(t_l, t_{j_1}, ..., t_{j_{k-2}}) \in \mathcal{T}_l^{\delta_1} \times \mathcal{T}_{j_1}^{\delta_1/2} \times \mathcal{T}_{j_2}^{\delta_1/2} ... \times \mathcal{T}_{j_{k-2}}^{\delta_1/2}$ **do**
4: Compute the sets $\mathcal{V}_i^{\delta_2} = V \cap B_{\delta_2}\big(y_i - (-t_l - \sum_{j'} t_{j_{j'}})\big)$
5: **for** $v_i \in \mathcal{V}_i^{\delta_2}$ **do**
6: Add $(t_{j_1}, ..., t_l, ...t_{j_{i-1}}, v_i + (-t_l - \sum_{j'} t_{j_{j'}}), t_{j_i}, ..., t_{j_{k-1}})$ to the list $\mathcal{T}$.
7: **end for**
8: **end for**
9: **end for**
10: **Return** $\mathcal{T}$.

Lemma 2 (Complexity with the splitting strategy)**.** *Algorithm 3, with the subroutine $SubR_2$ listed in Algorithm 4,*

outputs the set $\Gamma(V,\beta,k)_{\mathcal{P}} \cap B_\delta(y)$ *in worst-case time*

$$\mathfrak{C}_{A.3} = k\mathfrak{C}_{T\cap B_\delta(y)} + (k^2-k)\Big[l(T,\delta)l(T,\frac{\delta}{2})^{k-2}\mathfrak{C}_{V\cap B_{\frac{2}{3}\delta}(y)} + l(T,\frac{2}{3}\delta)l(T,\frac{\delta}{3})^{k-2}\mathfrak{C}_{V\cap B_\delta(y)}\Big]. \tag{28}$$

C. Decoding paradigm for $\Gamma(V,\alpha,\beta,k)$

The proposed decoding algorithm simply uses representation (15) of the k-ing construction: $\Gamma(V,\alpha,\beta,k)$ is decoded via $|\alpha|$ use of Algorithm 1, as described in Algorithm 5. The complexity of Algorithm 5 is

$$\mathfrak{C}_{A.5} = |\alpha|(k\mathfrak{C}^T_{dec} + k\mathfrak{C}^V_{dec}). \tag{29}$$

Algorithm 5 Decoder for $\Gamma(V,\alpha,\beta,k)$

Input: $y=(y_1,y_2,...,y_k)\in\mathbb{R}^{kn}$.

1: **for** $m\in\alpha$ **do**
2: $y' \leftarrow y - m^k$
3: Use Algorithm 1 with y' as input.
4: **end for**
5: **Return** the closest element of $\mathcal{T}$ to y.

V. Decoders for the parity lattices

A. Recursive decoding

The decoding paradigms presented in the previous section can be adapted to decode lattices recursively built from the single parity-check construction. As an example, we adapt Algorithm 2 in the recursive Algorithm 6 to decode the parity lattices $L_{kn} = \Gamma(\theta L_n,\beta,k)$. Hence, we have $T = L_n$ and $V = \theta L_n$. Since $l(L_n,\delta) = l(\theta L_n,\delta)$, we set $l(n,\delta) = L(n,r) = l(L_n,\delta)$ to simplify the notations. Moreover, we also write $\mathfrak{C}(\delta)$ for $\mathfrak{C}(\frac{n}{k},\delta)$ and $l(\delta)$ for $l(\frac{n}{k},\delta)$.

In Algorithm 6 the "removing" steps (Steps 14 and 15) are added to ensure that a list with no more than $l(n,\delta)$ elements is returned by each recursive call. This enables to control the complexity of the algorithm (see e.g. Section V-D3). However, we shall see that the step in bold is not always necessary for the Gaussian channel.

Note that this algorithm with $\delta = 1/4$ yields a recursive BDD whose complexity is provided by the next theorem.

Theorem 5. *Let* $n = c\cdot k^t$ *and* $y\in\mathbb{R}^n$. *If* $d(y,L_n) < \rho^2(L_n)$, *then Algorithm 6 with* $\delta = 1/4$ *outputs the closest lattice point to* y *in time*

$$\mathfrak{C}_{A.6}(n,\frac{1}{4}) = O(n^{1+\frac{1}{log_2 k}}). \tag{30}$$

Proof.

$$\mathfrak{C}(n,\frac{1}{4}) = 2k\mathfrak{C}(\frac{n}{k},\frac{1}{4}) + O(n) = O(n)\sum_{i=0}^{\log_k n}\left(\frac{2k}{k}\right)^i,$$
$$= O(n^{1+\frac{1}{log_2 k}}).$$

□

Algorithm 6 Recursive list dec. for $L_{kn} = \Gamma(\theta L_n,\beta,k)_{\mathcal{P}} \subset \mathbb{R}^{kn}$, $n = c\cdot k^{t-1}$

Function $ListRecL(y,t,\delta)$
Input: $y=(y_1,y_2,...,y_k)\in\mathbb{R}^{kn}$, $0\le t$, $0\le\delta$.

1: **if** $t=0$ **then**
2: $\mathcal{T} \leftarrow$ The set $\Lambda_c \cap B_\delta(y)$.
3: **else**
4: $\mathcal{T}_1 \leftarrow ListRecL(y_1,t-1,\delta)$, $\mathcal{T}_2 \leftarrow ListRecL(y_2,t-1,\delta)$,..., $\mathcal{T}_k \leftarrow ListRecL(y_k,t-1,\delta)$.
5: **for** $1\le i\le k$ **do**
6: Set $j_1 < j_2 < ... < j_{k-1}$, where $\{j_1,j_2,...,j_{k-1}\} = \{1,2,...,k\}\backslash\{i\}$.
7: **for** each $(t_{j_1},...,t_{j_{k-1}})\in\mathcal{T}_{j_1}\times\mathcal{T}_{j_2}...\times\mathcal{T}_{j_{k-1}}$ **do**
8: $\mathcal{V}_i \leftarrow ListRecL([y_i - (-\sum_{j'} t_{j_{j'}})]\cdot R(c\cdot k^t,\theta)^T, t-1,\delta)\cdot R(c\cdot k^t,\theta)$.
9: **for** $v_i\in\mathcal{V}_i$ **do**
10: Add $(t_{j_1},...,t_{j_{i-1}},v_i + (-\sum_{j'} t_{j_{j'}}),t_{j_i},...,t_{j_{k-1}})$ in the list $\mathcal{T}$.
11: **end for**
12: **end for**
13: **end for**
14: **Remove all elements in $\mathcal{T}$ at a relative distance $>\delta$ from y.**
15: Sort the remaining elements in $\mathcal{T}$ in a lexicographic order and remove all duplicates.
16: **end if**
17: **Return** $\mathcal{T}$.

B. Decoding performance on the Gaussian channel

Lemma 3. *Let* $x\in\Lambda\subset\mathbb{R}^n$ *and let* $y\in\mathbb{R}^n$ *be the point to decode. Let* $\mathcal{T}$ *denote the list outputted by a list-decoding algorithm. The point error probability under list decoding is bounded from above by:*

$$P_e(dec) \le P_e(opt) + P(x\notin\mathcal{T}). \tag{31}$$

Proof.

$$P_e(dec) = P(y\notin\mathcal{V}(x)) + P(x\notin\mathcal{T}\cap y\in\mathcal{V}(x)),$$
$$\le P(y\notin\mathcal{V}(x)) + P(x\notin\mathcal{T}). \tag{32}$$

□

In the sequel, we derive formulas to estimate the term $P(x\notin\mathcal{T})$.

1) Choosing the decoding radius for regular list decoding on the Gaussian channel: Consider the Gaussian channel where $y = x + w$, with $y\in\mathbb{R}^{kn}$, $x\in L_{kn}$, and $w\in\mathbb{R}^{kn}$ with i.i.d $\mathcal{N}(0,\sigma^2)$ components. With a regular list decoder $\mathcal{T} = \Lambda\cap B_\delta(y)$ and

$$P(x\notin\mathcal{T}) = P(||w||^2 > r). \tag{33}$$

Since $||w||^2$ is a Chi-square random variable with n degrees of freedom, $P(||w||^2>r) = F(n,r,\sigma^2)$, where, for n even :

$$F(n,r,\sigma^2) = e^{-\frac{r}{2\sigma^2}}\sum_{k=0}^{n/2-1}\frac{1}{k!}\left(\frac{r}{2\sigma^2}\right)^k. \tag{34}$$

Lemma 4. *Consider Algorithm 6 with the following input parameters. The point* $y = x + w$, *where* $y\in\mathbb{R}^{kn}$, $x\in L_{kn}$,

and $w \in \mathbb{R}^{kn}$ with i.i.d $\mathcal{N}(0,\sigma^2)$ components. Moreover, $t \geq 0$ and $\delta = r/d(L_{kn})$. We have

$$P(x \notin \mathcal{T}) = F(kn, r, \sigma^2). \tag{35}$$

Based on (31), quasi-optimal performance with regular list decoding is obtained by choosing a decoding radius $r = E[||w||^2](1+\epsilon) = n\sigma^2(1+\epsilon)$ such that $F(n,r,\sigma^2) < \eta \cdot P_e(opt,\sigma^2)$ (in practice $\eta = 1/2$ is good enough). Moreover, it is easy to show that $\epsilon \to 0$ when $n \to +\infty$. We denote by δ^* the relative decoding radius corresponding to this specific r:

$$\delta^* = \frac{n\sigma^2(1+\epsilon)}{d(\Lambda)}. \tag{36}$$

Of course, the greater δ^*, the greater the list-decoding complexity.

2) A modified list-decoding algorithm: Notice that due to the "removing step" (Steps 14, in bold, of Algorithms 6), if a point found at the last recursive step is at a distance greater than r from y, even if it is the unique point found, it is not kept and an empty list is returned: The decoding radius is r in V and $r/2$ in T, but only the points at a distance less than r from y are kept.

To avoid this situation, we remove Step 14 in Algorithm 6. We will see in the rest of the paper that this enables to choose smaller decoding radii for QMLD than with regular list decoding and reduce the complexity despite the absence of the removing step. In terms of error probability, decoding in a sphere is the best choice given a finite decoding volume around the received point y. However, there may be larger non-spherical volumes that achieve satisfactory performance but that are less complex to explore. This is the main idea behind this modified list-decoding algorithm. This subsection concentrates on the analysis of the error probability of the modified algorithm.

Theorem 6. *Consider Algorithm 6 without Step 14 with the following input parameters. The point y is obtained on a Gaussian channel with VNR $\Delta = \mathrm{vol}(L_{kn})^{2/kn}/2\pi e\sigma^2$ as $y = x + w$, where $y \in \mathbb{R}^{kn}$, $x \in L_{kn}$, and $w \in \mathbb{R}^{kn}$ with i.i.d $\mathcal{N}(0,\sigma^2)$ components. Moreover, $t \geq 0$ and δ is the relative decoding radius. We have*

$$P(x \notin \mathcal{T}) \leq U_{kn}(\delta, \Delta), \tag{37}$$

where

$$U_n(\delta,\Delta) = \min\Big\{ \binom{k}{2} U_{\frac{n}{k}}(\delta, \frac{\Delta}{2^{\frac{1}{k}}})^2 + kU_{\frac{n}{k}}(\delta, 2^{\frac{k-1}{k}}\Delta)(1 - U_{\frac{n}{k}}(\delta, \frac{\Delta}{2^{\frac{1}{k}}}))^{k-1},\ 1\Big\}. \tag{38}$$

The initial condition $U_c(\delta,\Delta)$ corresponds to the decoding performance in L_c: $U_c(\delta,\Delta) = P(x \notin \mathcal{T}_c)$, where $\mathcal{T}_c$ denotes the list of candidates obtained when list decoding in L_c.

To help the reader understand the result, we provide the beginning of the proof below.

Proof. If Step 14 is removed at the last recursive iteration of Algorithm 6 the sent point $x = (x_1, x_2, ..., x_k)$ is not in the outputted list if:

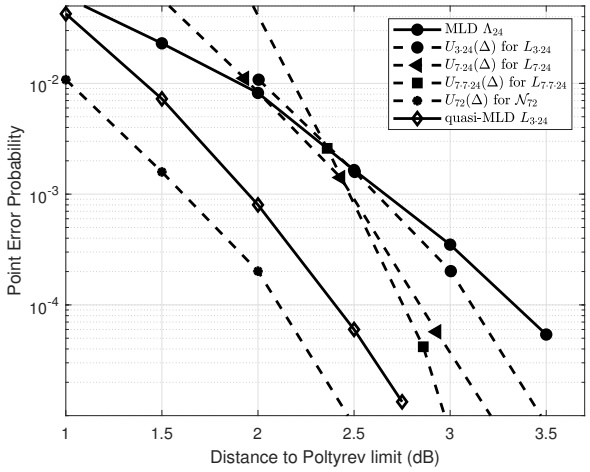

Fig. 2: Performance curves for Example 1.

- $x_i \notin \mathcal{T}_i$ for at least two lists $\mathcal{T}_i$ (at Step 4 of Algorithm 6),
- or if $x_1, ..., x_{j\neq i}, ..., x_k \in \mathcal{T}_1, ..., \mathcal{T}_j, ..., \mathcal{T}_k$, and $x_i - (-\sum_{j\neq i} x_j) \notin \mathcal{V}_i$ (for at least one i).

Let the noise $w = (w_1, ..., w_i, ..., w_k)$. Due to the i.i.d property of the noise, we have $P(||w_1||^2 > \frac{r}{2}) = P(||w_i||^2 > \frac{r}{2})$ for all $1 \leq i \leq k$. As a result, $P(x \notin \mathcal{T})$ becomes

$$\begin{aligned} P(x \notin \mathcal{T}) \leq & \binom{k}{2} P(||w_i||^2 > \frac{r}{2})^2 + \\ & kP(||w_i||^2 > r) P(||w_i||^2 < \frac{r}{2})^{k-1}, \\ = & \binom{k}{2} F(\frac{n}{2}, \frac{r}{2}, \sigma^2)^2 + kF(\frac{n}{2}, r, \sigma^2) F(\frac{n}{2}, \frac{r}{2}, \sigma^2)^{k-1}. \end{aligned} \tag{39}$$

The rest of the proof is provided in Appendix IX-B. □

For instance, a regular list decoder for L_c with relative decoding radius δ is used in Algorithm 6. Consequently, the initial condition would be

$$U_c(\delta,\Delta) = F(c, f(\delta), f(\Delta)), \tag{40}$$

with $f(\delta) = \delta \cdot d(L_c)$, and $f(\Delta) = \mathrm{vol}(L_c)^{\frac{2}{c}}/(2\pi e\Delta)$.

As illustrated by the next example, (38) means that the lattices of smaller dimensions are decoded with the same relative radius but with a VNR that is either greater, $2^{\frac{k-1}{k}}\Delta$, or smaller, $\Delta/2^{\frac{1}{k}}$. This result is a consequence of the following properties of the parity lattices: $\mathrm{vol}(L_n)^{\frac{2}{n}} = \mathrm{vol}(L_{kn})^{\frac{2}{kn}}/2^{\frac{1}{k}}$ and $d(L_n) = d(L_{kn})/2$ (see Appendix IX-B for justifications).

Example 1. *Let Λ_{24} be the Leech lattice and let $\beta = [\Lambda_{24}/\lambda\Lambda_{24}]$, where $\lambda = \frac{1+i\sqrt{7}}{2}$ (see Section III-C1 for more details on $\lambda\Lambda_{24}$). The k-parity-Leech lattices are defined as $L_{kn} = \Gamma(\lambda L_n, \beta, k)_{\mathcal{P}}$ with initial condition $\Lambda_c = \Lambda_{24}$. On the Gaussian channel and with Algorithm 6 without Step 14, the probability that the transmitted lattice point is not in the outputted list is given by* (38)*. We let the initial condition $U_{24}(\Delta)$ be the performance of the optimal decoder for Λ_{24} (δ is thus irrelevant in this case). It means that Step 2 of Algorithm 6 is modified by using a MLD decoder for Λ_{24}.*

For the lattice $L_{k\cdot 24}$, the value of $U_{k\cdot 24}(\Delta)$ is obtained (using (38)*) by adding the performance curves representing*

- $\binom{k}{2}\cdot(P_e^{\Lambda_{24}}(opt,\Delta))^2$*, shifted by* $10\log_{10}(2^{\frac{1}{k}})$ *dB to the right,*
- *and* $k\cdot P_e^{\Lambda_{24}}(opt,\Delta)$ *shifted by* $10\log_{10}(2^{\frac{k-1}{k}})$ *dB to the left (assuming that* $(1-U_{\frac{n}{k}}(\frac{\Delta}{2^{\frac{1}{k}}}))^{k-1}\approx 1$*).*

The results for $k=3$ and $k=7$ are shown by the dashed line in Figure 2. Since there is only one recursive step, Algorithm 6 is equivalent to Algorithm 2. The associated decoding complexity is obtained from (24) *where the term $l(T,\delta)=1$ (note that* (24) *reduces to* (22) *in this case). Consequently, we get*

$$\mathfrak{C}(L_{k\cdot 24})=2k\mathfrak{C}_{MLD}^{\Lambda_{24}}+O(k\cdot 24). \tag{41}$$

The probability $U_{k\cdot k\cdot 24}(\Delta)$ is obtained in a similar manner from $U_{k\cdot 24}(\Delta)$. For instance, $U_{7\cdot 7\cdot 24}(\Delta)$ is also plotted in the figure. The corresponding decoding complexity of $L_{k\cdot k\cdot 24}$ in this case is obtained from (24) *(with $l(T,\delta)=k$, the number of candidates obtained at the previous recursive step):*

$$\begin{aligned}\mathfrak{C}(L_{k\cdot k\cdot 24}) =& k(1+k^{k-1})\mathfrak{C}(L_{k\cdot 24})+O(k\cdot k\cdot 24),\\ \approx& 2k^{k+1}\mathfrak{C}_{MLD}^{\Lambda_{24}}+O(k^2 24).\end{aligned} \tag{42}$$

Figure 2 also depicts the QMLD performance of $L_{3\cdot 24}$ (obtained in Section V-C) for comparison.

We shall see in the next section that the Nebe lattice $\mathscr{N}_{72}$, constructed as $\Gamma(\lambda\Lambda_{24},\alpha,\beta,3)$, has the following properties: $\mathrm{vol}(\mathscr{N}_{72})^{\frac{2}{n=72}}=\mathrm{vol}(T=\Lambda_{24})^{\frac{2}{n/3}}$ *and* $d(\mathscr{N}_{72})=2d(\Lambda_{24})$*. Consequently,* (38) *becomes*

$$U_{n=72}(\Delta)=\min\Big\{3U_{\frac{n}{3}}(\Delta)^2+3U_{\frac{n}{3}}(2\Delta)(1-U_{\frac{n}{3}}(\delta,\Delta))^2,\ 1\Big\}. \tag{43}$$

Taking $U_{\frac{n}{3}}(\Delta)=P_e^{\Lambda_{24}}(opt,\Delta)$, we get a similar curve as $U_{3\cdot 24}$ for $L_{3\cdot 24}$ but shifted by $10\cdot\log_{10}(2^{1/3})=1$ dB to the left. The curve $U_{72}(\Delta)$ is shown in Figure 2. See Section VI-C and Figure 6 for more details on the quasi-MLD performance of $\mathcal{N}_{72}$.

If the (first) splitting strategy is considered (e.g. in a recursive version of Algorithm 3), the error probability is slightly greater due to specific cases, such as having simultaneously $\frac{2}{3}\frac{r}{2}<||w_j||<\frac{r}{2}$ and $\frac{2}{3}r<||w_i||<r$, which are not correctly decoded (whereas they were without the splitting strategy). For the case $k=2$, it is shown in Appendix IX-B that with the splitting strategy we get the recursion

$$\begin{aligned}U_n(\delta,\Delta)=\min\Big\{&U_{\frac{n}{2}}(\delta,\frac{\Delta}{\sqrt{2}})^2+\\ &2\Big[\big(U_{\frac{n}{2}}(\frac{2}{3}\delta,\frac{\Delta}{\sqrt{2}})-U_{\frac{n}{2}}(\delta,\frac{\Delta}{\sqrt{2}})\big)U_{\frac{n}{2}}(\frac{2}{3}\delta,\sqrt{2}\Delta)\\ &+(1-U_{\frac{n}{2}}(\frac{2}{3}\delta,\frac{\Delta}{\sqrt{2}}))U_{\frac{n}{2}}(\delta,\sqrt{2}\Delta)\Big],\ 1\Big\}.\end{aligned} \tag{44}$$

C. The 3-parity-Leech lattice in dimension 72

Consider the 3-parity-Leech lattice $L_{3\cdot 24}=\Gamma(\lambda\Lambda_{24},[\Lambda_{24}/\lambda\Lambda_{24}],3)_{\mathcal{P}}$ presented in Example 1. $L_{3\cdot 24}$ has the same minimum distance as $\mathscr{N}_{72}$ and a volume $\mathrm{vol}(L_{3\cdot 24})=\mathrm{vol}(\mathscr{N}_{72})\times|\alpha|$ (using (7)). Its fundamental coding gain is:

$$\gamma(L_{3\cdot 24})=\gamma(\Gamma(2S,T_{2\omega},T,3))\times\frac{1}{(2^{12})^{\frac{2}{72}}}\approx 6.35. \tag{45}$$

Lemma 5. *The kissing number of $L_{3\cdot 24}$ is 28,894,320.*

The proof is provided in Appendix IX-C. The kissing number is about $2^{7.75}$ smaller than the kissing number of $\mathscr{N}_{72}$ (which is $6,218,175,600$). As a result, one can state the following regarding the relative performance of these two lattices on the Gaussian channel: 1 dB is lost by the parity-Leech lattice due to a smaller γ, but using the rule of thumb that 0.1 dB is lost each time the kissing number is doubled [19], there is also an improvement of 0.8 dB. Overall, we expect the performance of these two lattices to be only 0.2 dB apart but where the decoding complexity of the 3-parity-Leech lattice is significantly reduced compared to $\mathscr{N}_{72}$ (see Section VI). The QMLD performance is shown in Figure 6 and it is indeed at 0.2 dB from the one of $\mathscr{N}_{72}$.

Consider Algorithm 2 for decoding. (38) yields

$$U_{3\cdot 24}(\delta,\Delta)=3U_{24}(\delta,\frac{\Delta}{2^{\frac{1}{3}}})^2+3U_{24}(\delta,2^{\frac{2}{3}}\Delta)(1-U_{24}(\delta,\frac{\Delta}{2^{\frac{1}{3}}}))^2. \tag{46}$$

A MLD decoder for Λ_{24} as subroutine is not powerful enough to get QMLD performance (see the curve for $U_{3\cdot 24}$ on Figure 2). We can for instance consider a sphere decoder computing $\Lambda_{24}\cap B_{\delta\cdot d(\Lambda_{24})}(y)$. Then $U_{24}(\delta,\Delta)=F(24,\delta\cdot d(\Lambda_{24}),\sigma^2)$ and the relative decoding radius δ^* should be chosen such that $3\cdot F(24,\delta^*\cdot d(\Lambda_{24}),\sigma^2)^2\approx 1/2\cdot P_e^{L_{3\cdot 24}}(opt,\sigma^2)$. We find $\delta^*\approx 25/64$. With Theorem 1 we get that $l(\Lambda_{24},\delta^*)=4$. The (worst-case) complexity of Algorithm 2, given by Lemma 1, becomes

$$\begin{aligned}\mathfrak{C}_{QMLD}^{L_{3\cdot 24}}=&3\mathfrak{C}_{\Lambda_{24}\cap B_{\delta^*\cdot d(\Lambda_{24})}(y)}+3l(\Lambda_{24},\delta^*)^2\mathfrak{C}_{\Lambda_{24}\cap B_{\delta^*\cdot d(\Lambda_{24})}(y)},\\ =&51\cdot\mathfrak{C}_{\Lambda_{24}\cap B_{\delta^*\cdot d(\Lambda_{24})}(y)}.\end{aligned} \tag{47}$$

D. Parity lattices with $k=2$: Barnes-Wall lattices

1) Existing algorithms: Several algorithms have been proposed to decode BW lattices. [16] uses the trellis representation of the two-level squaring construction to introduce an efficient MLD algorithm for the low dimension instances of BW_n. Nevertheless, the complexity of this algorithm is intractable for $n>32$: The number edges in the trellis is $2\cdot 2^{2n/8}+2\cdot 2^{3n/8}$, e.g. decoding in BW_{128} involves $2\cdot 2^{48}+2\cdot 2^{32}$ decoders of BW_{32}. Forney states in [16] : "The first four numbers in this sequence[6], i.e., 2, 4, 16, and 256, are well behaved, but then a combinatorial explosion occurs: 65 536 states for BW_{64}, which achieves a coding gain of 7.5 dB, and more than four billion states for BW_{128}, which achieves a coding gain of 9 dB. This explosion might have been expected from capacity and R_0 (cut-off rate) considerations".

Later, [33] proposed the first BDDs running in polynomial time; a parallel version of complexity $O(n^2)$ and a sequential one of complexity $O(n\log^2 n)$. The parallel decoder was generalized in [22] to work beyond the packing radius, still in polynomial time. It is discussed later in the paper. The sequential decoder uses the BW multilevel construction to perform multistage decoding: Each of the $\approx\log n$ levels is decoded with a Reed-Muller decoder of complexity $n\log n$.

[6]Forney refers to the number of states per section of the trellis, which is $2^{2n/8}$.

This decoder was also further studied, in [23], to design practical schemes for communications over the AWGN channel. However, the performance of this sequential decoder is far from MLD. A simple information-theoretic argument explains why multistage decoding[7] of BW lattices cannot be efficient: The rates of some component Reed-Muller codes exceed the channel capacities of the corresponding levels [20][47].

As a result, no BW decoder, being both practical and quasi-optimal on the Gaussian channel, have been designed and executed for dimensions greater than 32.

2) A new BDD: Algorithm 6 with $k = 2$ and $\delta = 1/4$ yields a new BDD for the Barnes-Wall lattices. We apply Theorem 5 for the case $k = 2$.

Corollary 1. *Let $n = 2^{t+1}$ and $y \in \mathbb{R}^n$. If $d(y, BW_n) < \rho^2(BW_n)$, then Algorithm 6 with $k = 2$ and $\delta = 1/4$ outputs the closest lattice point to y in time $O(n^2)$.*

3) List decoding BW lattices: The recursive version of Algorithm 3, with $k = 2$, can be used to list decode the BW lattices. For regular list decoding the removing and sorting steps, as in Algorithm 6, should also be added. Let us name it Algorithm 3'.

We investigate the complexity of the algorithm of [22] and Algorithm 3'. As mentioned at the beginning of the section, [22] adapts the parallel BDD of [33], which uses the automorphism group of BW_n, to output a list of all lattice points lying at a distance $r = d(BW_n)(1-\epsilon)$, $0 < \epsilon \leq 1$, from any $y \in \mathbb{R}^n$ in time

$$O(n^2) \cdot L(n, r^2)^2. \tag{48}$$

A critical aspect regarding the complexity of this decoder is therefore the list size. Theorem 1 provides bounds on the list size when $r \leq d(BW_n)/2$. The following lemma, addressing $r > d(BW_n)/2$, is proved in [22].

Lemma 6 (Results from [22])**.** *The list size of BW_n lattices is bounded as:*

- $L(n, r) = O(n^{\log_2 4\lfloor \frac{3}{4\epsilon} \rfloor})$ *if* $r \leq d(BW_n)(\frac{3}{4} - \epsilon)$, $0 < \epsilon < \frac{1}{4}$.
- $L(n, r) = O(n^{2\log_2 24})$ *if* $r = \frac{3}{4}d(BW_n)$.
- *And* $L(n, r) = O(n^{8\log_2 \frac{1}{\epsilon}})$, *if* $r \leq d(BW_n)(1-\epsilon)$, $0 < \epsilon < \frac{1}{4}$.

Lemma 6 shows that the list size of BW lattices is of the form $n^{O(\log \frac{1}{\epsilon})}$ and thus polynomial in the lattice dimension for any radius bounded away from the minimum distance. Combining the lemma with (48), the list decoder complexity becomes $n^{O(\log \frac{1}{\epsilon})}$ for any $r < d(BW_n)(1-\epsilon)$, $\epsilon > 0$. This result is of theoretical interest: It proves that there exists a polynomial time decoding algorithm (in the dimension) for any radius bounded away from the minimum distance. However, the quadratic dependence is a drawback: As already explained, finding an algorithm with quasi-linear dependence in the list size is stated as an open problem in [22].

In the following, we demonstrate that if we use Algorithm 3', rather than the automorphism group of BW_n for list decoding, we get complexity linear in the list size. This enables to both improve the regular list-decoding complexity and get a practical quasi-optimal decoding algorithm on the Gaussian channel up to $n = 128$.

We compute below the complexity of our algorithm for $\delta < 9/16$. The complexity analysis for larger δ (which is the proof of Theorem 7) is provided in Appendix IX-D.

If $\delta < \frac{3}{8}$ then $\frac{2}{3}\delta < \frac{1}{4}$ and we have $l(\delta) = O(1)$, $l(\frac{2}{3}\delta) = 1$. Moreover, $\mathfrak{C}(\frac{2}{3}\delta) \leq \mathfrak{C}(\frac{1}{4}) = O(n^2)$ (Theorem 1). The complexity becomes

$$\mathfrak{C}(n, \delta) = 4\mathfrak{C}(\delta) + l(\delta)O(n^2) = l(\delta)O(n^2 \log n) = \widetilde{O}(n^2). \tag{49}$$

If $\frac{3}{8} \leq \delta < \frac{9}{16}$ and $\mathfrak{C}(\frac{2}{3}\delta) \leq \mathfrak{C}(\frac{3}{8}) = O(n^2 \log n)$. We get

$$\begin{aligned} \mathfrak{C}(n, \delta) &= l(\delta)O(n^2 \log n) \sum_{i=0}^{\log_2 n} \left(\frac{2l(\frac{2}{3}\delta) + 2}{4} \right)^i, \\ &= l(\delta)O(n^{1+\log_2[1+l(\frac{2}{3}\delta)]} \log n), \\ &= l(\delta)\tilde{O}(n^{1+\log_2[1+l(\frac{2}{3}\delta)]}), \end{aligned} \tag{50}$$

which is $\tilde{O}(n^{1+\log_2 3})$ if $\delta < 1/2$.

Note that for these cases ($\delta < 1/2$) the decoder of [22] is more efficient: Indeed, Theorem 1 shows that when $\delta < 1/2$ then $l(n, \delta) = O(1)$ and the decoding complexity, given by (48), is $O(n^2)$. Nevertheless, the following theorem (proved in Appendix IX-D) shows that our decoder is better for larger values of δ and, as we shall see in the next subsection, is useful even when $\delta < 1/2$ for quasi-optimal decoding on the Gaussian channel.

Theorem 7. *Let $n = 2^{t+1}$, $y \in \mathbb{R}^n$. The set $BW_n \cap B_\delta(y)$ can be computed in worst-case time:*

- $O(n^2)$ *if* $\delta < \frac{1}{2}$ *(algorithm of [22]).*
- $l(\delta)O(n^{2+\log_2[\frac{l(\frac{2}{3}\delta)+1}{2}]}) \approx O(n^{1+\log_2 4\lfloor \frac{3}{4\epsilon} \rfloor^2])$ *if* $\delta = \frac{3}{4} - \epsilon$, $0 < \epsilon$.
- $O(n^{1+\log_2 432})$ *if* $\delta = \frac{3}{4}$.
- $l(\delta)O(n^2) = O(n^{8\log_2 \frac{1}{\epsilon}+2})$ *if* $\delta = 1 - \epsilon$, $0 < \epsilon < \frac{1}{4}$.

4) Decoding on the Gaussian channel: We apply the analysis presented in Section V-B to the case $k = 2$ to establish the smallest list-decoding radius δ required for quasi-optimal decoding. The first element needed is the MLD performance $P_e^{BW_n}(opt, n, \sigma^2)$ of BW_n. As mentioned earlier it is not known for $n > 32$. Nevertheless, $P_e^{\Lambda}(opt, n, \sigma^2)$ can be lower-bounded for *any* lattice Λ in n dimensions using the sphere lower bound [41] (see also [20] or [24]). Table I provides the sphere lower bound on the best performance achievable for $P_e^{\Lambda}(opt, n, \sigma^2) = 10^{-5}$. With (34), we can compute the smallest δ^*, for the corresponding values of σ^2, such that $P(x \notin \mathcal{T}) = P(||w||^2 > r) \lessapprox 10^{-5}$ with regular list decoding. Using δ^* yields quasi-optimal decoding performance, regardless of the MLD performance of BW_n. The values of δ^* as a function of n are presented in Figure 3. The corresponding (worst-case) decoding complexity is obtained with Theorem 7. It is super-quadratic for all $n \geq 16$.

Running the simulations (with δ^* found at the sphere bound) enables to estimate the MLD performance of BW_n lattices. The results are presented[8] in Table I, and are at ≈ 0.5 dB of

[7] Where only one candidate is decoded at each level.

[8] These estimations were not performed with the regular list decoder, but with the algorithm presented in the rest of the section.

Dimension n	16	32	64	128	256
Dist. to Polt. (dB) sphere bound	4.05	3.2	2.5	1.9	1.4
Dist. to Polt. (dB) MLD	4.5	3.7	3.1	2.3	?

TABLE I: Sphere lower bound on the best performance achievable by any lattice Λ for $P_e^{\Lambda}(opt, n, \sigma^2) = 10^{-5}$ and MLD performance of BW_n for $P_e(opt, n, \sigma^2) = 10^{-5}$.

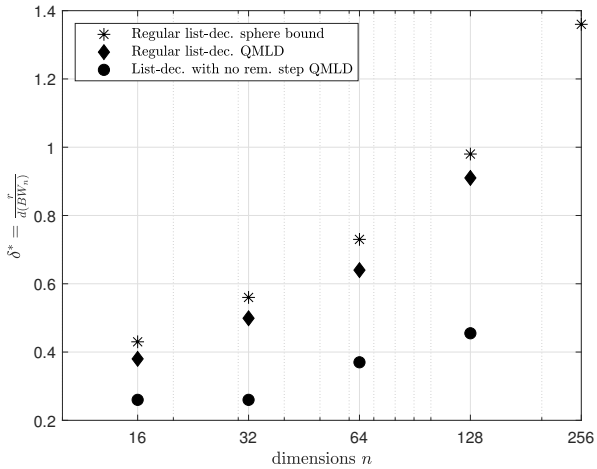

Fig. 3: Values of the list-decoding relative radius δ^*, for BW_n, such that $P(x \notin \mathcal{T}) \approx 10^{-5}$.

the sphere bound[9]. In Figure 3 the corresponding values of δ^* (still with regular list decoding) are depicted by the diamonds. For $n = 128$, we have $\delta^* > 3/4$. Even though the complexity of the regular list decoder is linear in the list size, it remains $O(n^{8\log_2 \frac{1}{\epsilon}+2})$ for $\delta = 1-\epsilon$, $0 < \epsilon < 1/4$. This is much too high to run on a computer in a reasonable time (linear in the list size doesn't mean low complexity in this case). Consequently, we consider a slightly different strategy to get a practical algorithm for the Gaussian channel.

As explained in Section V-B, the error probability of the list-decoding algorithm without the removing step can be estimated with Equation (38) without the splitting strategy or (44) with the first splitting strategy. Hence, we can also compute the smallest δ^* such that with this algorithm $P(x \notin \mathcal{T}^{\delta}) \lessapprox 10^{-5}$ (at the MLD performance). However, the decoding complexity should be updated to take into account the fact that there is no removing step in the algorithm even when $\delta < 1/2$.

We consider Algorithm 3' without the removing step. To mitigate the complexity and simplify the analysis, whenever $\delta \leq 1/4$ we shall use the BDD presented in Section V-D2 (i.e. when $\delta \leq 1/4$, we fix it to $1/4$). Hence, in (44), $U_n(\delta \leq \frac{1}{4}, \Delta) = P_e(BDD, n, \Delta)$. The error probability $P_e(BDD, n, \Delta)$ is shown in Figure 4. In the literature, the performance of BDDs is often estimated via the "effective error coefficient" [18] [39]. Nevertheless, it it not always accurate, especially in high dimensions. We therefore rely on the Monte Carlo simulations presented in the figure for

[9]We have not yet investigated the case $n = 256$.

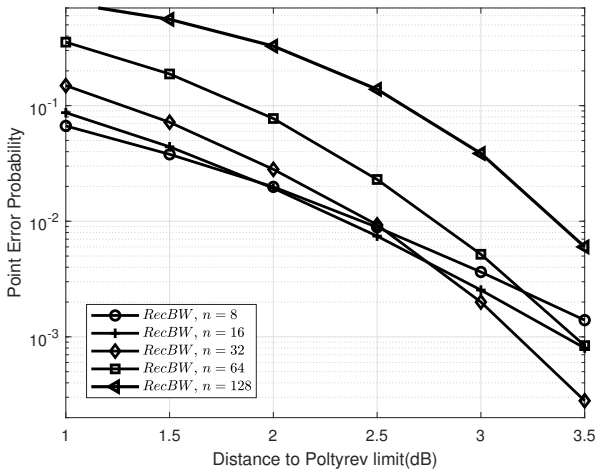

Fig. 4: Performance of the recursive BDD for the Barnes-Wall lattices on the Gaussian channel.

$P_e(BDD, n, \Delta)$. The estimated δ^* with this decoder, shown in Figure 3, are significantly smaller than the ones obtained with the regular list decoder. In particular, $\delta^* < 3/8$ for $n \leq 64$ and $\delta^* < 1/2$ for $n = 128$.

We now study the complexity of this latter algorithm. We shall use the notation $l'(n, \delta, y)$ to denote the number of elements returned by the decoding algorithm (without a removing step). If $\delta \leq 3/8$, we get

$$\begin{aligned} l'(n,\delta) &\leq 2\big[l'(\frac{1}{4})l'(\delta) + l'(\delta)l'(\frac{1}{4})\big] = 4l'(\delta) \\ &= 4^{\log_2 n} \cdot l(\mathbb{Z}_2, \delta) = O(n^2). \end{aligned} \tag{51}$$

However, considering the average complexity, and taking into account the fact that we remove the duplicates at each recursive step, one has

$$\begin{aligned} E_y[l'(n,\delta,y)] \leq & 2\big[l'(\frac{1}{4})E_y[l'(\delta)] + E_y[l'(\delta)]l'(\frac{1}{4}) - \\ & l'(\frac{1}{4})l'(\frac{1}{4})\big] - E_y^c. \\ = & 4E_y[l'(\delta)] - 2 - E_y^c, \end{aligned} \tag{52}$$

where E_y^c denotes the average number of common elements in the lists returned by the recursive calls. We observed experimentally that for $\delta \leq 3/8$, $E_y[l'(n, \delta, y)]$ is close to 1. This observation is not taken into account in the next theorem, which bounds the average list size and the average complexity. It is however in the interpretation following the theorem.

Theorem 8. *Let $E_y[l'(n, \delta, y)]$ be the average list size of Algorithm 3' without the removing step. Let η denote $E_y[l'(n, 3/8, y)]$. If $3/8 < \delta \leq 9/16$, $E_y[l'(n, \delta, y)]$ is bounded from above as*

$$E_y[l'(n,\delta,y)] = O(n^{2+\log_2 \eta}). \tag{53}$$

And the average complexity is bounded from above as:

- $E_y[\mathfrak{C}(n, \delta)] = \eta\widetilde{O}(n^2)$ *if* $\delta \leq 3/8$.
- $E_y[\mathfrak{C}(n, \delta)] = E_y[l'(\delta, y)]\widetilde{O}(n^{1+\log_2[1+\eta]})$ *if* $3/8 < \delta \leq 9/16$.

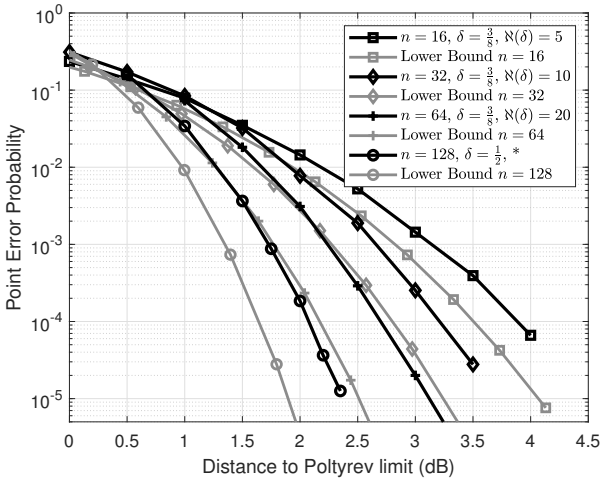

Fig. 5: Simulation results for the BW lattices up to $n = 128$ and the universal bounds of [41]. *For $n = 128$, $\aleph(\delta) = 1000$ and $\aleph(2/3\delta) = 4$.

See Appendix IX-E for the proof.

As a result, based on the observation that η is close to 1, the average complexity is estimated as:

- $E_y[\mathfrak{C}(n, \delta)] = \widetilde{O}(n^2)$ if $\delta \leq 3/8$.
- $E_y[\mathfrak{C}(n, \delta)] = \widetilde{O}(n^4)$ if $3/8 < \delta \leq 9/16$.

Since $\delta^* < 3/8$ for $n < 64$ and $\delta^* < 1/2$ for $n = 128$ for QMLD, we conclude that the decoding complexity is quadratic for $n \leq 64$ and quartic for $n = 128$.

For a practical implementation, we can use Algorithm 3' (without the removing step) whose advantage is to have the above theoretical analysis in terms of performance and complexity. Alternatively, we can perform the following minor modifications which improves the computational efficiency but makes a performance analysis not possible (the parameters mus be found via trial and error).

We can bound the maximum number of points kept at each recursive step: At the end of each recursive call, the $\aleph(\delta)$ best candidates are kept. The size of the list $\aleph(\delta)$, for a given δ, is a parameter to be fine tuned: For n =16, 32, 64, we set $\aleph(\delta)$ =5, 10, 20, respectively. For $n = 128$, choosing $\aleph(\delta) = 1000$ ($<<$ than our bound in $O(n^2)$ on $E_y[l'(\delta, y)]$) and $\aleph(2/3\delta) = 4$ yields quasi-MLD performance. Figure 5 depicts the simulation results for BW lattices up to $n = 128$.

VI. Decoders for Leech and Nebe lattices

A. Existing decoding algorithms for Λ_{24} and $\mathscr{N}_{72}$

1) History of the decoders of Λ_{24}: Λ_{24} appeared under many different forms in the literature (which may be equivalent to Turyn's construction). Among others, Λ_{24} can be obtained as (i) 8192 cosets of $4D_{24}$, (ii) 4096 cosets of $(\sqrt{2}E_8)^3$, (iii) 2 cosets of the half-Leech lattice H_{24}, where H_{24} is constructed by applying Construction B on the Golay code C_{24}, and (iv) 4 cosets of the quarter-Leech lattice, where quarter-Leech lattice is also built with Construction B but applied on a subcode of C_{24}. Finally, one of the simplest constructions is due to [5], where the Leech lattice is obtained via Construction A applied on the quaternary Golay code.

The history of maximum-likelihood decoding (MLD) algorithms for Λ_{24} starts with [8], where Conway and Sloane used (i) to compute the second moment of the Voronoi region of Λ_{24}. The first efficient decoder was presented in [9] by the same authors using construction (ii). Two years later, Forney reduced the complexity of the decoder by exploiting the same construction (ii), which he rediscovered in the scope of the "cubing construction", with a 256-state trellis diagram representation [16] (see Section IV-A for a presentation of trellis). A year later, it was further improved in [25] and [4] thanks to (iii) combined with an efficient decoder of C_{24}. Finally, (iv) along with the hexacode is used to build the fastest ever known MLD decoder by Vardy and Be'ery [44].

To further reduce the complexity, (suboptimal) BDD were also investigated based on the same constructions: e.g. [17] with (iii) and [2][45][18] with (iv). In these papers, it is shown that these BDD do not change the error exponent (i.e. the effective minimum distance is not diminished) but increase the "equivalent error coefficient". The extra loss is roughly 0.1 dB on the Gaussian channel compared to the optimal performance. As we shall see in the sequel, our decoding paradigm applied to the Leech lattice is more complex than the state-of-the-art decoders of Vardy [45][18] which requires only $\approx$ 300 real operations. But again, this latter decoder is specific to the Leech lattice whereas our decoder is more universal as it can be used, among others, to decode the Nebe lattice and the Barnes-Wall lattices.

2) The decoder of the Nebe lattice in [31]: While the decoding of Λ_{24} has been extensively studied, the literature on decoders for $\mathscr{N}_{72}$ is not as rich: Only [31] studied this aspect, but the proposed decoder is highly suboptimal.

First, notice that we can multiply (on the left) the matrix Pb given in (21) by a unimodular matrix to get the following matrix Pb':

$$Pb' = \begin{bmatrix} 1 & 1 & \lambda \\ 0 & \psi & \psi \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ \psi & -\lambda & 0 \end{bmatrix} \cdot Pb. \tag{54}$$

Similarly to (21), $Pb' \otimes G_S^{\mathbb{C}}$, $S \cong \Lambda_{24}$, is a basis for the Nebe lattice which induces the following structure:

$$\begin{aligned} \mathscr{N}_{72} = \{(a, b, c) \in \mathbb{C}^{36} : \\ a \in S, b - a \in T, c - (b - a) - \lambda a \in 2S\}, \end{aligned} \tag{55}$$

where (55) is derived from the columns of Pb'. A successive-cancellation-like algorithm can thus be considered: given $y = (y_1, y_2, y_3)$ in $\mathbb{C}^{36}$, y_1 is first decoded in S as t_1, $y_2 - t_1$ is then decoded in T as t_2, and $y_3 - t_2 - \lambda t_1$ is decoded in $2S$ as t_3. In [31], this successive-cancellation algorithm is proposed, with several candidates for t_1 which are obtained via sphere decoding with a given radius r. Among all resulting approximations, the closest to y is kept. It is proved in [31], that the lattice point $\hat{x}$ outputted by the algorithm using a decoding radius $r = R(S)$, the covering radius of S, has an approximation factor $||y - \hat{x}|| \leq \sqrt{7}||y - x_{opt}||$. Additionally, this algorithm is guaranteed to output the closest point $x_{opt} \in$

$\Gamma(2S, T_{2\omega}, T, 3)$ to y if $d(y, x_{opt}) \leq R(S)$, where $R(S)$ is unfortunately smaller by a factor $\sqrt{2}$ than the packing radius $\rho(\mathscr{N}_{72})$.

B. New decoders for $\Gamma(2S, T_{2\omega}, T, 3)$

We first adapt Algorithm 5 to $\Gamma(2S, T_{2\omega}, T, 3)$ by choosing the decoders for T and $V = 2S$ as BDDs. We name it Algorithm 5'.

Theorem 9. *Let $\Gamma(2S, T_{2\omega}, T, 3)$ and y be respectively a lattice and a point in $\mathbb{R}^{3n}$. If $d(y, \Gamma(2S, T_{2\omega}, T, 3)) < \rho^2(\Gamma(2S, T_{2\omega}, T, 3))$, then Algorithm 5' outputs the closest lattice point $x \in \Gamma(2S, T_{2\omega}, T, 3)$ to y in time*

$$\mathfrak{C}_{A.5'} = 6|\alpha|\mathfrak{C}^S_{BDD}. \tag{56}$$

Proof. We first show that x is the closest lattice point to y. Assume that, at Steps 1-2, m corresponds to the coset of the closest lattice point to y. Then, the result follows from Theorem 1 since Algorithm 5' is a special case of Algorithm 2 used α times.
Regarding the complexity, we use Equation (29) with $k = 3$ and where $\mathfrak{C}^S_{BDD} = \mathfrak{C}^{2S}_{BDD}$. □

It is insightful to compare Algorithm 5' to trellis decoding. The complexity is reduced from $\approx |\alpha||\beta|^2\mathfrak{C}^S_{CVP}$ to $\approx |\alpha|\mathfrak{C}^S_{BDD}$ (but where trellis decoding is optimal unlike Algorithm 5').

We name Algorithm 3" the list-decoding version of Algorithm 5': It consists in repeating $|\alpha|$ times (once for each coset of $\Gamma(2S, \beta, 3)_{\mathcal{P}}$) Algorithm 3, with k=3, using the first splitting strategy and the second splitting strategy (i.e. the function $SubR_2$). We use Lemma 2 to get the following theorem.

Theorem 10. *Let $\Gamma(2S, T_{2\omega}, T, 3)$ and y be respectively a lattice and a point in $\mathbb{R}^{3n}$. Algorithm 3" outputs the set $\Gamma(2S, T_{2\omega}, T, 3) \cap B_\delta(y)$ in worst-case time*

$$\begin{aligned}\mathfrak{C}_{A.3''}(\delta) = |\alpha|\big[&3\mathfrak{C}_{T\cap B_\delta(y)} + 6l(T, \delta)l(T, \frac{\delta}{2})\mathfrak{C}_{V\cap B_{\frac{2}{3}\delta}(y)} + \\ &6l(T, \frac{2}{3}\delta)l(T, \frac{\delta}{3})\mathfrak{C}_{V\cap B_\delta(y)}\big].\end{aligned} \tag{57}$$

Corollary 2. *Let $\Lambda_{24} = \Gamma(2S, T_{2\omega}, T, 3)$ (constructed as in Lemma 3). Algorithm 3" with a decoding radius $r = d(T) = d(E_8)$, i.e. $\delta = d(E_8)/d(\Lambda_{24}) = 1/2$, solves the CVP for Λ_{24} with worst-case complexity*

$$\begin{aligned}\mathfrak{C}_{A.3'}(\delta = \frac{1}{2}) =&|\alpha|\big[3\mathfrak{C}_{T\cap B_\delta(y)} + 6[2n2\mathfrak{C}_{E_8\cap B_{\frac{1}{3}}(y)} + 3\mathfrak{C}_{E_8\cap B_{\frac{1}{2}}(y)}]\big], \\ \lessapprox&2^4 \cdot 6 \cdot 2 \cdot 8 \cdot 2 \cdot \mathfrak{C}_{E_8\cap B_{\frac{1}{2}}(y)} \approx 2^{11}\mathfrak{C}_{E_8\cap B_{\frac{1}{2}}(y)}.\end{aligned} \tag{58}$$

Proof. If $S \cong E_8$, $d(T) = R^2(\Gamma(2S, T_{2\omega}, T, 3))$ (the covering radius). □

To the best of our knowledge, the covering radius of $\mathscr{N}_{72}$ appears nowhere in the literature. However, Gabriele Nebe showed in a private communication that it is greater than $\sqrt{2}\rho(\mathscr{N}_{72})$. The proof is available in Appendix IX-F. As a result, Algorithm 3" with $\delta = 1/2$ is not optimal for $\mathscr{N}_{72}$. The algorithm should be used with greater δ to ensure optimality.

C. Decoding Λ_{24} and $\mathscr{N}_{72}$ on the Gaussian channel

The analysis is similar to the one performed for BW lattices in Section V-D4. We will therefore be brief on the explanations.

The sphere lower bound for $P_e^\Lambda(opt, n, \sigma^2) = 10^{-4}$ in dimension 72 yields a distance to Poltyrev limit of 2.1 dB. The MLD performance of Λ_{24} for this error probability is 3.3 dB. Regarding the relative radius to ensure $P(x \notin \mathcal{T}^\delta) = P(||w||^2 > r) \lessapprox 10^{-4}$ with regular list decoding, we find with Equation (34) $\delta^* \approx 0.57$ for $\mathscr{N}_{72}$ and $\delta^* \approx 0.41$ for Λ_{24}.

An important observation (also made at the end of Example 1) when computing the performance of the modified list decoders on the Gaussian channel is the following. Let $T \in \mathbb{R}^n$. For Λ_{24} and $\mathscr{N}_{72}$ constructed as $\Gamma(2S, T_{2\omega}, T, 3)$, we have (see e.g. the proof of Theorem 3) $\mathrm{vol}(\Gamma(2S, T_{2\omega}, T, 3))^{\frac{2}{3n}} = \mathrm{vol}(T)^{\frac{2}{n}}$, whereas for the parity lattices, we have $\mathrm{vol}(L_{kn})^{\frac{2}{3n}} = 2^{\frac{1}{k}}\,\mathrm{vol}(T)^{\frac{2}{n}}$. This means that the equivalent VNR Δ is the same when decoding in $\Gamma(2S, T_{2\omega}, T, 3)$ and in T. This will be taken into account in the next formulas to estimate δ^*.

Decoding Λ_{24}.
Considering the list-decoding version of Algorithm 5' (without the splitting strategy), (38) becomes (see the proof of Theorem 6)

$$U_{24}(\delta, \Delta) = \min\{3U_8(\delta, \Delta)^2 + 3U_8(\delta, 2\Delta)(1 - U_8(\delta, \Delta))^2, 1\}. \tag{59}$$

Assume that $\delta^* \leq \frac{1}{4}$ with this algorithm. If this holds, $T, V \cong E_8$ can be decoded with the recursive BDD discussed in Section V-D (since $E_8 \cong BW_8$). Hence, $U_8(\frac{1}{4}, \Delta) = P_e(BDD, \Delta)$ is given by the curve $n = 8$ in Figure 4. With (59), for $\Delta = 3.3$ dB we find $U_{24}(\delta = \delta^*, \Delta) \leq 10^{-4}$, which confirms that $\delta^* < 1/4$. As a result, we can use Algorithm 5' for quasi-MLD decoding of Λ_{24}. The complexity of Algorithm 5' is

$$\begin{aligned}\mathfrak{C}^{\Lambda_{24}}_{QMLD} =&\mathfrak{C}_{A.5'}(\Lambda_{24}, \delta = \delta^*) = 2^4(3\mathfrak{C}(E_8) + 3\mathfrak{C}(RE_8)), \\ =&96\mathfrak{C}(E_8).\end{aligned} \tag{60}$$

Decoding $\mathscr{N}_{72}$.
Regarding $\mathscr{N}_{72}$, with the first decoder we have (similar to (59))

$$U_{72}(\delta, \Delta) = \min\{3U_{24}(\delta, \Delta)^2 + 3U_{24}(\delta, 2\Delta)(1 - U_{24}(\delta, \Delta))^2, 1\}. \tag{61}$$

Consider a MLD decoder for Λ_{24} such that $U_{24}(\delta, \Delta) = P_e^{\Lambda_{24}}(opt, \Delta)$. Then, when $\Delta > 1$, $U_{72}(\delta, \Delta) \approx 3(P_e^{\Lambda_{24}}(opt, \Delta))^2$. The performance of this decoder for $\mathcal{N}_{72}$ is shown by the curve $U_{72}(\Delta)$ on Figure 2 in Example 1. Unlike for the parity lattices, the curve for $P_e^{\Lambda_{24}}(opt, \Delta)$ should not be shifted to the right before squaring, as explained in Example 1. We easily see that this decoder is powerful enough to get quasi-MLD performance for $\mathscr{N}_{72}$. The complexity is then

$$\mathfrak{C}^{\mathscr{N}_{72}}_{QMLD} = 2^{12} \cdot [3\mathfrak{C}^{\Lambda_{24}}_{MLD} + 3\mathfrak{C}^{\Lambda_{24}}_{MLD}] = 2^{12} \cdot 6 \cdot \mathfrak{C}^{\Lambda_{24}}_{MLD}. \tag{62}$$

The curve of quasi-optimal performance of $\mathscr{N}_{72}$ on the Gaussian channel is depicted in Figure 6. The figure also shows the performance of $L_{3\cdot 24}$ (discussed in Section V-C). The performance of $\mathscr{N}_{72}$ is at a distance of 2.6 dB only from Poltyrev limit at around 10^{-5} of error per point.

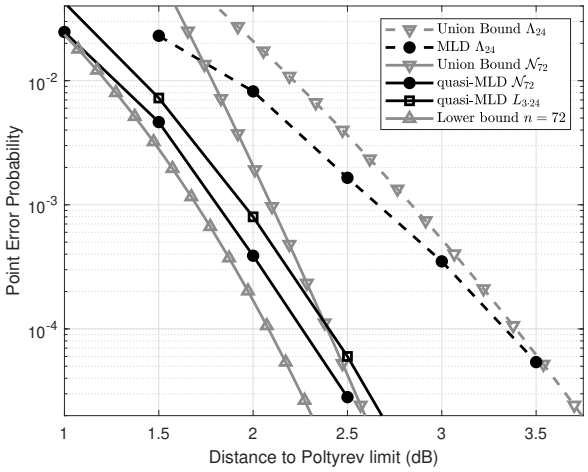

Fig. 6: Performance of $\mathscr{N}_{72}$ and $L_{3\cdot 24}$ on the Gaussian channel. The union bound is computed from the two first lattice shells of $\mathscr{N}_{72}$. The curves for Λ_{24} are also provided for comparison.

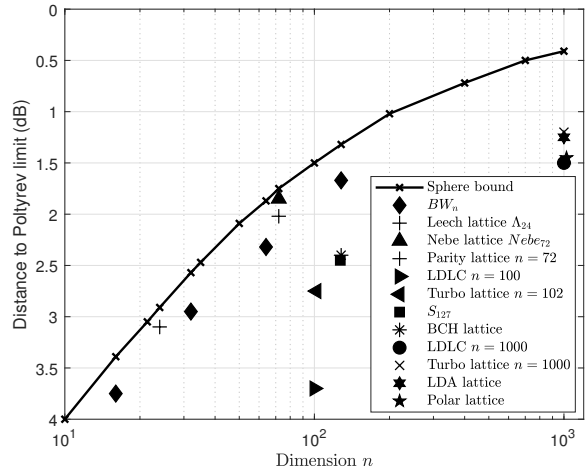

Fig. 7: Performance of different lattices for normalized error probability $P_e = 10^{-5}$.

VII. Lattice decoding benchmark

We compare the performance of lattices and decoders shown in the previous sections to existing schemes in the literature at $P_e = 10^{-5}$. For fair comparison at different dimensions, we let P_e be either the symbol-error probability or the normalized error probability, which is equal to the point-error probability divided by the dimension (as done in e.g. [41]). This approach enables to compare studies where the symbol-error probability is reported with studies where the point-error probability is reported.

First, several constructions have been proposed for block lengths around $n = 100$ in the literature. In [30] a two-level construction based on BCH codes with $n = 128$ achieves this error probability at 2.4 dB. The decoding involves an OSD of order 4 with 1505883 candidates. In [1] the multilevel (non-lattice packing) $\mathcal{S}_{127}$ ($n = 127$) has similar performance but with much lower decoding complexity via generalized minimum distance decoding. In [38] a turbo lattice with $n = 102$ and in [40] a LDLC with $n = 100$ achieve the error probability with iterative methods at respectively 2.75 dB, and 3.7 dB (unsurprisingly, these two schemes are efficient for larger block-lengths). All these schemes are outperformed by BW_{64}, the 3-parity-Leech lattices, and $\mathscr{N}_{72}$, where $P_e = 10^{-5}$ is reached at respectively 2.3 dB, 2.02 dB and 1.85 dB. Moreover, BW_{128} has $P_e = 10^{-5}$ at 1.7 dB, which is similar to many schemes with block length $n = 1000$ such as the LDLC (1.5 dB) [40], the turbo lattice (1.2 dB) [38], the polar lattice with $n = 1024$ (1.45 dB) [47], and the LDA lattice (1.27 dB) [14]. This benchmark is summarized on Figure 7.

VIII. Conclusions

In this paper, we present a unified framework for building lattices. It relies on a simple parity check, which can be applied recursively and combined to repetition coding. Famous lattices such as the Leech lattice in 24 dimensions, Nebe's extremal lattice in 72 dimensions, and Barnes-Wall lattices are obtained in this framework. A new decoding paradigm is established from this construction by taking into account the coset parity constraint. The paradigm leads to new bounded-distance decoders, list decoders, and quasi-optimal decoders on the Gaussian channel in terms of probability of error per lattice point. Quasi-optimal performance for BW_{64}, $\mathscr{N}_{72}$, and BW_{128} are shown to be achievable at reasonable complexity. A new parity lattice $L_{3\cdot 24}$ is also considered. It offers an excellent performance-complexity trade-off. The elegant single parity-check construction and its associated decoders are promising for the study of lattices in moderate and large dimensions.

IX. Appendix

A. Proof of Theorem 3

The following proof is not new, but it enables to make a clear link between the k-ing construction and Λ_{24} using our notations.

Proof. We let E_8 be scaled such that $d(E_8) = 2$ and $\mathrm{vol}(E_8) = 1$. This version of the Gosset lattice is even. Then, $S = \frac{1}{\sqrt{2}}E_8$ has $d(S) = 1$, $\mathrm{vol}(S) = 2^{-4}$ and $\mathrm{vol}(T) = \mathrm{vol}(T_{2\theta}) = 1$, $d(T) = d(T_{2\theta}) = 2$. Also, $|\alpha| = |\beta| = 2^4$ from (7).

Let $x = (a, b, c) \in \Gamma(2S, T_{2\omega}, T, 3)$. Firstly, using Theorem 2, we have $d(\Gamma(2S, T_{2\omega}, T, 3)) \geq 3$. Then, assume that $a = m + t_1$ and $b = m + t_2$ (with the notations of (13)) have both odd squared norms. This is equivalent to having the scalar products $\langle m, t_1\rangle = \frac{\nu}{2}$ and $\langle m, t_2\rangle = \frac{\nu'}{2}$, where ν and ν' are integers. Therefore, $\langle m, t_1 + t_2\rangle$ is integer and $c = m + t_1 + t_2$ has an even squared norm. We just proved that $\Gamma(2S, T_{2\omega}, T, 3)$ is even. This implies that $d(\Gamma(2S, T_{2\omega}, T, 3)) = 4$.

The last step aims at proving that $\Gamma(2S, T_{2\omega}, T, 3)$ has a unit volume. $\Gamma(2S, T_{2\omega}, T, 3)$ is obtained as the union of $|\alpha||\beta|^2 = 2^{12}$ cosets of $(2S)^3$. Hence, $\mathrm{vol}(\Gamma(2S, T_{2\omega}, T, 3)) = \mathrm{vol}((2S)^3)/2^{12} = 1$.

Finally, Λ_{24} is the unique lattice in dimension 24 with fundamental coding gain equal to 4. □

B. Proof of Theorem 6

Equation (39) can be generalized as

$$P(x \notin \mathcal{T}) \leq \binom{k}{2} P(x_j \notin \mathcal{T}_j)^2 + kP(x_i - (-\sum_{j \neq i} x_j) \notin \mathcal{V}_i)(1 - P(x_j \notin \mathcal{T}_j))^{k-1}. \tag{63}$$

This idea can be recursively applied if we remove Step 14 at each recursion. Let $U_n(r, \sigma^2)$ denote an upper-bound of $P(x \notin \mathcal{T})$. We have a recursion of the form

$$U_n(r, \sigma^2) = \binom{k}{2} U_{\frac{n}{k}}(\frac{r}{2}, \sigma^2)^2 + k \cdot U_{\frac{n}{k}}(r, \sigma^2)(1 - U_{\frac{n}{k}}(\frac{r}{2}, \sigma^2))^{k-1}, \tag{64}$$

where we set $U_n(r, \sigma^2) = 1$ if the right-hand term is greater than 1.

Note that $\mathrm{vol}(L_{\frac{n}{k}})^{\frac{2}{n/k}} = \mathrm{vol}(L_n)^{\frac{2}{n}}/2^{\frac{1}{k}}$, indeed:

$$\begin{aligned} \mathrm{vol}(L_{kn})^{\frac{2}{kn}} &= \left(\frac{\mathrm{vol}(\theta L_n)^k}{|\beta|^{k-1}}\right)^{\frac{2}{kn}} = \left(\frac{(\mathrm{vol}(L_n) \cdot 2^{\frac{n}{2}})^k}{2^{(\frac{n}{2})(k-1)}}\right)^{\frac{2}{kn}}, \\ &= \mathrm{vol}(L_n)^{\frac{2}{n}} \cdot 2^{\frac{1}{k}}. \end{aligned} \tag{65}$$

Moreover, we also have $d(\Gamma(V, \beta, k)_{\mathcal{P}}) = d(V) = 2d(T)$. Hence, if we express the recursion as a function of the VNR $\Delta = \frac{\mathrm{vol}(L_{\frac{n}{k}})^{\frac{2}{n/k}}}{2\pi e \sigma^2}$ and the relative radius δ, we get:

$$\begin{aligned} U_n(\delta, \Delta) =& \binom{k}{2} U_{\frac{n}{k}}(\delta, \frac{\Delta}{2^{\frac{1}{k}}})^2 + \\ & kU_{\frac{n}{k}}(\delta, 2^{\frac{k-1}{k}}\Delta)(1 - U_{\frac{n}{k}}(\delta, \frac{\Delta}{2^{\frac{1}{k}}}))^{k-1}. \end{aligned} \tag{66}$$

With the first splitting strategy (but not the second splitting strategy) the error probability is bounded from above as

$$\begin{aligned} P(n, \sigma^2, x \notin \mathcal{T}) \leq & \binom{k}{2} P(x_i \notin \mathcal{T}^\delta)^2 + \\ k\Big[& P(x_j \notin \mathcal{T}^{\frac{2}{3}\delta},\ x_j \in \mathcal{T}^\delta)^{k-1} P(x_i - (-\sum_{j \neq i} x_j) \notin \mathcal{V}_i^{2/3\delta}) \\ & + P(x_i - (-\sum_{j \neq i} x_j) \notin \mathcal{V}_i^\delta)) P(x_j \in \mathcal{T}^{\frac{2}{3}\delta})^{k-1} \Big], \end{aligned} \tag{67}$$

where

$$\begin{aligned} P(x_j \notin \mathcal{T}^{\frac{2}{3}\delta},\ x_j \in \mathcal{T}^\delta) =& (1 - \frac{P(x_j \in \mathcal{T}^{\frac{2}{3}\delta})}{P(x_j \in \mathcal{T}^\delta)}) P(x_j \in \mathcal{T}^\delta), \\ =& P(x_j \in \mathcal{T}^\delta) - P(x_j \in \mathcal{T}^{\frac{2}{3}\delta}). \end{aligned} \tag{68}$$

C. Proof of Lemma 5

Proof. The proof is similar to that of Theorem 3.3 in [35]. The vectors of squared norm 8 in $\Gamma(V, T, 3)_{\mathcal{P}}$ have only the following possible forms.

1) $(a, 0, 0), a \in V$ and $||a||^2 = 8$. The number of such vectors (counting the combinations) is $196560 \cdot 3$ vectors, i.e. the minimal vectors in V^3.
2) $(n_1, n_2, 0)$, $n_1, n_2 \in T, n_1 + n_2 \in V$ and $||n_1||^2 = ||n_2||^2 = 4$. The number of such vectors (counting the combinations) is $196560 \cdot 48 \cdot 3$. There are 196560 possibilities for n_1. Given n_1 how many choices are they for n_2? This is equivalent to asking the number of squared norm 8 vectors in the coset $m + V$, which are therefore congruent mod V. It is well-known (see Theorem 2 in [10, Chap.12]) that this number is 48, 24 mutually orthogonal pairs of vectors (one can check that $|T/V| = 2^{12} \cdot 48 = 196560$, the number of minimal vectors of Λ_{24}). Hence, there are 48 choices for n_2. Finally, the factor 3 comes from the combinations.

□

D. Proof of Theorem 7

To lighten the notations, we write $l(\delta)$ for $l(n/2, \delta)$ and $\mathfrak{C}(\delta)$ for $\mathfrak{C}(n/2, \delta)$.

• If $\delta < \frac{3}{8}$, the decoder of [22], whose complexity is given by (48), yields $O(n^2)$.

• If $\frac{3}{8} \leq \delta < \frac{1}{2}$, the decoder of [22], whose complexity is given by (48), yields $O(n^2)$.

• If $\delta = \frac{3}{4} - \epsilon$, $0 < \epsilon \leq 1/4$: Then, $\frac{2}{3}\delta < \frac{1}{2}$, $\frac{\delta}{2} < \frac{3}{8}$, $\frac{\delta}{3} < \frac{1}{4}$. We have $l(\frac{2}{3}\delta) = l(\frac{\delta}{2}) = O(1)$, $l(\frac{\delta}{3}) = 1$. We get

$$\begin{aligned} \mathfrak{C}(n, \delta) =& [2l(\frac{2}{3}\delta) + 2]\mathfrak{C}(\delta) + l(\delta)O(n^2), \\ =& l(\delta)O(n^2) \cdot \sum_{i=0}^{\log_2 n} \left(\frac{2l(\frac{2}{3}\delta) + 2}{4}\right)^i, \\ =& l(\delta)O(n^{2+\log_2[\frac{l(\frac{2}{3}\delta)+1}{2}]}) = l(\delta)O(n^{1+\log_2[\lfloor\frac{3}{4\epsilon}\rfloor+1]}). \end{aligned}$$

Consequently, if $\delta = \frac{1}{2}$, $l(\delta) \leq 2n$ and $\epsilon = \frac{1}{4}$. Then $\mathfrak{C}(n, \delta) = O(n^4)$. If $\delta > \frac{1}{2}$, we have $l(\delta) = O(n^{\log_2 4\lfloor\frac{3}{4\epsilon}\rfloor})$. Then, $\mathfrak{C}(n, \delta) = O(n^{1+\log_2 4\lfloor\frac{3}{4\epsilon}\rfloor^2})$, where we assumed that $\frac{3}{4\epsilon} >> 1$.

• If $\delta = \frac{3}{4}$: See Appendix F in [11] (long version on arXiv).

• If $\delta = 1 - \epsilon$, $0 < \epsilon < \frac{1}{4}$: See Appendix F in [11] (long version on arXiv).

E. Proof of Theorem 8

The result on the complexity is obtained by adapting (49), (50), and the complexity formulas in Theorem 7.

For $3/8 < \delta \leq 9/16$, we use the fact $E_y[l'(\frac{n}{2}, \frac{3}{8}, y)] \geq E_y[l'(\frac{n}{4}, \frac{3}{8}, y)] \geq ....$

$$\begin{aligned} E_y[l'(n, \delta, y)] \leq & 2\big[E_y[l'(\frac{3}{8}, y)]l(\delta) + l(\delta)E_y[l(\frac{3}{8}, y)]\big], \\ \leq & 4E_y[l'(\frac{3}{8}, y)]l(\delta) = O(n^{\log_2(4E_y[l'(\frac{3}{8}, y)])}), \\ =& O(n^{2+\log_2 E_y[l'(\frac{3}{8}, y)]}). \end{aligned} \tag{69}$$

F. A proof that $R(\mathscr{N}_{72}) > \sqrt{2}\rho(\mathscr{N}_{72})$

Lemma 7. $R(\mathscr{N}_{72}) > \sqrt{2}\rho(\mathscr{N}_{72})$.

The proof of this lemma is due to Gabriele Nebe (private communication).

Proof. Let $\mathscr{N}_{72}$ be scaled such that $\rho(\mathscr{N}_{72}) = \sqrt{2}$. The proof is done by contradiction. Assume that $R(\mathscr{N}_{72}) = \sqrt{2}\rho(\mathscr{N}_{72}) = 2$. Then, for any point $1/2v \in 1/2\mathscr{N}_{72}$, there is a point $x \in \mathscr{N}_{72}$ with

$||x - 1/2v|| \leq 2$. Squaring leads to $||2x - v||^2 \leq 16$. So each of the 2^{72} cosets of $2\mathscr{N}_{72}$ in $\mathscr{N}_{72}$ has to contain a point $w = 2x - v$ of squared norm smaller or equal to 16.

Now $\mathscr{N}_{72}$ has exactly 107502190683149087281 pairs $\pm w$ of squared norm ≤ 16 (obtained from the theta series of $\mathscr{N}_{72}$). This number is smaller than $|\mathscr{N}_{72}/2\mathscr{N}_{72}|$. Hence the covering radius of $\mathscr{N}_{72}$ is strictly larger than 2. □

REFERENCES

[1] D. Agrawal and A. Vardy, "Generalized minimum distance decoding in Euclidean space: performance analysis," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 60-83, Jan. 2000.

[2] O. Amrani, Y. Be'ery, A. Vardy, F.-W. Sun, and H. C. A. van Tilborg, "The Leech lattice and the Golay code: bounded-distance decoding and multilevel constructions," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1030-1043, Jul. 1994.

[3] E. S. Barnes and G. E. Wall, "Some extreme forms defined in terms of abelian groups," *Journal of the Australian Mathematical Society*, vol. 1, no. 1, pp. 47-63, Aug. 1959.

[4] Y. Be'ery, B. Shahar, and J. Snyders, "Fast decoding of the Leech lattice," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 959-967, Aug. 1989.

[5] A. Bonnecaze, P. Solé, and A. R. Calderbank, "Quaternary quadratic residue codes and unimodular lattices," *IEEE Transactions on Information Theory*, vol. 41, no. 2, pp. 366-377, Mar. 1995.

[6] H. Cohn, A. Kumar, S. D. Miller, D. Radchenko, and M. Viazovska, "The sphere packing problem in dimension 24," *Annals of Mathematics*, vol. 185, no. 3, pp. 1017-1033, Apr. 2017.

[7] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Transactions on Information Theory*, vol. 29, no. 6, pp. 820-824, Nov. 1983.

[8] J. H. Conway and N. J. A. Sloane, "On the Voronoi Regions of Certain Lattices," *SIAM Journal on Algebraic Discrete Methods*, vol. 5, no. 3, pp. 294-305, 1984.

[9] J. H. Conway and N. J. A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 41-50, Jan. 1986.

[10] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups,* 3rd edition, New York, USA: Springer-Verlag, 1999.

[11] V. Corlay, J.J. Boutros, P. Ciblat, and L. Brunel, "A new framework for building and decoding group codes," arXiv preprint arXiv:2012.06894v1 (first version), Dec. 2020.

[12] A. Desideri Bracco, "Treillis de codes quasi-cycliques," *European Journal of Combinatorics*, vol. 25, no. 4, pp. 505-516, May 2004.

[13] A. Desideri Bracco, A-M. Natividad, and P. Solé, "On quintic quasi-cyclic codes," *Discrete applied mathematics*, vol. 156, no. 18, pp. 3362-3375, Nov. 2008.

[14] N. di Pietro, J. J. Boutros, G. Zémor, and L. Brunel, "Integer low-density lattices based on construction A", *IEEE Information Theory Workshop*, Lausanne, Switzerland, pp. 422-426, Sep. 2012.

[15] G. D. Forney, Jr., "Coset codes. I. Introduction and geometrical classification," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1123-1151, Sep. 1988.

[16] G. D. Forney, Jr., "Coset codes. II. Binary lattices and related codes," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1152-1187, Sep. 1988.

[17] G. D. Forney, Jr., "A bounded-distance decoding algorithm for the Leech lattice, with generalizations," *IEEE Transactions on Information Theory*, vol. 35, no. 4, pp. 906-909, Jul. 1989.

[18] G. D. Forney, Jr., and A. Vardy, "Generalized minimum-distance decoding of Euclidean-space codes and lattices," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1992-2026, Nov. 1996.

[19] G. D. Forney, Jr., and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2384-2415, Oct. 1998.

[20] G. D. Forney, Jr., M. D. Trott, and S-Y. Chung, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 820-850, May. 2000.

[21] R. L. Griess, Jr., "Rank 72 high minimum norm lattices," *Journal of Number Theory*, vol. 130, no. 7, pp. 1512-1519, Jul. 2010.

[22] E. Grigorescu and C. Peikert, "List-Decoding Barnes-Wall Lattices," *Computational complexity*, vol. 26, pp. 365-392, June 2017.

[23] J. Harshan, E. Viterbo, and J. C. Belfiore, "Practical Encoders and Decoders for Euclidean Codes from Barnes-Wall Lattices," *IEEE Transactions on Information Theory*, vol. 61, no. 11, pp. 4417-4427, Nov. 2013.

[24] A. Ingber, R. Zamir, and M. Feder, "Finite-Dimensional Infinite Constellations," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1630-1656, Mar. 2013.

[25] G. R. Lang and F. M. Longstaff, "A Leech lattice modem," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 968-973, Aug. 1989.

[26] J. Leech, "Notes on sphere packings," *Canadian Journal of Mathematics*, vol. 19, pp. 251-257, 1967.

[27] J. Lepowsky and A. Meurman, "An E_8-approach to the Leech lattice and the Conway group," *Journal of Algebra*, vol. 77, no. 2, pp. 484-504, Aug. 1982.

[28] S. Ling and P. Solé, "On the algebraic structure of quasi-cyclic codes .I. Finite fields," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2751-2760, Nov. 2001.

[29] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes,* Amsterdam, The Netherland: North-Holland, 1977.

[30] T. Matsumine, B. M. Kurkoski, and H. Ochiai, "Construction D Lattice Decoding and Its Application to BCH Code Lattices," *2018 IEEE Global Communications Conference*, Abu Dhabi, United Arab Emirates, pp. 1-6, Dec. 2018.

[31] A. Meyer, "On the number of lattice points in a small sphere and a recursive lattice decoding algorithm," *Designs, Codes and Cryptography*, vol. 66, pp. 375–390, Jan. 2013.

[32] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of The Kluwer International Series in Engineering and Computer Science, Boston, Massachusetts:Kluwer Academic Publishers, 2002.

[33] D. Micciancio and A. Nicolosi, "Efficient bounded distance decoders for Barnes-Wall lattices," *2008 IEEE International Symposium on Information Theory*, Toronto, Canada, pp. 2484-2488, July 2008.

[34] G. Nebe, "A generalisation of Turyn's construction of self-dual codes," *RIMS workshop: Research into vertex operator algebras, finite groups and combinatorics*, Kyoto, pp. 51-59, Dec. 2010.

[35] G. Nebe, "An even unimodular 72-dimensional lattice of minimum 8," *Journal für die reine und angewandte Mathematik*, vol. 2012, no. 673, pp. 237-247, Dec. 2012.

[36] G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 409-417, Mar. 1994.

[37] H.-G. Quebbemann, "A construction of integral lattices," *Mathematika*, vol. 31, no. 1, pp. 137-140, Jun. 1984.

[38] A. Sakzad, M. Sadeghi, and D. Panario, "Turbo Lattices: Construction and Performance Analysis," Aug. 2011. Available: https://arxiv.org/abs/1108.1873.

[39] A. J. Salomon and O. Amrani, "Encoding and Decoding Binary Product Lattices," *IEEE Transactions on Information Theory*, vol. 52 , no. 12, pp. 5485-5495, Dec. 2006.

[40] N. Sommer, M. Feder, and O. Shalvi, "Low-Density Lattice Codes," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1561-1585, Apr. 2008.

[41] V. Tarokh, A. Vardy, and K. Zeger, "Universal bound on the performance of lattice codes," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 670-681, Mar. 1999.

[42] J. Tits, "Four presentations of Leech's lattice," *Finite simple groups, II, Proceedings of a London Mathematical Society Researsch Symposium*, Durham, United Kingdom, pp. 306-307, 1980.

[43] E. F. Assmus, Jr., H. F. Mattson, Jr., and R. J. Turyn, "Research to Develop the Algebraic Theory of Codes," *Report AFCRL-67-0365*, Air Force Cambridge Research Laboratories, Jun. 1967.

[44] A. Vardy and Y. Be'ery, "Maximum likelihood decoding of the Leech lattice," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1435-1444, July 1993.

[45] A. Vardy, "Even more efficient bounded-distance decoding of the hexacode, the Golay code, and the Leech lattice," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1495-1499, Sep. 1995.

[46] E. Viterbo and J. J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1639-1642, July 1999.

[47] Y. Yan, C. Ling, and X. Wu, "Polar lattices: Where Arıkan meets Forney," *2013 IEEE International Symposium on Information Theory*, Istanbul, Turkey, pp. 1292-1296, July 2013.

Vincent Corlay was born in Rennes, France, in 1993. He received the Engineering degree from Institut National des Sciences Appliquées (INSA) de Rennes in 2017, and the Ph.D. degree from Institut Polytechnique de Paris (at Telecom Paris) in 2020. In 2021, he received the best thesis award runner-up from Institut Polytechnique de Paris. Since 2020, he is a permanent researcher with Mitsubishi Electric R&D Centre Europe in Rennes, in the group Wireless Communication Systems.

Joseph Boutros

Philippe Ciblat was born in Paris, France, in 1973. He received the Engineering degree from Ecole Nationale Superieure des Telecommunications (ENST, now called Telecom Paris) and the M.Sc. degree in automatic control and signal processing from University Paris-Saclay, France, both in 1996, and the Ph.D. degree from University Gustave Eiffel, France, in 2000. He eventually received the HDR degree from University Gustave Eiffel, France, in 2007. In 2001, he was a Postdoctoral Researcher with University of Louvain, Belgium. In 2002, he joined Telecom Paris, as an Associate Professor. Since 2011, he has been (full) Professor in the same institution. He served as Associate Editor for the IEEE Communications Letters from 2004 to 2007. From 2008 to 2012, he served as Associate Editor and then Senior Area Editor for the IEEE Transactions on Signal Processing. From 2014, he is member of IEEE Technical Committee ”Signal Processing for Communications and Networking”. From 2018, he serves as Associate Editor for the IEEE Transactions on Signal and Information Processing over Networks. His research areas include statistical signal processing, distributed networks, and optimization for resource allocation.

Loïc Brunel