NEURAL LATTICE DECODERS

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

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

ABSTRACT

Lattice decoders constructed with neural networks are presented. Firstly, we show how the fundamental parallelotope is used as a compact set for the approximation by a neural lattice decoder. Secondly, we introduce the notion of Voronoi-reduced lattice basis. As a consequence, a first optimal neural lattice decoder is built from Boolean equations and the facets of the Voronoi region. This decoder needs no learning. Finally, we present two neural decoders with learning. It is shown that L1 regularization and *a priori* information about the lattice structure lead to a simplification of the model.

Index Terms— Closest Vector Problem, Neural Network, Machine Learning, Lattice Reduction.

1. NEURAL DECODING VIA A COMPACT SET

We restrict this paper to lattices in the n-dimensional real space $\mathbb{R}^n$. A lattice Λ is a free $\mathbb{Z}$-module in $\mathbb{R}^n$, or simply a discrete additive subgroup of $\mathbb{R}^n$. To generate such an infinite discrete set with an additive group structure, Λ requires a basis formed by linearly independent vectors. For a rank-n lattice in R^n, the rows of a $n \times n$ generator matrix G constitutes the basis of Λ and any lattice point x is obtained via $x = zG$, where $z \in \mathbb{Z}^n$. The Voronoi region of x is:

$$\mathcal{V}(x) = \{y \in \mathbb{R}^n : \|y - x\| \leq \|y - x'\|, \forall x' \in \Lambda\}. \quad (1)$$

For a given basis $\mathcal{B} = \{g_i\}_{i=1}^n$ forming the rows of G, the fundamental parallelotope of Λ is defined by

$$\mathcal{P}(\mathcal{B}) = \{y \in \mathbb{R}^n : y = \sum_{i=1}^{n} \alpha_i g_i,\ 0 \leq \alpha_i \leq 1\}. \quad (2)$$

The standard definition of $\mathcal{P}$ imposes $\alpha_i < 1$. Here, we deliberately made $\mathcal{P}$ compact in order to simplify our text. The fundamental volume of Λ is $\det(\Lambda) = |\det(G)| = \mathrm{Vol}(\mathcal{V}(x)) = \mathrm{Vol}(\mathcal{P}(\mathcal{B}))$. The first minimum of Λ, i.e. its minimum Euclidean distance, is given by

$$d_{min}(\Lambda) = \min_{x \neq x'} \|x - x'\| = 2\rho, \quad (3)$$

for $x, x' \in \Lambda$ and where ρ is the packing radius of the associated lattice sphere packing. Lattice decoding refers to the method of finding the closest lattice point, the closest in Euclidean distance sense. This problem is also known as the Closest Vector Problem (CVP). Its associated decision problem is NP-complete [12, Chap. 3]. Nevertheless, lattice decoding has been extensively studied in the literature for small and large dimensions. Optimal and quasi-optimal decoders are known for random lattices encountered in communications channels, typically for $n \leq 100$ [18][1][4], and for binary Construction-A lattices [6, Chap. 20]. Sub-optimal message-passing iterative decoders were very successful for non-binary Construction-A lattices in dimensions as high as 1 million [3][8]. The literature also includes extensive work on decoding multi-level coded modulations [20] that give rise to lattice (coset codes) and non-lattice constellations, e.g. see [17][14][16][19] for Leech and lattices based on low-density parity-check codes and polar codes.

• **Relation to Prior Work**. This is the first paper describing how artificial neural networks can be employed to solve the CVP for infinite lattice constellations. We build an optimal neural lattice decoder based on Voronoi-reduced lattice bases. This first neural lattice decoder needs no learning and is tractable in small dimensions. We also show two other types of neural lattice decoders obtained from training of unconstrained and constrained feed-forward networks. As cited above, previous published lattice decoders do not utilize neural networks techniques. However, the current literature on machine learning and deep learning includes interesting results with applications to communication theory and coding theory, e.g. [13][15][11]. These results motivated us to develop neural lattice decoders.

Neural network classifiers are trained to take the best decision about the value of a variable that belongs to a finite set, most frequently the binary set $\mathbb{F}_2$ [9]. Other learning models are trained to produce a good estimate for a real number characterizing one variable, e.g. the probability of a given event involving that variable. In other words, to our modest knowledge, it appears that feed-forward networks do not have the capability of observing the entire space $\mathbb{R}^n$ to infer the value of a variable that belongs to an infinite set such as $\mathbb{Z}$. Indeed, the majority of known lattice decoders search for the closest lattice point $\hat{x} = \hat{z}G$ by looking in $\mathbb{Z}^n$ for the best vector $\hat{z}$,

except for low-density lattices where message passing solves directly the coordinates of $\hat{x}$ [16]. In order to help a neural network solve or approximate the CVP, we force the decoder input to satisfy the assumptions of the Universal Approximation Theorem. This theorem was proved by G. Cybenko for sigmoid networks [7] and then generalized by K. Hornik to multilayer feed-forward architectures [10]. A version of this theorem can be stated as follows [2]:

Theorem. (Anthony & Bartlett 1999). *The two-layer sigmoid networks are "universal approximators", in a sense that, given any continuous function f defined on some compact subset $\mathcal{S}$ of $\mathbb{R}^n$, and any desired accuracy ϵ, there is a two-layer sigmoid network computing a function that is within ϵ of f at each point of $\mathcal{S}$.*

We focus on the fact that f is defined on a compact subset of $\mathbb{R}^n$. There are many ways to partition $\mathbb{R}^n$. Two obvious partitions inspired from the lattice structure are $\mathbb{R}^n = \bigcup_{x\in\Lambda} \mathcal{V}(x)$ and $\mathbb{R}^n = \bigcup_{x\in\Lambda}(\mathcal{P}(\mathcal{B}) + x)$, where one should be careful in assigning the facets to a single Voronoi region or a single translated parallelotope. Note that $\mathcal{V}(x) = \mathcal{V}(0) + x$. The partition based on Voronoi regions cannot be used because, given $y \in \mathbb{R}^n$, solving $\hat{x}$ where $y \in \mathcal{V}(\hat{x})$ is exactly the CVP that we aim to solve. On the other hand, it is easy to determine the translated parallelotope $\mathcal{P}(\mathcal{B}) + x$ to which y belongs. Hence, the lattice decoder input y is translated by $-x$ to let the neural lattice decoder operate in the compact region $\mathcal{P}(\mathcal{B}) + 0$.

2. VORONOI-REDUCED LATTICE BASIS

In the sequel, following the conclusion of the previous section, our neural lattice decoder shall operate as follows within the fundamental parallelotope (Step 2 below):

- Step 0: A noisy lattice point $y_0 = x + \eta$ is observed, where $x \in \Lambda$ and $\eta \in \mathbb{R}^n$ is an additive noise.
- Step 1: Compute $t = \lfloor y_0 G^{-1} \rfloor$ and get $y = y_0 - tG$ which is now inside $\mathcal{P}(\mathcal{B})$. Note: the floor function applied to a vector corresponds to its application on all its coordinates.
- Step 2: The neural lattice decoder finds $\hat{x}$ the closest lattice point to y.
- Step 3: The closest point to y_0 is $\hat{x}_0 = \hat{x} + tG$.

For mod-2 Construction-A lattices, y_0 can be folded inside the cube $[-1, +1]^n$ to decode the component error-correcting code and then find the closest lattice point [6, Chap. 20]. Thus, another option for the compact set to be used by the neural decoder of mod-2 Construction-A lattices is the cube $[-1, +1]^n$. In this paper, we only consider the compact region $\mathcal{P}(\mathcal{B})$ for Step 2 above.

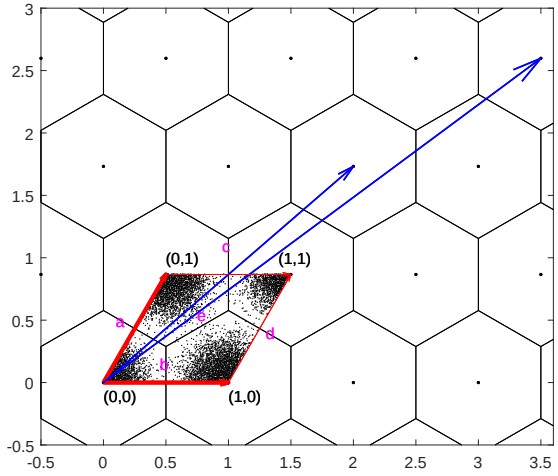

Fig. 1. Voronoi-reduced basis $\mathcal{B}_1$ for A_2 (in red) and a non-reduced basis $\mathcal{B}_2$ (in blue). $\mathcal{P}(\mathcal{B}_1)$ is partitioned into 4 parts included in the Voronoi regions of its corners. $\mathcal{P}(\mathcal{B}_2)$ has 10 parts involving 10 Voronoi regions.

Definition. *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 $\hat{z} \in \{0, 1\}^n$.*

Figure 1 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 $z = (0, 1)$. The basis $\mathcal{B}_1$ is Voronoi-reduced because

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

Lattice basis reduction is an important field in Number Theory. 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 [12][5]. The new Voronoi-reduced type defined in this paper should have interesting connections with the other well known types, one of them is stated in the following lemma. We omit the proof for the sake of space.

Lemma. *Assume that $\mathcal{B}$ is a basis of Λ with all its points located on the first lattice shell, i.e. all vectors in $\mathcal{B}$ have minimal norm. Then $\mathcal{B}$ is Voronoi-reduced.*

The reader may notice that a basis with all its vectors on the first lattice shell is Minkowski-reduced. In general, non-dense lattices do not admit a basis from the first shell. The basis $\mathcal{B}_1$ in Figure 1 is Minkowski-, KZ-, and Voronoi-reduced. The basis $\{v_1, v_1 + v_2\}$ of A_2 is not Minkowski, however it is Voronoi-reduced. Famous densest lattices listed in [6], D_4, E_8, Λ_{16}, and Λ_{24}, all have a Voronoi-reduced basis obtained from the lemma stated above.

3. A HYPERPLANE LOGICAL DECODER

In this section, we introduce a neural lattice decoder to find the closest point for small dimensions without learning. This decoder, referred to as the Hyperplane Logical Decoder (HLD), is Maximum-Likelihood (it exactly solves the CVP) for lattices admitting a Voronoi-reduced basis. It can also be applied to non-dense lattices, i.e. those not admitting a Voronoi-reduced basis, to yield near-Maximum-Likelihood performance in presence of additive white Gaussian noise.

The HLD shall operate in $\mathcal{P} = \mathcal{P}(\mathcal{B})$ as for Step 2 in the decoding steps listed in the previous section. $\mathcal{B}$ is assumed to be Voronoi-reduced. The exact CVP, or Maximum-Likelihood Decoding (MLD), is solved by comparing the position of y to all Voronoi facets partitioning $\mathcal{P}$. 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). Since $\mathcal{V}(x) = \mathcal{V}(0) + x$, orthogonal vectors to all facets partitioning $\mathcal{P}$ are determined from the facets of $\mathcal{V}(0)$, or equivalently these orthogonal vectors are lattice points of the first shell.

Example. *On Figure 1, let $\hat{z} = (\hat{z}_1, \hat{z}_2)$. The first component $\hat{z}_1$ is 1 (true) if y is at the right of hyperplane c, or to the right of b and below e simultaneously. We get the Boolean equation $\hat{z}_1 = c + b \cdot e$, where $+$ is a logical OR and $\cdot$ stands for a logical AND. Similarly, $\hat{z}_2 = d + a \cdot \overline{e}$, where $\overline{e}$ is the Boolean complement of e.*

For $\Lambda \subset \mathbb{R}^n$ of rank n, to find the Boolean equation of a coordinate $\hat{z}_k$, select the 2^{n-1} corners of $\mathcal{P}$ where $z_k = 1$ and perform the following steps:

- For each corner, move by $\rho + \epsilon$ in the direction of a lattice point from the first shell. Three possible situations are encountered. (i) The resulting point is outside $\mathcal{P}$. Hence, there is no decision boundary in this direction. (ii) If not outside $\mathcal{P}$, find the closest lattice point $x' = z'G$ by sphere decoding [18][1]. If $z'_k = 1$ then, again, there exists no decision boundary in this direction. (iii) $z'_k = 0$, a decision boundary orthogonal to this direction does exist.
- The Boolean equation of $\hat{z}_k$ contains a term with a Boolean AND of all decision boundaries found at the same corner. The equation is the Boolean OR of 2^{n-1} terms coming from all selected corners.

For clarity reasons, we omitted technical details in the above steps that involve facets of $\mathcal{P}$ with a tie. In practice, the constructed Boolean equation with its 2^{n-1} terms is significantly reduced into a simpler equation, mainly as a result of identical terms. If Λ does not admit a Voronoi-reduced basis, HLD should be constructed from several lattice shells (given the Theta series) and some coordinates of $\hat{z}$ are not binary anymore.

Once the Boolean equations and the decision boundaries are known, the HLD can be executed in three steps (a)-(c). By abuse of terminology, the inner product of two points in $\mathbb{R}^n$ refers to the inner product between the two vectors defined by these points: (a) Compute the inner product of y with the lattice points orthogonal to the decision boundaries. (b) Apply the Heaviside function on the resulting quantities to get its relative position under the form of Boolean variables. (c) Compute the logical equations associated to each coordinate.

Since (a) and (b) are simply an affine combination between two vectors followed by an activation function, the natural way to represent these steps is to use a perceptron [9], where the edges are labeled with the decision hyperplane parameters, i.e. the perceptron weights define the vector orthogonal to the decision hyperplane. (a) and (b) form the first layer of a neural network. The second layer implements the logical AND and the third layer the logical OR. As a result, the HLD can be thought of as a neural network with two hidden layers. Figure 2 illustrates the topology of the neural network obtained when applying the HLD to the lattice A_2.

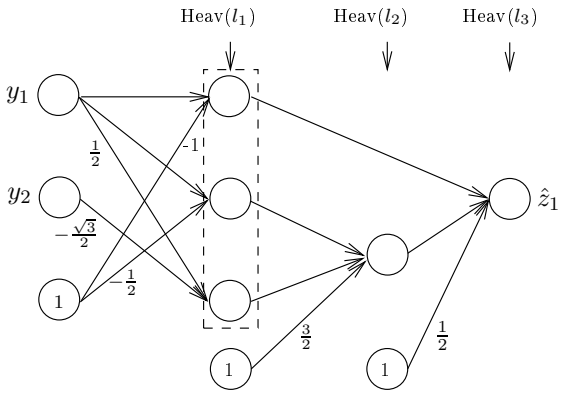

Fig. 2. Neural network performing HLD decoding on the first symbol z_1 of a point in $\mathcal{P}$ for the lattice A_2. Unlabeled edges have weight 1. The bias nodes are required to perform AND and OR. Heav($\cdot$) stands for Heaviside($\cdot$).

Figure 3 shows the point error-rate performance of MLD and HLD for the Schläfli lattice D_4 and the Gosset lattice E_8. All decoders perform exact CVP. However, HLD was running on a distinct machine with a different pseudo-random sequence of lattice points and noise samples. This explains the slight difference between HLD and MLD in Figure 3 due to the Monte Carlo method.

4. LEARNING TO DECODE

We discuss here two neural lattice decoders denoted by NLD2 and NLD3. The first neural lattice decoder (NLD) is the HLD of the previous section. Both models NLD2 and NLD3 need to acquire their weights via learning.

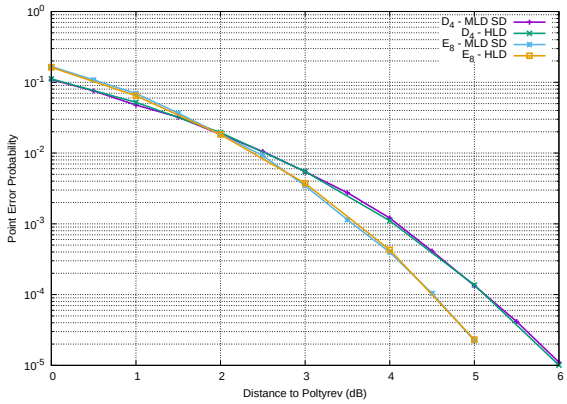

Fig. 3. HLD and MLD for lattices D_4 and E_8.

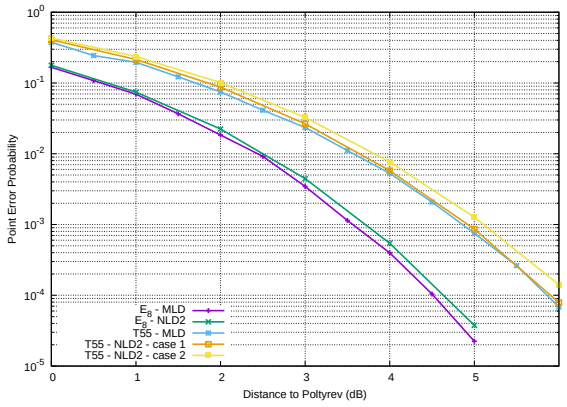

Fig. 4. Decoders with learning, NLD2, without constraint.

NLD2 is a standard fully-connected feed-forward sigmoid network [9] without any constraint on its architecture. The discussion of the sample complexity [2] of NLD2 is omitted due to lack of space. The performance of NLD2 is shown on Figure 4 for E_8 ($n = 8$) and the MIMO lattice $T55$ ($n = 16$) taken from [15]. For E_8, the NLD2 has three hidden layers each with 200 neurons. Its performance is very close to MLD but this model has $W = 83200$ parameters and is too complex relative to HLD for E_8. The ratio $\frac{\log_2(W)}{n} = 2.0$ (supra linear). The NLD2 neural network is not suited to decoding dense lattices. For $T55$, the NLD2 in case 1 has three hidden layers with 50-100-100 neurons respectively. In case 2, it is also made up of three hidden layers with 30-50-50 neurons respectively and $W = 6280$ parameters. The ratio $\frac{\log_2(W)}{n} = 0.78$ (sub-linear). From its complexity and its illustrated performance, we state that NLD2 is a competitive decoding algorithm for non-dense lattices.

In all NLD2 models, the size of first hidden layer is taken to be of the same order of magnitude as the lattice kissing number ($\tau(E_8) = 240$ and $\tau(T55) = 30$).

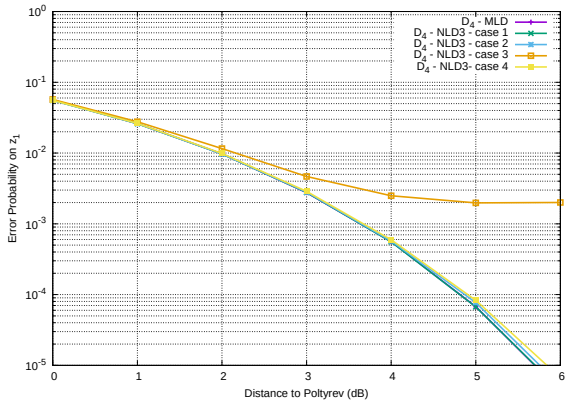

Fig. 5. Neural lattice decoder with learning, NLD3, with constraints and L1 regularization, applied to the lattice D_4.

Now, we introduce NLD3, a learning model with L1 regularization to simplify its structure. NLD3 shall have a structure constraint: its first hidden layer is fixed and taken from the HLD model. Indeed, the authors in [13] used the neural network representation of the Tanner graph for BCH codes to come up with an architecture exploiting *a priori* information on the code structure. The neural network being sub-optimal, they improved its performance via learning. We embrace a similar paradigm: use *a priori* information on the structure of the lattice to build the architecture. Nevertheless, in our case, the HLD neural network is already optimal (it cannot be improved). However, we let an HLD-initialized NLD3 model learn to simplify its structure while limiting the performance degradation.

The performance of NLD3 with D_4 is given in Figure 5. Cases 1 and 3 correspond to NLD3 neural networks with Heaviside activation functions (known as linear threshold networks). The threshold function is replaced by a sigmoid in cases 2 and 4. Of course, learning does not lead to the same model weights in these cases. For the first coordinate, HLD has the following Boolean equation:

$$z_1 = u_1 + u_2 \cdot u_3 \cdot u_4 \cdot u_5 \cdot u_6 + u_4 \cdot u_7 \cdot u_8 + u_4 \cdot u_7 \cdot u_9 + \cdot u_4 \cdot u_{10}.$$

The NLD3 model in case 1 simplifies the Boolean equation of z_1 to two terms only, the second hidden layer shrinked from five to two neurons. Its performance is still Maximum-Likelihood like the HLD. Case 2 is quasi-optimal, the slight loss is due to an imperfect training. For cases 3 and 4, in order to further simplify the model structure, three edges were pruned between the first and the second hidden layers. Case 3 generates an error-floor while case 4 exhibits a great robustness. Similar behavior of NLD3 was observed when utilized on other lattices.

5. 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, Aug. 2002.

[2] M. Anthony, P. Bartlett. *Neural Network Learning: Theoretical Fundations*. Cambridge University Press, 2009.

[3] J.J. Boutros, N. di Pietro, and Y.-C. Huang, "Spectral thinning in GLD lattices," in *Proc. ITA Workshop*, La Jolla (CA), USA, pp. 1-9, Feb. 2015.

[4] L. Brunel and J. J. Boutros, "Lattice decoding for joint detection in direct-sequence CDMA systems," *IEEE Trans. on Inf. Theory*, vol. 49, no. 4, pp. 1030-1037, April 2003.

[5] H. Cohen, *A course in computational algebraic number theory*. Berlin, Heidelberg, Germany: Springer-Verlag, 3rd ed., 1996.

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

[7] G. Cybenko, "Approximations by superpositions of sigmoidal functions," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303-314, Dec 1989.

[8] N. di Pietro and J.J. Boutros, "Leech constellations of Construction-A lattices," *IEEE Trans. on Communications*, vol. 65, no. 11, pp. 4622-4631, Nov. 2017.

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

[10] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, vol. 4, pp. 251257, 1991.

[11] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, P. Viswanath, "Communication Algorithms via Deep Learning," arXiv preprint arXiv:1805.09317, May 2018.

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

[13] 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.

[14] M.-R. Sadeghi, A. H. Banihashemi, and D. Panario, "Low-density parity-check lattices: construction and decoding analysis," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4481-4495, Oct. 2006.

[15] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," arXiv preprint arXiv:1706.01151, June 2017.

[16] N. Sommer, M. Feder, and O. Shalvi, "Low-Density Lattice Codes," *IEEE Trans. on Inf. Theory*, vol. 54, no. 4, pp 1561-1585, April 2008.

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

[18] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. on Inf. theory*, vol. 45, no. 5, pp. 1639-1642, July 1999.

[19] Y. Yan, L. Liu, C. Ling, and X. Wu, "Construction of capacity-achieving lattice codes: polar lattices," Nov. 2014. Available: http://arxiv.org/abs/1411.0187.

[20] U. Wachsmann, R.F.H. Fischer, and J.B. Huber, "Multilevel codes: theoretical concepts and practical design rules," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1361-1391, July 1999.