

Optimization and Identification of Lattice Quantizers

Downloaded from: https://research.chalmers.se, 2025-06-01 02:11 UTC

Citation for the original published paper (version of record):

Agrell, E., Pook-Kolb, D., Allen, B. (2025). Optimization and Identification of Lattice Quantizers. IEEE Transactions on Information Theory, In Press. http://dx.doi.org/10.1109/TIT.2025.3565218

N.B. When citing this work, cite the original published paper.

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

REVISION R1, JANUARY 25, 2025

Optimization and Identification of Lattice Quantizers

Erik Agrell, Fellow, IEEE, Daniel Pook-Kolb, and Bruce Allen, Member, IEEE

Abstract

Lattices with minimal normalized second moments are designed using a new numerical optimization algorithm. Starting from a random lower-triangular generator matrix and applying stochastic gradient descent, all elements are updated towards the negative gradient, which makes it the most efficient algorithm proposed so far for this purpose. A graphical illustration of the theta series, called theta image, is introduced and shown to be a powerful tool for converting numerical lattice representations into their underlying exact forms. As a proof of concept, optimized lattices are designed in dimensions up to 16. In all dimensions, the algorithm converges to either the previously best known lattice or a better one. The dual of the 15-dimensional laminated lattice is conjectured to be optimal in its dimension and its exact normalized second moment is computed.

Index Terms

Algorithm, laminated lattice, lattice design, lattice quantization, mean square error, moment of inertia, normalized second moment, numerical optimization, quantization constant, stochastic gradient descent, theta image, theta series, vector quantization, Voronoi region.

I. INTRODUCTION

Aclassical problem in geometry is how to construct a structure of infinitely many points in *n*-dimensional space so that the mean square distance between a random real vector and its closest member of the point structure is minimal, for a fixed density of points per unit volume. Apart from its fundamental theoretical value, the problem has important practical applications in, e.g., digital communications (both source [1] and channel coding [2, p. 70], [3, Ch. 9]), pattern recognition [4], cryptography [5], hashing [6], machine learning [7], and data analysis [8].

The construction of structures with small mean square distance is traditionally called the *quantizer problem* and the figure of merit is the (possibly normalized) *second moment*. The problem was first formulated in 1959 by Fejes Tóth, who also solved it for the two-dimensional case [9]. The optimal structure in two dimensions is, not surprisingly, the hexagonal lattice. Gersho in 1979 suggested lattice structures in three and four dimensions

The work of E. Agrell was supported by a Collaborating Scientist Grant from the Max Planck Institute for Gravitational Physics, Germany, which is gratefully acknowledged.

E. Agrell is with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: agrell@chalmers.se).

D. Pook-Kolb and B. Allen are with the Max Planck Institute for Gravitational Physics, 30167 Hannover, Germany, and Leibniz Universität Hannover (e-mail: daniel.pook.kolb@aei.mpg.de and bruce.allen@aei.mpg.de).

2

[10], which still today offer the smallest known normalized second moments (NSM) known in their respective dimensions. The proposed three-dimensional structure, which is the body-centered cubic lattice, was proved to be optimal among three-dimensional lattices by Barnes and Sloane [11]. Lattices with small NSM are called *lattice quantizers*.

In 1982, Conway and Sloane calculated the NSMs of the most common lattice families and tabulated the best known lattice quantizers in dimensions up to 10 [12]. They further conjectured that the best lattice quantizer in any dimension is the dual of the lattice that solves the more well-studied *packing problem*. Two years later, the same authors found improved lattices in dimensions 6 and 7 and numerically computed the NSMs of known lattices in dimensions 12, 16, and 24, which all supported their duality conjecture [13]. The pioneering lattice research by Conway and Sloane is summarized in their now classic textbook [2].

Improved lattice quantizers in dimensions 9 and 10 (as well as improved nonlattice structures in dimensions 7 and 9) were found by Agrell and Eriksson in 1998 using a numerical optimization technique [14]. These lattices are not duals of the densest lattice packings known, and thus indicate that Conway and Sloane's conjecture might not be true. Dutour Sikirić *et al.* found an improved lattice in 11 dimensions and evaluated its NSM exactly, thus providing further evidence against the conjecture [15]. Allen and Agrell described exactly the conjectured optimal 9-dimensional lattice quantizer, which was previously only approximated numerically [16]. Recently, some improved lattices in dimensions 12 and above have been reported [17]–[20].

While every lattice for which the exact NSM has been computed serves to constructively prove the achievability of that particular NSM value, there is no known "converse theorem" in the sense of proving that no better lattice exists. Lower bounds on the NSM are known or conjectured [21], but they are not tight in any dimension above 2.

In this paper, we take a fresh look at numerical optimization of lattice quantizers, which as far as we know has not seen any progress since 1998 [14]. A more efficient algorithm for the purpose is proposed. Like the previous algorithm, the new one employs stochastic gradient descent (SGD), which is a standard technique in numerical optimization. The main difference lies in how the gradient is computed. In the previous algorithm, one degree of freedom is reserved to maintain a fixed number of lattice points per unit volume, while the other degrees of freedom are updated guided by the gradient of the *unnormalized* second moment. Thereby the NSM decreases, but unfortunately not in the direction of its gradient. This shortcoming is resolved in the new algorithm, which makes it more efficient.

We also address the question of identifying an exact form for a lattice that is only known numerically. To this end, we introduce a new graphical tool, called *theta image*, and use it to derive a linear system of equations whose solution gives the desired exact form in many cases. Even in cases where the method does not yield an exact form, most of the unknown degrees of freedom are identified as functions of a small number of unknown parameters.

As a proof of concept, lattices with conjectured minimal NSMs are constructed in dimensions up to 16, and exact forms are identified for all of them. The results revisit previously best known lattice quantizers in most considered dimensions and reveals a new best lattice quantizer in 15 dimensions.

The paper is organized as follows. Sec. II compactly summarizes relevant lattice definitions and theory. The new algorithm is theoretically developed in Sec. III. Sec. IV provides pseudocode for the algorithm along with some

REVISION R1, JANUARY 25, 2025

implementational considerations. This section is mainly aimed at programmers and more or less stand-alone from the theoretical foundation in Sec. III. Other possible algorithms for numerical optimization of lattice quantizers are reviewed in Sec. V, where also the improvements over [14] are discussed. In Sec. VI, we discuss how the exact form of a lattice that is only known approximatively can be obtained using the theta series. Sec. VII, finally, demonstrates the potential of the new framework by constructing conjectured optimal lattice quantizers.

II. LATTICES AND THEIR SECOND MOMENTS

Notation: The norm (Euclidean length) of a vector x is denoted by ||x||. The $n \times n$ identity matrix is denoted by I_n . The *i*th element of a vector x is denoted by $[x]_i$, while the element in row *i*, column *j* of a matrix X is denoted by $[X]_{i,j}$. All vectors are row vectors.

Given a set of *n* linearly independent *basis vectors* in \mathbb{R}^m , a *lattice* \mathcal{L} is the set of all linear combinations of these vectors with integer coefficients. The basis vectors are collected as rows in an $n \times m$ generator matrix B, where the condition of linearly independent rows implies $m \ge n$. Expressed in terms of the generator matrix, the lattice \mathcal{L} consists of the points uB for all $u \in \mathbb{Z}^n$. By definition, the all-zero vector **0** belongs to any lattice. The inner products of all basis vectors with each other are collected in the symmetric, positive definite *Gram matrix* $A = BB^{T}$.

The basis vectors span an *n*-dimensional Euclidean space, which is a linear subspace of \mathbb{R}^m . The set of points in this subspace that are closer to **0** than to any other point in \mathcal{L} is the *Voronoi region* Ω of the lattice. The Voronoi region Ω is a *fundamental cell* of \mathcal{L} , which means that infinitely many copies of Ω , each translated by a vector in \mathcal{L} , fills the *n*-dimensional space in which the lattice resides without gaps or overlaps, except on the boundaries. Another fundamental cell of \mathcal{L} is the parallelotope $[0, 1)^n \mathbf{B}$.

The standard figures-of-merit for lattices can all be expressed in terms of properties of Ω . In this paper, we focus on the NSM [10], [2, pp. 34, 56–62]

$$G = \frac{1}{nV^{1+2/n}} \int_{\Omega} \|\boldsymbol{x}\|^2 \,\mathrm{d}\boldsymbol{x},\tag{1}$$

where $V = (\det A)^{1/2}$ is the *n*-volume of Ω . The normalization coefficient $1/V^{1+2/n}$ ensures that G is dimensionless and hence invariant to lattice scaling, while the coefficient 1/n ensures that lattices constructed as the product of identical lower-dimensional lattices have the same NSM as these constituent lattices [18, Sec. IV]. For example, the cubic lattice \mathbb{Z}^n has G = 1/12 for any *n*.

Two lattices are *equivalent* in Euclidean space if their Voronoi regions are congruent, and two generator matrices are equivalent if they generate equivalent lattices. Equivalent lattices have the same NSM. The packing density, covering density, kissing number, and all other metrics invariant to scaling and orthogonal transformation are also the same. Mathematically, two generator matrices B and B' of sizes $n \times m_1$ and $n \times m_2$, respectively, where $m_1 \le m_2$, are equivalent if and only if [2, p. 10]

$$B' = c UBR, (2)$$

where c is a real scalar, U is an $n \times n$ integer matrix with determinant ± 1 , and R is a real $m_1 \times m_2$ matrix with orthonormal rows, i.e., $\mathbf{RR}^{\mathrm{T}} = \mathbf{I}_{m_1}$. Left-multiplication by U corresponds to a basis change and right-

4

multiplication by R to an orthogonal transformation, i.e., rotation and/or reflection. The corresponding relation between Gram matrices is

$$A' = c^2 U A U^{\mathrm{T}}.$$
(3)

For every generator matrix B' of size $n \times m_2$, there exists a square generator matrix B that satisfies (2) with $m_1 = n$. The analysis of lattices in Euclidean space can therefore without loss of generality be confined to square generator matrices, which is exploited in this work.

Another important characteristic of a lattice \mathcal{L} is the *theta series* [2, pp. 44–47]

$$\theta(q) = \sum_{\boldsymbol{x} \in \mathcal{L}} q^{\|\boldsymbol{x}\|^2} = 1 + \tau q^{4\rho^2} + \cdots$$
(4)

where q is a dummy variable. It characterizes how the lattice points are distributed on spherical *shells* around the origin. The first term arises from the lattice vector **0**. The second term contains information about both the *kissing* number τ , i.e., the number of shortest nonzero lattice vectors, and the *packing radius* ρ , i.e., half the minimal distance between two lattice vectors. Equivalent lattices have the same theta series.

III. STOCHASTIC GRADIENT DESCENT

The starting point for minimization by stochastic gradient descent (SGD) is to write the objective function as the expected value of a real function of (i) a set of random variables with a given distribution and (ii) a set of deterministic variables, whose values are to be optimized. The gradient of this real function with respect to the deterministic variables is calculated, or if the exact gradient is not known, the gradient of a Monte-Carlo estimate of the objective function serves as a surrogate. The distribution of the random variables does not have to be known analytically, as long as an algorithmic procedure is available to generate random samples. Independent samples are generated sequentially using this procedure, and for every sample, the deterministic variables are shifted a small step in the direction of the negative gradient. In this section, theory is developed for applying SGD to NSM minimization. This leads to our lattice construction algorithm, whose implementation is described in the next section.

A. Second Moment Estimation

An elegant method to generate random vectors uniformly in the Voronoi region Ω of a given lattice \mathcal{L} was proposed in [13] for the purpose of NSM estimation. Let z be a random vector drawn uniformly from the unit *n*-cube $[0,1)^n$ and let, for a given generator matrix B,

$$\hat{\boldsymbol{u}} = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathbb{Z}^n} \| (\boldsymbol{z} - \boldsymbol{u}) \boldsymbol{B} \|^2.$$
(5)

Now $\hat{\boldsymbol{u}}\boldsymbol{B}$ is the lattice point closest to $\boldsymbol{z}\boldsymbol{B}$ (which is normally not a lattice point). Since the Voronoi region Ω and the parallelotope $[0,1)^n\boldsymbol{B}$ are both fundamental cells of \mathcal{L} , $\boldsymbol{e} = (\boldsymbol{z} - \hat{\boldsymbol{u}})\boldsymbol{B}$ is uniformly distributed in Ω . The generation of uniform random vectors in Ω is discussed in detail in [3, Sec. 4.2].

To calculate (5) requires solving the *closest point problem* for a given lattice. Algorithms for this purpose are available for classical, well-structured lattices [13], [22] as well as arbitrary lattices [23], [24].

REVISION R1, JANUARY 25, 2025

5

Using these definitions of z, \hat{u} , and e, the NSM in (1) can be written as

$$G = \mathbb{E}_{\boldsymbol{z}}[g(\boldsymbol{B}, \boldsymbol{z})],\tag{6}$$

where

$$g(\boldsymbol{B}, \boldsymbol{z}) = \frac{1}{n} V^{-2/n} \|\boldsymbol{e}\|^2.$$
 (7)

Here V is a function of B and e is a function of both B and z. Note that even if (5) has multiple solutions, g(B, z) is still well-defined, because $||e||^2$ has the same value for all solutions \hat{u} . These multiple solutions occur iff zB lies on the boundary of Ω (or its images under translation by the lattice).

If z_1, \ldots, z_T denote T independent realizations of z, then an unbiased estimate of G follows immediately from (6) as

$$\hat{G} = \frac{1}{T} \sum_{t=1}^{T} g(\boldsymbol{B}, \boldsymbol{z}_t).$$
(8)

To quantify the estimation accuracy, we use an unbiased estimator of the variance of \hat{G} [25, Sec. IV]

$$\hat{\sigma}^2 = \frac{1}{T-1} \left(\frac{1}{T} \sum_{t=1}^T g^2(\boldsymbol{B}, \boldsymbol{z}_t) - \hat{G}^2 \right) \,, \tag{9}$$

which is much more accurate than the "jackknife" estimator recommended in earlier literature.

It is easily verified that (8) remains unchanged if the lattice, represented by B, is rescaled. However, previous descriptions of the same NSM estimation method are valid only for lattices with V = 1. This is because of an unfortunate error in the original publication [13], where the right-hand sides of [13, Eqs. (2), (4)] are missing a factor corresponding to the volume of the Voronoi region (here denoted by V). This error appears to have propagated to [17, Eqs. (73)–(74)] and [25, Eqs. (12)–(15)].

B. Iterative basis vector update

If the gradient of the NSM $\nabla_{B}G$ were known analytically for arbitrary B, then this gradient could be used to minimize G. In a few special cases where we know G exactly, it is an infinitely differentiable (C^{∞}) function of certain elements of B in a certain domain, or *phase*, within which the overall topology of Ω (by which we mean the number of faces in any dimension and their pattern of intersection) is fixed. Furthermore, it is many times differentiable (approximately n times) on the boundaries between such domains, called *phase transitions* [16], [20]. If these examples are indicative of how G behaves in general, then the function $G(B): \mathbb{R}^{n \times m} \to \mathbb{R}$ is C^{∞} except on these phase transitions. Thus, generic values of B have open neighborhoods in which G is smooth, so gradient descent could be effective. We are however not aware of any general expression for $\nabla_B G$.

In the absence of the exact $\nabla_{B}G$, the NSM estimate \hat{G} in (8) can be straightforwardly differentiated and is well suited for minimization by SGD. Specifically, we generate a sequence of random vectors z, and for each z, we calculate \hat{u} and e. Then the generator matrix B is updated a small step in the direction that decreases g(B, z) most, which is the negative gradient with respect to the elements of B.

By a suitable choice of coordinate system, every lattice can be represented by a *square, lower-triangular* generator matrix with *positive diagonal elements*. Therefore, without loss of generality, our lattice construction method

6

considers only such generator matrices. From now on and until the end of Sec. IV, the generator matrix B has size $n \times n$ and its elements satisfy $[B]_{i,j} = 0$ for i < j and $[B]_{i,j} > 0$ for i = j. Apart from reducing the number of degrees of freedom from nm to n(n + 1)/2, these conditions greatly facilitate the calculation of the volume V as a function of B, which is now simply

$$V = \prod_{k=1}^{n} [B]_{k,k}.$$
 (10)

Furthermore, the closest-point algorithms we use to compute e from z need a triangular matrix B [23], [24].

We are now ready to calculate the gradient of g(B, z) in (7), which consists of the partial derivatives

$$\frac{\partial g(\boldsymbol{B}, \boldsymbol{z})}{\partial [\boldsymbol{B}]_{i,j}} = \frac{1}{n} V^{-2/n} \frac{\partial \|\boldsymbol{e}\|^2}{\partial [\boldsymbol{B}]_{i,j}} + \frac{1}{n} \|\boldsymbol{e}\|^2 \frac{\partial V^{-2/n}}{\partial [\boldsymbol{B}]_{i,j}}.$$
(11)

Note that for the multiple solution case described after (7), the first partial derivative on the right-hand side of (11) is undefined. For generic B, when zB crosses the boundary of Ω , the derivative steps abruptly from a well-defined value on one side of the boundary to a different well-defined value on the other side. For a given B, the set of z for which this occurs is a set of measure zero. Thus, from here forward, we assume that z denotes one of the T values $z_1, \ldots z_T$, which do not lie in this set of measure zero.

We define $y = z - \hat{u}$, so that e = yB. For a given z and infinitesimal changes of B, \hat{u} and y are unchanged. Hence $\partial y / \partial [B]_{i,j} = -\partial \hat{u} / \partial [B]_{i,j} = 0$ for all i and j. Therefore,

$$\frac{\partial \|\boldsymbol{e}\|^{2}}{\partial [\boldsymbol{B}]_{i,j}} = 2 \frac{\partial \boldsymbol{e}}{\partial [\boldsymbol{B}]_{i,j}} \boldsymbol{e}^{\mathrm{T}}
= 2[\boldsymbol{y}]_{i}[\boldsymbol{e}]_{j}, \quad \text{if } i \geq j,$$

$$\frac{\partial V^{-2/n}}{\partial [\boldsymbol{B}]_{i,j}} = \frac{\partial}{\partial [\boldsymbol{B}]_{i,j}} \prod_{k=1}^{n} [\boldsymbol{B}]_{k,k}
= 0, \quad \text{if } i > j.$$
(12)

For the diagonal elements,

$$\frac{\partial V^{-2/n}}{\partial [\boldsymbol{B}]_{i,i}} = \frac{\partial}{\partial [\boldsymbol{B}]_{i,i}} \prod_{k=1}^{n} [\boldsymbol{B}]_{k,k}^{-2/n}
= -\frac{2}{n} [\boldsymbol{B}]_{i,i}^{-1-2/n} \prod_{k\neq i} [\boldsymbol{B}]_{k,k}^{-2/n}
= -\frac{2V^{-2/n}}{n[\boldsymbol{B}]_{i,i}}.$$
(14)

Combining (11), (12), (13), and (14) yields

$$\frac{\partial g(\boldsymbol{B}, \boldsymbol{z})}{\partial [\boldsymbol{B}]_{i,j}} = \begin{cases} \frac{2}{n} V^{-2/n} [\boldsymbol{y}]_i [\boldsymbol{e}]_j, & i > j, \\ \frac{2}{n} V^{-2/n} \left([\boldsymbol{y}]_i [\boldsymbol{e}]_i - \frac{\|\boldsymbol{e}\|^2}{n[\boldsymbol{B}]_{i,i}} \right), & i = j. \end{cases}$$
(15)

Now introducing explicit notation for the steps in the SGD algorithm, we denote the T random realizations of z with z_t for t = 0, ..., T - 1, with t being referred to as "time." In each time instant t, the generator matrix $B = B_t$ is updated as

$$[\boldsymbol{B}_{t+1}]_{i,j} = [\boldsymbol{B}_t]_{i,j} - \epsilon \frac{\partial g(\boldsymbol{B}_t, \boldsymbol{z}_t)}{\partial [\boldsymbol{B}_t]_{i,j}}$$
(16)

DRAFT

REVISION R1, JANUARY 25, 2025

for i, j = 1, ..., n and $i \ge j$, where ϵ is a small step size.¹

As usual when gradient descent is applied to a smooth function, the iterations converge for small enough step sizes. If a too large step is taken in (16), then B may move to the other side of the "valley", where g(B, z) increases again. For very large steps, a diagonal element $[B]_{i,i}$ may even become negative, in which case the optimization fails. To avoid this, ϵ should be chosen small enough. What is small enough depends greatly on the structure of the generator matrix B. Generally, an ill-conditioned B is more sensitive to the size of ϵ , whereas a *reduced* B is more stable. A reduced generator matrix is one whose rows (basis vectors) are relatively short and orthogonal to each other, according to some heuristic criterion [26]. We therefore apply *lattice reduction* to B regularly throughout the iterations [14]. Reduction also brings the added benefit of speeding up the closest point search algorithm [23], which dominates the complexity of the optimization procedure. On the other hand, reduction generally brings the generator matrix away from the desired lower-triangular form, but this can be alleviated by rotating the coordinate system.

The elements of the initial generator matrix B_0 are chosen as independent zero-mean unit-variance Gaussian random variables, immediately followed by reduction and orthogonal transformation to bring it into the desired lower-triangular form with positive diagonal elements. Repeating the optimization procedure multiple times with different B_0 enables the algorithm to find local minima in different parts of the optimization space, which is essential when searching for a global minimum of the nonconvex function G. There are infinitely many global minima in any dimensions n, reflecting the fact that equivalent generator matrices (2) yield the same NSM G.

C. Step size

The convergence of the algorithm depends strongly on the choice of step size ϵ in (16). The step size may be selected as a function of B_t and t (but not z_t , i, or j). In our optimization algorithm, we define

$$\epsilon = (n/2)V^{2/n}\mu_t,\tag{17}$$

where μ_t is an annealing scheme to be discussed later.

The factor $V^{2/n}$ in (17) makes the algorithm insensitive to the scale of B. To see this, consider how the update rule (16) affects a family of generator matrices $B_t = B = c\bar{B}$ for any fixed generator matrix \bar{B} and a variable scale factor c > 0. For a given z, e scales linearly with c, V scales with c^n , and g(B, z) in (7) remains constant. Furthermore, the gradient (15) scales with c^{-1} and ϵ in (17) scales with c^2 . Therefore, B_{t+1} in (16) scales linearly with c if B_t scales with c, as desired. The coefficient n/2 is not important and only serves to simplify the notation in Sec. IV.

Substituting (17) into (16), the update rule becomes

$$[\boldsymbol{B}_{t+1}]_{i,j} = [\boldsymbol{B}_t]_{i,j} - \mu_t \frac{nV^{2/n}}{2} \frac{\partial g(\boldsymbol{B}_t, \boldsymbol{z}_t)}{\partial [\boldsymbol{B}_t]_{i,j}},$$
(18)

where the coefficient $(n/2)V^{2/n}$ is conveniently absorbed by its inverse in (15).

¹With time-dependent z and B, other quantities such as V and e also become time-dependent, although we will still write them without explicit subscripts t.

January 25, 2025

=

TABLE I
PARAMETERS OF ALGORITHM 1.

Parameter	Description	Fast	Medium	Slow
μ_0	Initial step size	0.005	0.001	0.0005
ν	Ratio between initial and final step size	200	500	1000
T	Number of steps	1 000 000	10 000 000	100 000 000
$T_{ m r}$	Reduction interval (should divide T)	100	100	100

The purpose of μ_t is that the speed of movement in the variable space should decrease gradually with time, which is why it is called an annealing (or cooling) scheme. A large step size is beneficial initially, allowing B_t to move fast to the vicinity of a minimum and escape possible plateaus and shallow local minima, whereas a small step size is better near the end of the optimization process, in order to fine-tune B_t towards the exact minimum.

As usual in SGD optimization, there is no guarantee that the algorithm will converge to the global optimum, or even to a local minimum. It may with a nonzero probability terminate literally anywhere in the vast space of generator matrices B. The annealing scheme and other optimization parameters should be chosen to maximize the probability of a successful outcome, i.e., a lattice very near a local optimum.

Lacking an analytic strategy to find the best annealing scheme for our purpose, we tested many options numerically. We focused on n = 8, where, according to a 1998 conjecture [14], there is a unique local optimum: the Gosset lattice E_8 . We designed many thousands of 8-dimensional lattices using the proposed SGD method and estimated how similar the obtained lattices were to E_8 . However, comparing two lattices is not a trivial task, since any given lattice, including E_8 , can be represented using infinitely many different generator matrices. Instead of comparing the generator matrices directly, we therefore numerically found the shortest nonzero lattice vectors in each of the obtained lattices. This was done using a modified version of the "kissing number" algorithm in [23, Sec. VI-B]. In the E_8 lattice, there are 240 shortest nonzero vectors, which have a squared norm of 2 if the lattice is normalized to unit volume, and it is the unique 8-dimensional lattice with this property [2, pp. 22, 120–123]. Our accuracy metric is therefore the mean square error of the squared Euclidean norms of the 240 shortest vectors, given that their true values should all be 2. This metric is independent of basis changes and orthogonal transformation (Uand R in (2)).

Three annealing schemes were tested, in which μ_t decreases linearly, quadratically, or exponentially with t. For each scheme, multiple step size parameters were tested. It turned out that the exponential decrease consistently showed better accuracy after a given number of iteration steps. We therefore apply an exponential annealing scheme in our proposed lattice construction algorithm, which is detailed in the next section.

IV. IMPLEMENTATION

In this section, we present the lattice construction algorithm from a purely practical perspective. The theory developed in the previous section is necessary to fully understand its operation, but not to implement it.

REVISION R1, JANUARY 25, 2025

9

Algorithm	1	Iterative	lattice	construction
-----------	---	-----------	---------	--------------

Input: Dimension *n* Output: Generator matrix B 1: $\boldsymbol{B} \leftarrow ORTH(RED(GRAN(n, n)))$ 2: $V \leftarrow \prod_{i=1}^{n} [\boldsymbol{B}]_{i,i}$ 3: $\boldsymbol{B} \leftarrow V^{-1/n} \boldsymbol{B}$ 4: for t = 0 to T - 1 do $\mu \leftarrow \mu_0 \nu^{-t/(T-1)}$ 5: $\boldsymbol{z} \leftarrow URAN(n)$ 6: $y \leftarrow z - CLP(B, zB)$ 7: 8: $e \leftarrow yB$ for i = 1 to n do 9: for j = 1 to i - 1 do 10: 11: $[\boldsymbol{B}]_{i,j} \leftarrow [\boldsymbol{B}]_{i,j} - \mu [\boldsymbol{y}]_i [\boldsymbol{e}]_j$ end for 12: $[\boldsymbol{B}]_{i,i} \leftarrow [\boldsymbol{B}]_{i,i} - \mu\left([\boldsymbol{y}]_i[\boldsymbol{e}]_i - \frac{\|\boldsymbol{e}\|^2}{n[\boldsymbol{B}]_{i,i}}\right)$ 13: 14: end for 15: if $(t \mod T_r) = T_r - 1$ then $\boldsymbol{B} \leftarrow ORTH(RED(\boldsymbol{B}))$ 16: $V \leftarrow \prod_{i=1}^{n} [\boldsymbol{B}]_{i,i}$ 17: $\boldsymbol{B} \leftarrow V^{-1/n} \boldsymbol{B}$ 18: end if 19: 20: end for

The iterative algorithm is detailed in Algorithm 1. Its operation is controlled by four parameters, which are summarized in Tab. I. As usual, there is a trade-off between time and quality. Suggested sets of parameters are listed in the last three table columns, of which the last one yields the most accurate results at the cost of a higher computational complexity. Generally speaking, one should shift from "Fast" to "Medium" to "Slow" with increasing dimension to achieve the same accuracy, or for improved accuracy at a fixed dimension.

The utility functions *GRAN*, *URAN*, *CLP*, *RED*, and *ORTH*, on which the algorithm relies, are described in the following paragraphs.

The function URAN(n) returns n random real numbers, which are uniformly distributed in [0, 1) and independent of each other. Interpreted as a vector, URAN(n) returns a random point uniformly distributed in an n-dimensional hypercube. We use the *permuted congruential generator* [27], which is well documented and fulfills advanced tests of randomness.² Furthermore, GRAN(n, m) returns an $n \times m$ matrix of random independent real numbers, each with a Gaussian zero-mean, unit-variance distribution. We use the Gaussian random number generator by Paley and Wiener [28, Eqs. (37.01), (37.03)].³

²This is not always the case for the built-in random number generators in various programming languages.

³The method has later become known as the Box-Muller transform.

10

The closest lattice point function CLP(B, x) finds the point in the lattice generated by B that is closest to x. The output is not the lattice point itself but rather its integer coordinates $\arg \min_{u \in \mathbb{Z}^n} ||x - uB||^2$. We use [24, Algorithm 5], which is the fastest general closest-point search algorithm known to us. It applies to square, lower-triangular generator matrices with positive diagonal elements, which is exactly how lattices are represented in Algorithm 1.

The *reduction* function RED(B) returns another generator matrix for the lattice generated by B, in which the rows (basis vectors) are shorter and more orthogonal to each other than in B. If no improved generator matrix is found, B is returned unchanged. A fast and popular algorithm for the purpose is the *Lenstra–Lenstra–Lovász algorithm* [26, Fig. 1], which we apply in this work. In the context of (2), reduction corresponds to finding a suitable U.

The orthogonal transformation function ORTH(B) rotates and reflects an arbitrary generator matrix into a square, lower-triangular form with positive diagonal elements. This corresponds to finding a new coordinate system for the lattice, in which the first *i* unit vectors span the subspace of the first *i* basis vectors, for i = 1, ..., n. We implement this function by Cholesky-decomposing [29, Sec. 14.5] the Gram matrix $A = BB^{T}$. In the context of (2), orthogonal transformation corresponds to right-multiplying the generator matrix by a matrix R with orthonormal rows.

After Step 13, we apply a sanity check that $[B]_{i,i}$ is still positive (not shown in Algorithm 1). If any diagonal element is zero or negative, then the algorithm is aborted and the construction has failed, which suggests that the parameters μ_0 , T_r , or both may have been set too large. We encountered this problem during algorithm development, but never when the parameters were chosen as in Tab. I.

Steps 15–19 can be omitted under some conditions (e.g., small dimensions n and small initial step sizes μ_0), but we do not recommend it. The usefulness of reduction is discussed in Sec. III-B.

Steps 2–3 and 17–18 normalize the lattice to unit volume, which is sometimes necessary to avoid numerical underflow in (10) for large-scale problems (large n and T). For small- and medium-scale problems, normalizing B does not affect the operation of the algorithm or the resulting NSM.

V. ALTERNATIVE ALGORITHMS

Algorithm 1 is not the only way to numerically optimize lattice quantizers. In this section, we review all alternative algorithms that we are aware of. We have implemented them all and confirmed that they work, but we have not compared their performance in sufficient detail to appoint a winner. A rigorous comparison would require careful meta-optimization to find the most efficient algorithm parameters, averaging over many trials in each dimension. This was done for Algorithm 1 in Sec. IV but not for the alternative algorithms reviewed in this section.

A. The Agrell–Eriksson algorithm

The first, and so far only, published algorithm for lattice quantizer optimization was proposed by Agrell and Eriksson in 1998 [14]. It serves as a benchmark for the present study. Like Algorithm 1, the 1998 benchmark restricts the generator matrix to a square, lower-triangular form and updates the elements by SGD. The first n - 1

REVISION R1, JANUARY 25, 2025

dimensions are updated in the direction of the negative gradient of the unnormalized second moment, whereas the last dimension (controlled by a single element in the bottom-right corner) in every iteration is set to maintain a unit volume. An example serves to highlight how this strategy differs from using SGD with the NSM gradient.

Example 1: We consider the first iteration when optimizing a four-dimensional lattice. For example, let n = 4, t = 0, and $\mathbf{B} = \mathbf{B}_0 = \mathbf{I}_4$. Assume that the first random vector is $\mathbf{z} = \mathbf{z}_0 = [0.6, 0.6, 0.0, 0.0]$. The closest point problem (5) is solved by $\hat{\mathbf{u}} = [1, 1, 0, 0]$. We subsequently obtain $\mathbf{e} = (\mathbf{z} - \hat{\mathbf{u}})\mathbf{B} = [-0.4, -0.4, 0.0, 0.0]$, $\|\mathbf{e}\|^2 = 0.32$, V = 1, and $g(\mathbf{B}, \mathbf{z}) = 0.08$.

If the benchmark method [14, Tab. I] is applied, then the updated generator matrix is

$$\mathbf{B}_{1}^{\prime} = \begin{bmatrix}
1 - 0.32\epsilon & 0 & 0 & 0 \\
-0.32\epsilon & 1 - 0.32\epsilon & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & \frac{1}{1 - 0.32\epsilon}
\end{bmatrix}$$

$$= \mathbf{B}_{0} - \epsilon \begin{bmatrix}
0.32 & 0 & 0 & 0 \\
0.32 & 0.32 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.64
\end{bmatrix} + O(\epsilon^{2})$$

$$= \mathbf{B}_{0} - \bar{\epsilon} \begin{bmatrix}
0.3780 & 0 & 0 & 0 \\
0.3780 & 0.3780 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.7559
\end{bmatrix} + O(\bar{\epsilon}^{2}), \quad (19)$$

where we in the last step substituted $\epsilon = 1.181\bar{\epsilon}$, so that $\bar{\epsilon}$ multiplies a matrix with unit Frobenius norm. With the B'_1 and the same z as before, we obtain $e = [-0.4 + 0.256\bar{\epsilon}, -0.4 + 0.256\bar{\epsilon}, 0, 0]$, V = 1, and $g(B, z) = 0.08 - 0.0907\bar{\epsilon} + 0.0286\bar{\epsilon}^2$. The negative coefficient of $\bar{\epsilon}$ shows that g(B, z) decreases for small values of $\bar{\epsilon}$, as expected.

REVISION R1, JANUARY 25, 2025

12

If instead Algorithm 1 is applied in the same scenario, the updated generator matrix is from (15)-(16)

г

$$\boldsymbol{B}_{1} = \boldsymbol{B}_{0} - \epsilon \begin{vmatrix} 0.04 & 0 & 0 & 0 \\ 0.08 & 0.04 & 0 & 0 \\ 0 & 0 & -0.04 & 0 \\ 0 & 0 & 0 & -0.04 \end{vmatrix}$$
$$= \boldsymbol{B}_{0} - \bar{\epsilon} \begin{vmatrix} 0.3536 & 0 & 0 & 0 \\ 0.7071 & 0.3536 & 0 & 0 \\ 0 & 0 & -0.3536 & 0 \\ 0 & 0 & 0 & -0.3536 \end{vmatrix},$$
(20)

٦

where we substituted $\epsilon = 8.839\bar{\epsilon}$ so that $\bar{\epsilon}$ in (20) as well as in (19) multiplies a matrix with unit Frobenius norm. After the update, we have $\boldsymbol{e} = (\boldsymbol{z} - \hat{\boldsymbol{u}})\boldsymbol{B}_1 = [-0.4 + 0.4243\bar{\epsilon}, -0.4 + 0.1414\bar{\epsilon}, 0, 0], V = (1 - 0.125\bar{\epsilon}^2)^2$, and from (7)

$$g(\boldsymbol{B}, \boldsymbol{z}) = \frac{0.32 - 0.4525\bar{\epsilon} + 0.2\bar{\epsilon}^2}{4(1 - 0.125\bar{\epsilon}^2)}$$
$$= 0.08 - 0.1131\bar{\epsilon} + 0.06\bar{\epsilon}^2 + O(\bar{\epsilon}^3)$$
(21)

for the given $z = z_0$. Again, the coefficient of $\bar{\epsilon}$ is negative, indicating a decreasing g(B, z).

The convergence rate for small step sizes $\bar{\epsilon}$ is governed by the linear term, which in this example is $-0.0907\bar{\epsilon}$ and $-0.1131\bar{\epsilon}$ for the two update rules, respectively. Evidently, g(B, z) decreases faster with Algorithm 1 than with the benchmark, which illustrates that the latter does not follow the gradient of the NSM. Therefore, the benchmark generally needs more iterations for the same accuracy than Algorithm 1, which we have also observed numerically. For other values of B and z than the ones considered in Example 1, the difference between the g(B, z) values obtained with the two update rules can be larger or smaller, but Algorithm 1 always reduces g(B, z) more than the benchmark does. Indeed, the coefficient of $\bar{\epsilon}$ in (21) is the smallest (most negative) possible and represents the direction of steepest descent subject to the lower-triangular constraint, since the elements of the right-hand matrix in (20) are the partial derivatives in (15).

Another indication that the benchmark performs suboptimally is that the third and fourth dimensions are updated differently in (19), even though the input data in this (artificial) example is fully symmetric with respect to these two dimensions. This is because the fourth dimension is used to maintain a constant volume. With Algorithm 1, in contrast, the third and fourth dimensions are updated equally, as evident from (20).

In the benchmark, as presented in [14], the initial generator matrix was set to the identity matrix. The motivation was that G as a function of B has a saddle point at I_n , from which subsequent iterations can bring B_t down into one of several different "valleys." Thereby, the algorithm is able to find different local minima in subsequent runs. However, although it seems intuitively likely and previous results show that it is the case in low dimensions [14], we cannot guarantee that there is a downhill route from the identity matrix into a global minimum. What if

REVISION R1, JANUARY 25, 2025

the global minimum lies on the other side of another saddle point, which the SGD algorithm does not have the momentum to overcome? To avoid this potential risk, the initial generator matrix is chosen randomly in Algorithm 1.

Other minor differences between the benchmark and Algorithm 1 are in the choices of step size annealing scheme (linear in the benchmark and exponential in Algorithm 1) and closest point search algorithm ([23] in the benchmark and [24] in Algorithm 1).

B. Full-matrix stochastic gradient descent

In Algorithm 1, the generator matrix is constrained to a lower-triangular structure, and the updates follow the negative gradient of the NSM subject to this constraint for complexity reasons. A different, and perhaps more natural, approach is to update an unconstrained $n \times n$ generator matrix according to the negative gradient of the NSM with respect to all matrix elements. Even if the initial matrix would be lower-triangular, the NSM gradient would soon bring it away from that structure. An optimization algorithm based on this principle can be defined as follows.

Let

$$g(\boldsymbol{B}, \boldsymbol{z}) = \frac{1}{n} |\det \boldsymbol{B}|^{-2/n} ||\boldsymbol{e}||^2,$$
(22)

where, in contrast to the analogous equation (7), the absolute value is needed because det B may here be positive or negative. As in Algorithm 1, z is uniform in $[0,1)^n$, $y = z - \hat{u}$ with \hat{u} given by (5), and e = yB. Again we neglect the zero-measure set of z for which (5) does not have a unique solution.

The gradient of (22) with respect to B is

$$\nabla g(\boldsymbol{B}, \boldsymbol{z}) = \frac{1}{n} |\det \boldsymbol{B}|^{-2/n} \nabla \|\boldsymbol{e}\|^2 + \frac{1}{n} \|\boldsymbol{e}\|^2 \nabla |\det \boldsymbol{B}|^{-2/n}.$$
(23)

The two gradients on the right-hand side of (23) are

$$\nabla \|\boldsymbol{e}\|^{2} = \nabla(\boldsymbol{y}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{y}^{\mathrm{T}})$$

$$= 2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{B}$$

$$= 2\boldsymbol{y}^{\mathrm{T}}\boldsymbol{e}, \qquad (24)$$

$$\nabla |\det \boldsymbol{B}|^{-2/n} = -\frac{2}{n} |\det \boldsymbol{B}|^{-2/n-1} \nabla |\det \boldsymbol{B}|$$

$$= -\frac{2}{n} |\det \boldsymbol{B}|^{-2/n-1} (\operatorname{sgn} \det \boldsymbol{B}) \nabla \det \boldsymbol{B}$$

$$= -\frac{2}{n} |\det \boldsymbol{B}|^{-2/n-1} (\operatorname{sgn} \det \boldsymbol{B}) (\operatorname{adj} \boldsymbol{B})^{\mathrm{T}}, \qquad (25)$$

where sgn denotes the signum function and adj the adjugate or adjoint matrix⁴ [29, Sec. 15.8]. Combining (23),

⁴Not to be confused with the conjugate transpose, which is also sometimes called adjoint.

REVISION R1, JANUARY 25, 2025

14

(24), and (25) yields after simplification

$$\nabla g(\boldsymbol{B}, \boldsymbol{z}) = \frac{2}{n} |\det \boldsymbol{B}|^{-2/n} \left(\boldsymbol{y}^{\mathrm{T}} \boldsymbol{e} - \frac{\|\boldsymbol{e}\|^{2}}{n \det \boldsymbol{B}} (\operatorname{adj} \boldsymbol{B})^{\mathrm{T}} \right)$$
$$= \frac{2}{n} |\det \boldsymbol{B}|^{-2/n} \left(\boldsymbol{y}^{\mathrm{T}} \boldsymbol{e} - \frac{\|\boldsymbol{e}\|^{2}}{n} (\boldsymbol{B}^{-1})^{\mathrm{T}} \right).$$
(26)

As an SGD update rule, analogous to (18), we can use

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t - \mu_t \frac{n |\det \boldsymbol{B}_t|^{2/n}}{2} \nabla g(\boldsymbol{B}_t, \boldsymbol{z}_t)$$
$$= \boldsymbol{B}_t - \mu_t \left(\boldsymbol{y}_t^{\mathrm{T}} \boldsymbol{e}_t - \frac{\|\boldsymbol{e}_t\|^2}{n} (\boldsymbol{B}_t^{-1})^{\mathrm{T}} \right).$$
(27)

The result depends on the choice of initial lattice, number of iterations, annealing scheme for μ_t , and frequency of reduction and renormalization. With our preliminary implementation, this algorithm appears to need fewer iterations than Algorithm 1 to converge to a locally optimal lattice quantizer with the same accuracy, but every iteration is more complex. This is because general-purpose closest lattice point search algorithms need a triangular generator matrix [23], [24]. Thus, *B* needs to be converted to lower-triangular form by Cholesky or QR decomposition in every iteration. The inverse in (27) also adds complexity compared with (15)–(18).

The more complex iterations can partly be mitigated by treating p independent samples $z_{t,i} \in [0,1)^n$ for $i = 1, \ldots, p$ in a batch:

$$\boldsymbol{y}_{t,i} = \boldsymbol{z}_{t,i} - \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathbb{Z}^n} \| (\boldsymbol{z}_{t,i} - \boldsymbol{u}) \boldsymbol{B}_t \|^2,$$
(28)

$$\boldsymbol{e}_{t,i} = \boldsymbol{y}_{t,i} \boldsymbol{B}_t, \tag{29}$$

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t - \frac{\mu_t}{p} \sum_{i=1}^p \left(\boldsymbol{y}_{t,i}^{\mathrm{T}} \boldsymbol{e}_{t,i} - \frac{\|\boldsymbol{e}_{t,i}\|^2}{n} (\boldsymbol{B}_t^{-1})^{\mathrm{T}} \right).$$
(30)

With this modification, the triangular decomposition and matrix inverse need to be calculated only for every pth closest point search.

We do not know if this SGD variant performs better than Algorithm 1 if both are configured for the same execution time. It depends on the algorithm parameters, the employed hardware and software, and the dimension.

C. Whitening the quantization noise

In 1996, Zamir and Feder proved that the optimal lattice quantizer in every dimension has white quantization noise; i.e., a covariance matrix

$$\boldsymbol{U} = \frac{1}{V} \int_{\Omega} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{x} \,\mathrm{d}\boldsymbol{x}$$
(31)

proportional to the identity [30]. Furthermore, if a lattice \mathcal{L} has a covariance matrix U that is not proportional to the identity, then $\mathcal{L}U^{-1/2}$ has a lower NSM than \mathcal{L} . The whiteness property was generalized to locally optimal lattices in [18] and applied to optimize a parametric lattice in [20]. The second moment matrix can be empirically estimated as $\hat{U} = E^{T}E/p$, where E is a $p \times n$ vector whose rows are independent random vectors, uniformly distributed in Ω , and $p \ge n$. The generation of uniform random vectors in Ω is discussed in Sec. III-A.

REVISION R1, JANUARY 25, 2025

15

A simple lattice optimization algorithm follows straightforwardly from the white noise property. The idea is to repeatedly estimate the covariance matrix and update the generator matrix. Every update consists of computing

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \left(\frac{\boldsymbol{E}_t^{\mathrm{T}} \boldsymbol{E}_t}{p}\right)^{-1/2},\tag{32}$$

where E_t is composed of p vectors in the Voronoi region of the lattice generated by B_t , possibly followed by reduction and renormalization. Much fewer iterations are needed than with any of the previously described algorithms, but every iteration is more complex, involving p closest-point searches. Like in the previous algorithm, the generator matrix needs to be rotated into lower-triangular form in every iteration to enable closest point search.

Good results were obtained when p increases with t. Apart from this simple observation, we leave the metaoptimization of the algorithm parameters as an open question.

D. Simulated annealing

Simulated annealing is a classical heuristic in numerical optimization. The idea is to accept suboptimal updates with some probability, so that the algorithm will sometimes move "uphill," away from local minima.⁵ The probability is decreased with time as the optimization progresses and is sometimes referred to as "temperature." Simulated annealing has been successfully applied in the design of several types of codes and quantizers [31], [32], but not lattices, as far as we know.

All algorithms mentioned in this paper can be modified to include some aspects of simulated annealing. We propose to introduce it in the closest lattice point search, which consumes a large share of the computational resources in all considered optimization algorithms. Such search methods [23], [24] work by enumerating lattice points inside a sphere and reducing the size of the sphere every time a lattice point is found, until the sphere is empty. Simulated annealing can be implemented in this context by aborting the search with some probability every time a new lattice point is found. Thereby we achieve not only the desired erratic movement but also reduced computational complexity. If the probability, or temperature, is 100%, then the returned vector is the so-called *Babai point*, which is found very fast [23], [33], and if it is 0%, then the search method reverts to its normal, optimal behavior.

VI. IDENTIFICATION OF EXACT LATTICES

When a numerical lattice optimization algorithm terminates, the result is a generator matrix with numerical entries. Such matrices can be used to numerically estimate various lattice parameters, such as the NSM, packing radius, and covering radius. They do not immediately offer any geometrical insights about for example symmetries or algebraic construction methods, but they can pave the way for such progress if their underlying exact forms can be identified. Therefore, the algorithm's outputs should if possible be refined. We do this in three steps: first, computing and visualizing the theta series; second, replacing the numerically defined lattice with a similar but exact lattice, whose theta series has a certain desired form; and third, checking the obtained exact lattice for consistency.

⁵The annealing scheme used to adjust the step size in Sec. III-C is not simulated annealing in this sense.



Fig. 1. The evolution of a 10-dimensional lattice during numerical optimization.

A. Theta Image: Visualizing Lattice Structure

To monitor the progression during optimization and visualize the convergence, we find it informative to study the cumulative distribution of lattice point norms

$$N(\boldsymbol{B}, r) = \left| \left\{ \boldsymbol{u} \in \mathbb{Z}^n : \left\| \boldsymbol{u} \boldsymbol{B} \right\| \le r \right\} \right|.$$
(33)

The function can be numerically computed for any B and moderate values of r by trivial modifications of the kissing number algorithm in [23, Sec. VI-B]. We call a plot of N(B, r) versus r^2 a *theta image*, because it illustrates the *theta series* (4) of the lattice generated by B. For any lattice, N(B, r) = 1 for $0 < r < 2\rho$ and $N(B, 2\rho) = 1 + \tau$, where, as defined in Sec. II, ρ is the packing radius and τ is the kissing number.

Fig. 1 illustrates how the theta image of a lattice evolves during the iterations of Algorithm 1 and converges to a local minimum. The dimension was set to n = 10 and the optimization parameters were taken from the column "Medium" in Tab. I. As the algorithm progresses, the theta image approaches a staircase-like curve, composed of very steep and very flat sections, which indicates that the lattice points move towards a small number of spherical shells around the origin. We have observed this behavior in all dimension, provided that the optimization parameters are well chosen. The height of each vertical step indicates the number of lattice points in the corresponding shell and the horizontal location of the step indicates its squared radius.

The convergence towards discrete shells around the origin is not surprising, given that good lattices usually have a high degree of symmetry. Known theta series, which are listed for many classical lattices in [2, Ch. 4], confirm that many lattice points have equal norms in such lattices.

B. From Approximate to Exact Theta Image

The observation that numerically optimized lattices tend to have many lattice vectors with similar norms suggests a systematic approach to the generation of exact lattice representations. In this section, we describe a technique to perturb the Gram matrix so that all lattice points whose norms are numerically *almost* equal become *exactly* equal. REVISION R1, JANUARY 25, 2025



Fig. 2. The exact theta image of D_{10}^+ , towards which the curves in Fig. 1 converge.

The squared norm of a lattice vector uB is the quadratic form $||uB||^2 = uAu^T$, where $u \in \mathbb{Z}^n$. If A is not known exactly but only through an approximation \tilde{A} , then we can use the theta image of \tilde{A} to identify a set of integer vectors u_1, \ldots, u_M for which $u_1 \tilde{A} u_1^T \approx \cdots \approx u_M \tilde{A} u_M^T$. Thereafter we replace \tilde{A} with an unknown, generic A of the same dimension and solve the equations $u_1 A u_1^T = \cdots = u_M A u_M^T$ with the same integer vectors. Additional equations can be obtained from other vertical steps in the theta image. We do not impose a certain value for the norms, only that they should be equal to each other. The first diagonal element of A is set to 1 to fix the scale and A is symmetric, giving a system of equations with n(n+1)/2 - 1 unknowns.

The system of equations is linear and can be solved exactly using standard methods. One of three possible outcomes can occur: one solution, multiple solutions, or no solution. If a single solution occurs, then it is presumably a locally optimal lattice quantizer and the investigation can proceed to the next stage, which is to characterize its properties and compare it with known lattices. If multiple solutions occur, then the system of equations was underdetermined and it may help to extract equations from additional vertical steps in the theta image. If these additional equations do not reduce the number of solutions, it may be because the symmetry group of the underlying exact lattice is separable into disjoint groups involving separate lower-dimensional rotations. The solution may still identify lower-dimensional sublattices and thus reduce the number of unknowns from n(n+1)/2 - 1, even though at least one unknown remains. Such *parametric lattices* require additional numerical and analytical methods to be identified exactly [20]. The smallest dimension for which this occurs is n = 9, where a single parameter cannot be identified by symmetry considerations [14], [16]. The third and final outcome is that no solution is returned. Then the system was overdetermined and a plausible explanation is that what looks like a single vertical step in the theta image may theoretically consist of two steps at different but similar norms. We have not yet encountered this outcome in any practical trials.

C. Validation and Identification

An important consistency check is now to compute the theta series of the obtained exact lattice, including more terms than those considered in the equation-solving framework described in Sec. VI-B. If those additional terms

18

REVISION R1, JANUARY 25, 2025



Fig. 3. Theta images of numerically optimized lattices in dimensions 10–16 and of the exact lattices towards which these optimized lattices DRAFT January 25, 2025 January 25, 2025

REVISION R1, JANUARY 25, 2025

match vertical steps in the theta image of the numerically optimized lattice, then it is a strong indication that its exact form has been correctly determined. Furthermore, the theta series can be compared with other theta series found numerically, analytically, or in the literature, to help identify equivalences.

We return to the 10-dimensional lattice example of Sec. VI-A. When the optimization algorithm terminates, it has 180 points with squared norms between 1.952 and 2.043 and 512 points with squared norms between 2.460 and 2.544. The lattice has no points with squared norms between 2.043 and 2.460, and no nonzero points with squared norm less than 1.952. The two point sets, which are visualized in the first two vertical steps in Fig. 1, provide together 179 + 511 = 690 linear equations in 54 unknowns. The system of equations has a unique solution, apart from an overall scale factor, which gives a Gram matrix with rational elements. The mean square error between the elements of the numerical and exact Gram matrices is $7.0 \cdot 10^{-4}$, when both matrices are scaled to determinant 1. When instead the "Slow" parameters in Tab. I were used, a mean square error of $7.4 \cdot 10^{-5}$ was obtained. These error values should be interpreted as mere indications, since they were not averaged over many trials.

The theta image of the resulting lattice, scaled to det A = 1, is shown in Fig. 2. Indeed all vertical steps, not only the first two, agree perfectly with steps in Fig. 1. The theta series corresponding to Fig. 2 is

$$1 + 180q^2 + 512q^{5/2} + 3380q^4 + 5120q^{9/2} + \cdots$$
(34)

This result suggests that the obtained lattice might be D_{10}^+ , which has exactly the same theta series [2, p. 120] and has been previously suggested as the best 10-dimensional lattice quantizer [14].

VII. LATTICE QUANTIZERS IN DIMENSIONS 10-16

As a proof of concept, lattices were numerically optimized in dimensions 10-16. Algorithm 1 was implemented on a high-performance computer cluster, running on 100 cores in parallel. Since the NSM is a nonconvex function of the elements of \boldsymbol{B} with multiple minima, we designed 100 lattices in each dimension. The optimization parameters are listed under "Slow" in Tab. I.

The NSM G of each lattice was numerically estimated as described in Sec. III-A. To compute \hat{G} in (8), lines 6–8 of Algorithm 1 were repeated many times (in this case, 10^8 times), the obtained squared vector norms $||e||^2$ were averaged, and the result was normalized. The variance $\hat{\sigma}^2$ of the NSM estimate \hat{G} was estimated according to (9). In each dimension, several lattices with the smallest \hat{G} were selected for further analysis and possible identification.

For the selected lattices in each dimension n = 10, ..., 16, the theta image was computed and plotted as described in Sec. VI-A. It turned out that these lattices have almost identical theta images, which suggests that they represent equivalent lattices, although their Gram (and generator) matrices are quite different. Specifically, the number of hits into the best equivalence class ranged from 16 (for n = 15) to 96 (for n = 16). It therefore seems plausible that 100 trials in each dimension is more than enough to find the optimal lattices, although there is no guarantee.

For the best lattices in each dimension, vertical steps in their theta images were identified visually and exploited to define a system of equations as described in Sec. VI-B. In five of the seven cases, the system of equations had a unique solution, which indicated the exact lattice towards which the optimization process had converged. In two cases, more advanced identification methods were needed [20]. The theta images of the obtained exact lattices

n	$\hat{G}\pm 2\hat{\sigma}$	heta(q)	Converged lattice	G
10	0.070811 ± 0.000003	$1 + 180q^2 + 512q^{5/2} + 3380q^4 + 5120q^{9/2} + \cdots$	D_{10}^{+}	0.070813818
11	0.070424 ± 0.000002	$1 + 132q^{3\alpha} + 990q^{4\alpha} + 2970q^{6\alpha} + 12408q^{7\alpha} + \cdots$	A_{11}^{3}	0.070426259
12	0.070029 ± 0.000002	$1 + 120q^2 + 768q^{5/2} + 1024q^3 + 4104q^4 + \cdots$	Glued $D_6 \times D_6$	0.070031226
13	0.069696 ± 0.000002	$1 + 56q^{1.989} + 40q^{2.006} + 2q^{2.049} + 256q^{2.412} + \cdots$	Glued $A_7 \times D_5 \times \mathbb{Z}$	0.069697639
14	0.069261 ± 0.000002	$1 + 24q^{1.996} + 270q^{2.311} + 1920q^{2.731} + 960q^{3.467} + \cdots$	Glued $K'_{10} \times D_4$	0.069261779
15	0.068869 ± 0.000002	$1 + 280q^{12\beta} + 2048q^{15\beta} + 1710q^{16\beta} + 5376q^{20\beta} + \dots$	Λ^*_{15}	0.068871726
16	0.068296 ± 0.000002	$1 + 4320q^{2\sqrt{2}} + 61440q^{3\sqrt{2}} + 522720q^{4\sqrt{2}} + \cdots$	Λ_{16}	0.068297622
		0/11 10/11 10/5		

 TABLE II

 Conjectured optimal lattice quantizers obtained by numerical optimization.

where $\alpha = 2^{9/11} \cdot 3^{-10/11}$ and $\beta = 2^{-12/5}$

matches those of the numerically optimized lattices excellently, as shown in Fig. 3. As sanity checks, we confirmed that the volume of the Voronoi region in each case is $V = \det B$ and that the normalized second moment matrix [30] is

$$\frac{1}{V} \int_{\Omega} \boldsymbol{x}^T \boldsymbol{x} \, \mathrm{d}\boldsymbol{x} = G V^{2/n} \boldsymbol{I}_n.$$
(35)

The former condition holds for any lattice (see Sec. II) and the latter holds for any locally optimal n-dimensional lattice quantizer [18]. We conjecture that all seven lattices are optimal in their respective dimensions.

The next step was to investigate if the found lattices are equivalent to any known lattices. The theta series proved useful for this purpose too, because having identical theta series is a necessary (but not sufficient) condition for equivalence between two lattices [2, pp. xxix, 47]. We computed the theta series of approximately 110 known lattices in dimensions 10–16, all normalized to unit volume, and compared with the theta series of the found lattices. If an identical theta series was found, we verified the equivalence by finding explicit matrices U and R that transformed one of the generator matrices into the other according to (2). The results are summarized in Tab. II.

In dimensions 10–12, the algorithm converged to the previously best known lattice quantizers, suggesting that these might indeed be optimal in their respective dimensions. For n = 10, D_{10}^+ was proposed in [14] and its exact NSM was computed in [15]. For n = 11, the Coxeter lattice A_{11}^3 was proposed and its exact NSM was computed in [15]. For n = 12, a lattice constructed by gluing four copies of $D_6 \times D_6$ was proposed and its exact NSM was computed in [19].

In dimensions 13 and 14, the obtained lattices have lower NSMs than the previously best reported values [17]. These new lattices are parametric and cannot be fully identified by the technique in Sec. VI-B. The 13-dimensional lattice is composed of eight copies of $A_7 \times D_5 \times \mathbb{Z}$ and the 14-dimensional one of four copies of $K'_{10} \times D_4$, with the constituent sublattices scaled separately. These lattices, including their analysis methods and optimal scale factors, are described in [20].

In dimension 15, the algorithm unexpectedly converged to Λ_{15}^* , the dual of the laminated lattice Λ_{15} defined in

REVISION R1, JANUARY 25, 2025

21

[34]. Among four inequivalent laminated lattices denoted by Λ_{15} in [35], the relevant one is labelled 15.1. As far as we know, neither Λ_{15} nor its dual has been previously considered for quantization. A previously unpublished generator matrix for Λ_{15}^* is

	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0
	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
B =	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0
	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0
	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0
	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0
	$\frac{1}{2}$														

We computed its NSM numerically and exactly. A numerical estimate of the NSM using the method described in Sec. III-A but with a higher accuracy than in Tab. II (10^{12} random samples) is $\hat{G} \pm 2\hat{\sigma} = 0.06887171 \pm 0.00000002$. The Voronoi region of Λ_{15}^* was completely charted using the methods in [25], [36]. This computation was more demanding than expected, because the Voronoi region turned out to have 17 675 520 000 vertices. For comparison, the best 16-dimensional lattice has only 201 343 200 vertices [25]. The kissing number of Λ_{15}^* is 280. Its Voronoi region has 40 134 facets and altogether 142 450 921 496 795 faces in all dimensions from 0 to 15. These are split into 3 784 046 equivalence classes under its symmetry group, whose order of 41 287 680 agrees with the value for Λ_{15} given in [35, Table 1]. The numbers of classes in dimensions 0 through 15 are, respectively, 820, 13 701, 92 610, 330 483, 703 579, 956 395, 862 632, 526 405, 219 298, 62 987, 12 823, 1 994, 272, 40, 6, and 1. The exact NSM of Λ_{15}^* is

$$G = \frac{306\,171\,250\,566\,121\,481\,924\,149\,919}{3\,369\,080\,934\,926\,681\,018\,204\,160\,000\cdot2^{2/5}} \approx 0.0688717257963967, \tag{37}$$

which matches the numerical estimate very well. This is a clear improvement over the previously best known lattice quantizer, whose NSM is approximately 0.07037 [17].

In dimension 16, finally, the optimization process again converged to the previously best known lattice quantizer. The 16-dimensional Barnes–Wall lattice, which is also equivalent to the laminated lattice Λ_{16} , was proposed for quantization already in [13] and its exact NSM was computed in [25].

REVISION R1, JANUARY 25, 2025

VIII. CONCLUSIONS

Using a new numerical optimization algorithm and a new technique to derive exact lattices from numerically defined ones, conjectured optimized lattice quantizers are obtained in selected dimensions up to 16. In dimensions 13 and 14, new parametric lattices are discovered. The dual of the 15-dimensional laminated lattice emerges as a new best lattice in dimension 15. In dimensions 10, 11, 12, and 16, numerical evidence is obtained that the previously best known lattices might indeed be optimal.

It took 27 years until the benchmark algorithm was improved and the range of numerically designed lattice quantizers was extended from 10 dimensions up to 16. If anyone in the next decades would embark on a journey towards dimensions beyond 16, we believe that the algorithms reviewed in Sec. V may deserve to be carefully explored and refined.

ACKNOWLEDGMENT

The authors are grateful to Or Ordentlich for insightful feedback on an earlier version of the manuscript and for suggesting and deriving the alternative optimization algorithms in Sec. V-B and V-C.

REFERENCES

- R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998. [Online]. Available: https://doi.org/10.1109/18.720541
- J. H. Conway and N. J. A. Sloane, Sphere Packings, Lattices and Groups, 3rd ed. New York, NY: Springer, 1999. [Online]. Available: https://doi.org/10.1007/978-1-4757-6568-7
- [3] R. Zamir, Lattice Coding for Signals and Networks. Cambridge, UK: Cambridge University Press, 2014. [Online]. Available: https://doi.org/10.1017/CBO9781139045520
- [4] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in *Proc. IEEE Int. Conf. Computer Vision*, Rio de Janeiro, Brazil, Oct. 2007. [Online]. Available: https://doi.org/10.1109/ICCV.2007.4408924
- [5] Q. Guo, T. Johansson, and P. Stankovski, "Coded-BKW: solving LWE using lattice codes," in Proc. Annual Cryptology Conference (CRYPTO), 2015, pp. 23–42. [Online]. Available: https://doi.org/10.1007/978-3-662-47989-6_2
- [6] H. Jégou, L. Amsaleg, C. Schmid, and P. Gros, "Query-adaptive locality sensitive hashing," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc., Las Vegas, NV, Mar.–Apr. 2008, pp. 825–828. [Online]. Available: https://doi.org/10.1109/ICASSP.2008.4517737
- [7] A. Sablayrolles, M. Douze, C. Schmid, and H. Jégou, "Spreading vectors for similarity search," in *Proc. Int. Conf. Learning Representations*, New Orleans, LA, May 2019. [Online]. Available: https://openreview.net/forum?id=SkGuG2R5tm
- [8] B. Allen, "Optimal template banks," *Phys. Rev. D*, vol. 104, p. 042005, Aug. 2021. [Online]. Available: https://doi.org/10.1103/PhysRevD. 104.042005
- [9] L. Fejes Tóth, "Sur la représentation d'une population infinie par un nombre fini d'éléments," Acta Mathematica Hungarica, vol. 10, no. 3–4, pp. 299–304, Sept. 1959, in French. [Online]. Available: https://doi.org/10.1007/BF02024494
- [10] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inf. Theory*, vol. IT-25, no. 4, pp. 373–380, July 1979. [Online]. Available: https://doi.org/10.1109/TIT.1979.1056067
- [11] E. S. Barnes and N. J. A. Sloane, "The optimal lattice quantizer in three dimensions," SIAM J. Alg. Disc. Meth., vol. 4, no. 1, pp. 30–41, Mar. 1983. [Online]. Available: https://doi.org/10.1137/0604005
- [12] J. H. Conway and N. J. A. Sloane, "Voronoi regions of lattices, second moments of polytopes, and quantization," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 211–226, Mar. 1982. [Online]. Available: https://doi.org/10.1109/TIT.1982.1056483
- [13] —, "On the Voronoi regions of certain lattices," SIAM J. Alg. Disc. Meth., vol. 5, no. 3, pp. 294–305, Sept. 1984. [Online]. Available: https://doi.org/10.1137/0605031
- [14] E. Agrell and T. Eriksson, "Optimization of lattices for quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1814–1828, Sept. 1998. [Online]. Available: https://doi.org/10.1109/18.705561

22

REVISION R1, JANUARY 25, 2025

- [15] M. Dutour Sikirić, A. Schürmann, and F. Vallentin, "Complexity and algorithms for computing Voronoi cells of lattices," *Mathematics of Computation*, vol. 78, no. 267, pp. 1713–1731, July 2009. [Online]. Available: https://doi.org/10.1090/S0025-5718-09-02224-8
- [16] B. Allen and E. Agrell, "The optimal lattice quantizer in nine dimensions," Annalen der Physik, vol. 533, no. 12, p. 2100259, Dec. 2021. [Online]. Available: https://doi.org/10.1002/andp.202100259
- [17] S. Lyu, Z. Wang, C. Ling, and H. Chen, "Better lattice quantizers constructed from complex integers," *IEEE Trans. Commun.*, vol. 70, no. 12, pp. 7932–7940, Dec. 2022. [Online]. Available: https://doi.org/10.1109/TCOMM.2022.3215685
- [18] E. Agrell and B. Allen, "On the best lattice quantizers," *IEEE Trans. Inf. Theory*, vol. 69, no. 12, pp. 7650–7658, Dec. 2023. [Online]. Available: https://doi.org/10.1109/TIT.2023.3291313
- [19] E. Agrell, D. Pook-Kolb, and B. Allen, "Glued lattices are better quantizers than K₁₂," *IEEE Trans. Inf. Theory*, vol. 70, no. 11, pp. 8414–8418, Nov. 2024. [Online]. Available: https://doi.org/10.1109/TIT.2024.3398421
- [20] D. Pook-Kolb, E. Agrell, and B. Allen, "Parametric lattices are better quantizers in dimensions 13 and 14," preprint, Nov. 2024. [Online]. Available: https://arxiv.org/abs/2411.19250
- [21] J. H. Conway and N. J. A. Sloane, "A lower bound on the average error of vector quantizers," *IEEE Trans. Inf. Theory*, vol. IT-31, no. 1, pp. 106–109, Jan. 1985. [Online]. Available: https://doi.org/10.1109/TIT.1985.1056993
- [22] —, "Fast quantizing and decoding algorithms for lattice quantizers and codes," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 227–232, Mar. 1982. [Online]. Available: https://doi.org/10.1109/TIT.1982.1056484
- [23] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002. [Online]. Available: https://doi.org/10.1109/TIT.2002.800499
- [24] A. Ghasemmehdi and E. Agrell, "Faster recursions in sphere decoding," IEEE Trans. Inf. Theory, vol. 57, no. 6, pp. 3530–3536, June 2011. [Online]. Available: https://doi.org/10.1109/TIT.2011.2143830
- [25] D. Pook-Kolb, E. Agrell, and B. Allen, "The Voronoi region of the Barnes–Wall lattice Λ₁₆," J. Sel. Areas Inf. Theory, vol. 4, pp. 16–23, 2023. [Online]. Available: https://doi.org/10.1109/JSAIT.2023.3276897
- [26] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, pp. 515–534, Dec. 1982. [Online]. Available: https://doi.org/10.1007/BF01457454
- [27] M. E. O'Neill, "PCG: a family of simple fast space-efficient statistically good algorithms for random number generation," Harvey Mudd College, Claremont, CA, Tech. Rep. HMC-CS-2014-0905, Sept. 2014. [Online]. Available: https://pcg-random.org/
- [28] R. E. A. C. Paley and N. Wiener, *Fourier Transforms in the Complex Domain*. Providence, RI: American Mathematical Society, 1934, vol. 19. [Online]. Available: https://doi.org/10.1090/coll/019
- [29] D. A. Harville, Matrix Algebra from a Statistician's Perspective. New York, NY: Springer, 1997. [Online]. Available: https://doi.org/10.1007/b98818
- [30] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Trans. Inf. Theory*, vol. 42, no. 4, pp. 1152–1159, July 1996. [Online]. Available: https://doi.org/10.1109/18.508838
- [31] A. A. El Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using simulated annealing to design good codes," *IEEE Trans. Inf. Theory*, vol. IT-33, no. 1, pp. 116–123, Jan. 1987. [Online]. Available: https://doi.org/10.1109/TIT.1987.1057277
- [32] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inf. Theory*, vol. 36, no. 4, pp. 799–809, July 1990. [Online]. Available: https://doi.org/10.1109/18.53739
- [33] L. Babai, "On Lovász' nearest lattice reduction and the lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, Mar. 1986. [Online]. Available: https://doi.org/10.1007/BF02579403
- [34] J. H. Conway and N. J. A. Sloane, "Laminated lattices," Annals of Mathematics, vol. 116, no. 3, pp. 593–620, Nov. 1982. [Online]. Available: https://doi.org/10.2307/2007025
- [35] W. Plesken and M. Pohst, "Constructing integral lattices with prescribed minimum. II," *Mathematics of Computation*, vol. 60, no. 202, pp. 817–825, Apr. 1993. [Online]. Available: https://doi.org/10.1090/S0025-5718-1993-1176715-1
- [36] D. Pook-Kolb, B. Allen, and E. Agrell, "Exact calculation of quantizer constants for arbitrary lattices," 2022–2024. [Online]. Available: https://arxiv.org/abs/2211.01987