Neural network approaches to point lattice decoding

Vincent Corlay, Joseph J Boutros, Philippe Ciblat, Loïc Brunel

Neural network approaches to point lattice decoding

Vincent Corlay, Joseph J. Boutros, Philippe Ciblat, and Loïc Brunel

Abstract

We characterize the complexity of the lattice decoding problem from a neural network perspective. The notion of Voronoi-reduced basis is introduced to restrict the space of solutions to a binary set. On the one hand, this problem is shown to be equivalent to computing a continuous piecewise linear (CPWL) function restricted to the fundamental parallelotope. On the other hand, it is known that any function computed by a ReLU feed-forward neural network is CPWL. As a result, we count the number of affine pieces in the CPWL decoding function to characterize the complexity of the decoding problem. It is exponential in the space dimension n, which induces shallow neural networks of exponential size. For structured lattices we show that folding, a technique equivalent to using a deep neural network, enables to reduce this complexity from exponential in n to polynomial in n. Regarding unstructured MIMO lattices, in contrary to dense lattices many pieces in the CPWL decoding function can be neglected for quasi-optimal decoding on the Gaussian channel. This makes the decoding problem easier and it explains why shallow neural networks of reasonable size are more efficient with this category of lattices (in low to moderate dimensions).

Index Terms

Neural network, dense lattice, MIMO lattice, continuous piecewise linear function, basis reduction.

I. Introduction

In 2012 Alex Krizhevsky and his team presented a revolutionary deep neural network in the ImageNet Large Scale Visual Recognition Challenge [16]. The network largely outperformed all the competitors. This event triggered not only a revolution in the field of computer vision but has also affected many different engineering fields, including the field of digital communications.

In our specific area of interest, the physical layer, countless studies have been published since 2016. For instance, reference papers such as [15] gathered more than 800 citations in less than three years. However, most of these papers present simulation results: e.g. a decoding problem is set and different neural network architectures are heuristically considered. Learning via usual gradient-descent-like techniques is performed and the results are presented.

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 this paper was presented at the IEEE International Symposium on Information Theory, Paris, France, July 2019.

Our approach is different: we try to characterize the complexity of the decoding problem that should be solved by the neural network.

Neural network learning is about two key aspects: first, finding a function class $\Phi = \{f\}$ that contains a function "close enough" to a target function f^*. Second, finding a learning algorithm for the class Φ. Naturally, the less "complex" the target function f^*, the easier the problem is. We argue that understanding this function f^* encountered in the scope of the decoding problem is of interest to find new efficient solutions.

Indeed, the first attempts to perform decoding operations with "raw" neural networks (i.e. without using the underlying graph structures of existing sub-optimal algorithms, see below) were unsuccessful. For instance, an exponential number of neurons in the network is needed in [13] to achieve satisfactory performance when decoding small length polar codes. We made the same observation when we tried to decode dense lattices typically used for channel coding [9]. So far, it was not clear whether such a behavior is due to either an unadapted learning algorithm or a consequence of the complexity of the function to learn. However, unlike for channel decoding (i.e. dense lattice decoding), neural networks can sometimes be successfully trained in the scope of multiple-input multiple-output (MIMO) detection [24] [9].

In this paper, the problem of neural-network lattice decoding is investigated. Lattices are well-suited to understand these observed differences as they can be used both for channel coding and for modelling MIMO channels.

We embrace a feed-forward neural network perspective. These neural networks are aggregation of perceptrons and compute a composition of the functions executed by each perceptron. For instance, if the activation functions are rectified linear unit (ReLU), each perceptron computes a piecewise affine function. Consequently, all functions in the function class Φ of this feed-forward neural network are CPWL.

We shall see that, under some assumptions, the lattice decoding problem is equivalent to computing a CPWL function. The target f^* is thus CPWL. The complexity of f^* can be assessed, for instance, by counting its number of affine pieces.

It has been shown that the minimum size of shallow neural networks, such that Φ contains a given CPWL function f^*, directly depends on the number of affine pieces of f^* whereas deep neural networks can "fold" the function and thus benefit of an exponential complexity reduction [19]. On the one hand, it is critical to determine the number of affine pieces in f^* to figure out if shallow neural networks can solve the decoding problem. On the other hand, when this is not the case, we can investigate if there exist preproccessing techniques to reduce the number of pieces in the CPWL function. We shall see that these preprocessing techniques are sequential and thus involve deep neural networks. Due to the nature of feed-forward neural networks, our approach is mainly geometric and combinatorial. It is restricted to low and moderate dimensions.

Note that neural networks have been applied in several studies to improve the efficiency of existing decoding algorithms. This approach, to be opposed with the use of "raw" neural networks mentioned above, is in general more successful. For instance, it is possible to unfold existing iterative algorithms to establish the neural network structure for MIMO detection as done in [14]. In the field of decoding, this idea was originally proposed in [20] for short block-length codes. For lattices in reasonable number of dimensions, one can maintain sphere decoding but tune its parameters via a neural network [18]. A similar approach is investigated in [3], where the authors use

deep neural network to improve the design and control of the sphere radius for sphere decoding. Paper [28] utilizes a neural network to perform predictions on the sub-trees to be explored by the sphere decoder, which avoids some unnecessary operations. However, our main contribution is not to present new decoding algorithms but to provide a better understanding of the decoding/detection problem from a neural network perspective. Consequently, the cited applications of neural networks are outside the context of our study.

The paper is organized as follows. Preliminaries are found in Section II. We show in Section III how the lattice decoding problem can be restricted to the compact set $\mathcal{P}(\mathcal{B})$. This new lattice decoding problem in $\mathcal{P}(\mathcal{B})$ induces a new type of lattice-reduced basis. The category of basis, called Voronoi-reduced basis, is presented in Section IV. In Section V, we introduce the decision boundary to decode componentwise. The discrimination with respect to this boundary can be implemented via the hyperplane logical decoder (HLD) also presented in this section. It is proved that, under some assumptions, this boundary is a CPWL function with an exponential number of pieces. Finally, we show in Section VI that this function can be computed at a reduced complexity via folding with deep neural networks, for some famous dense lattices. We also argue that the number of pieces to be considered for quasi-optimal decoding is reduced for MIMO lattices on the Gaussian channel, which makes the problem easier.

We summarize below the main contributions of the paper.

- We first state a new closest vector problem (CVP), where the point to decode is restricted to the fundamental parallelotope $\mathcal{P}(\mathcal{B})$. See Problem 1. This problem naturally induces a new type of lattice basis reduction, where the corresponding basis is called Voronoi-reduced basis. See Definition 1. In Section IV, we prove that some famous dense lattices admit a Voronoi-reduced basis. We also show that it is easy to get quasi-Voronoi-reduced bases for random MIMO lattices up to dimension $n = 12$.
- A new paradigm to address the CVP problem in $\mathcal{P}(\mathcal{B})$ is presented. We introduce the notion of decision boundary in order to decode componentwise in $\mathcal{P}(\mathcal{B})$. This decision boundary partition $\mathcal{P}(\mathcal{B})$ into two regions. The discrimination of a point with respect to this boundary enables to decode. The hyperplane logical decoder (HLD, see Algorithm 2) is a brute-force algorithm which computes the position of a point with respect to this decision boundary. The HLD can be viewed as a shallow neural network.
- In Section V-E, we show that the number of affine pieces in the decision boundary grows exponentially with the dimension for some basic lattices such as A_n, D_n, and E_n (see e.g. Theorem 5). This induces both a HLD of exponential complexity and a shallow (one hidden layer) neural network of exponential size (Theorem 6).
- In Section VI-A, in order to compute the decision boundary function in polynomial time, the folding strategy is utilized (see Theorems 9-11 for new results of folding applied to lattices). The folding strategy can be naturally implemented by a deep neural network. For instance, the decision boundary function (which enables to optimally decode) for A_n can be computed by a ReLU network of depth $\mathcal{O}(n^2)$ and width $\mathcal{O}(n)$.
- Regarding less structured lattices such as those considered in the scope of MIMO, we argue that the decoding problem on the Gaussian channel, to be addressed by a neural network, is easier compared to decoding dense lattices (in low to moderate dimensions). Namely, only a small fraction of the total number of pieces in the decision boundary function should be considered for quasi-optimal decoding. As a result, smaller shallow neural networks can be considered for random MIMO lattices, which makes the training easier and the decoding

complexity reasonable.

II. Preliminaries

This section is intended to introduce the notations for readers with a sufficient background in lattice theory. It is also useful as a short introduction to lattices for newcomers to whom we suggest reading chapters 1-4 in [6]. Additional details on all elements of this section are found in [6] and [10].

Lattice. A lattice Λ is a discrete additive subgroup of $\mathbb{R}^n$. For a rank-n lattice in $\mathbb{R}^n$, the rows of a $n \times n$ generator matrix G constitute a basis of Λ and any lattice point x is obtained via $x = z \cdot G$, where $z \in \mathbb{Z}^n$. The Gram matrix is $\Gamma = G \cdot G^T = (GQ) \cdot (GQ)^T$, where Q is any $n \times n$ orthogonal matrix. All bases defined by a Gram matrix are equivalent modulo rotations and reflections. A lower triangular generator matrix is obtained from the Gram matrix by Cholesky decomposition [7, Chap. 2]. For a given basis $\mathcal{B} = \{g_i\}_{i=1}^n$ forming the rows of G, the fundamental parallelotope of Λ is defined by

$$\mathcal{P}(\mathcal{B}) = \{y \in \mathbb{R}^n : y = \sum_{i=1}^{n} \alpha_i g_i,\ 0 \leq \alpha_i < 1\}. \tag{1}$$

The Voronoi region of x is:

$$\mathcal{V}(x) = \{y \in \mathbb{R}^n : \|y - x\| \leq \|y - x'\|, \forall x' \neq x,\ x, x' \in \Lambda\}. \tag{2}$$

A Voronoi facet of $\mathcal{V}(x)$ denotes a subset of the points

$$\{y \in \mathcal{V}(x) : \|y - x\| = \|y - x'\|, \forall x' \neq x,\ x, x' \in \Lambda\}, \tag{3}$$

which are in a common hyperplane. The boundary of $\mathcal{V}(x)$ is formed by all facets.

$\mathcal{P}(\mathcal{B})$ and $\mathcal{V}(x)$ are fundamental regions of the lattice: one can perform a tessellation of $\mathbb{R}^n$ with these regions. The fundamental volume of Λ is $\mathrm{Vol}(\mathcal{V}(x)) = \mathrm{Vol}(\mathcal{P}(\mathcal{B})) = |\det(G)|$.

The minimum Euclidean distance of Λ is $d(\Lambda) = 2\rho(\Lambda)$, where $\rho(\Lambda)$ is the packing radius. The nominal coding gain γ of a lattice Λ is given by the following ratio [10]

$$\gamma(\Lambda) = \frac{d^2(\Lambda)}{\mathrm{vol}(\Lambda)^{\frac{2}{n}}}. \tag{4}$$

A vector $v \in \Lambda$ is called Voronoi vector if the hyperplane [5]

$$\{y \in \mathbb{R}^n\ :\ y \cdot v = \frac{1}{2}||v||^2\} \tag{5}$$

has a non empty intersection with $\mathcal{V}(0)$. The vector is said relevant [6, Chap. 2] if the intersection includes a $(n-1)$-dimensional face of $\mathcal{V}(0)$. We denote by τ_f the number of relevant Voronoi vectors, referred to as the Voronoi number in the sequel. For root lattices [6], the Voronoi number is equal to the kissing number τ, defined as the number of points at a distance $d(\Lambda)$ from the origin. For random lattices, we have $\tau_f = 2^{n+1} - 2$ (with probability 1) [5]. The set $\mathcal{T}_f(x)$, for $x \in \Lambda$, denotes the set of lattice points having a common Voronoi facet with x. The theta series of Λ is [6, Chap. 2, Section 2.3]

$$\Theta_\Lambda(q) = \sum_{x \in \Lambda} q^{\|x\|^2} = \sum_{\ell=0}^{\infty} \tau_\ell q^\ell, \tag{6}$$

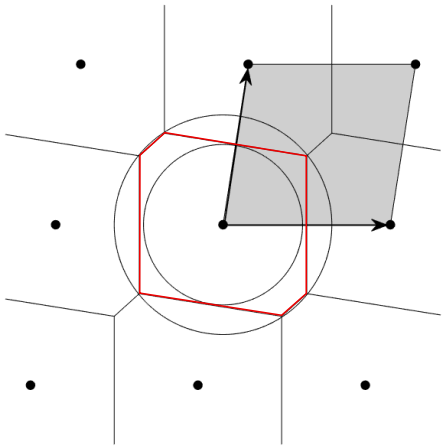


Fig. 1: Illustration of the main parameters of a lattice.

where τ_ℓ represents the number of lattice points of norm ℓ in Λ (with $\tau_{4\rho^2} = \tau$). Moreover, a lattice shell denotes the set of τ_i lattice points at a distance i from the origin. For instance, the first non-zero term of the series is $\tau q^{4\rho^2}$ as there are τ lattice points at a distance $d(\Lambda)$ from the origin. These lattice points constitute the first lattice shell.

For any lattice Λ the dual lattice Λ^* is defined as follows [6, Chap. 2, Section 2.6, (65)]:

$$\Lambda^* = \{u \in \mathbb{R}^n : u \cdot x \in \mathbb{Z},\ \forall\ x \in \Lambda\}. \tag{7}$$

Hence if G is a square generator matrix for Λ, then $(G^{-1})^T$ is a generator matrix for Λ^*. Moreover, if a lattice is $equivalent$ to its dual, it is called a self-dual (or unimodular) lattice. For instance, E_8 and Λ_{24} are self-dual.

The main lattice parameters are depicted on Figure 1. The black arrows represent a basis $\mathcal{B}$. The shaded area is the parallelotope $\mathcal{P}(\mathcal{B})$. The facets of the Voronoi region are shown in red. In this example, the Voronoi region has six facets generated by the perpendicular bisectors with six neighboring points. The two circles represent the packing sphere of radius $\rho(\Lambda)$ and the covering sphere of radius $R(\Lambda)$ respectively, $R(\Lambda) > \rho(\Lambda)$. The kissing number τ of this lattice is 2 and the Voronoi number τ_f is 6. In this case, all Voronoi vectors are relevant.

Geometry. Let $\overline{\mathcal{P}}(\mathcal{B})$ be the topological closure of $\mathcal{P}(\mathcal{B})$ and $\mathring{\mathcal{P}}(\mathcal{B})$ the interior of $\mathcal{P}(\mathcal{B})$. A k-dimensional element of $\overline{\mathcal{P}}(\mathcal{B}) \setminus \mathring{\mathcal{P}}(\mathcal{B})$ is referred to as k-face of $\mathcal{P}(\mathcal{B})$. There are 2^n 0-faces, called corners or vertices. This set of corners is denoted $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$. The subset of $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$ obtained with $z_i = 1$ is $\mathcal{C}^1_{i,\mathcal{P}(\mathcal{B})}$ and $\mathcal{C}^0_{i,\mathcal{P}(\mathcal{B})}$ for $z_i = 0$. To lighten the notations, we shall sometimes use $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ and $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$.

The remaining k-faces of $\mathcal{P}(\mathcal{B})$, $k > 0$, are parallelotopes. For instance, a $(n-1)$-face of $\mathcal{P}(\mathcal{B})$, say $\mathcal{F}_i$, is itself a parallelotope of dimension $n-1$ defined by $n-1$ vectors of $\mathcal{B}$. Throughout the paper, the term facet refers to a $n-1$-face.

Let v_j denote the vector orthogonal to the hyperplane

$$\{y \in \mathbb{R}^n : \ y \cdot v_j - p_j = 0\}. \tag{8}$$

A polytope (or convex polyhedron) is defined as the intersection of a finite number of half-spaces (as in e.g. [11])

$$P_o = \{x \in \mathbb{R}^n : \ x \cdot A \leq b, \ A \in \mathbb{R}^{n \times m}, \ b \in \mathbb{R}^m\}, \tag{9}$$

where the columns of the matrix A are m vectors v_j.

Since a parallelotope is a polytope, it can be alternatively defined from its bounding hyperplanes. Note that the vectors orthogonal to the facets of $\mathcal{P}(\mathcal{B})$ are basis vectors of the dual lattice. Hence, a second useful definition for $\mathcal{P}(\mathcal{B})$ is obtained through the basis of the dual lattice:

$$\begin{aligned} \mathcal{P}(\mathcal{B}) = &\{x \in \mathbb{R}^n : \ x \cdot G^{-1} \geq 0 \ , \quad x \cdot G^{-1} \leq 1, \\ &G \in \mathbb{R}^{n \times n}\}, \end{aligned} \tag{10}$$

where each column vector of G^{-1} is orthogonal to two facets of $\mathcal{P}(\mathcal{B})$ and $(G^{-1})^T$ is a basis for the dual lattice of Λ.

We say that a function $g : \mathbb{R}^{n-1} \to \mathbb{R}$ is CPWL if there exists a finite set of polytopes covering $\mathbb{R}^{n-1}$, and g is affine over each polytope. The number of pieces of g is the number of distinct polytopes partitioning its domain.

$\vee$ and $\wedge$ denote respectively the maximum and the minimum operator. We define a convex (resp. concave) CPWL function formed by a set of affine functions related by the operator $\vee$ (resp. $\wedge$). If $\{g_k\}$ is a set of K affine functions, the function $f = g_1 \vee ... \vee g_K$ is CPWL and convex.

Lattice decoding. Optimal lattice decoding refers to finding the closest lattice point, the closest in Euclidean distance sense. This problem is also known as the CVP. Its associated decision problem is NP-complete [17, Chap. 3].

Let $x \in \Lambda$ and η be a Gaussian vector where each component is i.i.d $\mathcal{N}(0, \sigma^2)$. Consider $y \in \mathbb{R}^n$ obtained as

$$y = x + \eta. \tag{11}$$

Since this model is often used in digital communications, x is referred to as the transmitted point, y the received point, and the process described by (11) is called a Gaussian channel. Given equiprobable inputs, maximum-likelihood decoding (MLD) on the Gaussian channel is equivalent to solving the CVP. Moreover, we say that a decoder is quasi-MLD (QMLD) if $\mathcal{P}_{dec}(\sigma^2) \leq \mathcal{P}_{opt}(\sigma^2) \cdot (1 + \epsilon)$, where $\epsilon > 0$.

In the scope of (infinite) lattices, the transmitted information rate and the signal-to-noise ratio based on the second-order moment are pointless. Poltyrev introduced the generalized capacity [22] [29], the analog of Shannon capacity for lattices. The Poltyrev limit corresponds to a noise variance of $\sigma^2_{max} = \mathrm{Vol}(\Lambda)^{\frac{2}{n}}/(2\pi e)$. 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) [29], i.e.

$$\Delta = \frac{\sigma^2_{max}}{\sigma^2}. \tag{12}$$

The reader should not confuse this VNR Δ with the standart notation of the lattice sphere packing density as in Section 1.2 of [6]. Using the union bound with the Theta series (see (6)), the MLD probability of error per lattice point of lattice Λ can be bounded from above by [6, Chap. 3, Section 1.3, (19)]

$$P_e(opt) \leq P_e(ub), \tag{13}$$

where [6, Chap. 3, Section 1.4, (19) and (35)]

$$P_e(ub) = \frac{1}{2}\Theta_\Lambda\left(\exp(-\frac{1}{8\sigma^2})\right) - \frac{1}{2} = \frac{1}{2}\sum_{x\in\Lambda\setminus\{0\}} \exp\left(-\frac{\|x\|^2}{8\sigma^2}\right). \tag{14}$$

It can be easily shown that $\frac{\rho^2}{2\sigma^2} = \frac{\pi e\Delta\gamma}{4}$. For $\Delta \to \infty$, the term $\tau q^{4\rho^2}$ dominates the sum in $\Theta_\Lambda(q)$ [6, Chap. 3, Section 1.4, (21)]. As proven in Appendix A, (14) becomes

$$P_e(ub) = \frac{\tau}{2}\exp(-\frac{\pi e\Delta\gamma}{4}) + o\left(\exp(-\frac{\pi e\Delta\gamma}{4})\right). \tag{15}$$

Finally, lattices are often used to model MIMO channels [23, Chap. 15]. Consider a flat quasi-static MIMO channel with $n/2$ transmit antennas and $n/2$ receive antennas. Any complex matrix of size $n/2$ can be trivially transformed into a real matrix of size n. Let G be the $n \times n$ real matrix representing the channel coefficients. Let $z \in \mathbb{Z}^n$ be the channel input, i.e., z is the uncoded information sequence. The input message yields the output $y \in \mathbb{R}^n$ via the standard flat MIMO channel equation,

$$y = \underbrace{z \cdot G}_{x} + \eta.$$

A MIMO lattice shall refer to a lattice generated by a matrix G representing a MIMO channel.

Neural networks. Given n scalar inputs $y_1, ..., y_n$ a perceptron performs the operation $\sigma(\sum_i w_i \cdot y_i)$ [12, Chap. 1]. The parameters w_i are called the weights or edges of the perceptron and $\sigma(\cdot)$ is the activation function. The activation function $\sigma(x) = \max(0, x)$ is called ReLU. A perceptron can alternatively be called a neuron.
Given the inputs $y = (y_1, ..., y_n)$, a feed-forward neural network simply performs the operation [12, Chap. 6]:

$$\hat{z} = \sigma_d(...\sigma_2(\sigma_1(y \cdot G_1 + b_1) \cdot G_2 + b_2) \cdot ... \cdot G_d + b_d), \tag{16}$$

where:

- d is the number of layers of the neural network.
- Each layer of size m_i is composed of m_i neurons. The weights of the neurons in the ith layer are stored in the m_i columns of the matrix G_i. The vector b_i represents m_i biases.
- The activation functions σ_i are applied componentwise.

III. From the CVP in $\mathbb{R}^n$ to the CVP in $\mathcal{P}(\mathcal{B})$.

It is well known in lattice theory that $\mathbb{R}^n$ can be partitioned as $\mathbb{R}^n = \bigcup_{x\in\Lambda}(\mathcal{P}(\mathcal{B}) + x)$. The parallelotope to which a point $y_0 \in \mathbb{R}^n$ belongs is:

$$y_0 \in \mathcal{P}(\mathcal{B}) + x, \tag{17}$$

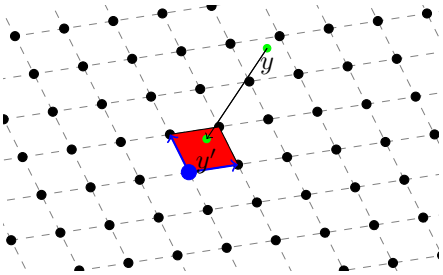


Fig. 2: Translation of a noisy point into the fundamental parallelotope.

with

$$x = \lfloor y_0 G^{-1} \rfloor \cdot G, \tag{18}$$

where the floor function $\lfloor \cdot \rfloor$ is applied componentwise. This floor function should not be confused with the round function $\lfloor \cdot \rceil$. Hence, a translation of y_0 by $-x$ results in a point y located in the fundamental parallelotope $\mathcal{P}(\mathcal{B})$. An instance of this operation is illustrated on Figure 2. The point $y \in \mathcal{P}(\mathcal{B}) + x$ is translated in the fundamental parallelotope (in red on the figure) to get the point $y' \in \mathcal{P}(\mathcal{B})$. The blue arrows represent a basis $\mathcal{B}$ of the lattice. As a result, a point y_0 to decode (e.g. in the scope of the CVP) can be processed as follows:

Parallelotope-Based Decoding.

- Step 0: a noisy lattice point $y_0 = x + \eta$ is observed, where $x \in \Lambda$ and $\eta \in \mathbb{R}^n$ is any additive noise.
- Step 1: compute $t = \lfloor y_0 \cdot G^{-1} \rfloor$ and get $y = y_0 - t \cdot G$ which now belongs to $\mathcal{P}(\mathcal{B})$.
- Step 2: find $\hat{z}$, where $\hat{x} = \hat{z} \cdot G$ is the closest lattice point to y.
- Step 3: the closest point to y_0 is $\hat{x}_0 = \hat{x} + t \cdot G$.

Since Step 1 and Step 3 have negligible complexity, an equivalent problem to the CVP (in $\mathbb{R}^n$) is the CVP in $\mathcal{P}(\mathcal{B})$ (Step 2 above), which can simply be stated as follows.

Problem 1. *(CVP in $\mathcal{P}(\mathcal{B})$) Given a point $y \in \mathcal{P}(\mathcal{B})$, find the closest lattice point $\hat{x} = \hat{z} \cdot G$.*

Remark 1. *Consider a point $y = x + \eta$, where $\eta = \epsilon_1 g_1 + ... + \epsilon_n g_n$, $x \in \Lambda$, $0 \leq \epsilon_1, ..., \epsilon_n < 1$, $g_1, ..., g_n \in \mathcal{B}$. Obviously, $y \in x + \mathcal{P}(\mathcal{B})$. The well-known Zero-Forcing (ZF) decoding algorithm computes*

$$\hat{z} = \lfloor y \cdot G^{-1} \rceil = \lfloor y_0 \cdot G^{-1} \rceil + x G^{-1}. \tag{19}$$

In other words, it simply replaces each ϵ_i by the closest integer, i.e. 0 or 1. The solution provided by this algorithm is one of the corners of the parallelotope $x + \mathcal{P}(\mathcal{B})$.

Remark 2. *From a complexity theory view point, Problem 1 is NP-hard. Indeed, since the above Steps 0, 1, and 3 are of polynomial complexity, the CVP, which is known to be NP-hard [17, Chap. 3], is polynomially reduced to Problem 1.*

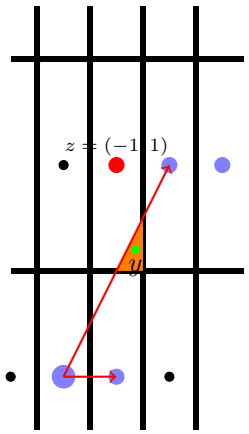


Fig. 3: Example of a non-VR basis.

IV. Voronoi-reduced lattice basis

A. Voronoi- and quasi-Voronoi-reduced basis

The natural question arising from Problem 1 is the following: Is the closest lattice point to any point $y \in \mathcal{P}(\mathcal{B})$ one of the corners of $\mathcal{P}(\mathcal{B})$? Unfortunately, as illustrated in Figure 3, this is not always the case. The red arrows in the figure represent the basis vectors. The orange area in $\mathcal{P}(\mathcal{B})$ belongs to the Voronoi region of the point $x = z \cdot G$, where $z = (-1, 1)$ (in red on the figure). Since this lattice point is not a corner of $\mathcal{P}(\mathcal{B})$, any point in this orange area, such as y, is not decoded to one of the corner of $\mathcal{P}(\mathcal{B})$ (the four blue points on the figure). Consequently, we introduce a new type of basis reduction.

Definition 1. *Let $\mathcal{B}$ be the $\mathbb{Z}$-basis of a rank-n lattice Λ in $\mathbb{R}^n$. $\mathcal{B}$ is said Voronoi-reduced if, for any point $y \in \mathcal{P}(\mathcal{B})$, the closest lattice point $\hat{x}$ to y is one of the 2^n corners of $\mathcal{P}(\mathcal{B})$, i.e. $\hat{x} = \hat{z}G$ where $\hat{z} \in \{0, 1\}^n$.*

We will use the abbreviation *VR basis* to refer to a Voronoi-reduced basis. Figure 4 shows the hexagonal lattice A_2, its Voronoi regions, and the fundamental parallelotope of the basis $\mathcal{B}_1 = \{v_1, v_2\}$, where $v_1 = (1, 0)$ corresponds to $z = (1, 0)$ and $v_2 = (\frac{1}{2}, \frac{\sqrt{3}}{2})$ corresponds to $z = (0, 1)$. $\mathcal{P}(\mathcal{B}_1)$ is partitioned into 4 parts included in the Voronoi regions of its corners. $\mathcal{P}(\mathcal{B}_2)$ has 10 parts involving 10 Voronoi regions. The small black dots in $\mathcal{P}(\mathcal{B})$ represent Gaussian distributed points in $\mathbb{R}^2$ that have been aliased in $\mathcal{P}(\mathcal{B})$. The basis $\mathcal{B}_1$ is Voronoi-reduced because

$$\mathcal{P}(\mathcal{B}_1) \subset \mathcal{V}(0) \cup \mathcal{V}(v_1) \cup \mathcal{V}(v_2) \cup \mathcal{V}(v_1 + v_2). \tag{20}$$

Lattice basis reduction is an important field in Number Theory. In general, a lattice basis is said to be of good quality when the basis vectors are relatively short and close to being orthogonal. We cite three famous types of reduction to get a good basis: Minkowski-reduced basis, Korkin-Zolotarev-reduced (or Hermite-reduced) basis, and LLL-reduced basis for Lenstra-Lenstra-Lovász [17] [7]. A basis is said to be LLL-reduced if it has been processed by the LLL algorithm. This algorithm, given an input basis of a lattice, outputs a new basis in polynomial time where the new basis respects some criteria, see e.g. [7]. The LLL-reduction is widely used in practice to improve the quality of a basis. The basis $\mathcal{B}_1$ in Figure 4 is Minkowski-, KZ-, and Voronoi-reduced.

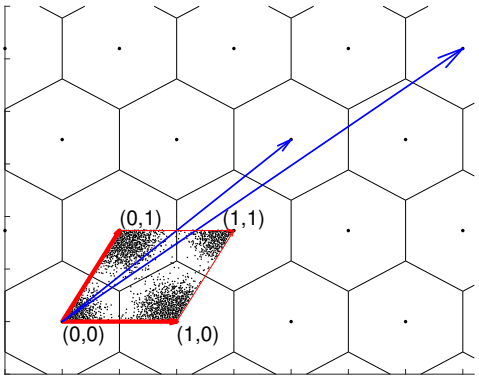


Fig. 4: Voronoi-reduced basis $\mathcal{B}_1$ for A_2 (in red) and a non-reduced basis $\mathcal{B}_2$ (in blue).

Note that this new notion ensures that the closest lattice point $\hat{x}$ to any point $y \in \mathcal{P}(\mathcal{B})$ is obtained with a vector $\hat{z}$ having only binary values (where $\hat{x} = \hat{z} \cdot G$). As a result, it enables to use a decoder with only binary outputs to optimally solve the CVP in $\mathcal{P}(\mathcal{B})$.

Unfortunately, not all lattices admit a VR basis (see the following subsection). Nevertheless, as we shall see in the sequel, some famous dense lattices listed in [6] admit a VR basis. Also, in some cases the LLL-reduction leads to a $quasi$-VR basis. Indeed, the strong constraint defining a VR basis can be relaxed as follows.

Definition 2. *Let $\mathcal{C}(\mathcal{B})$ be the set of the 2^n corners of $\mathcal{P}(\mathcal{B})$. Let $\mathscr{O}$ be the subset of $\mathcal{P}(\mathcal{B})$ that is covered by Voronoi regions of points not belonging to $\mathcal{C}(\mathcal{B})$, namely*

$$\mathscr{O} = \mathcal{P}(\mathcal{B}) \setminus \left(\bigcup_{x \in \mathcal{C}(\mathcal{B})} V(x) \right). \tag{21}$$

The basis $\mathcal{B}$ is said quasi-Voronoi-reduced if $\mathrm{Vol}(\mathscr{O})$ is negligible compared to $\mathrm{Vol}(\Lambda)$, i.e. $\mathrm{Vol}(\mathscr{O}) \ll \mathrm{Vol}(\Lambda)$.

Let

$$d^2_{\mathscr{O}C}(\mathcal{B}) = \min_{x \in \mathscr{O}, x' \in \mathcal{C}(\mathcal{B})} \|x - x'\|^2 \tag{22}$$

be the minimum squared Euclidean distance between $\mathscr{O}$ and $\mathcal{C}(\mathcal{B})$. The sphere packing structure associated to Λ guarantees that $d^2_{\mathscr{O}C} \geq \rho^2$. Let $P_e(\mathcal{B})$ be the probability of error for a decoder where the closest corner of $\mathcal{P}(\mathcal{B})$ to y is decoded. In other words, the space of solution for this decoder is restricted to $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$. The following lemma tells us that a quasi-Voronoi-reduced basis exhibits quasi-optimal performance on a Gaussian channel at high signal-to-noise ratio. In practice, the quasi-optimal performance is also observed at moderate values of signal-to-noise ratio.

Lemma 1. *The error probability on the Gaussian channel when decoding a lattice Λ in $\mathcal{P}(\mathcal{B})$ can be bounded from above as*

$$P_e(\mathcal{B}) \leq P_e(ub) + \frac{\mathrm{Vol}(\mathscr{O})}{\det(\Lambda)} \cdot (e\Delta)^{n/2} \cdot \exp(-\frac{\pi e \Delta \gamma}{4} \cdot \frac{d^2_{\mathscr{O}C}}{\rho^2}), \tag{23}$$

for Δ large enough and where $P_e(ub)$ is defined by (15)*.*

Proof. If $\mathcal{B}$ is Voronoi-reduced and the decoder works inside $\mathcal{P}(\mathcal{B})$ to find the nearest corner, then the performance is given by $P_e(opt)$.

If $\mathcal{B}$ is quasi-Voronoi-reduced and the decoder only decides a lattice point from $\mathcal{C}(\mathcal{B})$, then an error shall occur each time y falls in $\mathscr{O}$. We get

$$\begin{aligned} P_e(\mathcal{B}) &\leq P_e(opt) + P_e(\mathscr{O}), \\ &\leq P_e(ub) + P_e(\mathscr{O}). \end{aligned} \tag{24}$$

where

$$\begin{aligned} P_e(\mathscr{O}) &= \int \cdots \int_{\mathscr{O}} \frac{1}{\sqrt{2\pi\sigma^2}^n} \exp(-\frac{\|x\|^2}{2\sigma^2})\, dx_1 \ldots dx_n \\ &\leq \frac{1}{\sqrt{2\pi\sigma^2}^n} \exp(-\frac{d^2_{\mathscr{O}C}}{2\sigma^2})\ \ \mathrm{Vol}(\mathscr{O}) \\ &= \frac{\mathrm{Vol}(\mathscr{O})}{\det(\Lambda)} \cdot (e\Delta)^{n/2} \cdot \exp(-\frac{\pi e \Delta \gamma}{4} \cdot \frac{d^2_{\mathscr{O}C}}{\rho^2}). \end{aligned}$$

This completes the proof. □

B. Some examples

1) Structured lattices: We first state the following three theorems on the existence of VR bases for some famous lattices. The proofs are provided in Appendix B.

Consider a basis for the lattice A_n with all vectors from the first lattice shell. Also, the angle between any two basis vectors is $\pi/3$. Let J_n denote the $n \times n$ all-ones matrix and I_n the identity matrix. The Gram matrix is

$$\Gamma_{A_n} = G \cdot G^T = J_n + I_n = \begin{pmatrix} 2 & 1 & 1 & ... & 1 \\ 1 & 2 & 1 & ... & 1 \\ 1 & 1 & 2 & ... & 1 \\ . & . & . & ... & . \\ 1 & 1 & 1 & ... & 2 \end{pmatrix}. \tag{25}$$

Theorem 1. *A lattice basis of A_n defined by the Gram matrix* (25) *is Voronoi-reduced.*

Consider the following Gram matrix of E_8.

$$\Gamma_{E_8} = \begin{pmatrix} 4 & 2 & 0 & 2 & 2 & 2 & 2 & 2 \\ 2 & 4 & 2 & 0 & 2 & 2 & 2 & 2 \\ 0 & 2 & 4 & 0 & 2 & 2 & 0 & 0 \\ 2 & 0 & 0 & 4 & 2 & 2 & 0 & 0 \\ 2 & 2 & 2 & 2 & 4 & 2 & 2 & 0 \\ 2 & 2 & 2 & 2 & 2 & 4 & 0 & 2 \\ 2 & 2 & 0 & 0 & 2 & 0 & 4 & 0 \\ 2 & 2 & 0 & 0 & 0 & 2 & 0 & 4 \end{pmatrix}. \tag{26}$$

Theorem 2. *A lattice basis of E_8 defined by the Gram matrix* (26) *is Voronoi-reduced with respect to $\mathring{\mathcal{P}}(\mathcal{B})$.*

Theorem 3. *There exists no Voronoi-reduced basis for Λ_{24}.*

Unfortunately, for most lattices such theorems can not be proved. However, quasi-Voronoi-reduced bases can sometimes be obtained. For instance, the following Gram matrix corresponds to a quasi-Voronoi-reduced basis of E_6:

$$\Gamma_{E_6} = \begin{pmatrix} 3 & \frac{3}{2} & 0 & 0 & \frac{3}{2} & \frac{3}{2} \\ \frac{3}{2} & 3 & 0 & 0 & \frac{3}{2} & \frac{3}{2} \\ 0 & 0 & 3 & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} \\ 0 & 0 & \frac{3}{2} & 3 & \frac{3}{2} & \frac{3}{2} \\ \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & 3 & \frac{3}{2} \\ \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & \frac{3}{2} & 3 \end{pmatrix}, \tag{27}$$

with $\frac{d^2_{\mathscr{O}C}}{\rho^2} = 1.60$ (2dB of gain) and $\frac{\mathrm{Vol}(\mathscr{O})}{\det(\Lambda)} = 2.47 \times 10^{-3}$. The ratio of $P_e(ub)$ by the second term of the right-hand side of (23) is about 10^{-4} at $\Delta = 1$ (0 dB) then vanishes further for increasing Δ.

Obviously, the quasi-VR property is good enough to allow the application of a decoder working with $\mathcal{C}(\mathcal{B})$.If an optimal decoder is required, e.g. in specific applications such as lattice shaping and cryptography, the user should let the decoder manage extra points outside $\mathcal{C}(\mathcal{B})$. For example, the disconnected region $\mathscr{O}$ (see (21)) for E_6 defined by Γ_{E_6} includes extra points where $z_i \in \{-1, 0, 1, +2\}$ instead of $\{0, 1\}$ as for $\mathcal{C}(\mathcal{B})$.

2) Unstructured MIMO lattices: We investigate the VR properties of typical random MIMO lattices where the lattice is generated by a real matrix G whose associated $n/2 \times n/2$ complex matrix has i.i.d. circular symetric $\mathcal{CN}(0, 1)$ entries. The basis obtained via this random process is in general of poor quality. As mentioned in the previous subsection, the standard and cheap process to obtained a basis of better quality is to apply the LLL algorithm. As a result, we are interested in the following question: Is a LLL-reduced random MIMO lattice quasi-Voronoi-reduced?

In the previous subsection, we highlighted that two specific quantities characterize the loss in the error probability on the Gaussian channel ($P_e(O)$, see Equation (24)) due to non-VR parts of $\mathcal{P}(\mathcal{B})$: $\mathrm{Vol}(O)$ and $d_{OC}(\mathcal{B})$.

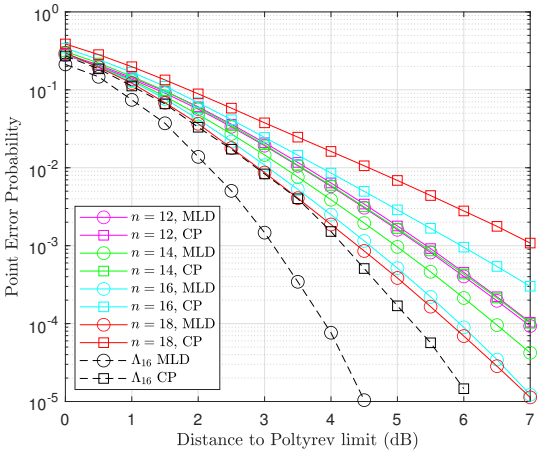


Fig. 5: Assessment of the performance loss due to non-VR parts of $\mathcal{P}(\mathcal{B})$.

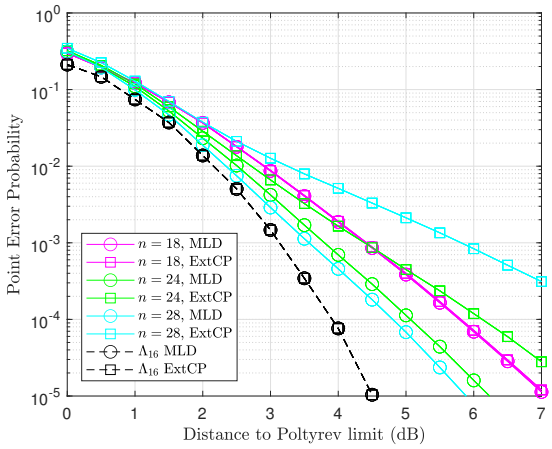


Fig. 6: Assessment of the performance loss with the extended corner points.

Unfortunately, for a given basis, these quantities are in general difficult to compute because it requires sampling in a n-dimensional space. In fact, one can directly estimate the term $P_e(O)$, without evaluate numerically these two quantities via Monte Carlo simulations. It is simpler to directly compute $P_e(O)$. Noisy points $y_0 = x + \eta$ are generated as in Step 0 of the parallelotope-based decoding in Section III, then the shifted versions of $\mathcal{P}(\mathcal{B})$ containing y_0 are determined as in Step 1 of the parallelotope-based decoding, and finally y_0 points are decoded with an optimal algorithm. If the decoded point is not a corner of $\mathcal{P}(\mathcal{B})$, i.e. $\hat{z} \notin \{0,1\}^n$, we declare an error. However, if the decoded point is a corner of $\mathcal{P}(\mathcal{B})$ but it is different from the transmitted lattice point x, we also declare an error. This is shown by the curves with caption named CP (for Corner Points) in Figure 5. Comparing the resulting performance with the one obtained with the optimal algorithm enables to assess the term $P_e(O)$ and observe the loss in the error probability on the Gaussian channel caused by the non-VR parts of $\mathcal{P}(\mathcal{B})$.

The simulation results are depicted on Figure 5 where we show performance loss, on the Gaussian channel, due to non-VR parts of $\mathcal{P}(\mathcal{B})$ for LLL-reduced random MIMO lattices. For each point, we average the performance over 1000 random generator matrices G. Up to dimension $n = 12$, considering only the corners of $\mathcal{P}(\mathcal{B})$ yields no significant loss in performance. We can conclude that, on average for the considered model, a LLL-reduced basis for $n \leq 12$ is quasi-VR. However, for larger dimensions, the loss increases and becomes significant. On the figure, we also added the performance of the dense lattice Λ_{16} (also called Barnes-Wall lattice in dimension 16 [6, Chap. 4]) for comparison. Obviously, the basis considered in not VR.

Figure 6 shows the performance of a decoder with extended corner points (ExtCP) versus the maximum-likelihood decoder (MLD). The VR concept assumes $z_i \in \{0,1\}$. Here, the ExtCP decoder looks for the nearest lattice point slightly beyond the corners of $\mathcal{P}(\mathcal{B})$ by considering $z_i \in \{-1,0,1,2\}$. This illustrates that the VR notion can be extended to consider z_i values belonging to a larger set.

In summary, the VR approximation can be made for a LLL-reduced random MIMO lattice up to dimension 12

(6 antennas) and the extended corner-points decoding is quasi-optimal up to dimension 18 (9 antennas).

V. Finding the closest corner of $\mathcal{P}(\mathcal{B})$ for decoding

Thanks to the previous section, we know that the CVP in $\mathcal{P}(\mathcal{B})$, with a VR basis, can be optimally solved with an algorithm having only binary outputs. In this section, we show how each z_i can be decoded independently in $\mathcal{P}(\mathcal{B})$ via a decision boundary. Our main objective shall be to characterize this decision boundary. The decision boundary enables to find, componentwise, the closest corner of $\mathcal{P}(\mathcal{B})$ to any point $y \in \mathcal{P}(\mathcal{B})$. This process exactly solves the CVP if the basis is VR. This discrimination can be implemented with the hyperplane logical decoder (HLD). It can also be applied to lattices admitting only a quasi-VR basis to yield quasi-MLD performance in presence of additive white Gaussian noise. The complexity of the HLD depends on the number of affine pieces in the decision boundary, which is exponential in the dimension. More generally, we shall see that this exponential number of pieces induces shallow neural networks of exponential size.

A. The decision boundary

We show how to decode one component of the vector $\hat{z}$. Without loss of generality, if not specified, the integer coordinate to be decoded for the rest of this section is $\hat{z}_1$. The process presented in this section should be repeated for each z_i, $1 \leq i \leq n$ to recover all the components of $\hat{z}$. Given a lattice with a VR basis, exactly half of the corners of $\mathcal{P}(\mathcal{B})$ are obtained with $z_1 = 1$ and the other half with $z_1 = 0$. Therefore, one can partition $\mathcal{P}(\mathcal{B})$ in two regions, where each region is:

$$\mathcal{R}_{\mathcal{C}^i_{\mathcal{P}(\mathcal{B})}} = \bigcup_{x \in \mathcal{C}^i_{\mathcal{P}(\mathcal{B})}} \mathcal{V}(x) \cap \mathcal{P}(\mathcal{B}), \tag{28}$$

with $i = 1$ or 0. The intersections between $\mathcal{R}_{\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}}$ and $\mathcal{R}_{\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}}$ define a boundary. This boundary splitting $\mathcal{P}(\mathcal{B})$ into two regions $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ and $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$, is the union of some of the Voronoi facets of the corners of $\mathcal{P}(\mathcal{B})$. Each facet can be defined by an affine function over a compact subset of $\mathbb{R}^{n-1}$, and the boundary is locally described by one of these functions.

Obviously, the position of a point to decode with respect to this boundary determines whether $\hat{z}_1$ should be decoded to 1 or 0. For this reason, we call this boundary the decision boundary. Moreover, the hyperplanes involved in the decision boundary are called boundary hyperplanes. An instance of a decision boundary is illustrated on Figure 7 where the green point y, $\hat{z}_1$ should be decoded to 1 because y is above the decision boundary.

B. Decoding via a Boolean equation

Let $\mathcal{B}$ be VR basis. The CVP in $\mathcal{P}(\mathcal{B})$ is solved componentwise, by comparing the position of y with the Voronoi facets partitioning $\mathcal{P}(\mathcal{B})$. This can be expressed in the form of a Boolean equation, where the binary (Boolean) variables are the positions with respect to the facets (on one side or another). Therefore, one should compute the position of y relative to the decision boundary via a Boolean equation to guess whether $\hat{z}_1 = 0$ or $\hat{z}_1 = 1$.

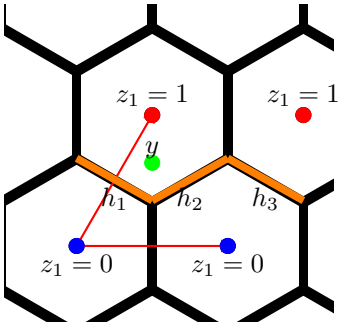


Fig. 7: The hexagonal lattice A_2 with a VR basis. The two upper corners of $\mathcal{P}(\mathcal{B})$ (in red) are obtained with $z_1 = 1$ and the two other ones with $z_1 = 0$ (in blue). The decision boundary is illustrated in orange.

Consider the orthogonal vectors to the hyperplanes containing the Voronoi facet of a point $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ and a point from $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. These vectors are denoted by v_j as in (8). A Boolean variable $u_j(y)$ is obtained as:

$$u_j(y) = \text{Heav}(y \cdot v_j - p_j) \in \{0, 1\}, \tag{29}$$

where Heav$(\cdot)$ stands for the Heaviside function. Since $\mathcal{V}(x) = \mathcal{V}(0) + x$, orthogonal vectors v_j to all facets partitioning $\mathcal{P}(\mathcal{B})$ are determined from the facets of $\mathcal{V}(0)$.

Example 1. *Let $\hat{z} = (\hat{z}_1, \hat{z}_2)$ and $y \in \mathcal{P}(\mathcal{B})$ the point to be decoded. Given the red basis on Figure 7, the first component $\hat{z}_1$ is 1 (true) if y is above hyperplanes h_1 and h_2 simultaneously or above h_3. Let $u_1(y), u_2(y)$, and $u_3(y)$ be Boolean variables, the state of which depends on the location of y with respect to the hyperplanes h_1, h_2, and h_3, respectively. We get the Boolean equation $\hat{z}_1 = u_1(y) \cdot u_2(y) + u_3(y)$, where $+$ is a logical OR and $\cdot$ stands for a logical AND.*

Given a lattice $\Lambda \subset \mathbb{R}^n$ of rank n, Algorithm 1 enables to find the Boolean equation of a coordinate $\hat{z}_i$. It also finds the equation of each hyperplane needed to get the value of the Boolean variables involved in the equation. This algorithm can be seen as a "training" step to "learn" the structure of the lattice. It is a brute-force search that may quickly become too complex as the dimension increases. However, we shall see in Section V-D and V-E that these Boolean equations can be analyzed without this algorithm, via a study of the basis. Note that the decoding complexity does not depend on the complexity of this search algorithm.

C. The HLD

The HLD is a brute-force algorithm to compute the Boolean equation provided by Algorithm 1. The HLD can be executed via the three steps summarized in Algorithm 2.

1) Implementation of the HLD: Since Steps 1-2 are simply linear combinations followed by activation functions, these operations can be written as:

$$l_1 = \sigma(y \cdot G_1 + b_1), \tag{30}$$

Algorithm 1 Brute-force search to find the Boolean equation of a coordinate $\hat{z}_i$ for a lattice Λ

Select the 2^{n-1} corners of $\mathcal{P}(\mathcal{B})$ where $z_i = 1$ and all relevant Voronoi vectors of Λ.
for each of the 2^{n-1} corners where $z_i = 1$ **do**
  for each relevant Voronoi vector of Λ **do**
    Move in the direction of the selected relevant Voronoi vector by half its norm + ϵ (ϵ being a small number).
    if The resulting point is outside $\mathcal{P}(\mathcal{B})$. **then**
      Do nothing. //There is no decision boundary hyperplane in this direction.
    else
      Find the closest lattice point $x' = z'G$ (e.g. by sphere decoding [1]).
      if $z'_i = 1$ **then**
        Do nothing. //There is no decision boundary hyperplane in this direction.
      else
        Store the decision boundary orthogonal to this direction. //$z'_i = 0$
      end if
    end if
  end for
  for each decision boundary hyperplane found (at this corner) **do**
    Associate and store a Boolean variable to this hyperplane (corresponding to the position of the point to be decoded with respect to the hyperplane).
  end for
  The Boolean equation of $\hat{z}_i$ contains a Boolean AND of these variables.
end for
The equation is the Boolean OR of the 2^{n-1} AND coming from all corners.

Algorithm 2 HLD

Compute the inner product of y with the vectors orthogonal to the decision boundary hyperplanes.
Apply the Heaviside function on the resulting quantities to get its relative positions under the form of Boolean variables.
Compute the logical equations associated to each coordinate.

where σ is the Heaviside function, G_1 a matrix having the vectors v_j as columns, and b_1 a vector of biases containing the p_j. Equation (30) describes the operation performed by a layer of a neural network (see (16)) . The layer l_1 is a vector containing the Boolean variables $u_j(y)$.

Let l_{i-1} be a vector of Boolean variables. It is well known that both Boolean AND and Boolean OR can be expressed as:

$$l_i = \sigma(l_{i-1} \cdot G_i + b_i),$$

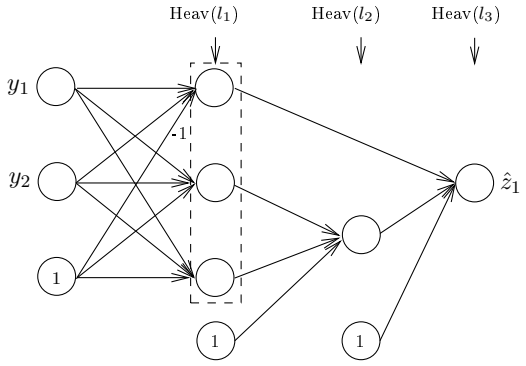


Fig. 8: Neural network performing HLD decoding on the first symbol $\hat{z}_1$ of a point in $\mathcal{P}(\mathcal{B})$ for the lattice A_2 (see Example 1).

where G_i a matrix composed of 0 and 1, and b_i a vector of biases. Therefore, the mathematical expression of the HLD is:

$$z_1 = \sigma(\sigma(\sigma(y \cdot G_1 + b_1) \cdot G_2 + b_2) \cdot G_3 + b_3). \tag{31}$$

Equation (31) is exactly the definition of a feed-forward neural network (see (16)) with three layers. Figure 8 illustrates the topology of the neural network obtained when applying the HLD to the lattice A_2. Heav(·) stands for Heaviside(·). The first part of the network computes the position of y with respect to the boundary hyperplanes to get the variables $u_j(y)$. The second part (two last layers) computes the Boolean ANDs and Boolean ORs of the decoding Boolean equation.

D. The decision boundary as a piecewise affine function

In order to better understand the decision boundary, we characterize it as a function rather than a Boolean equation. We shall see in the sequel that it is sometimes possible to efficiently compute this function and thus reduce the decoding complexity.

Let $\{e_i\}_{i=1}^n$ be the canonical orthonormal basis of the vector space $\mathbb{R}^n$. For $y \in \mathbb{R}^n$, the i-th coordinate is $y_i = y \cdot e_i$. Denote $\tilde{y} = (y_2, \ldots, y_n) \in \mathbb{R}^{n-1}$ and let $\mathcal{H} = \{h_j\}$ be the set of affine functions involved in the decision boundary. The affine boundary function $h_j : \mathbb{R}^{n-1} \to \mathbb{R}$ is

$$h_j(\tilde{y}) = y_1 = \left(p_j - \sum_{k \neq 1} y_k v_j^k\right) / v_j^1, \tag{32}$$

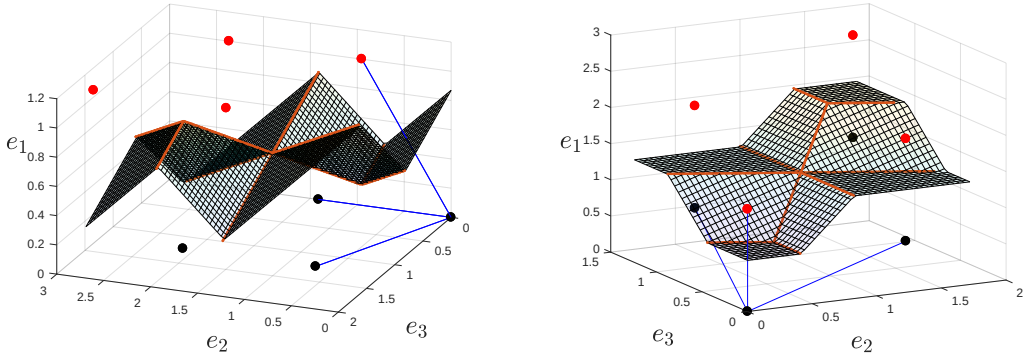
where v_j^k is the kth component of vector v_j. For the sake of simplicity, in the sequel h_j shall denote the function defined in (32) or its associated hyperplane depending on the context.

Theorem 4. *Consider a lattice defined by a VR basis $\mathcal{B} = \{g_i\}_{i=1}^n$. Let $\mathcal{H} = \{h_j\}$ be the set of affine functions involved in the decision boundary. Assume that $g_1^1 > 0$. Suppose also that $x_1 > \lambda_1$ (in the basis $\{e_i\}_{i=1}^n$), $\forall x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ and $\forall \lambda \in \mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. Then, the decision boundary is given by a CPWL function $f : \mathbb{R}^{n-1} \to \mathbb{R}$, expressed as*

$$f(\tilde{y}) = \wedge_{m=1}^{M} \{\vee_{k=1}^{l_m} h_{m,k}(\tilde{y})\}, \tag{33}$$

where $h_{m,k} \in \mathcal{H}$, $1 \leq l_m < \tau_f$, and $1 \leq M \leq 2^{n-1}$.

The proof is provided in Appendix C. In the previous theorem, the orientation of the axes relative to $\mathcal{B}$ does not require $\{g_i\}_{i=2}^n$ to be orthogonal to e_1. This is however the case for the next corollary, which involves a specific rotation satisfying the assumption of the previous theorem. Indeed, with the following orientation, any point in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is in the hyperplane $\{y \in \mathbb{R}^n : \ y \cdot e_1 = 0\}$ and has its first coordinate equal to 0, and $g_1^1 > 0$ (if it is negative, simply multiply the basis vectors by -1).



(a) The orientation of the basis satisfies the assumptions of Theorem 4 and Corollary 1.

(b) The orientation of the basis satisfies the assumptions of Theorem 4 but not the ones of Corollary 1.

Fig. 9: CPWL decision boundary function for A_3. The basis vectors are represented by the blue lines. The corner points in $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ are in red and the corner points in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ in black.

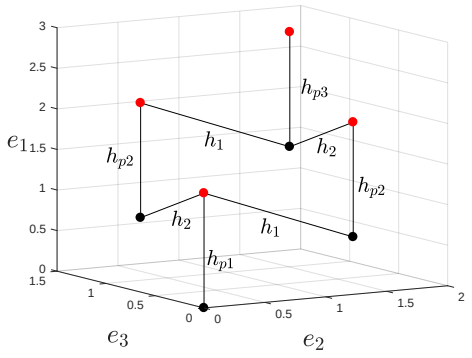


Fig. 10: "Neighbor figure" of $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$ for the lattice A_3.

Corollary 1. *Consider a lattice defined by a VR basis $\mathcal{B} = \{g_i\}_{i=1}^n$. Suppose that the $n-1$ points $\mathcal{B}\backslash\{g_1\}$ belong to the hyperplane $\{y \in \mathbb{R}^n : \ y \cdot e_1 = 0\}$. Then, the decision boundary is given by a CPWL function as in (33).*

Example 2. *Consider the lattice A_3 defined by the Gram matrix* (25)*. To better illustrate the symmetries we rotate the basis*[1] *to have g_1 colinear with e_1. Theorem 4 ensures that the decision boundary is a function. The rotated function is illustrated in Figure 9b and the non-rotated version in Figure 9a. On Figure 10 each edge is orthogonal to a local affine function of f. The edges are labeled with the name of the corresponding affine function. Each edge connects a point $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ to an element of $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. Consequently, each edge is orthogonal to a local affine function of the decision boundary function f. The edges are labeled with the names of the corresponding affine functions. Theorem 5 and its proof show that each set $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ generates a convex part of the decision boundary function f with $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}|$ pieces. E.g. on the figure there are one set with $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = 3$, two with $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = 2$, and one with $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = 1$, thus f has 8 pieces. The equation of the function is (we omit the $\tilde{y}$ in the formula to lighten the notations):*

$$f = \Big[h_{p_1} \vee h_1 \vee h_2\Big] \wedge \Big[\,(h_{p_2} \vee h_1) \wedge (h_{p_2} \vee h_2)\,\Big] \wedge \Big[h_{p_3}\Big],$$

where h_{p_1}, h_{p_2} and h_{p_3} are hyperplanes orthogonal to g_1 (the p index stands for plateau) and the $[\cdot]$ groups all the set of convex pieces of f that includes the same h_{p_j}. Functions for higher dimensions (i.e. $A_n, n \geq 3$) are available in Appendix D.

The notion of decision boundary function can be generalized to non-VR basis under the assumptions of the following definition. A surface in $\mathbb{R}^n$ defined by a function g of $n-1$ arguments is written as $\text{Surf}(g) = \{y = (g(\tilde{y}), \tilde{y}) \in \mathbb{R}^n, \tilde{y} \in \mathbb{R}^{n-1}\}$.

Definition 3. *Let $\mathcal{B}$ be a is quasi-Voronoi-reduced basis of Λ. Assume that $\mathcal{B}$ and $\{e_i\}_{i=1}^n$ have the same orientation as in Corollary 1. The basis is called semi-Voronoi-reduced (SVR) if there exists at least two points $x_1, x_2 \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ such that Surf$(\vee_{k=1}^{\ell_1} g_{1,k}) \bigcap$ Surf$(\vee_{k=1}^{\ell_2} g_{2,k}) \neq \varnothing$, where $\ell_1, \ell_2 \geq 1$, $g_{1,k}$ are the facets between x_1 and all points in $\mathcal{T}_f(x_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, and $g_{2,k}$ are the facets between x_2 and all points in $\mathcal{T}_f(x_2) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$.*

The above definition of a SVR basis imposes that the boundaries around two points of $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$, defined by the two convex functions $\vee_{k=1}^{\ell_m} h_{m,k}$, $m = 1, 2$, have a non-empty intersection. Consequently, the min operator $\wedge$ leads to a boundary function as in (33).

Corollary 2. *$\mathcal{P}(\mathcal{B})$ for a SVR basis $\mathcal{B}$ admits a decision boundary defined by a CPWL function as in (33).*

From now on, the default orientation of the basis with respect to the canonical axes of $\mathbb{R}^n$ is assumed to be the one of Corollary 1. We call f the decision boundary function. The domain of f (its input space) is $\mathcal{D}(\mathcal{B}) \subset \mathbb{R}^{n-1}$. The domain $\mathcal{D}(\mathcal{B})$ is the projection of $\mathcal{P}(\mathcal{B})$ on the hyperplane $\{e_i\}_{i=2}^n$. It is a bounded polyhedron that can be partitioned into convex regions which we call linear regions. For any $\tilde{y}$ in one of these regions, f is described by a unique local affine function h_j. The number of those regions is equal to the number of affine pieces of f.

[1]Note that the orientation of the basis does not satisfy the assumptions of Corollary 1.

E. Complexity analysis: the number of affine pieces of the decision boundary

An efficient neural lattice decoder should have a reasonable size, i.e. a reasonable number of neurons. Obviously, the size of the neural network implementing the HLD (such as the one of Figure 8) depends on the number of affine pieces in the decision boundary function. It is thus of high interest to characterize the number of pieces in the decision boundary as a function of the dimension. Unfortunately, it is not possible to treat all lattices in a unique framework. Therefore, we investigate this aspect for some well-known lattices.

The lattice A_n

We count the number of affine pieces of the decision boundary function f obtained for z_1 with the basis defined by the Gram matrix (25).

Theorem 5. *Consider an A_n-lattice basis defined by the Gram matrix* (25)*. Let o_i denote the number of sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$, where $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = i$. The decision boundary function f has a number of affine pieces equal to*

$$\sum_{i=1}^{n} i \cdot o_i, \tag{34}$$

with $o_i = \binom{n-1}{n-i}$.

Proof. For any given point $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$, each element in the set $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ generates a Voronoi facet of the Voronoi region of x. Since any Voronoi region is convex, the $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = i$ facets are convex. Consequently, the set $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ generates a convex part of the decision boundary function with i pieces.

We now count the number of sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ with cardinality i. It is obvious that $\forall x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$: $x + g_1 \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$. We walk in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ and for each of the 2^{n-1} points $x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ we investigate the cardinality of the set $\mathcal{T}_f(x + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. This is achieved via the following property of the basis.

$$\begin{aligned} &\forall x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})},\ x' \in A_n \backslash \{g_j, 0\},\ 2 \le j \le n : \\ &x + g_j \in \mathcal{T}_f(x + g_1),\ x + x' \notin \mathcal{T}_f(x + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}. \end{aligned} \tag{35}$$

Starting from the lattice point 0, the set $\mathcal{T}_f(0 + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is composed of 0 and the $n-1$ other basis vectors. Then, for all g_{j_1}, $2 \le j_1 \le n$, the sets $\mathcal{T}_f(g_{j_1} + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are obtained by adding any the $n-2$ remaining basis vectors to g_{j_1}. Indeed, if we add g_{j_1} to g_{j_1}, the resulting point is outside of $\mathcal{P}(\mathcal{B})$. Hence, the cardinality of these sets is $n-1$ and there are $\binom{n-1}{1}$ ways to choose g_{j_1}: any basis vectors except g_1. Similarly, for $g_{j_1} + g_{j_2}$, $j_1 \neq j_2$, the cardinality of the sets $\mathcal{T}_f(g_{j_1} + g_{j_2} + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is $n-2$ and there are $\binom{n-1}{2}$ ways to choose $g_{j_1} + g_{j_2}$. More generally, there are $\binom{n-1}{k}$ sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ of cardinality $n-k$.

Summing over $k = n - i = 0 \dots n-1$ gives the announced result. □

Theorem 5 implies that the HLD, applied on A_n, induces a neural network (having the form given by (31)) of exponential size. Indeed, remember that the first layer of the neural network implementing the HLD performs projections on the orthogonal vectors to each affine piece.

Nevertheless, one can wonder whether a neural network with a different architecture can compute the decision boundary more efficiently. We first address another category of shallow neural networks: ReLU neural networks with two layers. Deep neural networks shall be discussed later in the paper. Note that in this case we do not consider a single function computed by the neural network, like the HLD, but any function that can be computed by this class of neural network.

Theorem 6. *A ReLU neural network with two layers needs at least*

$$\sum_{i=2}^{n}(i-1)\times\binom{n-1}{n-i} \tag{36}$$

neurons for optimal decoding of the lattice A_n.

The proof is provided in Appendix E. Consequently, this class of neural networks is not efficient. However, we shall see in the sequel that deep neural networks are better suited.

Other dense lattices

Similar proof techniques can be used to compute the number of pieces obtain with some bases of other dense lattices such as $D_n, n \geq 2$, and E_n, $6 \leq n \leq 8$.

Consider the Gram matrix of D_n given by (37). All basis vectors have the same length but we have either $\pi/3$ or $\pi/2$ angles between the basis vectors. This basis is not VR but SVR. It is defined by the following Gram matrix.

$$\Gamma_{D_n} = \begin{pmatrix} 2 & 0 & 1 & ... & 1 \\ 0 & 2 & 1 & ... & 1 \\ 1 & 1 & 2 & ... & 1 \\ . & . & . & ... & . \\ 1 & 1 & 1 & ... & 2 \end{pmatrix}. \tag{37}$$

Theorem 7. *Consider a D_n-lattice basis defined by the Gram matrix* (37)*. Let o_i denote the number of sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$, where:*

- $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = \underbrace{[1+(n-2-i)]}_{(l_i)}$, *and*
- $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = \underbrace{\left[\underbrace{1+2(n-2-i)}_{(1)} + \underbrace{\binom{n-2-i}{2}}_{(2)}\right]}_{(ll_i)}$.

The decision boundary function f has a number of affine pieces equal to

$$\sum_{i=0}^{n-2}((l_i)+(ll_i))\times o_i - 1, \tag{38}$$

with $o_i = \binom{n-2}{i}$.

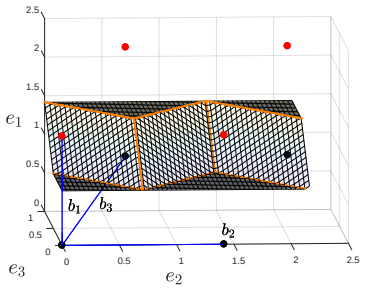


Fig. 11: CPWL boundary function for D_3. The basis is rotated to better illustrate the symmetry: g_1 is colinear with e_1.

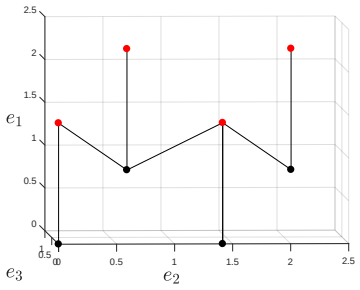


Fig. 12: "Neighbor figure" of $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$ for D_3. Each edge connects a point $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ to an element of $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$.

We presents the two different "neighborhood patterns" encountered with this basis of D_n (this gives (l_i) and (ll_i)). In the proof available in Appendix F, we then count the number of simplices (i.e. (o_i)) in each of these two categories.

The decision boundary function for D_3 is illustrated on Figure 11. We investigate the different "neighborhood patterns" by studying Figure 12: I.e. we are looking for the different ways to find the neighbors of $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ in $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, depending on x. In the sequel, (l_i), (ll_i), and (1), (2) refer to Equation (38) and $\sum_j g_j$ denotes any sum of points in the set $\{0, g_j\}_{j=3}^n$, where g_2 is the basis vector orthogonal to g_1. We recall that adding g_1 to any point $x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ leads to a point in $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$.

(l_i) This pattern is the same as the (only) one encountered for A_n with the basis given by Equation (25). We first consider any point in $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ of the form $\sum_j g_j + g_1$. Its neighbors in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are $\sum_j g_j$ and any $\sum_j g_j + g_i$, where g_i is any basis vector having an angle of $\pi/3$ with g_1 such that $\sum_j g_j + g_i$ is not outside $\mathcal{P}(\mathcal{B})$. Hence, $|\mathcal{T}_f(\sum_{j=1}^i g_j + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = 1 + n - 2 - i$. E.g. for $n = 3$, the closest neighbors of $0 + g_1$ in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are 0 and g_3. g_2 is perpendicular to g_1 and is not a closest neighbor of g_1.

(ll_i) The second pattern is obtained with any point of the form $\sum_j g_j + g_2 + g_1$ and its neighbors in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. $\sum_j g_j + g_2$ and any $\sum_j g_j + g_2 + g_i$, $\sum_j g_j + g_k$ are neighbors of this point in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, where g_i, g_k are any basis vectors having an angle of $\pi/3$ with g_1 such that (respectively) $\sum_j g_j + g_2 + g_i$, $\sum_j g_j + g_k$ are not outside $\mathcal{P}(\mathcal{B})$. This terms generate the (1) in the formula. E.g. for $n = 3$, the closest neighbors of $0 + g_2 + g_1$ in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are g_2, $g_2 + g_3$, and g_3. Moreover, for $n = 3$ one "neighborhood case" is not happening: from $n = 4$, the points $g_i + g_j \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, $3 \leq i < j \leq n$, are also closest neighbors of $g_2 + g_1$. This explains the binomial coefficient (2). Hence, $|\mathcal{T}_f(\sum_{j=1}^i g_j + g_2 + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = 1 + 2(n - 2 - i) + \binom{n-2-i}{2}$.

Finally, we investigate E_n, $6 \leq n \leq 8$. E_8 is one of the most famous and remarkable lattices due to its exceptional density relatively to its dimension (it was recently proved that E_8 is the densest packing of congruent spheres in 8-dimensions [27]). The basis we consider is almost identical to the basis of D_n given by (37), except one main difference: there are two basis vectors orthogonal to g_1 instead of one. This basis is not VR but SVR. It is defined by the following Gram matrix.

$$\Gamma_{E_n} = \begin{pmatrix} 2 & 0 & 0 & 1 & ... & 1 \\ 0 & 2 & 1 & 1 & ... & 1 \\ 0 & 1 & 2 & 1 & ... & 1 \\ 1 & 1 & 1 & 2 & ... & 1 \\ . & . & . & . & ... & . \\ 1 & 1 & 1 & 1 & ... & 2 \end{pmatrix}. \tag{39}$$

Theorem 8. *Consider an E_n-lattice basis, $6 \leq n \leq 8$, defined by the Gram matrix* (37)*. The decision boundary function f has a number of affine pieces equal to*

$$\sum_{i=0}^{n-3} \Big(\underbrace{[1+(n-3-i)]}_{(l_i)} + \underbrace{2\left[1+2(n-3-i)+\binom{n-3-i}{2}\right]}_{(ll_i)} + \underbrace{\left[\underbrace{1+3(n-3-i)}_{(1)} + \underbrace{3\binom{n-3-i}{2}}_{(2)} + \underbrace{\binom{n-3-i}{3}}_{(3)}\right]}_{(lll_i)} \Big) \underbrace{\binom{n-3}{n-i}}_{(o_i)} - 3. \tag{40}$$

We first highlight the similarities with the function of D_n defined by (37). As with D_n, we have case (l_i). Case (ll_i) of D_n is also present but obtained twice because of the two orthogonal vectors. The terms $n-2-i$ in (l_i) and (ll_i) of Equation (38) are replaced by $n-3-i$ also because of the additional orthogonal vector.

Then, there is a new pattern (lll_i): Any point of the form $\sum_j g_j + g_3 + g_2 + g_1$ and its neighbors in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, where $\sum_j g_j$ represents any sum of points in the set $\{0, g_j\}_{j=4}^n$. For instance, the closest neighbors in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ of $g_3+g_2+g_1 \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ are the following points, which we can sort in three groups as on Equation (40): (1) g_2+g_j, g_3+g_j, $g_2+g_3+g_j$, (2) g_j+g_k, $g_2+g_j+g_k$, $g_3+g_j+g_k$, (3) $g_j+g_i+g_k$, $4 \leq i < j < k \leq n$. The formal proof is available in Appendix H.

VI. Complexity reduction

In this section, we first show that a technique called the folding strategy enables to compute the decision boundary function at a reduced (polynomial) complexity. The folding strategy can be seen as a preprocessing step to simplify the function to compute. The implementation of this technique involves a deep neural network. As a result, the exponential complexity of the HLD is reduced to a polynomial complexity by moving from a shallow neural network to a deep neural network. The folding strategy and its implementation is first presented for the lattice A_n. We then show that folding is also possible for D_n and E_n.

In the second part of the section, we argue that, on the Gaussian channel, the problem to be solved by neural networks is easier for MIMO lattices than for dense lattices: In low to moderate dimensions, many pieces of the decision boundary function can be neglected for quasi-optimal decoding. Assuming that usual training techniques

naturally neglect the useless pieces, this explains why neural networks of reasonable size are more efficient with MIMO lattices than with dense lattices.

A. Folding strategy

1) The algorithm: Obviously, at a location $\tilde{y}$, we do not want to compute all affine pieces in (33), whose number is for instance given by (34) for A_n. To reduce the complexity of this evaluation, the idea is to exploit the symmetries of f by "folding" the function and mapping distinct regions of the input domain to the same location. If folding is applied sequentially, i.e. fold a region that has already been folded, the gain becomes exponential. The notion of folding the input space in the context of neural networks was introduced in [25] and [19]. We first present the folding procedure for the lattice A_n and explain how this translates into a deep neural networks. We then show that this strategy can also be applied to the other dense lattices studied in Section V-E.

Folding of A_n

The input space $\mathcal{D}(\mathcal{B})$ is defined as in Section V-D. Given the basis orientation as in Corollary 1, the projection of g_j on $\mathcal{D}(\mathcal{B})$ is g_j itself, for $j \geq 2$. We also denote the bisector hyperplane between two vectors g_j, g_k by $BH(g_j, g_k)$ and its normal vector is taken to be $v_{j,k} = g_j - g_k$. Let $\tilde{y} \in \mathcal{D}(\mathcal{B})$ and let $\tilde{v}_{j,k}$ be a vector with the $n-1$ last coordinates of $v_{j,k}$. First, we define the function $F_{j,k}$, where $2 \leq j < k \leq n$, which performs the following reflection. Compute $\tilde{y} \cdot \tilde{v}_{j,k}$. If the scalar product is non-positive, replace $\tilde{y}$ by its mirror image with respect to $BH(g_j, g_k)$. Since $2 \leq j < k \leq n$, there are $\binom{n-1}{2} = (n-1)(n-2)/2$ functions $F_{j,k}$. The function F_{A_n} performs sequentially these $O(n^2)$ reflections:

$$F_{A_n} = F_{2,2} \circ F_{2,3} \circ F_{3,3} \circ ... \circ F_{n,n}, \tag{41}$$

and

$$F_{A_n} : \mathcal{D}(\mathcal{B}) \to \mathcal{D}(\mathcal{B})'. \tag{42}$$

Theorem 9. *Let us consider the lattice A_n defined by the Gram matrix* (25)*. We have (i) $\mathcal{D}(\mathcal{B})' \subset \mathcal{D}(\mathcal{B})$, (ii) for all $\tilde{y} \in \mathcal{D}(\mathcal{B})$, $f(\tilde{y}) = f(F_{A_n}(\tilde{y}))$ and (iii) f has exactly*

$$2n - 1 \tag{43}$$

pieces on $\mathcal{D}(\mathcal{B})'$.

Equation (43) is to be compared with (34).

Example 2 (Continued)**.** *The function f for A_3 restricted to $\mathcal{D}(\mathcal{B})'$ (i.e. the function to evaluate after folding), say $f_{\mathcal{D}(\mathcal{B})'}$, is*

$$f_{\mathcal{D}(\mathcal{B})'} = \Big[h_{p1} \vee h_1\Big] \wedge \Big[h_{p2} \vee h_2\Big] \wedge \Big[h_{p3}\Big]. \tag{44}$$

The general expression of $f^n_{\mathcal{D}(\mathcal{B})'}$ for any dimension n is

$$f^n_{\mathcal{D}(\mathcal{B})'} = \Big[h_{p_1} \vee h_1\Big] \wedge \Big[h_{p_2} \vee h_2\Big] \wedge \ldots \wedge \Big[h_{p_{n-1}} \vee h_{n-1}\Big] \wedge \Big[h_{p_n}\Big].$$

Proof. To prove (i) we use the fact that $BH(g_j, g_k)$, $2 \leq j < k \leq n$, is orthogonal to $\mathcal{D}(\mathcal{B})$, then the image of $\tilde{y}$ via the folding F is in $\mathcal{D}(\mathcal{B})$.

(ii) is the direct result of the symmetries in the A_n basis where the n vectors have the same length and the angle between any two basis vectors is $\pi/3$. A reflection with respect $BH(g_j, g_k)$ switches g_j and g_k in the hyperplane containing $\mathcal{D}(\mathcal{B})$ and orthogonal to e_1. Switching g_j and g_k does not change the decision boundary because of the basis symmetry, hence f is unchanged.

Now, for (iii), how many pieces are left after all reflections? Similarly to the proof of Theorem 5, we walk in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ and for a given point $x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ we count the number of elements of $\mathcal{T}_f(x + b_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ (via Equation (35)) that are on the proper side of all bisector hyperplanes. Starting with $\mathcal{T}_f(x + b_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, only 0 and g_2 are on the proper side: any other point g_j, $j \geq 3$, is on the other side of the the bisector hyperplanes $BH(g_2, g_j)$. Hence, the lattice point g_1, which had n neighbors in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ before folding, only has 2 now. f has only two pieces around g_1 instead of n. Then, from g_2 one can add g_3 but no other for the same reason. The point $g_2 + g_1$ has only 2 neighbors in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ on the proper side. The pattern replicates until the last corner reaching $g_1 + g_2 + \ldots + g_n$ which has only one neighbor. So we get $2(n-1)+1$ pieces. □

From folding to a deep ReLU neural network

For sake of simplicity and without loss of generality, in addition to the standard ReLU activation function $\text{ReLU}(a) = \max(0, a)$, we also allow the function $\max(0, -a)$ and the identity as activation functions in the neural network.

To implement a reflection $F_{j,k}$, one can use the following strategy.

- Step 1: rotate the axes to have the ith axis e_i perpendicular to the reflection hyperplane and shift the point (i.e. the ith coordinate) to have the reflection hyperplane at the origin.
- Step 2: take the absolute value of the ith coordinate.
- Step 3: do the inverse operation of step 1.

Now consider the ReLU neural network[2] illustrated in Figure 13. The edges between the input layer and the hidden layer (the dashed square) represent the rotation matrix (Step 1), where the ith column is repeated twice, and p is a bias applied on the ith coordinate. Within the dashed square, the absolute value of the ith coordinate is computed and shifted by $-p$. The activation functions in the dashed square, $\max(0, a)$, $\max(0, -a)$, and a, implement the absolute value operation (Step 2). Finally, the edges between the hidden layer and the output layer represent the inverse rotation matrix (Step 3). This ReLU neural network computes a reflection $F_{j,k}$. We call it a reflection block. Note that the width of a reflection block is $O(n)$.

[2]This neural network uses both ReLU and linear activation functions. It can still be considered as a ReLU neural network as a linear activation function can be implemented with ReLU neurons.

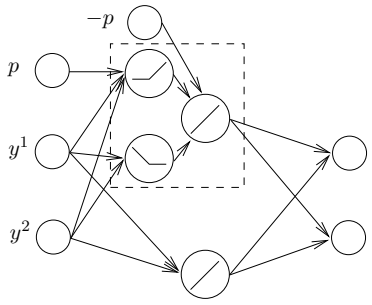


Fig. 13: Reflection ReLU neural network (called reflection block).

The function F_{A_n} can be implemented by a simple concatenation of reflection blocks. This leads to a very deep and narrow neural network of depth $\mathcal{O}(n^2)$ (the number of functions $F_{j,k}$) and width $\mathcal{O}(n)$ (the width of a reflection block is linear in n).

Regarding the $2n-1$ remaining pieces after folding, we have two options (in both cases, the number of operations involved is negligible compared to the previous folding operations). To directly discriminate the point with respect to f, we implement the HLD on these remaining pieces with two additional hidden layers (as in Figure 8): project y_{folded} on the $2n-1$ hyperplanes (see Theorem 9), with one layer of width $2n+1$, and compute the associated Boolean equation with an additional hidden layer. If needed, we can evaluate $f(\tilde{y})$ via $\mathcal{O}(\log(n))$ additional hidden layers. First, compute the $n-1$ 2-$\vee$ via two layers of size $\mathcal{O}(n)$ containing several "max ReLU neural networks" (see e.g. Figure 3 in [2]). Then, compute the n-$\wedge$ via $\mathcal{O}(\log(n))$ layers.

Consequently, f can be computed by a ReLU network of depth $\mathcal{O}(n^2)$ and width $\mathcal{O}(n)$. Note that the induced decoding complexity, while being polynomial, remains higher than state-of-art algebraic decoder for A_n. In [4], Conway and Sloane describe an algorithm which takes $O(n \log n)$ steps.

Folding of other dense lattices

We now present the folding procedure for other lattices.

First, we consider D_n defined by the Gram matrix (37). F_{D_n} is defined as F_{A_n} except that we keep only the $F_{j,k}$ for $j,k \geq 3$. Moreover, the g_i are now the basis vectors of D_n instead of A_n, where g_2 is the basis vector orthogonal to g_1. There are $\binom{n-2}{2} = (n-2)(n-3)/2$ functions $F_{j,k}$ and the function F_{D_n} performs sequentially the $O(n^2)$ reflections.

Theorem 10. *Let us consider the lattice D_n defined by the Gram matrix* (37)*. We have (i) for all $\tilde{y} \in \mathcal{D}(\mathcal{B})$, $f(\tilde{y}) = f(F_{D_n}(\tilde{y}))$ and (ii) f has exactly*

$$6n - 12 \tag{45}$$

pieces on $\mathcal{D}'(\mathcal{B})$.

Equation (45) is to be compared with (38).

Sketch of proof. To count the number of pieces of f, defined on $\mathcal{D}'(\mathcal{B})$, we need to enumerate the cases where both $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ and $x' \in \mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are on the non-negative side of all reflection hyperplanes. Among the points in $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$ only the points

1) $x_1 = g_3 + ... + g_{i-1} + g_i$ and $x_1 + g_1$,

2) $x_2 = g_3 + ... + g_{i-1} + g_i + g_2$ and $x_2 + g_1$,

$i \leq n$, are on the non-negative side of all reflection hyperplanes. It is then easily seen that the number of pieces of f, defined on $\mathcal{D}'(\mathcal{B})$, is given by equation (38) reduced as follows. The three terms $(n-2-i)$ (i.e. $2(n-2-i)$ counts for two), the term $\binom{n-2-i}{2}$, and the term $\binom{n-2}{i}$ become 1 at each step i, for all $0 \leq i \leq n-3$ (except $\binom{n-2-i}{2}$ which is equal to 0 for $i = n-3$). Hence, (38) becomes $(n-3)\times(2+4)+(2+3)+1$, which gives the announced result.

Consequently, f can be computed by a ReLU network of depth $\mathcal{O}(n^2)$ and width $\mathcal{O}(n)$ (i.e. the same size as the one for A_n).

Second, we show how to fold the function for E_n. F_{E_n} is defined as F_{A_n} except that, for the functions $F_{j,k}$, $4 \leq j < k \leq n$ and $j = 2, k = 3$ instead of $2 \leq j < k \leq n$, where g_2, g_3 are the basis vectors orthogonal to g_1. There are $\binom{n-3}{2} + 1 = (n-3)(n-4)/2 + 1$ functions $F_{j,k}$ and the function F_{E_n} performs sequentially the $O(n^2)$ reflections.

Theorem 11. *Let us consider the lattice E_n, $6 \leq n \leq 8$, defined by the Gram matrix* (7)*. We have (i) for all $\tilde{y} \in \mathcal{D}(\mathcal{B})$, $f(\tilde{y}) = f(F_{E_n}(\tilde{y}))$ and (ii) f has exactly*

$$12n - 40 \tag{46}$$

pieces on $\mathcal{D}'(\mathcal{B})$.

Equation (46) is to be compared with (40). Consequently, f can be computed by a ReLU network of depth $\mathcal{O}(n^2)$ and width $\mathcal{O}(n)$.

B. Neglecting many affine pieces in the decision boundary

In the previous section, we showed that complexity reduction can be achieved for some structured lattices by exploiting their symmetries. What about unstructured lattices? We consider the problem of decoding on the Gaussian channel. The goal is to obtain quasi-MLD performance.

1) Empirical observations: In [9], we performed several computer simulations with dense lattices (e.g. E_8) and MIMO lattices (such as the ones considered in [24]), which are typically not dense in low to moderate dimensions. We aimed at minimizing the number of parameters in a standard fully-connected feed-forward sigmoid neural network [12] while maintaining quasi-MLD performance. The training was performed with usual gradient-descent-like techniques [12]. The network considered is shallow, similar to the HLD, as it contains only three hidden layers.

Dimension n	10	12	14	16
Average number of points	59	109	201	361

TABLE I: Average number of points in a sphere of squared radius $2 \cdot d^2(\Lambda)$ centered at the origin for random MIMO lattices Λ .

Let W be the number of parameters in the neural networks (i.e. the number of edges). To be competitive, W should be smaller than 2^n. For E_8 we obtained a complexity ratio $\frac{\log_2 W}{n} = 2.0$ whereas for the MIMO lattice the ratio is $\frac{\log_2 W}{n} = 0.78$.

We also compared the decoding complexity of MIMO lattices and dense lattices (BW_{16} in this case) in [8], with a different network architectures (but still having the form of a feed-forward neural network). The conclusion was the same: While it is possible to get a reasonable complexity for MIMO lattices, it is much more challenging for dense lattices.

2) Explanation: We explained in the first part of this paper that all pieces of the decision boundary function are facets of Voronoi regions. As a result, the (optimal) HLD needs to consider all Voronoi relevant vectors, which is equal to $\tau_f = 2^{n+1} - 2$ for random lattices. However, (14) shows that a term in the union bound decreases exponentially with $\|x\|^2$, which is a standard behavior on the Gaussian channel. Numerical evaluations of a union bound truncated at a squared distance of $2 \cdot d^2(\Lambda)$ (3dB margin in VNR) yield very tight results at moderate and high VNR. Therefore, only the first lattice shells need to be considered for quasi-MLD performance on the Gaussian channel.

Consequently, we performed simulations to know how many Voronoi facets contribute to the 3dB-margin quasi-MLD error probability for random MIMO lattices generated by a matrix G with random i.i.d $\mathcal{N}(0,1)$ components. We numerically generated 200000 random MIMO lattices Λ and computed the average number of lattice points in a sphere of squared radius $2 \cdot d^2(\Lambda)$ centered at the origin. The results are reported in Table I. Figure 14 also provide the distribution for $n = 14$. The random lattices in dimension $n = 14$ are generated by a matrix G with random i.i.d. $\mathcal{N}(0,1)$ components. We numerically generated 200000 random lattices in dimension $n = 14$ to estimate the probability distribution. For comparison, the number of points in such a sphere is 25201 for the dense Coxeter-Todd lattice in dimension 12 and 588481 for the dense Barnes-Wall lattice in dimension 16 [6, Chap. 4]. Note however that while the numbers shown in Table I are relatively low, the increase seems to be exponential: The number of lattice points in the sphere almost doubles when adding two dimensions.

This means that the number of Voronoi facets significantly contributing to the error probability is much smaller for random unstructured MIMO lattices compared to structured lattices in these dimensions. As a result, the number of hyperplanes that should be taken into account for quasi-MLD is much smaller for random unstructured MIMO lattices. In other words, the function to compute for quasi-optimal decoding is "simpler": A piecewise linear boundary with a relatively low amount of affine pieces can achieve quasi-MLD for random MIMO lattices.

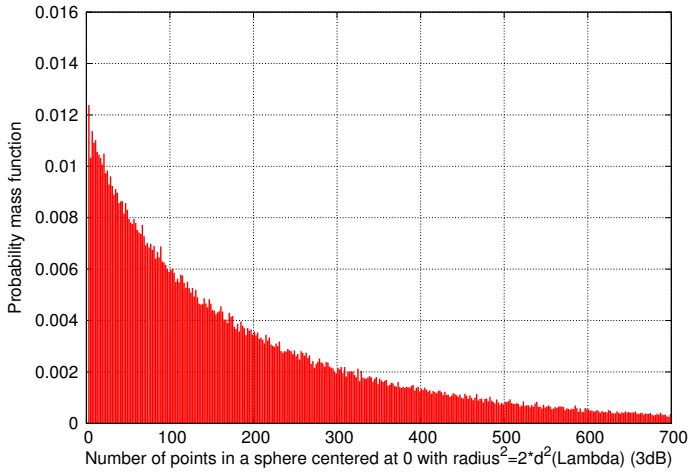


Fig. 14: Distribution of the number of lattice points in a sphere of squared radius $2d^2(\Lambda)$ for $n = 14$.

C. Learning perspective

We argue that regular learning techniques for shallow neural networks, such as gradient-descent, using Gaussian distributed data at moderate SNR for the training, naturally selects the Voronoi facets contributing to the error probability. We estimated in the previous subsection, via computer search, that the number of Voronoi facets from this category is low for unstructured MIMO lattices. This explains why, for quasi-optimal decoding in low to moderate dimensions, shallow neural networks can achieve satisfactory performance at reasonable complexity with unstructured MIMO lattices. However, the number of Voronoi facets to consider is much higher for structured lattices. This elucidates why it is much more challenging to train a shallow neural network with structured lattices.

In the first part of this section, we explained that for this latter category of lattices, such as A_n, one should consider a deep neural network. It is thus legitimate to suppose that training a deep neural network to decode A_n should be successful. However, when this category of neural networks is used, even when we know that their function class contains the target function, the training is much more challenging. In particular, even learning simple one dimensional oscillatory function, such as the triangle wave function illustrated on Figure 15, is very difficult whereas they can be easily computed via folding. This can only be worst for high-dimensional oscillatory functions such as the boundary decision functions.

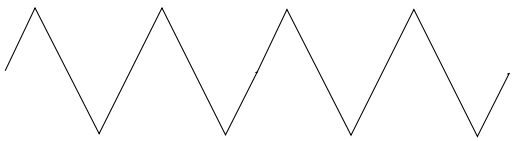


Fig. 15: Simple one-dimensional function which is challenging to learn via usual techniques.

This might explain the success of model-based techniques, where the neural network architectures are established by unfolding known decoding algorithms and where the weights are initialized based on these algorithms [20].

Learning is then used to explore the functions in the function class of the neural network that are not "too far" from the initial point in the optimization space. Nevertheless, the initial point should already be of good quality to get satisfactory performance and learning amounts to fine tuning the algorithm.

VII. Conclusions

The decoding problem has been investigated from a neural network perspective. We discussed what can and cannot be done with feed-forward neural networks in light of the complexity of the decoding problem. We have highlighted that feed-forward neural networks should compute a CPWL boundary function to decode. When the number of pieces in the boundary function is too high, the size of the shallow neural networks becomes prohibitive and deeper neural networks should be considered. For dense structured lattices, this number of pieces is high even in moderate dimensions whereas it remains reasonable in low and moderate dimensions for unstructured random lattices.

Appendix

A. Proof of Equation (15)

$P_e(ub) = \frac{1}{2}\sum_{x\in\Lambda\setminus\{0\}} \exp\left(-\frac{\|x\|^2}{8\sigma^2}\right) = \frac{1}{2}\sum_{x\in\Lambda\setminus\{0\}} \exp\left(-\frac{\Delta}{8\sigma_{max}^2}\cdot\|x\|^2\right)$, where the signal-to-noise ratio, here called VNR, is $\Delta = \sigma_{max}^2/\sigma^2$. After grouping the lattice points shell by shell, with shell of index k located at distance d_k from the origin, we obtain

$$P_e(ub) = f(\Delta) = \sum_{k=1}^{\infty} \tau_k \exp(-\frac{\Delta}{8\sigma_{max}^2}\cdot d_k^2), \tag{47}$$

where $\tau_1 = \tau$ is the kissing number and $d_1 = d(\Lambda) = 2\rho(\Lambda)$ is the lattice minimum distance. It is well-known that the series $f(\Delta)$ converges for $\Delta > 0$, because the Theta series itself converges for $|q| < 1$ and it is holomorphic in z for $q = e^{i\pi z}$ and $\Im z \geq 0$ [6, Chap.2, Sec.2.3]. Another direct method is to upperbound τ_k, for k large, by the number of points on a sphere in $\mathbb{R}^n$ of radius d_k where each point is occupying an area given by a sphere in $\mathbb{R}^{n-1}$ of radius ρ to prove that τ_k is polynomial in d_k. The sequence d_k is unbounded and strictly increasing, hence $f(\Delta)$ converges for $\Delta > 0$. We will be just using the fact that $f(1)$ is finite to prove (15). Indeed, we can write

$$\begin{aligned}
\frac{\sum_{k=2}^{\infty}\tau_k \exp(-\frac{\Delta}{8\sigma_{max}^2}\cdot d_k^2)}{\tau_1\exp(-\frac{\Delta}{8\sigma_{max}^2}\cdot d_1^2)} &= \sum_{k=2}^{\infty}\frac{\tau_k}{\tau_1}\exp\left(-\frac{\Delta}{8\sigma_{max}^2}\cdot(d_k^2-d_1^2)\right)\\
&= \sum_{k=2}^{\infty}\frac{\tau_k}{\tau_1}\left[\exp\left(-\frac{d_k^2-d_1^2}{8\sigma_{max}^2}\right)\right]^1\left[\exp\left(-\frac{d_k^2-d_1^2}{8\sigma_{max}^2}\right)\right]^{\Delta-1}\\
&\leq f(1)\cdot\left[\exp\left(-\frac{d_2^2-d_1^2}{8\sigma_{max}^2}\right)\right]^{\Delta-1},
\end{aligned}$$

where the latest right term vanishes for $\Delta\to\infty$. This proves that $P_e(ub) = f(\Delta) = \tau_1\exp(-\frac{\Delta}{8\sigma_{max}^2}\cdot d_1^2) + o\left(\exp(-\frac{\Delta}{8\sigma_{max}^2}\cdot d_1^2)\right)$ with the Bachmann-Landau small o notation. This is (15) after replacing $-\frac{\Delta}{8\sigma_{max}^2}\cdot d_1^2$ by $-\frac{\pi e\Delta\gamma}{4}$. The interpretation of (15) is that the error-rate performance of a lattice on a Gaussian channel is dominated by the nearest neighbors in the small-noise regime.

B. Proofs of Section IV-B1

1) Proof of Theorem 1: We need to show that none of $y \in \mathcal{V}(x)$, $x \in \Lambda \backslash \mathcal{C}_{\mathcal{P}(\mathcal{B})}$, crosses a facet of $\overline{\mathcal{P}}(\mathcal{B})$. In this scope, we first find the closest point to a facet of $\overline{\mathcal{P}}(\mathcal{B})$ and show that its Voronoi region do not cross $\overline{\mathcal{P}}(\mathcal{B})$. It is sufficient to prove the result for one facet of $\overline{\mathcal{P}}(\mathcal{B})$ as the landscape is the same for all of them.

Let $H_{\mathcal{F}_1}$ denote the hyperplane defined by $\mathcal{B}\backslash g_1$ where the facet $\mathcal{F}_1$ of $\overline{\mathcal{P}}(\mathcal{B})$ lies. While g_1 is in $\overline{\mathcal{P}}(\mathcal{B})$ it is clear that $-g_1$ is not in $\overline{\mathcal{P}}(\mathcal{B})$. Adding to $-g_1$ any linear combination of the $n-1$ vectors generating $\mathcal{F}_1$ is equivalent to moving in a hyperplane, say H_{P_1}, parallel to $\mathcal{F}_1$ and it does not change the distance from $H_{\mathcal{F}_1}$. Additionally, any integer multiplication of $-g_1$ results in a point which is further from the hyperplane (except by ± 1 of course). Note however that the orthogonal projection of $-g_1$ onto $H_{\mathcal{F}_1}$ is not in $\mathcal{F}_1$. The only lattice point in H_{P_1} having this property is obtained by adding all g_j, $2 \leq j \leq n$, to $-g_1$, i.e. it is the point $-g_1 + \sum_{j=2}^{n} g_j$.

This closest point to $\overline{\mathcal{P}}(\mathcal{B})$, along with the points $\mathcal{B}\backslash g_1$, form a simplex. The centroid of this simplex is a hole of the lattice (but it is not a deep hole of A_n for $n \geq 3$). It is located at a distance of $\alpha/(n+1)$, $\alpha > 0$, to the center of any facet of the simplex and thus to $\mathcal{F}_1$ and $\overline{\mathcal{P}}(\mathcal{B})$.

2) Proof of Theorem 2: In this appendix, we prove Lemma 2. One can check that any generator matrix G obtained from the following Gram matrix generates E_8 and satisfies the assumption of Lemma 2. Consequently, it proves Theorem 2.

$$\Gamma_{E_8} = \begin{pmatrix} 4 & 2 & 0 & 2 & 2 & 2 & 2 & 2 \\ 2 & 4 & 2 & 0 & 2 & 2 & 2 & 2 \\ 0 & 2 & 4 & 0 & 2 & 2 & 0 & 0 \\ 2 & 0 & 0 & 4 & 2 & 2 & 0 & 0 \\ 2 & 2 & 2 & 2 & 4 & 2 & 2 & 0 \\ 2 & 2 & 2 & 2 & 2 & 4 & 0 & 2 \\ 2 & 2 & 0 & 0 & 2 & 0 & 4 & 0 \\ 2 & 2 & 0 & 0 & 0 & 2 & 0 & 4 \end{pmatrix}. \tag{48}$$

Lemma 2. *Let G be a generator matrix of E_8, where the set of basis vectors $\mathcal{B}$ are all from the first lattice shell. Let $H_{\mathcal{F}_i}$ denote the hyperplane defined by $\mathcal{B}\backslash g_i$ where the facet $\mathcal{F}_i$ of $\overline{\mathcal{P}}(\mathcal{B})$ lies. Let $\mathring{\mathcal{P}}(\mathcal{B})$ 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 $\mathring{\mathcal{P}}$.*

To prove Lemma 2, we need the next lemma.

Lemma 3. *Let G be a generator matrix of a lattice Λ, where the rows of G form a basis $\mathcal{B}$ of Λ with lattice points from the first shell. Let $H_{\mathcal{F}_i}$ denote the hyperplane defined by $\mathcal{B}\backslash g_i$ where the facet $\mathcal{F}_i$ of $\overline{\mathcal{P}}(\mathcal{B})$ lies. If $(G^{-1})^T$ generates Λ^* with lattice points from the first shell of this dual lattice, then the minimum distance between any $H_{\mathcal{F}_i}$ and a lattice point in $\Lambda \backslash \overline{\mathcal{P}}(\mathcal{B})$ is*

$$d(\Lambda \backslash \overline{\mathcal{P}}(\mathcal{B})), H_{\mathcal{F}_i}) = \frac{d(\Lambda)}{\sqrt{\gamma(\Lambda^*)} \times \sqrt{\gamma(\Lambda)}}. \tag{49}$$

Proof. We derive the minimum distance between a lattice point outside of $\overline{\mathcal{P}}(\mathcal{B})$, $x \in \Lambda \backslash \overline{\mathcal{P}}(\mathcal{B})$, and $H_{\mathcal{F}_i}$. This involves two steps: First, we find one of the closest lattice point by showing that any other lattice point is at the same distance or further and then we compute the distance between this point and $H_{\mathcal{F}_i}$. In the following, u_i is the basis vector of the dual lattice Λ^* orthogonal to $\mathcal{F}_i$ and g_i the only basis vector of Λ where $u_i \cdot g_i \neq 0$, $g_i \in \mathcal{B}$.

As explained in the proof for A_n, while g_i is in $\overline{\mathcal{P}}(\mathcal{B})$ it is clear that $-g_i$ is not in $\overline{\mathcal{P}}(\mathcal{B})$. Adding any linear combination of the $n-1$ vectors generating the facet is equivalent to moving in a hyperplane parallel to $H_{\mathcal{F}_i}$. It does not change the distance from $H_{\mathcal{F}_i}$. Additionally, any integer multiplication of $-g_i$ results in a point which is further from the facet (except by ± 1 of course). Therefore, $-g_i$ is one of the closest lattice points in $\Lambda \backslash \overline{\mathcal{P}}(\mathcal{B})$ from $H_{\mathcal{F}_i}$.

How far is this point from $\overline{\mathcal{P}}(\mathcal{B})$? This distance is obtained by projecting $-g_i$ on u_i, the vector orthogonal to $\mathcal{F}_i$

$$d(\Lambda \backslash \overline{\mathcal{P}}(\mathcal{B}), H_{\mathcal{F}_i}) = \frac{|g_i \cdot u_i|}{||u_i)||}. \tag{50}$$

First, the term $g_i \cdot u_i = 1$ since $G \cdot G^{-1} = I$. Second, from the Hermite constant of the dual lattice Λ^*, and using $\det G \ \cdot \ \det G^{-1} = 1$, we get:

$$d(\Lambda^*) = \frac{\sqrt{\gamma(\Lambda^*)}}{|\det G|^{1/n}}. \tag{51}$$

Since all vectors of Λ^* are from the first shell (i.e. their norm is $d(\Lambda^*)$, assumption of the lemma), (50) becomes

$$d(\Lambda \backslash \mathcal{P}(\mathcal{B}), H_{\mathcal{F}_i}) = \frac{1}{d(\Lambda^*)} = \frac{|\det G|^{1/n}}{\sqrt{\gamma(\Lambda^*)}}. \tag{52}$$

The result follow by expressing $\det G$ as a function of $\gamma(\Lambda)$ and $d(\Lambda)$.

□

We are now ready to prove Lemma 2.

Proof (of Lemma 2). g_i, u_i, and $H_{\mathcal{F}_i}$ are defined as in the previous proof. We apply (49) to E_8. Since this lattice is self-dual, $\gamma(E_8^*) = \gamma(E_8) = 2$ and (49) becomes

$$d(E_8 \backslash \overline{\mathcal{P}}(\mathcal{B}), H_{\mathcal{F}_i}) = \frac{d(E_8)}{2} = \rho(E_8),$$

As a result, the closest lattice point outside of $\overline{\mathcal{P}}(\mathcal{B})$ is at a distance equal to the packing radius. Since the covering radius is larger than the packing radius, the basis is VR only if the Voronoi region of the closest points have a specific orientation relatively to the parallelotope.

The rest of the proof consists in showing that $H_{\mathcal{F}_i}$ is a reflection hyperplane for $-g_i$. Indeed, this would mean that there is a lattice point of E_8 on the other side of $H_{\mathcal{F}_i}$, located at a distance $\rho(E_8)$ from $H_{\mathcal{F}_i}$. It follows that this lattice point is at a distance $d(E_8)$ from $-g_i$ and is one of its closest neighbor. Hence, one of the facet of its Voronoi region lies in the hyperplane perpendicular to the vector joining the points, at a distance $\rho(E_8)$ from the two lattice points. Consequently, this facet and $H_{\mathcal{F}_i}$ lie in the same hyperplane. Finally, the fact that a Voronoi region is a convex set implies that the basis is VR.

To finish the proof, we show that $H_{\mathcal{F}_i}$ is indeed a reflection hyperplane for $-g_i$. The reflection of a point with respect to the hyperplane perpendicular to u_i (i.e. $H_{\mathcal{F}_i}$) is expressed as

$$s_{u_i}(-g_i) = -g_i + 2 \cdot \frac{u_i \cdot g_i}{||u_i||^2} \cdot u_i.$$

We have to show that this point belongs to E_8. The dual of the dual of a lattice is the original lattice. Hence, if the scalar product between $s_{u_i}(-g_i)$ and all the vectors of the basis of E_8^* is an integer, it means that this point belongs to E_8.

$$s_{u_i}(-g_i) \cdot u_j = -g_i \cdot u_j + 2 \cdot \frac{u_i \cdot g_i}{||u_i||^2} \cdot u_i \cdot u_j.$$

We analyse the terms of this equation: $g_i \cdot u_j \in \mathbb{Z}$ since they belong to dual lattices. We already know that $u_i \cdot g_i = 1$. Also $u_i \cdot u_j \in \mathbb{Z}$ as E_8^* is an integral lattice. With Equation (51), we get that $\frac{2}{||u_i||^2} = 1$. We conclude that $s_{u_i}(-g_i) \cdot u_j \in \mathbb{Z}$. □

3) Proof of Theorem 3:

Proof. Λ_{24} is self-dual with $\gamma(\Lambda_{24}) = 2$ and $d(\Lambda_{24}) = 2$. Asumme that we have two generator matrices G and G^{-1} satisfying the assumption of Lemma 3. Equation (49) gives

$$d(\Lambda_{24} \backslash \mathcal{P}(\mathcal{B}), H_{\mathcal{F}_i}) = \frac{d(\Lambda_{24})}{4} = \frac{\rho(\Lambda_{24})}{2}. \tag{53}$$

This distance is clearly smaller than the packing radius of Λ_{24}.

Moreover, Equation (50) shows that if G^{-1} contains a point which is not from the first shell, $\min_i d(\Lambda \backslash \overline{\mathcal{P}}(\mathcal{B}), H_{\mathcal{F}_i})$ becomes smaller has $\max_i ||u_i||$ is greater. Hence, (53) is an upper bound on $d(\Lambda_{24} \backslash \mathcal{P}(\mathcal{B}), H_{\mathcal{F}_i})$. □

C. Proof of Theorem 4

All Voronoi facets of f associated to a same point of $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ form a polytope. The variables within a AND condition of the HLD discriminate a point with respect to the boundary hyperplanes where these facets lie: The condition is true if the point is on the proper side of all these facets. For a given point $y \in \mathcal{P}(\mathcal{B})$, we write a AND condition m as Heav$(yA_m + q_m) > 0$, where $A_m \in \mathbb{R}^{n \times l_m}$, $q_m \in \mathbb{R}^{l_m}$. Does this convex polyhedron lead to a convex CPWL function?

Consider Equation (29). The direction of any v_j is chosen so that the Boolean variable is true for the point in $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ whose Voronoi facet is in the corresponding boundary hyperplane. Obviously, there is a boundary hyperplane, which we name ψ, between the lattice point $0 \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ and $g_1 \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$. This is also true for any $x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ and $x + g_1 \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$. Now, assume that one of the vector v_j has its first coordinate v_j^1 negative. It implies that for a given location $\tilde{y}$, if one increases y_1 the term $y \cdot v_j^T - p_j$ decreases and eventually becomes negative if it was positive. Note that the Voronoi facet corresponding to this v_j is necessarily above ψ, with respect to the first axis e_1, as the Voronoi region is convex. It means that there exists $\tilde{y}$ where one can do as follows. For a given y_1 small enough, y is in the decoding region $z_1 = 0$. If one increases this value, y will cross ψ and be in the decoding region $z_1 = 1$. If one keeps increasing the value of y_1, y eventually crosses the second hyperplane and is back in the region $z_1 = 0$. In this case f has two different values at the location $\tilde{y}$ and it is not a function. If no v_j^1 is negative, this situation is not possible. All v_j^1 are positive if and only if all $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ have their first coordinates x_1 larger than the first coordinates of all $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. Hence, the convex polytope leads to a function if and only if this condition is respected. If this is the case, we can write Heav$(yA_m + q) > 0 \Leftrightarrow \wedge_{k=1}^{l_m} y \cdot a_{m,k} + q_{m,k} > 0$, $a_{m,k}, q_{m,k} \in \{v_j, p_j\}$.

We want $y_1 > h_{m,k}(\tilde{y})$, for all $1 \leq k \leq l_m$, which is achieved if y_1 is greater than the maximum of all values. The maximum value at a location $\tilde{y}$ is the active piece in this convex region and we get $y_1 = \vee_{k=1}^{l_m} h_{m,k}(\tilde{y})$.

A Voronoi facet of a neighboring Voronoi region is concave with the facets of the other Voronoi region it intersects. The region of f formed by Voronoi facets belonging to distinct points in $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ form concave regions that are linked by a OR condition in the HLD. The condition is true if y is in the Voronoi region of at least one point of $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$: $\vee_{m=1}^{M}\{\wedge_{k=1}^{l_m} y \cdot a_{m,k} + q_{m,k}\} > 0$. We get $f(\tilde{y}) = \wedge_{m=1}^{M}\{\vee_{k=1}^{l_m} h_{m,k}(\tilde{y})\}$.

Finally, l_m is strictly inferior to τ_f because all Voronoi facets lying in the affine function of a convex part of f are facets of the same corner point. Regarding the bound on M, the number of logical OR term is upper bounded by half of the number of corner of $\mathcal{P}(\mathcal{B})$ which is equal to 2^{n-1}.

D. First order terms of the decision boundary function before folding for A_n

The equations of the boundary function for A_n are the following.

$$f^{n=2} = \Big[h_{p1} \vee h_1\Big] \wedge \Big[h_{p2}\Big].$$

$$f^{n=3} = \Big[h_{p1} \vee h_1 \vee h_2\Big] \ \wedge \Big[\ (h_{p2} \vee h_1) \wedge (h_{p2} \vee h_2)\ \Big] \ \wedge \Big[h_{p3}\Big].$$

$$\begin{aligned} f^{n=4} = &\Big[h_{p1} \vee h_1 \vee h_2 \vee h_3\Big] \wedge \Big[\ (h_{p2} \vee h_1 \vee h_2) \wedge (h_{p2} \vee h_2 \vee h_3) \wedge (h_{p2} \vee h_1 \vee h_3)\ \Big] \wedge \\ &\Big[\ (h_{p3} \vee h_1) \wedge (h_{p3} \vee h_2) \wedge (h_{p3} \vee h_3)\ \Big] \wedge \Big[h_{p4}\Big]. \end{aligned}$$

E. Proof of Theorem 6

A ReLU neural network with n inputs and W_1 neurons in the hidden layer can compute a CPWL function with at most $\sum_{i=0}^{n} \binom{W_1}{i}$ pieces [21]. This is easily understood by noticing that the non-differentiable part of $\max(0, a)$ is a $n-2$-dimensional hyperplane that separates two linear regions. If one sums W_1 functions $\max(0, d_i \cdot y)$, where d_i, $1 \leq i \leq w_1$, is a random vector, one gets W_1 of such $n-2$-hyperplanes. The result is obtained by counting the number of linear regions that can be generated by these W_1 hyperplanes.

The proof of the theorem consists in finding a lower bound on the number of such $n-2$-hyperplanes (or more accurately the $n-2$-faces located in $n-2$-hyperplanes) partitioning $\mathcal{D}(\mathcal{B})$. This number is a lower-bound on the number of linear regions. Note that these $n-2$-faces are the projections in $\mathcal{D}(\mathcal{B})$ of the $n-2$-dimensional intersections of the affine pieces of f.

We show that many intersections between two affine pieces linked by a $\vee$ operator (i.e. an intersection of affine pieces within a convex region of f) are located in distinct $n-2$-hyperplanes. To prove it, consider all sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ of the form $\{x, x+g_1, x+g_j\}$, $x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, $x+g_j \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. The part of decision boundary function f generated by any of these sets has 2 pieces and their intersection is a $n-2$-hyperplane. Consider the set $\{0, g_1, g_2\}$. Any other set is obtained by a composition of reflections and translations from this set. For two $n-2$-hyperplanes associated to different sets to be the same, the second set should be obtained from the first one by a translation

along a vector orthogonal to the 2-face defined by the points of this first set. However, the allowed translations are only in the direction of a basis vector. None of them is orthogonal to one of of these sets.

Finally, note that any set $\{x \cup (\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})})\}$ where $|\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}| = i$, encountered in the proof of Theorem 5, can be decomposed into $i-1$ of such sets (i.e. of the form $\{x, x-g_1, x-g_1+g_j\}$). Hence, from the proof of Theorem 5, we get that the number of this category of sets, and thus a lower bound on the number of $n-2$-hyperplanes, is $\sum_{k=0}^{n-1}(n-1-k)\binom{n-1}{k}$. Summing over $k = n-i = 0 \dots n-1$ gives the announced result.

F. Proof of Theorem 7

We count the number of sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ with cardinality i. We walk in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ and for each of the 2^{n-1} points $x \in \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ we investigate the cardinality of the set $\mathcal{T}_f(x+\ g_1) \ \cap \ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. In this scope, the points in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ can be sorted into two categories: (l_i) and (ll_i). In the sequel, $\sum_j g_j$ denotes any sum of points in the set $\{0, g_j\}_{j=3}^n$. These two categories and their properties (see also the explanations below Theorem 7), are:

$$\begin{aligned}(l_i)\ \forall\ x = \sum_j g_j \in\ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})},\ x' \in D_n \backslash \{g_k, 0\},\ 3 \le k \le n :\\ x + g_k \in \mathcal{T}_f(x+g_1),\ x + x' \notin \mathcal{T}_f(x+g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}.\end{aligned} \tag{54}$$

$$\begin{aligned}(ll_i)\ \forall\ x = \sum_j g_j + g_2 \in\ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})},\\ x' \in D_n \backslash \{g_i, -g_2+g_i, -g_2+g_i+g_k, 0\},\ 3 \le i < k \le n :\\ (1)\ (a)\ x+g_i \in \mathcal{T}_f(x+g_1),\ (b)\ x-g_2+g_i \in \mathcal{T}_f(x+g_1),\\ (2)\ x-g_2+g_i+g_k \in \mathcal{T}_f(x+g_1),\\ (3)\ x+x' \notin \mathcal{T}_f(x+g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}.\end{aligned} \tag{55}$$

We count the number of sets $\mathcal{T}_f(x)\ \cap\ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ with cardinality i per category.

(l_i) is like A_n. Starting from the lattice point 0, the set $\mathcal{T}_f(0+g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is composed of 0 and the $n-2$ other basis vectors (i.e. without g_2 because it is perpendicular to g_1). Then, for all g_{j_1}, $3 \le j_1 \le n$, the sets $\mathcal{T}_f(g_{j_1}+g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are obtained by adding any of the $n-3$ remaining basis vectors to g_{j_1} (i.e. not g_1, g_2, or g_{j_1}). Indeed, if we add again g_{j_1}, the resulting point is outside $\mathcal{P}(\mathcal{B})$ and should not be considered. Hence, the cardinality of these sets is $n-2$ and there are $\binom{n-2}{1}$ ways to choose g_{j_1}: any basis vectors except g_1 and g_2. Similarly, for $g_{j_1}+g_{j_2}$, $j_1 \neq j_2$, the cardinality of the sets $\mathcal{T}_f(g_{j_1}+g_{j_2}+g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is $n-3$ and there are $\binom{n-2}{2}$ ways to choose $g_{j_1}+g_{j_2}$. More generally, there are $\binom{n-2}{i}$ sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ of cardinality $n-1-i$.

(ll_i) To begin with, we are looking for the neighbors of g_2+g_1. First (i.e. property (1)), we have the following $1+2\times(n-2)$ points in $\mathcal{T}_f(g_2+g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$: g_2, any g_j+g_2, $3 \le j \le n$, and any g_j, $3 \le j \le n$. Second (i.e. property (2)), the $\binom{n-2}{2}$ points g_j+g_k, $3 \le j < k \le n$, are also neighbors of g_2+g_1. Hence, g_2+g_1 has $1+2\times(n-2)+\binom{n-2}{2}$ neighbors in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. Then, the points $g_1+g_2+g_{j_1}$, $3 \le j_1 \le n$, have $1+2\times(n-2-1)+\binom{n-2-1}{2}$ neighbors of this kind, using the same arguments, and there are $\binom{n-2}{1}$ ways to chose g_{j_1}. In general, there are $\binom{n-2}{i}$ sets of cardinality $1+2\times(n-2-i)+\binom{n-2-i}{2}$.

To summarize, each set replicates $\sum_i \binom{n-2}{i}$ times, where for each i we have both (l_i) sets of cardinality $1+(n-2-i)$ and (ll_i) sets of cardinality $1+2\times(n-2-i)+\binom{n-2-i}{2}$. As a result, the total number of pieces of f is obtained as

$$\sum_{i=0}^{n-2}\left(\underbrace{[1+(n-2-i)]}_{(l_i)}+\underbrace{\left[\underbrace{1+2(n-2-i)}_{(1)}+\underbrace{\binom{n-2-i}{2}}_{(2)}\right]}_{(ll_i)}\right)\times\underbrace{\binom{n-2}{i}}_{(o_i)}-1, \tag{56}$$

where the -1 comes from the fact that for $i=n-2$, the piece generated by (l_i) and the piece generated by (ll_i) are the same. Indeed, the bisector hyperplane of x, $x+g_1$ and the bisector hyperplane of $x+g_2$, $x+g_2+g_1$ are the same since g_2 and g_1 are perpendicular.

G. Proof of Theorem 10

Lemma 4. *Among the elements of $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$, only the points of the form*

1) $x_1=g_3+...+g_{i-1}+g_i$ *and* x_1+g_1,
2) $x_2=g_3+...+g_{i-1}+g_i+g_2$ *and* x_2+g_1,

$i\leq n$, are on the non-negative side of all $BH(g_j,g_k)$, $3\leq j<k\leq n$.

Proof. In the sequel, $\sum_i g_i$ denotes any sum of points in the set $\{0,g_i\}_{i=3}^n$. For 1), consider a point of the form $g_3+...+g_{j-1}+g_{j+1}+...+g_{i-1}+g_i$, $j+1<i-1\leq n-1$. This point is on the negative side of all $BH(g_j,g_k)$, $j<k\leq i$. More generally, any point $\sum_i g_i$, where $\sum_i g_i$ includes in the sum g_k but not g_j, $j<k\leq n$, is on the negative side of $BH(g_j,g_k)$. Hence, the only points in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ that are on the non-negative side of all hyperplanes have the form $g_3+...+g_{i-1}+g_i$, $i\leq n$.

Moreover, if $x\in\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is on the negative side of one of the hyperplanes $BH(g_j,g_k)$, $3\leq j<k\leq n$, so is $x+g_1$ since g_1 is in all $BH(g_j,g_k)$.

2) is proved with the same arguments. □

Proof. (of Theorem 10) (i) The folding via $BH(g_j,g_k)$, $3\leq j<k\leq n$, switches g_j and g_k in the hyperplane containing $\mathcal{D}(\mathcal{B})$, which is orthogonal to e_1. Switching g_j and g_k does not change the decision boundary because of the basis symmetry, hence f is unchanged.

Now, for (ii), how many pieces are left after all reflections? To count the number of pieces of f, defined on $\mathcal{D}'(\mathcal{B})$, we need to enumerate the cases where both $x\in\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ and $x'\in\mathcal{T}_f(x)\cap\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are on the non-negative side of all reflection hyperplanes.

Firstly, we investigate the effect of the folding operation on the term $\sum_{i=0}^{n-2}[1+(n-2-i)]\times\binom{n-2}{i}$ in Equation (56). Remember that it is obtained via (l_i) (i.e. Equation (54)). Due to the reflections, among the points in $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ of the form $\sum_j g_j+g_1$ only $x=g_3+g_4+...+g_{i-1}+g_i+g_1$, $j\leq n$, is on the non-negative side of all reflection hyperplanes (see result 1. of Lemma 4). Similarly, among the elements in $\mathcal{T}_f(x)\cap\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$, only $x-g_1$ and $x-g_1+g_{i+1}$ (instead of $x-g_1+g_k$, $3\leq k\leq n$) are on the non-negative side of all reflection hyperplanes. Hence, at each step i, the

term $[1+(n-2-i)]$ becomes 2 (except for $i=n-2$ where it is 1). Therefore, the folding operation reduced the term $\sum_{i=0}^{n-2}[1+(n-2-i)]\times\binom{n-2}{i}$ to $(n-2)\times 2+1$.

Secondly, we investigate the reduction of the term $\sum_{i=0}^{n-2}\left[1+2(n-2-i)+\binom{n-2-i}{2}\right]\times\binom{n-2}{i}$ obtained via (ll_i) (i.e. Equation 55). The following results are obtained via item 2. of Lemma 4. Among the points denoted by $\sum_j g_j+g_2+g_1\in\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ only $x=g_3+g_4+...+g_{i-1}+g_i+g_2+g_1$ is on the proper side of all reflection hyperplanes. Among the neighbors of any of these points, of the form $(ll_i)-(2)$, only $x+g_{i+1}+g_{i+2}$ is on the proper side of all hyperplanes. Additionally, among the neighbors of the form $(ll_i)-(1)$ and $(ll_i)-(b)$, i.e. $x+g_k$ or $x-g_2+g_k$, $3\leq k\leq n$, g_k can only be g_{i+1}. Therefore, the folding operation reduces the term $\sum_{i=0}^{n-2}[1+2(n-2-i)+\binom{n-2-i}{2}]\times\binom{n-2}{i}$ to $(n-3)\times 4+3+1$.

□

H. Proof of Theorem 8

Proof. We count the number of sets $\mathcal{T}_f(x)\cap\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ with cardinality i. We walk in $\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ and for each of the 2^{n-1} points $x\in\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ we investigate the cardinality of the set $\mathcal{T}_f(x+g_1)\cap\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. In this scope, we group the lattice points $x\in\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ in three categories. The numbering of these categories matches the one given in the sketch of proof (see also Equation 61 below). $\sum_j g_j$ denotes any sum of points in the set $\{0,g_j\}_{j=4}^n$.

$$\begin{aligned}(l_i)\ \forall\ x=\sum_j g_j\in\ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})},\ x'\in D_n\backslash\{g_j,0\},\ 4\leq k\leq n:\\ x+g_k\in\mathcal{T}_f(x+g_1),\ x+x'\notin\mathcal{T}_f(x+g_1)\cap\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}.\end{aligned}\tag{57}$$

$$\begin{aligned}(ll_i)-A\ \forall\ x=\sum_j g_j+g_2\in\ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})},\\ x'\in D_n\backslash\{g_i,-g_2+g_i,-g_2+g_i+g_k,0\},\ 4\leq i<k\leq n:\\ (1)\ x+g_i\in\mathcal{T}_f(x+g_1),\ x-g_2+g_i\in\mathcal{T}_f(x+g_1),\\ (2)\ x-g_2+g_i+g_k\in\mathcal{T}_f(x+g_1),\\ (3)\ x+x'\notin\mathcal{T}_f(x+g_1)\cap\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}.\end{aligned}\tag{58}$$

$$\begin{aligned}(ll_i)-B\ \forall\ x=\sum_j g_j+g_3\in\ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})},\\ x'\in D_n\backslash\{g_i,-g_3+g_i,-g_3+g_i+g_k,0\},\ 4\leq i<k\leq n:\\ (1)\ x+g_i\in\mathcal{T}_f(x+g_1),\ x-g_3+g_i\in\mathcal{T}_f(x+g_1),\\ (2)\ x-g_3+g_i+g_k\in\mathcal{T}_f(x+g_1),\\ (3)\ x+x'\notin\mathcal{T}_f(x+g_1)\cap\mathcal{C}^0_{\mathcal{P}(\mathcal{B})}.\end{aligned}\tag{59}$$

$$
\begin{aligned}
&(lll_i)\ \forall\ x = \sum_j g_j + g_2 + g_3 \in\ \mathcal{C}^0_{\mathcal{P}(\mathcal{B})},\\
&\quad x' \in D_n \backslash \{g_i, g_i + g_k, g_i + g_k + g_l, 0\},\ 4 \le i < k < l \le n :\\
&\quad (1)\ x - g_2 + g_k \in \mathcal{T}_f(x + g_1),\ x - g_3 + g_k \in \mathcal{T}_f(x + g_1),\\
&\quad x + g_k \in \mathcal{T}_f(x + g_1),\\
&\quad (2)\ x - g_3 - g_2 + g_i + g_k \in \mathcal{T}_f(x + g_1),\\
&\quad x - g_2 + g_i + g_k \in \mathcal{T}_f(x + g_1),\ x - g_3 + g_i + g_k \in \mathcal{T}_f(x + g_1),\\
&\quad (3)\ x + g_i + g_k + g_l \in \mathcal{T}_f(x + g_1),\\
&\quad (4)\ x + x' \notin \mathcal{T}_f(x + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}.
\end{aligned}
\tag{60}
$$

We count the number of i-simplices per category.

(l_i) is like A_n. Starting from the lattice point 0, the set $\mathcal{T}_f(0 + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is composed of 0 and the $n-3$ other basis vectors (i.e. without g_2 and g_3 because they are perpendicular to g_1). Then, for all g_{j_1}, $4 \le j_1 \le n$, the sets $\mathcal{T}_f(g_{j_1} + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are obtained by adding any of the $n-4$ remaining basis vectors to g_{j_1} (i.e. not g_1, g_2, g_3 or g_{j_1}). Hence, the cardinality of these sets is $n-3$ and there are $\binom{n-3}{1}$ ways to choose g_{j_1}: any basis vectors except g_1, g_2, and g_3. Similarly, for $g_{j_1} + g_{j_2}$, $j_1 \neq j_2$, the cardinality of the sets $\mathcal{T}_f(g_{j_1} + g_{j_2} + g_1) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ is $n-4$ and there are $\binom{n-3}{2}$ ways to choose $g_{j_1} + g_{j_2}$. More generally, there are $\binom{n-3}{i}$ sets $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ of cardinality $n-2-i$.

(ll_i) is like the basis of D_n (see (ll_i) in the proof in Appendix F), repeated twice because we now have two basis vectors orthogonal to g_1 instead of one. Hence, we get that there are $\binom{n-3}{i}$ sets of cardinality $2 \times \left(1 + 2(n-3-i) + \binom{n-3-i}{2}\right)$.

(lll_i) is the new category. We investigate the neighbors of a given point $x = \sum_j g_j + g_3 + g_2 + g_1$. First (1), any $\sum_j g_j + g_3 + g_2$ is in $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. Any $\sum_j g_j + g_2 + g_k$, $\sum_j g_j + g_3 + g_k$, and $\sum_j g_j + g_3 + g_2 + g_k$, where $4 \le k \le n$ and $k \notin \{j\}$ are also in $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. Hence, there are $3 \times (n-3-i)$ of such neighbors, where $i = |\{j\}|$ (in $\sum_j g_j$). Then, (2) any $\sum_j g_j + g_i + g_k$, $\sum_j g_j + g_2 + g_i + g_k$, and $\sum_j g_j + g_3 + g_i + g_k$, where $4 \le i < k \le n$ and $i, k \notin \{j\}$, are in $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. There are $3 \times \binom{n-3-i}{2}$ possibilities, where $i = |\{j\}|$. Finally (3), any $\sum_j g_j + g_i + g_k + g_l$, $4 \le i < k < l \le n$ and $i, k, l \notin \{j\}$ are in $\mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$. There are $\binom{n-3-i}{3}$ of them, where $i = |\{j\}|$.

To summarize, each set replicates $\sum_i \binom{n-3}{i}$ times, where for each i we have (l_i) sets of cardinality $1 + n - 3 - i$, (ll_i) $2 \times \left(1 + 2(n-3-i) + \binom{n-3-i}{2}\right)$, and (lll_i) $1 + 3 \times (n-3-i) + 3 \times \binom{n-3-i}{2} + \binom{n-3-i}{3}$. As a result, the total number of pieces of f is obtained as

$$
\sum_{i=0}^{n-3} \Bigg(\underbrace{[1 + (n-3-i)]}_{(l_i)} + \underbrace{2\left[1 + 2(n-3-i) + \binom{n-3-i}{2}\right]}_{(ll_i)} + \tag{61}
$$

$$
\underbrace{\left[\underbrace{1 + 3(n-3-i)}_{(1)} + \underbrace{3\binom{n-3-i}{2}}_{(2)} + \underbrace{\binom{n-3-i}{3}}_{(3)} \right]}_{(lll_i)} \times \underbrace{\binom{n-3}{n-i}}_{(o_i)} \Bigg) - 3, \tag{62}
$$

where the -3 comes from the fact that for $i = n-3$, the four pieces generated by (l_i), (ll_i), and (lll_i) are the same. Indeed, the bisector hyperplane of x, $x+g_1$, is the same as the one of $x+g_2$, $x+g_2+g_1$, of $x+g_3$, $x+g_3+g_1$, and of $x+g_2+g_3$, $x+g_2+g_3+g_1$, since both g_2 and g_3 are perpendicular to g_1. □

I. Proof of Theorem 11

Lemma 5. *Among the elements of $\mathcal{C}_{\mathcal{P}(\mathcal{B})}$, only the points of the form*

1) $x_1 = g_4 + ... + g_{i-1} + g_i$ *and* $x_1 + g_1$,
2) $x_2 = g_4 + ... + g_{i-1} + g_i + g_2$ *and* $x_2 + g_1$,
3) $x_3 = g_4 + ... + g_{i-1} + g_i + g_2 + g_3$ *and* $x_3 + g_1$,

$i \leq n$, are on the non-negative side of all $BH(g_j, g_k)$, $4 \leq j < k \leq n$.

Proof. See the proof of Lemma 4. □

Proof. (of Theorem 11) (i) The folding via $BH(g_j, g_k)$, $4 \leq j < k \leq n$ and $j = 2, k = 3$, switches g_j and g_k in the hyperplane containing $\mathcal{D}(\mathcal{B})$, which is orthogonal to e_1. Switching g_j and g_k does not change the decision boundary because of the basis symmetry, hence f is unchanged.

Now, for (ii), how many pieces are left after all reflections? To count the number of pieces of f, defined on $\mathcal{D}'(\mathcal{B})$, we need to enumerate the cases where both $x \in \mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$ and $x' \in \mathcal{T}_f(x) \cap \mathcal{C}^0_{\mathcal{P}(\mathcal{B})}$ are on the non-negative side of all reflection hyperplanes.

Firsly, we investigate the effect of the folding operation on the term $\sum_{i=0}^{n-3}[1+n-3-i] \times \binom{n-3}{i}$ in Equation (61). Remember that it is obtained via (l_i) (i.e. Equation (57)). Due to result 1 of Lemma 5 and similarly to the corresponding term in the proof of Theorem 10, this term reduces to $(n-3) \times 2 + 1$.

Secondly, we investigate the reduction of the term $2\left[1 + 2(n-3-i) + \binom{n-3-i}{2}\right] \times \binom{n-3}{i}$, obtained via (ll_i) (i.e. Equation (58)). The following results are obtained via item 2 of Lemma 5. $\binom{n-3}{i}$ reduces to 1 at each step i because in $\mathcal{C}^1_{\mathcal{P}(\mathcal{B})}$, only the points $x = g_2 + g_3 + g_{i-1} + g_i + g_1$ are on the non-negative side of all hyperplanes, $i \leq n$. Then, since any $\sum_j g_j + g_3 + g_1$ is on the negative side of the hyperplane $BH(g_2, g_3)$, $(ll_i) - (B)$ generates no piece in f (defined to $\mathcal{D}'(\mathcal{B})$). $(ll_i) - (A)$ is the same situation as the situation (ll_i) in the proof of Theorem 10. Hence, the term reduces to $(n-3) \times (4) + 3 + 1$.

Finally, what happens to the term $\left[1 + 3(n-3-i) + 3\binom{n-3-i}{2} + \binom{n-3-i}{3}\right]\binom{n-3}{n-i}$, obtained via (lll_i) (i.e. Equation (59))? The following results are obtained via item 3 of Lemma 5. As usual, $\binom{n-3}{n-i}$ reduces to 1 at each step i. Then, $3(n-3-i)$, due to $(lll_i) - (1)$, becomes 2×1 at each step i because any $x - g_2 + g_k$ (in $(lll_i) - (1)$), $k \leq 4 \leq n$, is on the negative side of $BH(g_2, g_3)$. For $x - g_3 + g_k$ and $x + g_k$, only one valid choice of g_k remains at each step i, as explained in the proof of Theorem 10. Regarding the term $3\binom{n-3-i}{2}$, due to $(lll_i) - (2)$, any point $x - g_2 + g_i + g_k$ (in $(lll_i) - (2)$) is on the negative side of $BH(g_2, g_3)$ and at each step i there is only one valid way to chose g_j and g_k for both $x - g_3 - g_2 + g_j + g_k$ and $x - g_3 + g_j + g_k$. Eventually, for the last term due to $(lll_i) - (3)$ only one valid choice remain at each step i. Therefore, the term due to (lll_i) is reduced to to $(n-4) \times 6 + 5 + 3 + 1$. □

REFERENCES

[1] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. on Inf. Theory*, vol. 48, no. 8, pp. 2201-2214, 2002.

[2] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," *International Conference on Learning Representations*, Nov. 2016.

[3] A. Askri and G. R. Othman, "DNN assisted Sphere Decoder," *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1172-1176, Jul. 2019.

[4] J. H. Conway and N. J. A. Sloane, "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes," *IEEE Trans. on Inf. Theory*, vol. 28, no. 2, Mar. 1982.

[5] J. H. Conway and N. J. A. Sloane, "Low-Dimensional Lattices. VI. Voronoi Reduction of Three-Dimensional Lattices," *Proceedings: Mathematical and Physical Sciences by the Royal Society*, vol. 436, no. 1896, pp. 55-68, Jan. 1992.

[6] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*. Springer-Verlag, New York, 3rd ed., 1999.

[7] H. Cohen, *A course in computational algebraic number theory*. Springer-Verlag, New York, 1993.

[8] V. Corlay, J.J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO Detection with Deep Learning," 52^{th} *Asilomar Conference on Signals, Systems and Computers*, May 2018.

[9] V. Corlay, J.J. Boutros, P. Ciblat, and L. Brunel, "Neural Lattice Decoders," *6th IEEE Global Conference on Signal and Information Processing*, Nov. 2018.

[10] 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.

[11] H. Coxeter. *Regular Polytopes*. Dover, New York, 3rd ed., 1973.

[12] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016.

[13] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," *Conference on Information Sciences and Systems*, March 2017.

[14] H. He, C.-K. Wen, S. Jin, G. Ye Li, "Model-Driven Deep Learning for MIMO Detection," *IEEE Trans. on Signal Processing*, vol. 68 , pp. 1702-1715, Feb. 2020.

[15] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563 - 575, Dec. 2017.

[16] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems 25*, pp. 1097-1105, 2012.

[17] D. Micciancio and S. Goldwasser, *Complexity of lattice problems, a cryptographic perspective*. Kluwers Academic Publishers, 2002.

[18] M. Mohammadkarimi, M. Mehrabi, M. Ardakani and Y. Jing, "Deep Learning-Based Sphere Decoding," *IEEE Trans. on Wireless Communications*, vol. 18, no. 9, pp. 4368-4378, Sept. 2019.

[19] G. Montùfar, R. Pascanu, K. Cho, and Y. Bengio, "On the Number of Linear Regions of Deep Neural Networks," *Advances in neural information processing systems*, Dec. 2014.

[20] E. Nachmani, Y. Be'ery and D. Burshtein, "Learning to decode linear codes using deep learning," *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, Illinois, pp. 341-346, Sept. 2016.

[21] R. Pascanu, G. Montufar, and Y. Bengio, "On the number of inference regions of deep feed forward with piece-wise linear activations," arXiv preprint arXiv:1312.6098, 2013.

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

[23] J.G. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill, 5th ed., 2008.

[24] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications*, July 2017.

[25] L. Szymanski and B. McCane, "Learning in deep architectures with folding transformations," *International Joint Conference on Neural Networks* , Aug. 2013.

[26] M. Telgarsky, "Benefits of depth in neural networks," *29th Annual Conference on Learning Theory*, June 2016.

[27] M. Viazovska, "The sphere packing problem in dimension 8", *Annals of Mathematics*, vol. 185, no. 2, pp. 991–1015, 2017.

[28] Doyeon Weon and Kyungchun Lee, "Learning-Aided Deep Path Prediction for Sphere Decoding in Large MIMO Systems", *Access IEEE*, vol. 8, pp. 70870-70877, 2020.

[29] R. Zamir, *Lattice coding for signals and networks*. Cambridge University Press, 2014.