A Turyn-based neural Leech decoder

Vincent Corlay[†*], Joseph J. Boutros[‡], Philippe Ciblat[†], and Loïc Brunel[*]

[†] Telecom ParisTech, 46 Rue Barrault, 75013 Paris, [‡] Texas A&M University, Qatar,
[*]Mitsubishi Electric R&D, Rennes, France. v.corlay@fr.merce.mee.com

Abstract. A new decoder for the Leech lattice is presented. This quasi-optimal decoder utilizes a re-encoding paradigm, where candidates are obtained via a shallow neural network. This implies easy parallelization and low latency. The decoder exploits the fact that the Leech lattice is obtained from the direct sum of three polarized Gosset 8-dimensional lattices. This Turyn's construction was used in 2010 by G. Nebe to build the extremal even unimodular lattice in dimension 72 from three copies of the Leech lattice. Thus, we view this work as a first step towards the implementation of an efficient decoder for the Nebe 72-dimensional lattice.

1 Introduction

The Leech lattice Λ_{24} was discovered at the dawn of the communications era [15]. Recently, it was proved that Λ_{24} is the densest packing of congruent spheres in 24-dimensions [4]. Between these two major events, it has been subject to countless studies. This 24-dimensional lattice is exceptionally dense for its dimension and, unsurprisingly, has a remarkable structure. For instance, it contains the densest known lattices in all lower dimensions and it can be obtained in many 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 [11]. Of course, decoding the Leech lattice is not just a useless game between engineers as it has many practical interests: its high nominal coding gain of 6dB makes it a good candidate for high spectral efficiency short block length channel coding and its spherical-like Voronoi region of 16969680 facets [7] enables to get state-of-the-art performance for operations such as vector quantization or lattice shaping.

Among others, Λ_{24} can be obtained as (i) 8192 cosets of $4D_{24}$, (ii) 4096 cosets of $\sqrt{2}E_8 \oplus \sqrt{2}E_8 \oplus \sqrt{2}E_8$, (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 Q_{24} is also built with Construction B but applied on a subcode of C_{24}. Finally, one of the simplest construction is due to [23], where the Leech lattice is obtained via Construction A applied on the quaternary Golay code.

The history of maximum-likelihood decoding algorithms for Λ_{24} starts with [5], 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 [6] by the same authors using construction (ii). Two years latter, 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 [10]. A year later, it was further improved in [14] and [3] thanks to (iii) combined with an efficient decoder of C_{24}. Finally, (iv) along with the Hexacode is used to build a state-of-the-art decoder in [26].
To further reduce the complexity, suboptimal bounded-distance decoders were also investigated based on the same constructions: e.g. [11] with (iii) and [2] [27] with (iv). Most of these bounded-distance decoders don't change the error exponent (i.e. the effective minimum distance is not diminished) but increase the equivalent error coefficient. The extra loss is roughly 0.1dB on the Gaussian channel.

Besides lattice decoders tuned to the specific algebraic or geometric structure of a point lattice in the real Euclidean space, there exist universal lattice decoders, i.e. sphere decoders, based on point enumeration [28][1]. Sphere decoding is extremely fast at low noise level but it may get stuck (i.e. tragically slow) in looking for the closest point at high noise level for dense lattices in dimensions beyond 64. The parallelization of sphere decoding may help in overcoming this defect for high dimensions but its implementation over multiple processors was never made yet.

In this work, unlike the latest algebraic decoders, we don't use the Golay code to build our decoder but rather the multiple occurrence of E_8 in Λ_{24}, similarly to [6][10]. This construction of the Leech lattice usually goes under the name of Turyn's construction. This structure enables to utilize a re-encoding process to correct errors. Since the decoder requires candidates, a simple list decoder for E_8, based on a neural network, is presented: it uses the hyperplane logical decoder (HLD) [8]. The advantages of the HLD are its low-latency and its hardware-friendly architecture. Moreover, to ensure the optimality of the HLD, we state a new theorem providing sufficient conditions for a basis of E_8 to be Voronoi-reduced. The performance of the proposed algorithm is investigated on the Gaussian channel.

2 Turyn's Construction

The story of Turyn's construction starts in 1967, when Turyn constructed the Golay code from two versions of the extended Hamming code [17]. According to Nebe [18], it has then been remarked independently in [25], [16], and [22] that there is an analogous construction of Λ_{24} based on E_8. Turyn's construction reappeared in [6] under the form of 4096 cosets of the lattice generated by glued copies of E_8 (construction (ii) above). Finally, it was rediscovered under both forms in the scope of the "cubing construction" [10]. We briefly describe below the Turyn's construction from E_8 and prove that the constructed lattice is the Leech Λ_{24}. Our proof relies on simpler but less general arguments than those found in [22][16].

In the sequel, $\mathrm{vol}(\Lambda)$ denotes the fundamental volume of Λ, i.e. the volume of its Voronoi cell. The squared minimal distance (or minimal squared norm) of Λ will be denoted $d^2_{min}(\Lambda)$. We say that an integral lattice is even if $\|x\|^2$ is even for any x in Λ. Let E_8 be a version of the Gosset lattice with squared minimal norm $d^2_{min}(E_8) = 2$ and a fundamental volume equal to unity, i.e. an even unimodular version. Consider $L = \frac{1}{\sqrt{2}}E_8$ with minimal norm 1 and volume 2^{-4}. Starting from the quotient $L/2L$, we determine two versions M and N of E_8 satisfying

$$2L \subset M \subset L, \quad \text{and} \quad 2L \subset N \subset L,$$

with $\mathrm{vol}(M) = \mathrm{vol}(N) = 1$, $d^2_{min}(M) = d^2_{min}(N) = 2$, and $M \bigcap N = 2L$, to get the following polarisation of L [18]:

$$L = M + N.$$

M and N are integral even unimodular lattices. Now define the quotient groups $\mathcal{M} = M/2L$ and $\mathcal{N} = N/2L$, $|\mathcal{M}| = |\mathcal{N}| = 2^4$.

Theorem 1. *Using the above notations, the lattice defined as*

$$\begin{aligned}\Lambda = \{(a, b, c) :\ & a = m + n_1,\ b = m + n_2,\ c = m + n_3,\\ & m \in M,\ n_1, n_2, n_3 \in N,\ n_1 + n_2 + n_3 \in 2L\}\end{aligned}$$

is the even unimodular Leech lattice Λ_{24}.

Proof. Firstly, assume that $m + n_1$ and $m + n_2$ both have odd squared norms. This is equivalent to having the scalar products $\langle m, n_1\rangle = \frac{\lambda}{2}$ and $\langle m, n_2\rangle = \frac{\lambda'}{2}$, where λ and λ' are odd integers. Then using $n_1+n_2+n_3 \in 2L$, we get that $\langle m, n_3\rangle$ is integer. Thus $m+n_3$ has an even squared norm. We just proved that Λ is even.

Secondly, let us prove that $d^2_{min}(\Lambda) = 4$.
Let $x = (a, b, c) \in \Lambda$. Given the symmetry with respect to the three sets of eight coordinates, we shall distinguish three cases according to the number of non-zero components:

1. $x = \{(a, 0, 0)\} \Rightarrow a \in 2L \Rightarrow \|x\|^2 \geq d^2_{min}(2L) = 4$.
2. $x = \{(a, b, 0)\} \Rightarrow a, b \in N \Rightarrow \|x\|^2 \geq 2d_{min}(N) = 4$.
3. $x = \{(a, b, c)\} \Rightarrow \|x\|^2 \geq 3d^2_{min}(L) = 3$. But Λ is even, then $\|x\|^2 \geq 4$.

This implies that Λ has $d^2_{min} = 4$.
The last step aims at proving that Λ has a unit volume. Indeed, the definition of Λ is rewritten by developing a, b, c modulo $2L$,

$$\begin{aligned}\Lambda = \{(a, b, c) :\ & a = d_1 + \underbrace{m' + n'_1}_{p_1},\ b = d_2 + \underbrace{m' + n'_2}_{p_2},\ c = d_3 + \underbrace{m' + n'_3}_{p_3},\\ & d_1, d_2, d_3 \in 2L,\ m' \in \mathcal{M},\ n'_1, n'_2, n'_3 \in \mathcal{N},\ n'_1 + n'_2 + n'_3 = 0\},\end{aligned} \tag{1}$$

where $p_1, p_2, p_3 \in L/2L$. (1) shows that Λ is obtained by the union of cosets of $(2L)^3$. The number of those cosets is determined by m', n'_1, and n'_2, with $n'_3 = -n'_1 - n'_2$. Hence, there are $|\mathcal{M}| \times |\mathcal{N}|^2 = 2^{12}$ such cosets. Finally, $\text{vol}(\Lambda) = \text{vol}((2L)^3)/2^{12} = 1$. The Leech lattice is the unique lattice in dimension 24 with Hermite constant $d_{min}^2/vol^{2/24} = 4$. ■

As stated in the introduction, a new extremal even unimodular lattice in dimension 72 of minimum 8 was discovered by Nebe in 2010 [19]. Given Nebe's lattice Hermite constant of 8 (9 dB) and its huge kissing number, we expect a performance about 2.5dB from Poltyrev limit [21] at a point error probability of 10^{-5} on an additive white Gaussian noise channel. For a gentle introduction on this lattice and its Turyn's construction, the reader is invited to refer to [18]. Based on arguments similar to those in Theorem 1, we state the next lemma.

Lemma 1. *$Nebe_{72}$ can be obtained as the union of 2^{32} cosets of $2L' \oplus 2L' \oplus 2L'$, where L' is a specific version of Λ_{24} (with minimum 2) used in Turyn's construction of $Nebe_{72}$.*

Thanks to this lemma, it might be possible to use the algorithm presented in the next sections, with some modifications, to decode $Nebe_{72}$.

3 A new Turyn-based Leech decoder

In [6] a point in $\mathbb{R}^{24}$ is decoded in all 2^{12} cosets of $2L \oplus 2L \oplus 2L$ and the best candidate is kept. However, as we shall see in the sequel it is not necessary to investigate all the cosets to get quasi-optimal performance on the Gaussian channel. To explain our decoder, we first introduce a naive decoder.

The main idea of this naive decoder is to generate several candidates for each of the three 8 dimensions of Λ_{24} by decoding in L and keep only the combinations resulting in a valid point. The best point among these final candidates is then kept. To run the decoder, we first need to pre-compute the following elements.

1. Pick generator matrices for L, M, N.
2. Choose 256 coset leaders of $2L$ in L, a set denoted by $\mathcal{C}$. For instance, they can be chosen with the "maximally biased method" of Forney [12].
3. Generate $\mathcal{M}$ and $\mathcal{N}$ based on $\mathcal{C}$ (they are the unique 16 integers 8-tuples when the coset leaders in $\mathcal{C}$ are multiplied by the inverse of the generator matrix of M or N).
4. Find the unique map between the 256 elements of $\mathcal{C}$ and the 256 elements of $\mathcal{M} + \mathcal{N} \bmod 2L$ (create a look-up table).

We are now ready to present the naive decoder. To decode a point $y = (y_1, y_2, y_3) \in \mathbb{R}^{24}$, the algorithm implements the following steps.

1. Generate $\aleph$ candidates $t_1 \in \mathcal{T}_1$, $t_2 \in \mathcal{T}_2$, $t_3 \in \mathcal{T}_3$, $t_1, t_2, t_3 \in L$ for each y_1, y_2, y_3 (i.e. $|\mathcal{T}_1| = \aleph$, $|\mathcal{T}_2| = \aleph$, and $|\mathcal{T}_3| = \aleph$).

2. For each of these candidates, find the coset of $2L$ it belongs to (i.e. find the proper coset leader in $\mathcal{C}$).
3. For each of these coset leaders, find the unique corresponding elements $m'_{t_i} \in \mathcal{M}$ and $n'_{t_i} \in \mathcal{N}$.
4. For each of the $\aleph^3$ combinations of (t_1, t_2, t_3), check if $m'_{t_1} = m'_{t_2} = m'_{t_3}$ and $n'_{t_1} + n'_{t_2} + n'_{t_3} = 0 \pmod{2L}$. If the conditions are satisfied then store this point of Λ_{24}.
5. For each point found, compute its distance to the received point. Keep the closest point.

Important choices are the rule to generate the candidates and the size of $\aleph$. We choose the candidates as the $\aleph$ closest lattice points in L from y_i.

Figure 1 illustrates the performance of the naive decoding algorithm on the Gaussian channel for several values of $\aleph$. Unsurprisingly, the performance is disappointing: if the noise realization is strong and concentrated on the 8 dimensions of a y_i, even if y is within the decoding capability of Λ_{24}, the proper t_i is unlikely to be in the corresponding list, even if the list is large.

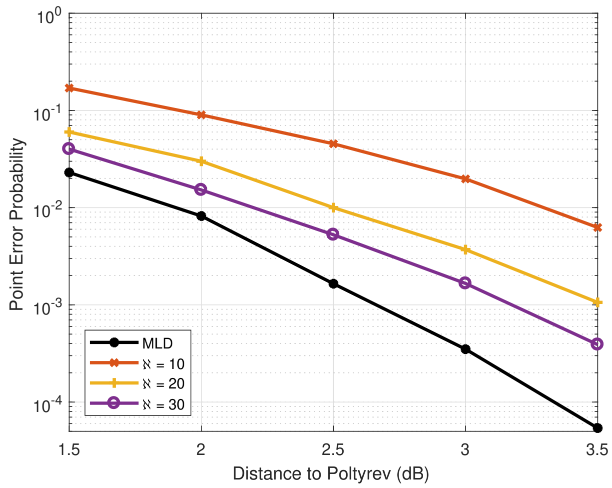

Fig. 1. Performance of the naive decoder versus the optimal maximum-likelihood decoder (MLD) with different $\aleph$. The candidates are chosen to be the $\aleph$ closest lattice points.

Nevertheless, the noise is unlikely to be strong on two of the three 8 dimensions and almost null on the remaining 8 dimensions. Hence, if the list size is large enough (but not necessarily very large as shown below), at least two of the three lists probably contain the good point. Moreover, given a and b, thanks to (1), we know in which coset of $2L$ is located c (via the uniqueness of n'_3 given n'_1 and n'_2). In other words, we can use these constraints to re-encode c based on a and b: once p_3 is computed, one can find d_3 by decoding y_3 in this coset of $2L$. Note that the equivalent minimum distance of $2L$ is the same as Λ_{24}. Consequently, if one decodes y_3 in the proper coset and the point y is within the decoding capability of Λ_{24}, the closest lattice point from y_3 found in this coset of $2L$ is always the correct d_3.

The second decoder (our main decoder) exploits these observations. It needs an additional element compared to the first decoder: a look-up table that gives the only valid $n'_i \in \mathcal{N}$ given one of the 256 possible tuples of 2 elements in $\mathcal{N}$. We are now ready to present our main decoder.

1. Generate $\aleph$ candidates $t_1 \in \mathcal{T}_1$, $t_2 \in \mathcal{T}_2$, $t_3 \in \mathcal{T}_3$, $t_1, t_2, t_3 \in L$ for each y_1, y_2, y_3 (i.e. $|\mathcal{T}_1| = \aleph$, $|\mathcal{T}_2| = \aleph$, and $|\mathcal{T}_3| = \aleph$).
2. For each of these candidates, find the coset of $2L$ it belongs to (i.e. find the proper coset leader in $\mathcal{C}$).
3. For each of these coset leaders, find the unique corresponding elements $m'_{t_i} \in \mathcal{M}$ and $n'_{t_i} \in \mathcal{N}$.
4. For each of the $3\aleph^2$ combinations of (t_i, t_j), if $m'_{t_i} = m'_{t_j}$ find n'_k and generate the coset leader p_k. Find the closest lattice point d_k to y_k in this coset of $2L$ and compute $t_k = d_k + p_k$. Store the resulting point $(t_i, t_j, t_k) \in \Lambda_{24}$ (with the proper arrangement of t_i, t_j, t_k).
5. For each point found, compute its distance to the received point. Keep the closest point.

Figure 2 shows the very satisfactory performance of this decoding scheme for several values of $\aleph$. The candidates are the $\aleph$ closest lattice points in L from y_i.

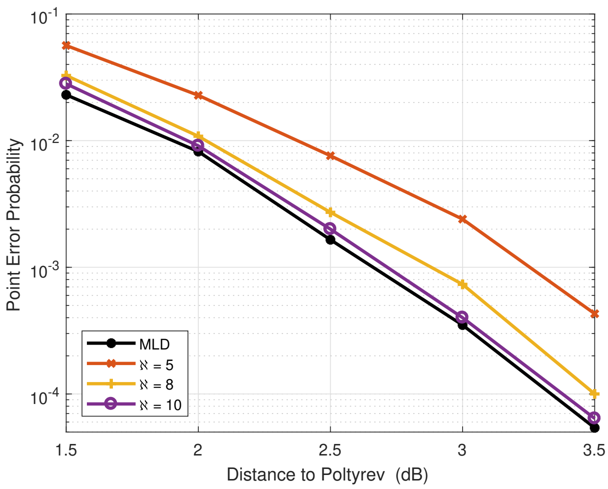

Fig. 2. Performance of the main decoder versus the optimal maximum-likelihood decoder (MLD) with different $\aleph$. The candidates are chosen to be the closest lattice points.

Note that the paradigm of this decoder is similar to the one of the Ordered Statistics Decoder [13], namely use a reliable subset of the received symbols to re-encode and correct errors.

4 A method to generate the list of candidates in L

In this section, we present a simple list decoder for E_8. It is used in step 1 of the above main decoder.

Our strategy involves an optimal decoder for E_8. We use the HLD because of its low-latency and its hardware-friendly architecture: it can be implemented via a neural network with only two hidden layers. This decoder operates in the fundamental parallelotope $\mathcal{P}$ and considers only the corners of $\mathcal{P}$. Hence, to make sure that the HLD is optimal, we must prove that the closest lattice point to any point in $\mathcal{P}$ is one of the corner of $\mathcal{P}$: i.e. we must prove that E_8 admits a Voronoi-reduced (VR) basis [8].

Theorem 2. *Let G be a generator matrix of E_8, where all the basis vectors are from the first lattice shell. Let $\mathring{\mathcal{P}}$ be the interior of the fundamental parallelotope of E_8. If $(G^{-1})^T$ is a generator matrix of E_8 with basis vectors from the first shell, then the G basis is Voronoi-reduced with respect to $\mathring{\mathcal{P}}$.*

The proof of this theorem is omitted due to the paper length. An example of a Voronoi-reduced basis for E_8 satisfying the sufficient conditions given by Theorem 2 can be found in [8].

We are now ready to present the list decoder:

1. Find the closest lattice point $x_1 \in L$ to the point to decode y_i via the HLD. Compute $y' = R \times (y_i - x_1)/||y_i - x_1||$ and find x_2, the closest lattice point of y', also via the HLD ($R = (1+\epsilon) \times d_{min}/\sqrt{2}$, where $d_{min}/\sqrt{2}$ is the covering radius of E_8).
2. Compute $x_3 = (x_2 + x_1)/2$. Two situations can be encountered: (i) if $||x_2 - x_1|| = \sqrt{2} \times d_{min}$, then there are 14 lattice points at equal distance from x_3. (ii) If $||x_2 - x_1|| = d_{min}$, then there are 56 lattice points at equal distance from x_3. In both cases, the list is formed by x_1, x_2, and the other $\aleph - 2$ points, among the 14 or 56 points at equal distance from x_3, that are the closest to y_i.

With $\aleph = 11$, the performance of the main decoder using this list decoder is almost the same as one depicted in Figure 2 by the purple curve.

5 Summary and complexity analysis

The decoder is summarized in Figure 3.

Nowadays, the CPU time of an algorithm is not as crucial as it used to be: with the advent of GPUs, parallelization is often more important than the raw amount of flops. This former aspect is clearly a strength of the proposed

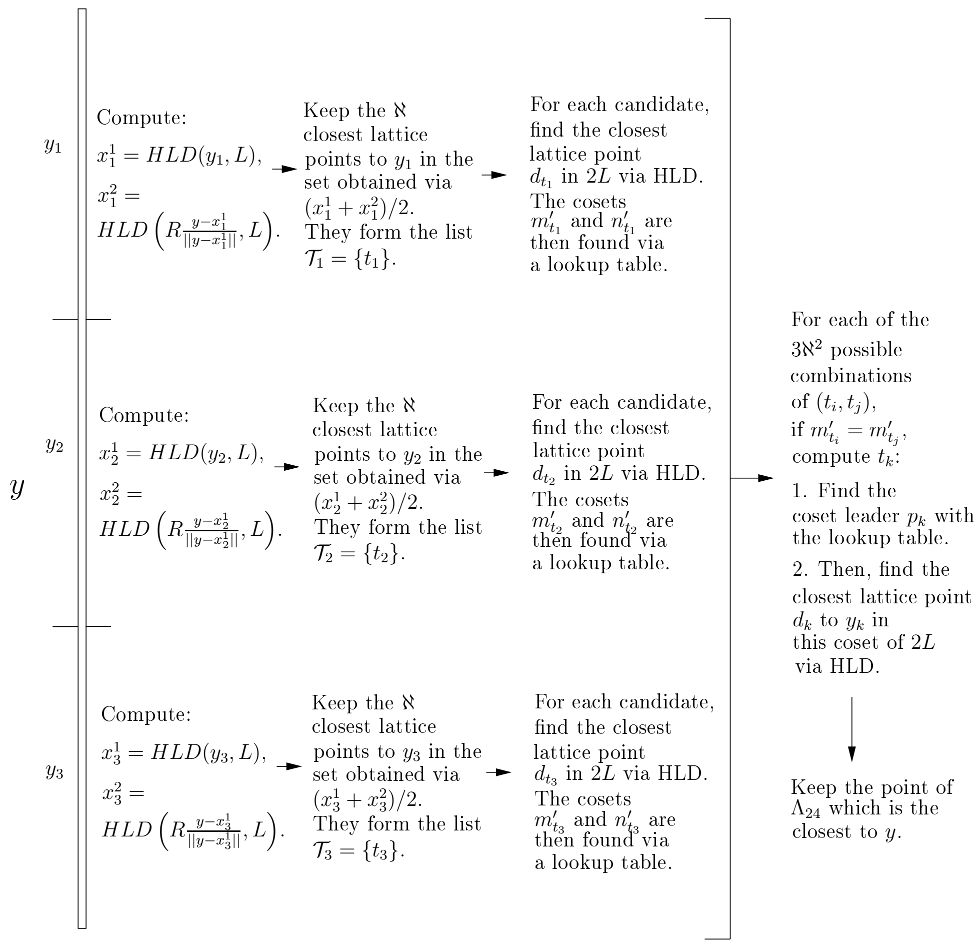

Fig. 3. Overview of the proposed decoder. From left to right: the point to decode $y \in \mathbb{R}^{24}$ is split into three points $y_1, y_2, y_3 \in \mathbb{R}^8$, which are then processed independently. A list of $|\{t_i\}| = \aleph$ candidates for each y_i is generated. For each valid combination of (t_i, t_j), a lattice point of Λ_{24} is obtained. The closest point to y among these lattice points is the decoded point.

algorithm, mainly due to the parallel processing of the three 8 dimensions of the point to decode $y \in \mathbb{R}^{24}$ and the shallow structure of the HLD.

The CPU time could be optimized, but at the cost of latency: for instance, a large amount of computations in our algorithm is due to the multiple use of the HLD. Hence, the number of flops could be reduced via folding techniques, such as the one presented in [9], decreasing the complexity of the HLD. Nevertheless, this implies using neural networks which cannot be parallelized as efficiently due to a large depth. Similarly, one could also replace the HLD by the low-complexity E_8 decoder presented in [6] but this would also increase the latency.

A rough estimate of the CPU time $\mathcal{C}$ of our Leech decoder is the following. Let $\mathcal{C}(HLD)$ denote the complexity of the maximum-likelihood decoder of E_8. The largest amount of operations are due to: (i) finding the two maximum-likelihood points in the list decoder of E_8, (ii) finding the coset of $2L$ each candidate belongs to, and (iii) re-encoding and processing the $3\aleph^2$ points. We get:

$$\mathcal{C} \approx 3 \times 2 \times \mathcal{C}(HLD) + 3 \times \aleph \times \mathcal{C}(HLD) + 3\aleph^2(a + \mathcal{C}(HLD)) + b,$$

where a and b are small constants. Our decoder is more complex than the state-of-the art decoder of Vardy [27] which requires only 331 real operations. However, this latter decoder is specific to Λ_{24} whereas our decoder is more universal: the re-encoding paradigm can be used whenever some dimensions of the lattice are explicitly constrained by the others, which is the case of the Nebe 72-dimensional lattice.

References

1. E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, pp. 2201-2214, 2002.
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 Trans. Inf. Theory*, vol. 40, 1030–1043, 1994.
3. Y. Be'ery, B. Shahar, and J. Snyders, "Fast decoding of the Leech lattice," *IEEE J. Select. Areas Com.*, vol. 7, pp. 959-967, 1989.
4. H. Cohn, A. Kumar, S. D. Miller, D. Radchenko, M. Viazovska, "The sphere packing problem in dimension 24," *Ann. Math.*, vol. 183, pp. 1017-1033, 2017.
5. J. H. Conway and N. J. A. Sloane, "On the voronoi regions of certain lattices," *SIAM Journ. on Algeb. Disc. Meth.*, vol. 5, pp. 294-305, 1984.
6. J. H. Conway and N. J. A. Sloane, "Soft decoding techniques for codes and lattices, including the Golay code and the Leech lattice," *IEEE Trans. Inf. Theory*, vol. 32, pp. 41-50, 1986.
7. J. Conway and N. Sloane. *Sphere packings, lattices and groups.* Springer-Verlag, New York, 3rd edition, 1999.
8. V. Corlay, J.J. Boutros, P. Ciblat, and L. Brunel, "Neural Lattice Decoders," arXiv preprint arXiv:1807.00592, July 2018.
9. V. Corlay, J.J. Boutros, P. Ciblat, and L. Brunel, "A lattice-based approach to the expressivity of deep ReLU neural networks," arXiv preprint arXiv:1902.11294 , Feb. 2019.

10. G.D Forney, Jr., “Coset codes II: Binary lattices and related codes,” *IEEE Trans. Inf. Theory*, vol. 34, pp. 1152-1187, 1988.
11. G.D. Forney, Jr., “A bounded distance decoding algorithm for the Leech lattice, with generalizations,” *IEEE Trans. Inf. Theory*, vol. 35, pp. 906-909, 1989.
12. G.D. Forney, Jr., “Multidimensional Constellations - part II: Voronoi Constellations,” *IEEE J. Select. Areas Com.*, vol. 7, pp. 941-958, 1989.
13. M. P.C. Fossorier and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” *IEEE Trans. Inf. Theory*, vol. 41, pp. 1379-1396, 1995.
14. G. R. Lang and F. M. Longstaff, “A Leech lattice modem,” *IEEE journ. of selec. areas in com.*, vol. 7, pp. 968-973, 1989.
15. J. Leech, “Notes on sphere packings,” *Can J. Math.*, vol. 19, pp. 251-257, 1967.
16. J. Lepowsky and A. Meurman, “An E_8-approach to the Leech lattice and the Conway group,” *J. Algebra*, vol. 77, pp. 484-504, 1982.
17. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes.* Amsterdam, The Netherlands: North-Holland, 1977.
18. G. Nebe, “A generalisation of Turyn’s construction of self-dual codes,” *RIMS workshop: Research into vertex operator algebras, finite groups and combinatorics*, Kyoto, Dec. 2010.
19. G. Nebe, “An even unimodular 72-dimensional lattice of minimum 8,” *J. Reine Angew. Math.*, vol. 673, pp. 237-247, 2012.
20. G. Nebe and R. Parker, “On extremal even unimodular 72-dimensional lattices,” *Mathematics of Computation*, vol. 83, pp. 287, 2014.
21. G. Poltyrev, “On coding without restrictions for the AWGN channel,” *IEEE Trans. Inf. Theory*, vol. 40, pp. 409-417, Mar. 1994.
22. H.-G. Quebbemann, “A construction of integral lattices,” *Mathematika*, vol. 31, pp. 137-140, 1984.
23. A. Bonnecaze, P. Solé, and A. R. Calderbank, “Quaternary Quadratic Residue Codes and Unimodular Lattices”, *IEEE Trans. Inf. Theory*, vol. 41, 1995.
24. N. Secord and R. De Buda, “Demodulation of a Gosset Lattice Code Having a Spectral Null at DC,” *IEEE Trans. Inf. Theory*, vol. 35, 1989.
25. J. Tits, “Four presentations of Leech’s lattice,” *M.J.Collins (Ed.) Finite simple groups, II, Proc. LMS Researsch Symp.* Durham, 1978, Academic Press (1980), pp. 306-307.
26. A. Vardy and Y. Be’ery, “Maximum-likelihood decoding of the Leech lattice,” *IEEE Trans. Inf. Theory*, vol. 39, pp. 1435-1444, 1993.
27. A. Vardy, “Even more efficient bounded-distance decoding of the hexacode, the golay code, and the leech lattice,” *IEEE Trans. Inf. Theo*, vol. 41, pp. 1495-1499, 1995.
28. E. Viterbo and J. J. Boutros, “A universal lattice code decoder for fading channels,” *IEEE Trans. Inf. Theory*, vol. 45, pp. 1639-1642, 1999.