Approximate Voronoi cells for lattices, revisited

Research Article

Thijs Laarhoven*

Approximate Voronoi cells for lattices, revisited

https://doi.org/10.1515/jmc-2020-0074
Received Jun 05, 2019; accepted Jul 01, 2019

Abstract: We revisit the approximate Voronoi cells approach for solving the closest vector problem with preprocessing (CVPP) on high-dimensional lattices, and settle the open problem of Doulgerakis–Laarhoven–De Weger [PQCrypto, 2019] of determining exact asymptotics on the volume of these Voronoi cells under the Gaussian heuristic. As a result, we obtain improved upper bounds on the time complexity of the randomized iterative slicer when using less than $2^{0.076d+o(d)}$ memory, and we show how to obtain time–memory trade-offs even when using less than $2^{0.048d+o(d)}$ memory. We also settle the open problem of obtaining a continuous trade-off between the size of the advice and the query time complexity, as the time complexity with subexponential advice in our approach scales as $d^{d/2+o(d)}$, matching worst-case enumeration bounds, and achieving the same asymptotic scaling as average-case enumeration algorithms for the closest vector problem.

Keywords: Voronoi cells, polytopes, volume estimation, lattices, closest vector problem

2010 Mathematics Subject Classification: 11H06, 52B11, 52C07, 94A60

1 Introduction

Ever since the discovery of polynomial-time quantum attacks on widely deployed public-key cryptosystems [36], researchers have been looking for ways to construct cryptographic schemes whose security relies on problems which remain hard even when large-scale quantum computers become a reality [8, 14, 29]. A prominent class of potentially "post-quantum" cryptosystems [2, 33, 38] relies on the hardness of lattice problems, such as the shortest (SVP) and closest vector problems (CVP). Understanding their hardness is essential for an efficient and reliable deployment of lattice-based cryptographic schemes in practice.

Over time, the practical hardness of SVP and CVP has been quite well studied, with two classes of algorithms emerging as the most competitive: *enumeration* [5, 6, 15, 16, 19, 27], running in superexponential time $2^{\Theta(d \log d)}$ in the lattice dimension d (the main security parameter), using a negligible amount of space; and *sieving* [3, 4, 13, 18, 20, 26, 28], running in only exponential time $2^{\Theta(d)}$, but also requiring an amount of memory scaling as $2^{\Theta(d)}$. The best asymptotic time complexities for enumeration ($d^{d/2e+o(d)}$ for SVP, $d^{d/2+o(d)}$ for CVP [17]) and sieving ($(3/2)^{d/2+o(d)}$ for both SVP and CVP [7, 21]) have remained unchanged since 2007 and 2016 respectively,[1] and recent work has mainly focused on decreasing second-order terms in the time and space complexities [4, 5, 13, 16, 22].

A close relative to CVP, the closest vector problem with preprocessing (CVPP), has received far less attention [1, 10, 24, 39] – from a practical point of view, only a few recent works have studied how preprocessing can be used to speed up CVP [12, 21]. Since a fast CVPP algorithm would imply faster lattice enumeration

The author is supported by a Veni grant from NWO under project number 016.Veni.192.005.

***Corresponding Author: Thijs Laarhoven:** Eindhoven University of Technology, Eindhoven, The Netherlands; Email: mail@thijs.com

1 This statement concerns classical complexities; for quantum complexities, see e.g. [6, 23].

algorithms for SVP/CVP [12, 16, 21], faster approximate-SVP algorithms for ideal lattices [30, 39], and even faster isogeny-based cryptography [9], a better understanding of the hardness of CVPP is needed.

1.1 Approximate Voronoi cells

A natural approach for solving nearest-point queries for large data sets is to use *Voronoi cells*; partitioning the space in regions, where each cell contains all points closer to the point in this cell than to any other point in the data set. Micciancio–Voulgaris [25] proposed an algorithm for constructing the Voronoi cell $\mathcal{V}$ of a lattice in time $2^{2d+o(d)}$ and space $2^{d+o(d)}$, which can then be used to solve CVPP in time $2^{2d+o(d)}$. Bonifas–Dadush [10] later improved the query time complexity to only $2^{d+o(d)}$, but with the best heuristic algorithms for CVP running in time and space less than $2^{0.3d+o(d)}$, using exact Voronoi cells seems impractical.

To make the Voronoi cells approach practical, Laarhoven [21] and Doulgerakis–Laarhoven–De Weger (DLW) [12] proposed constructing *approximate Voronoi cells* of the lattice, and using a randomized version of the iterative slicer algorithm of Sommer–Feder–Shalvi [37] for solving CVP queries. These cells $\mathcal{V}_L$, defined by a list of lattice vectors $L \subset \mathcal{L}$, can be seen as rough, low-memory approximations to the exact Voronoi cell $\mathcal{V}$ – low-quality representations of the same object, which attempt to model the object as well as possible within the limited space available. These approximate representations are lossy, but are also smaller and easier to store (less memory) and faster to process (less time).

For analyzing the performance of this approach, DLW conjectured a relation between the performance of the algorithm and how well $\mathcal{V}_L$ approximates $\mathcal{V}$:

$$p = \Pr(\text{the iterative slicer, with input } L, \text{ solves CVP}) \overset{?}{\approx} \frac{\mathrm{vol}(\mathcal{V})}{\mathrm{vol}(\mathcal{V}_L)}. \tag{1}$$

They then obtained upper bounds on the volume of $\mathcal{V}_L$ relative to $\mathcal{V}$ by studying the success probability of the randomized slicer. An open problem from DLW was to better study the volumes of these approximate Voronoi cells, as this may lead to tighter bounds on their CVPP algorithm. Furthermore, the time–space trade-offs from DLW seemed somewhat unnatural — the query time complexity diverges when the memory is less than $2^{0.05d+o(d)}$ — and a second open problem was to obtain time complexities scaling as $2^{\Theta(d)}$ for arbitrary memory complexities $2^{\Omega(d)}$.

1.2 Volumes of approximate Voronoi cells

In this paper we take a fundamental approach to studying the shape of approximate Voronoi cells. We model this problem as estimating the volume of the intersection of a large number of random half-spaces, and we solve the latter problem exactly for the main regimes of interest. In particular, without any heuristic assumptions, we prove the following result regarding the volume of a random polytope obtained by intersecting a large number of random half-spaces. Assuming that the distribution of lattice points inside a large ball can be approximated well by a uniform distribution over the ball, this then leads to a tight asymptotic estimate of the volume of approximate Voronoi cells.

Theorem 1.1 (Volume of approximate Voronoi cells) Let $\alpha > 1$, and let $L \subset \mathcal{L} \setminus \{\mathbf{0}\}$ consist of the α^d shortest non-zero vectors of a lattice $\mathcal{L}$. Then, assuming the Gaussian heuristic holds, with probability $1 - o(1)$ we have:

$$\alpha \le \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = \left(\frac{\alpha^4}{4\alpha^2 - 4}\right)^{d/2+o(d)} \mathrm{vol}(\mathcal{V}); \tag{2}$$

$$\alpha \ge \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = (1 + o(1))^{d+o(d)}\, \mathrm{vol}(\mathcal{V}). \tag{3}$$

Assuming [12, Heuristic assumption 1] holds (which has been restated here as Heuristic 4.2), this result would then imply what are the exact asymptotic time and space complexities of the randomized slicer. However,

under the same assumption, DLW derived the following asymptotic upper bound on the relative volume of approximate Voronoi cells, for $\alpha \in (1, \sqrt{2})$:

$$\frac{\mathrm{vol}(\mathcal{V}_L)}{\mathrm{vol}(\mathcal{V})} \overset{?}{\leq} \left(\frac{16\alpha^4 \left(\alpha^2 - 1\right)}{-9\alpha^8 + 64\alpha^6 - 104\alpha^4 + 64\alpha^2 - 16} \right)^{d/2+o(d)}. \tag{4}$$

Looking closely, (2) in fact *contradicts* the above upper bound for $\alpha > \frac{1}{3}\sqrt{10} \approx 1.054$. The source of this contradiction can be found in [12, Heuristic assumption 1]: while this assumption states that the success probability p of the randomized slicer is *exactly* $p = \mathrm{vol}(\mathcal{V})/\mathrm{vol}(\mathcal{V}_L)$, the randomized slicer is in fact *more likely* to converge to short solutions than to long solutions: we may well have $p \gg \mathrm{vol}(\mathcal{V})/\mathrm{vol}(\mathcal{V}_L)$, and the gap between both quantities may be exponentially large in d. A lower bound on p therefore does not necessarily translate to a lower bound on $\mathrm{vol}(\mathcal{V})/\mathrm{vol}(\mathcal{V}_L)$, or to an upper bound on its reciprocal.

1.3 Application to CVPP

Although (4) is incorrect as an upper bound on the volume of approximate Voronoi cells, on closer inspection we see that to bound the complexity of their algorithm, DLW in fact proved that p is at most the RHS of (4): the bound on the volume of the approximate Voronoi cell was then only obtained through transitivity by applying [12, Heuristic assumption 1]. Thus, letting p_α denote the success probability of the randomized slicer when using a list of the $n = \alpha^d$ shortest non-zero vectors in the lattice, we now have two heuristic lower bounds on p_α:

$$\text{(DLW)} \qquad p_\alpha \geq \left(\frac{-9\alpha^8 + 64\alpha^6 - 104\alpha^4 + 64\alpha^2 - 16}{16\alpha^4 \left(\alpha^2 - 1\right)} \right)^{d/2+o(d)}; \tag{5}$$

$$\text{(ours)} \qquad p_\alpha \geq \left(\frac{4\alpha^2 - 4}{\alpha^4} \right)^{d/2+o(d)}. \tag{6}$$

These bounds are both conditional on the Gaussian heuristic, and the second result holds conditional on $p_\alpha \geq \mathrm{vol}(\mathcal{V})/\mathrm{vol}(\mathcal{V}_L)$. By applying similar techniques from [12], we obtain the following CVPP complexities, where $\delta = \sqrt{\alpha^2 - 1}/\alpha$.

Theorem 1.2 (CVPP complexities) Let $\alpha \in (1, \sqrt{2})$ and $u \in (\delta, \frac{1}{\delta})$. Then we can heuristically solve CVPP with query space and time S and T, where:

$$\mathrm{S} = \left(\frac{\alpha}{\alpha - (\alpha^2 - 1)(\alpha u^2 - 2u\sqrt{\alpha^2 - 1} + \alpha)} \right)^{d/2+o(d)}, \tag{7}$$

$$\mathrm{T} = \left(\frac{\alpha^4}{4\alpha^2 - 4} \cdot \frac{\alpha + u\sqrt{\alpha^2 - 1}}{-\alpha^3 + \alpha^2 u\sqrt{\alpha^2 - 1} + 2\alpha} \right)^{d/2+o(d)}. \tag{8}$$

The best query complexities (S, T) together form the blue curve in Figure 1.

As we can see in Figure 1, for the low-memory regime of less than $2^{0.076d+o(d)}$ memory, we obtain strictly better query time complexities than [12]. The trade-offs from [12] were further limited to the regime of using at least $2^{0.048d+o(d)}$ memory, whereas Theorem 1.2 describes a continuous trade-off between the query time and space complexities: for arbitrary memory complexities $2^{\varepsilon d+o(d)}$ with $\varepsilon > 0$, we obtain a query time complexity $2^{\Theta(d)}$. Extending Theorem 1.2 to the regime of $\alpha = 1 + o(1)$, we obtain the following result.

Corollary 1.3 (Polynomial advice for CVPP). *Using $d^{\Theta(1)}$ memory, we can heuristically solve CVPP in time $d^{d/2+o(d)}$.*

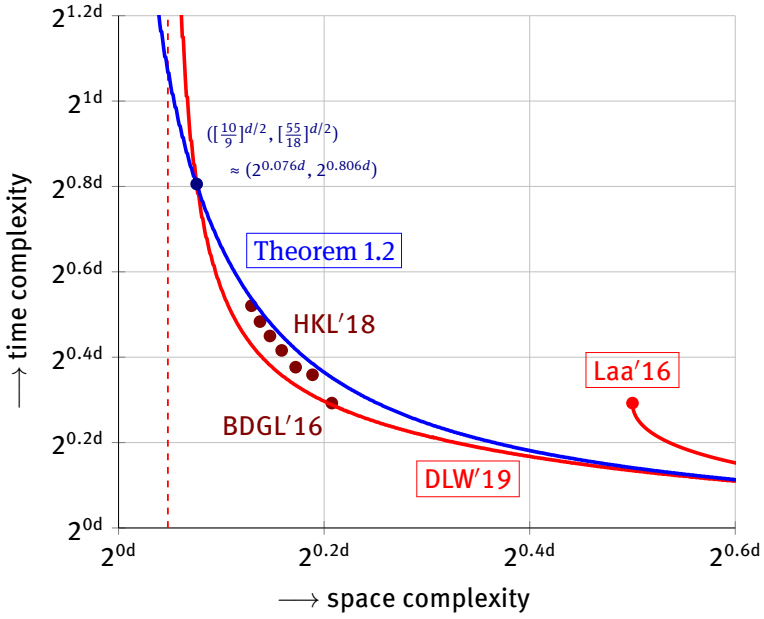

Figure 1: Query complexities for solving CVPP. The labeled curves and points correspond to the papers [7, 12, 18, 21]. Our new upper bound on the query time complexity improves upon DLW when using less than $(10/9)^{d/2+o(d)} \approx 2^{0.076d+o(d)}$ memory. Note that, whereas the red DLW-curve diverges as the memory approaches the dashed asymptote $2^{0.048d+o(d)}$ from above, our trade-offs heuristically continue all the way to the regime of subexponential memory.

This matches the asymptotic worst-case time complexities for solving CVP with enumeration of Hanrot–Stehlé [17], and with an average-case scaling for enumeration of $d^{d/(2e)+o(d)}$, this is only off by a factor $1/e$ in the exponent compared to practical enumeration methods. We further see that if we use a preprocessed list of size e.g. $2^{\Theta(d^\gamma)}$ for constant $\gamma \in (0, 1)$, we heuristically obtain a CVPP time complexity scaling as $2^{\frac{1}{2}(1-\gamma)d \log_2 d + o(d \log d)}$.

Outline.
Section 2 first defines notation and preliminary results. Section 3 studies the volume of intersections of random halfspaces. Section 4 describes the application of these results to solving CVPP and the resulting trade-offs. The appendices describe further details on prior work, to make the paper self-contained.

2 Preliminaries

Given a set $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d\} \subset \mathbb{R}^d$ of linearly independent vectors, we define $\mathcal{L} = \mathcal{L}(\boldsymbol{B}) := \{\sum_{i=1}^d \lambda_i \boldsymbol{b}_i : \boldsymbol{\lambda} \in \mathbb{Z}^d\}$ as the lattice generated by $\boldsymbol{B}$. We write $\|\cdot\|$ for the Euclidean norm. Given a basis of a lattice and a target vector $\boldsymbol{t} \in \mathbb{R}^d$, the closest vector problem (CVP) is to find the vector $\boldsymbol{v} \in \mathcal{L}$ closest to $\boldsymbol{t}$. In the preprocessing version (CVPP), the problem is split into two parts: the preprocessing phase (without knowing $\boldsymbol{t}$) and the query phase (with knowledge of $\boldsymbol{t}$). For CVPP, the task is to do preprocessing such that CVP queries can then be answered more efficiently than when solving CVP directly.

Let us define some basic high-dimensional objects below, where $\boldsymbol{v} \in \mathbb{R}^d$.

$$\text{(unit sphere)} \qquad \mathcal{S} := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| = 1\}, \tag{9}$$

$$\text{(unit ball)} \qquad \mathcal{B} := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq 1\}, \tag{10}$$

$$\text{(half-space)} \qquad \mathcal{H}_{\boldsymbol{v}} := \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq \|\boldsymbol{x} - \boldsymbol{v}\|\}, \tag{11}$$

$$\text{(convex polytope)} \qquad \mathcal{V}_L := \bigcap_{\boldsymbol{v}\in L} \mathcal{H}_{\boldsymbol{v}}, \qquad (\mathbf{0} \notin L) \tag{12}$$

$$\text{(spherical cap)} \qquad \mathcal{C}_{\boldsymbol{v}} := \overline{\mathcal{H}_{\boldsymbol{v}}} \cap \mathcal{B}, \tag{13}$$

$$\text{(Voronoi cell)} \qquad \mathcal{V} := \mathcal{V}_{\mathcal{L}\setminus\{\mathbf{0}\}}. \tag{14}$$

We further define the complements $\overline{\mathcal{H}_{\boldsymbol{v}}} := \mathbb{R}^d \setminus \mathcal{H}_{\boldsymbol{v}}$ and $\overline{\mathcal{V}_L} := \mathbb{R}^d \setminus \mathcal{V}_L$ in $\mathbb{R}^d$, and $\overline{\mathcal{C}_{\boldsymbol{v}}} := \mathcal{B} \setminus \mathcal{C}_{\boldsymbol{v}}$ on the ball. Note that the definition of a polytope $\mathcal{V}_L$ is generic, and the list L need not be from a lattice. $\mathcal{V}_L$ may further be unbounded (and its volume may be infinite), although for sufficiently large, randomly chosen lists L it will usually be finite. For $L \subset \mathcal{L} \setminus \{\mathbf{0}\}$, the polytope $\mathcal{V}_L$ defines an *approximate Voronoi cell* of the lattice $\mathcal{L}$ [12], satisfying $\mathcal{V} \subseteq \mathcal{V}_L$ with equality iff $\mathcal{R} \subseteq L$, where $\mathcal{R}$ is the set of *relevant vectors* of the lattice [25].

To analyze volumes of intersections on the ball, we will use the following asymptotic formula [34, Equation (28)], where $\alpha = \frac{1}{2}\|\boldsymbol{v}\| \in (0, 1)$:

$$C(\alpha) := \frac{\mathrm{vol}(\mathcal{C}_{\boldsymbol{v}})}{\mathrm{vol}(\mathcal{B})} \sim \sqrt{\frac{1-\alpha^2}{2\pi\alpha^2 d}} \cdot (1-\alpha^2)^{d/2}. \qquad (d \to \infty) \tag{15}$$

For constant $\alpha \in (0, 1)$ and large d, Equation (15) can alternatively be written as $C(\alpha) = O((1-\alpha^2)^{d/2}/\sqrt{d}) = (1-\alpha^2)^{d/2+o(d)}$.

Finally, the *Gaussian heuristic* states that for sufficiently smooth and random regions $\mathcal{K} \subset \mathbb{R}^d$, the number of lattice points inside $\mathcal{K}$ scales as $\mathrm{vol}(\mathcal{K})/\mathrm{vol}(\mathcal{V})$.

3 Volumes of random polytopes

To study the asymptotic behavior of volumes of approximate Voronoi cells, we will first study the more fundamental problem of estimating the volume of polytopes $\mathcal{V}_L$ defined as the intersection of a large number of random half-spaces. We will study two specific cases for the list L below:

1. Uniformly random points from the (unit) sphere;
2. Uniformly random points from the (unit) ball.

The volume of such random polytopes has been previously studied in e.g. [31, 32, 35, 40, 41], and in particular the case of points from the sphere was analyzed in [31]. For the application to approximate Voronoi cells we need bounds for the case when points are drawn uniformly at random from a ball, which to the best of our knowledge has not been explicitly studied before. For completeness, and to illustrate how the analysis changes between the case of the sphere and the ball, we treat the case of random points from the unit sphere here as well.

3.1 Uniformly random points from the (unit) sphere

First, let us study the case where L is sampled uniformly at random from the unit sphere $\mathcal{S}$. This setting was previously studied in [31, Section 3.2], but for extending the analysis to the case of the unit ball we explicitly analyze this problem here as well. Note that for $L \subseteq \mathcal{S}^{d-1}$ we have the trivial lower bound $\mathrm{vol}(\mathcal{V}_L) \geq 2^{-d}\,\mathrm{vol}(\mathcal{B})$, as $\frac{1}{2}\mathcal{B} \subseteq \mathcal{V}_L$. For a slightly less trivial upper bound, note that the polytope $\mathcal{V}_L$ is unbounded iff all points in L lie in a certain hemisphere. The probability that this happens was computed by Wendel [42] as:

$$\Pr_{L\sim\mathcal{S}}\Big(\mathrm{vol}(\mathcal{V}_L) < \infty\Big) = 1 - 2^{-n+1}\sum_{k=0}^{d-1}\binom{n-1}{k}. \tag{16}$$

In particular, it is extremely unlikely that for lists of size $n = \omega(d)$, the corresponding polytopes are unbounded. For lists of exponential size, we obtain the following result, similar to [31, Theorem 3.9].

Theorem 3.1 (Random points from the sphere) Let $\alpha > 1$, and let $L \subset \mathcal{S}$ consist of $n = \alpha^d$ uniformly random vectors from $\mathcal{S}$. Then, with probability $1 - o(1)$ over the randomness of L, we have:

$$\mathrm{vol}(\mathcal{V}_L) = \left(\frac{\alpha^2}{4\alpha^2 - 4}\right)^{d/2+o(d)} \mathrm{vol}(\mathcal{B}). \tag{17}$$

Proof. To prove Theorem 3.1, we will prove the following, equivalent statement:

$$\mathrm{vol}(\mathcal{V}_L) = \mathrm{vol}(r_0\mathcal{B})^{1+o(1)}, \qquad r_0 = \sqrt{\frac{\alpha^2}{4\alpha^2 - 4}}\,. \tag{18}$$

Note that $\mathrm{vol}(r\mathcal{B}) = r^d \mathrm{vol}(\mathcal{B})$ for arbitrary r, hence the equivalence. Below we will further use the quantity $\mathcal{V}_L^{(r)} = \mathcal{V}_L \cap r\mathcal{B} \subseteq \mathcal{V}_L$ as the intersection of the polytope with the ball of radius $r > 0$. Observe that for sufficiently small $r \ll r_0$ we have $\mathcal{V}_L^{(r)} = r\mathcal{B} \subset \mathcal{V}_L$ while for large $r \gg r_0$ we have $\mathcal{V}_L^{(r)} = \mathcal{V}_L \subset r\mathcal{B}$. The quantity r_0 is intuitively the radius r for which $\mathrm{vol}(\mathcal{V}_L^{(r)}) \approx \mathrm{vol}(\mathcal{V}_L) \approx \mathrm{vol}(r\mathcal{B})$.

First, some simple manipulations give:

$$\mathcal{V}_L^{(r)} = \bigcap_{\boldsymbol{v}\in L} \mathcal{H}_{\boldsymbol{v}} \cap (r\mathcal{B}) = \bigcap_{\boldsymbol{v}\in L} \left(r\mathcal{B} \setminus r\mathcal{C}_{\boldsymbol{v}/r}\right) = \underbrace{r\left(\mathcal{B} \setminus \bigcup_{\boldsymbol{v}\in L} \mathcal{C}_{\boldsymbol{v}/r}\right)}_{\mathcal{K}}. \tag{19}$$

Note that the vectors $\boldsymbol{v}/r$ all have norm $1/r$, and the spherical caps $\mathcal{C}_{\boldsymbol{v}/r}$ thus have a fixed base radius of $1/(2r)$. To prove the lower bound on $\mathrm{vol}(\mathcal{V}_L)$, we will use elementary volume arguments to argue that $\mathrm{vol}(\mathcal{K}) \approx \mathrm{vol}(\mathcal{B})$. For the upper bound, we have $\mathrm{vol}(\mathcal{K}) \le \mathrm{vol}(\mathcal{B})$, and we will argue that with high probability over the randomness of L, $\mathrm{vol}(\mathcal{V}_L) \approx \mathrm{vol}(\mathcal{V}_L^{(r)})$.

Lower bound ($\ge$): Ignoring spherical cap intersections, we have:

$$\mathrm{vol}(\mathcal{K}) \ge \mathrm{vol}(\mathcal{B}) - n \cdot \mathrm{vol}(\mathcal{C}_{\boldsymbol{v}/r}) = \mathrm{vol}(\mathcal{B})\left[1 - \alpha^d\left(1 - \tfrac{1}{4r^2}\right)^{d/2+o(d)}\right]. \tag{20}$$

For $1/\alpha^2 = 1 - 1/(4r^2) + o(1)$, or equivalently $r = r_0 - o(1)$, we thus get $\mathrm{vol}(\mathcal{K}) \ge (1 - o(1)) \cdot \mathrm{vol}(\mathcal{B})$.

Upper bound ($\le$): Clearly $\mathrm{vol}(\mathcal{V}_L^{(r)}) = r^d \mathrm{vol}(\mathcal{K}) \le r^d \mathrm{vol}(\mathcal{B})$; the difficulty lies in showing that $\mathrm{vol}(\mathcal{V}_L) \approx \mathrm{vol}(\mathcal{V}_L^{(r)})$. Note that when n is large, then the spherical caps in (19) will cover (almost) the entire surface of $\mathcal{B}$ – if e.g. only a fraction $2^{-\Theta(d^2)}$ of the sphere remains uncovered, then the parts of $\mathcal{V}_L$ extending beyond $r\mathcal{B}$ will contribute a negligible amount to the volume of $\mathcal{V}_L$.

Given a point on $\mathcal{S}$, the probability of it not being covered by one of n spherical caps $\mathcal{C}_{\boldsymbol{v}/r}$ is given by $[1 - \mathrm{vol}(\mathcal{C}_{\boldsymbol{v}/r})/\mathrm{vol}(\mathcal{B})]^n$. For $n = \mathrm{vol}(\mathcal{B})/\mathrm{vol}(\mathcal{C}_{\boldsymbol{v}/r})$, this can be upper bounded by $1/e$, hence for $n = 2d^2\,\mathrm{vol}(\mathcal{B})/\mathrm{vol}(\mathcal{C}_{\boldsymbol{v}/r})$ the expected quantity not covered on the sphere is at most e^{-2d^2}. By Markov's inequality, the probability that more than a fraction e^{-d^2} of the sphere is covered is at most $e^{-2d^2+d^2} = e^{-d^2}$, and so the upper bound follows. □

3.2 Uniformly random points from the (unit) ball

As sampling from $\mathcal{B}$ and $\mathcal{S}$ is similar in high-dimensional spaces (almost all the volume of the ball is concentrated near the surface of the sphere), in most cases the asymptotics for the unit sphere and the unit ball are the same. However, when n is very large, a significant number of vectors will have norm significantly less than 1, and these will then determine the shape of the resulting polytope.

The following main result shows that if $n \gg 2^{d/2}$, then the volume of the Voronoi cell for $\mathbf{0}$ scales like $\mathrm{vol}(\mathcal{B})/n$. Note that $\mathcal{V}_L$ can be seen as the Voronoi cell for $\mathbf{0}$ in the data set $L \cup \{\mathbf{0}\}$, and for $n \gg 2^{d/2}$ the Voronoi cell of the $\mathbf{0}$-vector is therefore no larger than the Voronoi cells of the other n points in the ball – each of the points covers an equal fraction $\mathrm{vol}(\mathcal{B})/n$ of the ball. For small n, the portion of the ball covered by $\mathbf{0}$ is an exponential factor larger than the average.

Theorem 3.2 (Random points from the unit ball) Let $\alpha > 1$, and let $L \subset \mathcal{B}$ consist of $n = \alpha^d$ uniformly random vectors from $\mathcal{B}$. Then, with probability $1 - o(1)$ over the randomness of L, we have:

$$\alpha \leq \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = \left(\frac{\alpha^2}{4\alpha^2 - 4}\right)^{d/2+o(d)} \mathrm{vol}(\mathcal{B}); \tag{21}$$

$$\alpha \geq \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = \left(\frac{1}{\alpha^2}\right)^{d/2+o(d)} \mathrm{vol}(\mathcal{B}). \tag{22}$$

Proof. For $\gamma < 1$ close to 1, let us divide the set L into sets $L_i = \{\boldsymbol{v} \in L : \gamma^i \leq \|\boldsymbol{v}\| \leq \gamma^{i-1}\}$, for $i = 0, 1, \ldots$, i.e. we partition L_i according to a sequence of thin spherical shells. With high probability over the randomness of L, each of these lists L_i will contain $(\gamma^i\alpha)^{d+o(d)}$ vectors. The original polytope can now equivalently be described as $\mathcal{V}_L = \bigcap_{i=0}^{\infty} \mathcal{V}_{L_i}$. To estimate the volume of $\mathcal{V}_L$, note that by Theorem 3.1, each of these cells $\mathcal{V}_{L_i}$ is roughly shaped like a ball of a certain radius r_i. As a result, the volume of $\mathcal{V}_L$ is determined by the smallest radius $\min_{i\in\mathbb{N}} r_i$ of these balls, corresponding to one of the lists L_i.

To find the list L_i defining the smallest polytope, recall that by applying Theorem 3.1 with $n_i = (\gamma^i\alpha)^{d+o(d)}$ vectors to a sphere of radius γ^i, we have the following relation, where $\beta = \gamma^{2i}$:

$$\mathrm{vol}(\mathcal{V}_{L_i}) = \left(\tfrac{\alpha^2\gamma^{2i}}{4\alpha^2\gamma^{2i}-4}\right)^{d/2+o(d)} \mathrm{vol}(\gamma^i\mathcal{B}) = \underbrace{\left(\tfrac{\beta^2\alpha^2}{4\beta\alpha^2-4}\right)}_{f(\beta)}^{d/2+o(d)} \mathrm{vol}(\mathcal{B}). \tag{23}$$

To find the value β resulting in the smallest radius, note that the derivative of $f(\beta)$ satisfies $f'(\beta) = -\beta\alpha^2(2 - \beta\alpha^2)/4(\beta\alpha^2-1)^2$, which is negative for small $\beta < 2/\alpha^2$, i.e. $f(\beta)$ is decreasing with β, and the volume of the $\mathcal{V}_{L_i}$ increases with i. Now $f'(\beta) = 0$ has one solution at $\beta = 2/\alpha^2$, which is attained by one of the lists L_i iff $\alpha \geq \sqrt{2}$. In the regime $\alpha < \sqrt{2}$, the smallest radius is obtained for the first list L_0, resulting in the same bound as in Theorem 3.1, while for $\alpha \geq \sqrt{2}$ the non-trivial minimum value lies at $\beta = \gamma^{2i} = 2/\alpha^2$, resulting in $f(\beta) = 1/\alpha^2$ and $\mathrm{vol}(\mathcal{V}_L) = \alpha^{-d+o(d)} \mathrm{vol}(\mathcal{B})$. □

Let us finally state separately what happens when we draw points uniformly at random from a ball of a different radius. This directly follows from Theorem 3.2.

Corollary 3.3 (Random points from the β-ball)**.** *Let $\alpha > 1$, and let $L \subset \mathcal{B}$ consist of $n = \alpha^d$ uniformly random vectors from $\beta \cdot \mathcal{B}$. Then, with probability $1 - o(1)$ over the randomness of L, we have:*

$$\alpha \leq \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = \left(\frac{\alpha^2\beta^2}{4\alpha^2 - 4}\right)^{d/2+o(d)} \mathrm{vol}(\mathcal{B}); \tag{24}$$

$$\alpha \geq \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = \left(\frac{\beta^2}{\alpha^2}\right)^{d/2+o(d)} \mathrm{vol}(\mathcal{B}). \tag{25}$$

Proof. Relative to the β-ball, we have $\mathrm{vol}(\mathcal{V}_L) = r^{d+o(d)} \mathrm{vol}(\beta\mathcal{B})$ with r as in Theorem 3.2. Noting that $\mathrm{vol}(\beta\mathcal{B}) = \beta^d \mathrm{vol}(\mathcal{B})$, the result follows. □

4 Approximate Voronoi cells, revisited

With the results from Section 3, we can immediately deduce asymptotics for the volume of approximate Voronoi cells, where these results can now be derived using only the Gaussian heuristic, which has been used and verified on far more occasions than [12, Heuristic 1].[2]

2 By applying the Gaussian heuristic to balls of different radii, we can derive the density of norms of lattice vectors, while spherical symmetry of the distribution of lattice vectors then implies that the lattice vectors inside a ball must follow a uniform distribution.

Corollary 4.1 (Points from a lattice)**.** *Let $\alpha > 1$, and let $L \subset \mathcal{L} \setminus \{\mathbf{0}\}$ consist of the α^d shortest non-zero vectors of a lattice $\mathcal{L}$. Then, assuming the Gaussian heuristic holds, with probability $1 - o(1)$ we have:*

$$\alpha \le \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = \left(\frac{\alpha^4}{4\alpha^2 - 4}\right)^{d/2+o(d)} \mathrm{vol}(\mathcal{V}); \tag{26}$$

$$\alpha \ge \sqrt{2} \implies \mathrm{vol}(\mathcal{V}_L) = (1 + o(1))^{d/2+o(d)} \mathrm{vol}(\mathcal{V}). \tag{27}$$

Proof. Without loss of generality, suppose that $\mathrm{vol}(\mathcal{V}) = \mathrm{vol}(\mathcal{B})$. Under the Gaussian heuristic, the points L are then essentially uniformly distributed in the ball of radius α. Applying Corollary 3.3 with $\alpha = \beta$, the result then follows. □

4.1 Heuristic assumptions

Assuming that [12, Heuristic assumption 1] holds, as discussed in the introduction this would give us tight bounds on the success probability of the randomized iterative slicer from [12]. However, these results would then contradict the claimed lower bound on the success probability from [12, Equation (37)]. The source of this contradiction is [12, Heuristic assumption 1], which reads as follows.[3]

Heuristic assumption 4.2 (Randomized slicing, DLW) For $L \subset \mathcal{L}$ and large s,

$$\Pr_{\boldsymbol{t}' \sim D_{\boldsymbol{t}+\mathcal{L},s}} \left[\mathrm{Slice}_L(\boldsymbol{t}') \in \mathcal{V} \right] \approx \frac{\mathrm{vol}(\mathcal{V})}{\mathrm{vol}(\mathcal{V}_L)}. \tag{28}$$

In fact, the randomized slicer is biased towards finding as short solutions as possible, and the probability of returning the unique representative from $\mathcal{V}$ may be much larger than $\mathrm{vol}(\mathcal{V})/\mathrm{vol}(\mathcal{V}_L)$. We therefore propose using the following heuristic assumption instead:

Heuristic assumption 4.3 (Randomized slicing, new) For $L \subset \mathcal{L}$ and large s,

$$\Pr_{\boldsymbol{t}' \sim D_{\boldsymbol{t}+\mathcal{L},s}} \left[\mathrm{Slice}_L(\boldsymbol{t}') \in \mathcal{V} \right] \gtrsim \frac{\mathrm{vol}(\mathcal{V})}{\mathrm{vol}(\mathcal{V}_L)}. \tag{29}$$

To motivate this new assumption, consider the reverse process of starting at the sliced solution vector $\boldsymbol{t}'' = \mathrm{Slice}_L(\boldsymbol{t}')$, and adding lattice vectors of length at most $\alpha\lambda_1(\mathcal{L})$ to obtain longer and longer vectors in the coset $\boldsymbol{t} + \mathcal{L}$. Now, given an initial sampled vector $\boldsymbol{t}' \sim D_{\boldsymbol{t}+\mathcal{L},s}$, the probability of reaching $\boldsymbol{t}''$ out of all possible solution vectors in $\boldsymbol{t} + \mathcal{L}$ is essentially proportional to the number of paths from $\boldsymbol{t}''$ to $\boldsymbol{t}'$ through the above process of adding lattice vectors of length at most $\alpha\lambda_1(\mathcal{L})$ to $\boldsymbol{t}''$. Starting from a shorter vector, the tree of potential paths to $\boldsymbol{t}''$ is likely to be wider, and there are likely more such paths reaching $\boldsymbol{t}''$.

Assuming that indeed, the success probability is *at least* proportional to the ratio of these volumes, we obtain the CVPP complexities described in Theorem 1.2 in the introduction. Here we simply replaced the upper bound on p_α from [12] by the upper bound obtained via the volume of approximate Voronoi cells, and otherwise applied the same techniques of nearest neighbor speed-ups.

4.2 The low-memory regime

As Theorem 1.2 describes complexities even for the regime of $2^{\varepsilon d+o(d)}$ memory with small ε, let us study the asymptotic behavior as the memory is actually subexponential or even polynomial in d.

3 For details and definitions of $D_{\boldsymbol{t}+\mathcal{L},s}$ and $\mathrm{Slice}_L(\boldsymbol{t}')$, we refer the reader to [12].

First, note that for the lower bound on the volume, we essentially only needed Equation (15), which holds even when $\alpha = o(1)$ scales with d. (See also [31, Lemmas 4.1 and 4.2] for absolute bounds.) For the upper bounds, we needed that the list L properly covers the sphere, and we argued that $n = 2d^2 \operatorname{vol}(\mathcal{B})/\operatorname{vol}(\mathcal{C}_{\boldsymbol{v}})$ suffices to cover enough of the sphere with high probability. We can therefore extend these results all the way up to the regime of polynomial space. Note that for small $\alpha = 1 + \varepsilon$, Theorem 3.2 gives:

$$\operatorname{vol}(\mathcal{V}_L) = \left(\frac{1}{\sqrt{8\varepsilon}} + O(\sqrt{\varepsilon})\right)^{d+o(d)} \operatorname{vol}(\mathcal{B}). \tag{30}$$

Substituting suitable values of α, we get the following results.

Proposition 4.4 (Polynomially many points from the unit ball)**.** *Let $L \subset \mathcal{B}$ consist of $n = d^{\Theta(1)}$ uniformly random vectors from $\mathcal{B}$. Then, with probability $1-o(1)$ over the randomness of L, we have* $\operatorname{vol}(\mathcal{V}_L) = 2^{\frac{1}{2}d\log_2 d+o(d\log d)} \operatorname{vol}(\mathcal{B})$.

Proof. This follows from substituting $\alpha = d^{\Theta(1/d)} = 1 + \Theta(\log d)/d$. □

In the application of CVPP algorithms, Proposition 4.4 shows that heuristically, we obtain a smooth trade-off between enumeration and using exact Voronoi cells – Hanrot–Stehlé [17, Theorem 4] previously showed that enumeration has a cost of $d^{d/2+o(d)}$ time for solving CVP in the worst case, with polynomial memory.

Proposition 4.5 (Subexponentially many points from the unit ball)**.** *Let $L \subset \mathcal{B}$ consist of $n = 2^{\Theta(d^\gamma)}$ uniformly random vectors from $\mathcal{B}$. Then, with probability $1 - o(1)$ over the randomness of L, we have* $\operatorname{vol}(\mathcal{V}_L) = 2^{\frac{1}{2}(1-\gamma)d\log_2 d+o(d\log d)} \operatorname{vol}(\mathcal{B})$.

Proof. This follows from substituting $\alpha = \exp\Theta(d^{\gamma-1}) = 1 + \Theta(d^{\gamma-1})$. □

This matches results from e.g. [11]. To illustrate Proposition 4.5 with an example, we expect to be able to solve CVPP with query time $d^{d/4+o(d)}$ when using $2^{\Theta(\sqrt{d})}$ memory, or we can match the average-case complexity of enumeration with a query time complexity of $d^{d/(2e)+o(d)}$ using $2^{\Theta(d^{1-1/e})} \approx 2^{\Theta(d^{0.63})}$ memory.

Acknowledgement: The author thanks Léo Ducas for insightful discussions on the topic of approximate Voronoi cells.

References

[1] Dorit Aharonov and Oded Regev, Lattice problems in NP ∩ coNP, in: *FOCS*, pp. 362–371, 2004.
[2] Miklós Ajtai and Cynthia Dwork, A public-key cryptosystem with worst-case/average-case equivalence, in: *STOC*, pp. 284–293, 1997.
[3] Miklós Ajtai, Ravi Kumar and Dandapani Sivakumar, A sieve algorithm for the shortest lattice vector problem, in: *STOC*, pp. 601–610, 2001.
[4] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn Postlethwaite and Marc Stevens, The general sieve kernel and new records in lattice reduction, in: *EUROCRYPT*, pp. 717–746, 2019.
[5] Yoshinori Aono and Phong Q. Nguyên, Random sampling revisited: lattice enumeration with discrete pruning, in: *EUROCRYPT*, pp. 65–102, 2017.
[6] Yoshinori Aono, Phong Q. Nguyen and Yixin Shen, Quantum lattice enumeration and tweaking discrete pruning, in: *ASIACRYPT*, pp. 405–434, 2018.
[7] Anja Becker, Léo Ducas, Nicolas Gama and Thijs Laarhoven, New directions in nearest neighbor searching with applications to lattice sieving, in: *SODA*, pp. 10–24, 2016.
[8] Daniel J. Bernstein, Johannes Buchmann and Erik Dahmen (eds.), *Post-quantum cryptography*, Springer, 2009.
[9] Ward Beullens, Thorsten Kleinjung and Frederik Vercauteren, CSI-FiSh: Efficient isogeny based signatures through class group computations, *Cryptology ePrint Archive, Report 2019/498* (2019).
[10] Nicolas Bonifas and Daniel Dadush, Short paths on the Voronoi graph and the closest vector problem with preprocessing, in: *SODA*, pp. 295–314, 2015.

[11] Daniel Dadush, Oded Regev and Noah Stephens-Davidowitz, On the closest vector problem with a distance guarantee, in: *CCC*, pp. 98–109, 2014.
[12] Emmanouil Doulgerakis, Thijs Laarhoven and Benne de Weger, Finding closest lattice vectors using approximate Voronoi cells, in: *PQCrypto*, 2019.
[13] Léo Ducas, Shortest vector from lattice sieving: a few dimensions for free, in: *EUROCRYPT*, pp. 125–145, 2018.
[14] The European Telecommunications Standards Institute (ETSI), *Quantum-Safe Cryptography*, 2019.
[15] Ulrich Fincke and Michael Pohst, Improved methods for calculating vectors of short length in a lattice, *Mathematics of Computation* **44** (1985), 463–471.
[16] Nicolas Gama, Phong Q. Nguyên and Oded Regev, Lattice enumeration using extreme pruning, in: *EUROCRYPT*, pp. 257–278, 2010.
[17] Guillaume Hanrot and Damien Stehlé, Improved analysis of Kannan's shortest lattice vector algorithm, in: *CRYPTO*, pp. 170–186, 2007.
[18] Gottfried Herold, Elena Kirshanova and Thijs Laarhoven, Speed-ups and time-memory trade-offs for tuple lattice sieving, in: *PKC*, pp. 407–436, 2018.
[19] Ravi Kannan, Improved algorithms for integer programming and related lattice problems, in: *STOC*, pp. 193–206, 1983.
[20] Thijs Laarhoven, Sieving for shortest vectors in lattices using angular locality-sensitive hashing, in: *CRYPTO*, pp. 3–22, 2015.
[21] Thijs Laarhoven, Sieving for closest lattice vectors (with preprocessing), in: *SAC*, pp. 523–542, 2016.
[22] Thijs Laarhoven and Artur Mariano, Progressive lattice sieving, in: *PQCrypto*, pp. 292–311, 2018.
[23] Thijs Laarhoven, Michele Mosca and Joop van de Pol, Finding shortest lattice vectors faster using quantum search, *Designs, Codes and Cryptography* **77** (2015), 375–400.
[24] Daniele Micciancio, The hardness of the closest vector problem with preprocessing, *IEEE Transactions on Information Theory* **47** (2001), 1212–1215.
[25] Daniele Micciancio and Panagiotis Voulgaris, A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations, in: *STOC*, pp. 351–358, 2010.
[26] Daniele Micciancio and Panagiotis Voulgaris, Faster exponential time algorithms for the shortest vector problem, in: *SODA*, pp. 1468–1480, 2010.
[27] Daniele Micciancio and Michael Walter, Fast lattice point enumeration with minimal overhead, in: *SODA*, pp. 276–294, 2015.
[28] Phong Q. Nguyên and Thomas Vidick, Sieve algorithms for the shortest vector problem are practical, *Journal of Mathematical Cryptology* **2** (2008), 181–207.
[29] The National Institute of Standards and Technology (NIST), *Post-Quantum Cryptography*, 2017.
[30] Alice Pellet-Mary, Guillaume Hanrot and Damien Stehlé, Approx-SVP in ideal lattices with pre-processing, in: *EUROCRYPT*, pp. 685–716, 2019.
[31] Peter Pivovarov, Volume thresholds for Gaussian and spherical random polytopes and their duals, *Studia Mathematica* **183** (2007), 15–34.
[32] Peter Pivovarov, *Volume distribution and the geometry of high-dimensional random polytopes*, Ph.D. thesis, 2010.
[33] Oded Regev, On lattices, learning with errors, random linear codes, and cryptography, in: *STOC*, pp. 84–93, 2005.
[34] Claude E. Shannon, Probability of error for optimal codes in a Gaussian channel, *Bell System Technical Journal* **38** (1959), 611–656.
[35] Maria Shcherbina and Brunello Tirozzi, On the volume of the intersection of a sphere with random half spaces, *Comptes Rendus Mathematique* **334** (2002), 803–806.
[36] Peter W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, in: *FOCS*, pp. 124–134, 1994.
[37] Naftali Sommer, Meir Feder and Ofir Shalvi, Finding the closest lattice point by iterative slicing, *SIAM Journal of Discrete Mathematics* **23** (2009), 715–731.
[38] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka and Keita Xagawa, Efficient public key encryption based on ideal lattices, in: *ASIACRYPT*, pp. 617–635, 2009.
[39] Noah Stephens-Davidowitz, A time-distance trade-off for GDD with preprocessing - Instantiating the DLW heuristic, in: *CCC*, 2019.
[40] Michel Talagrand, Intersecting random half-spaces: toward the Gardner–Derrida formula, *The Annals of Probability* **28** (2000), 725–758.
[41] Nicola Turchi, *High-dimensional asymptotics for random polytopes*, Ph.D. thesis, 2019.
[42] J. G. Wendel, A problem in geometric probability, *Mathematica Scandinavica* **11** (1962), 109–112.

A The Sommer–Feder–Shalvi iterative slicer

We briefly describe some more details on previous, related work in these appendices, starting with the iterative slicer of Sommer–Feder–Shalvi [37]. This algorithm provides an elementary, greedy strategy to attempt to find a closest vector to a given target vector $\boldsymbol{t}$, given a list of lattice points $L \subset \mathcal{L}$, which always finds a solution when $L = \mathcal{R}$ is the set of relevant vectors of the lattice. To do this, note that the shortest representative $\boldsymbol{t}'$ in the coset of the lattice $\boldsymbol{t} + \mathcal{L}$ is necessarily contained in the Voronoi cell of the lattice, and therefore $\mathbf{0}$ is the closest lattice vector to $\boldsymbol{t}'$. This implies that $\boldsymbol{t} - \boldsymbol{t}'$ is the closest lattice vector to $\boldsymbol{t}$, and so finding the shortest representative $\boldsymbol{t}' \in \boldsymbol{t} + \mathcal{L}$ is equivalent to solving CVP for $\boldsymbol{t}$.

To find this shortest representative, given $\boldsymbol{t}$ and a list of lattice vectors $L \subset \mathcal{L}$, the algorithm follows the same approach of e.g. lattice sieving algorithms [21, 26, 28]: we start with $\boldsymbol{t}' = \boldsymbol{t}$, and we repeatedly try to find vectors $\boldsymbol{v} \in L$ such that $\boldsymbol{t}' \leftarrow \boldsymbol{t}' - \boldsymbol{v}$ is a shorter vector in the coset $\boldsymbol{t} + \mathcal{L}$. If no more such reductions can be done, we terminate and hope that the algorithm found the shortest representative.

Summarizing, the iterative slicer can be succinctly described through the pseudocode of Algorithm 1.

Algorithm 1 The Sommer–Feder–Shalvi iterative slicer [37]

Require: The relevant vectors $\mathcal{R} \subset \mathcal{L}$ and a target $\boldsymbol{t} \in \mathbb{R}^d$
Ensure: The algorithm outputs a closest lattice vector $\boldsymbol{s} \in \mathcal{L}$ to $\boldsymbol{t}$

```
Initialize t' ← t
for each r ∈ R do
    if ||t' − r|| < ||t'|| then
        Replace t' ← t' − r and restart the for-loop
    end if
end for
return s = t − t'
```

B The Doulgerakis–Laarhoven–De Weger randomized slicer

As the iterative slicer of Sommer–Feder–Shalvi often does not succeed, when using as input only a subset of the relevant vectors of the lattice, Doulgerakis–Laarhoven–De Weger proposed the following heuristic variant of the slicer. Instead of using the list of relevant vectors for reductions, first we only use a subset of the relevant vectors. Since there is no guarantee that the slicer then returns a vector from the exact Voronoi cell, and the output may not be a solution, we repeat the algorithm many times on rerandomized versions of the same target vector. What this means is that instead of reducing $\boldsymbol{t}' = \boldsymbol{t}$ with the iterative slicer, we sample $\boldsymbol{t}' \sim \boldsymbol{t} + \mathcal{L}$ at random (e.g. from a discrete Gaussian distribution over the coset $\boldsymbol{t} + \mathcal{L}$) and repeat the algorithm on many such samples. This algorithm is given in pseudocode in Algorithm 2.

In the worst case, each of these reductions will end up on the same path and reduce to the same, wrong solutions, thus making no progress. In practice however it was observed that, if the iterative slicer find a solution in a single run with probability $p \ll 1$, then repeating the algorithm K times with such randomized target vectors leads to an overall success probability proportional to $K \times p$. This is purely an experimental, heuristic tweak – there are no theoretical guarantees that reducing such a shifted target vector gives "fresh" results.

Algorithm 2 The Doulgerakis–Laarhoven–De Weger randomized slicer [12]

Require: A list $L \subset \mathcal{L}$ and a target $\boldsymbol{t} \in \mathbb{R}^d$
Ensure: The algorithm outputs a closest lattice vector $\boldsymbol{s} \in \mathcal{L}$ to $\boldsymbol{t}$

$\boldsymbol{s} \leftarrow \boldsymbol{0}$
repeat
  Sample $\boldsymbol{t}' \sim D_{\boldsymbol{t}+\mathcal{L},s}$
  for each $\boldsymbol{r} \in L$ **do**
    if $\|\boldsymbol{t}' - \boldsymbol{r}\| < \|\boldsymbol{t}'\|$ **then**
      Replace $\boldsymbol{t}' \leftarrow \boldsymbol{t}' - \boldsymbol{r}$ and restart the **for**-loop
    end if
  end for
  if $\|\boldsymbol{t}' - \boldsymbol{0}\| < \|\boldsymbol{t} - \boldsymbol{s}\|$ **then**
    $\boldsymbol{s} \leftarrow \boldsymbol{t} - \boldsymbol{t}'$
  end if
until $\boldsymbol{s}$ is a closest lattice vector to $\boldsymbol{t}$
return $\boldsymbol{s}$

C The Doulgerakis–Laarhoven–De Weger complexity analysis

To analyze the heuristic time and space complexities of the randomized slicer, Doulgerakis–Laarhoven–De Weger made the following assumptions. First, the vectors from the list $L \subset \mathcal{L}$ are assumed to follow a spherically symmetric distribution, and their lengths are assumed to follow the prediction obtained via the Gaussian heuristic. Similarly, the exact Voronoi cell of the lattice is modeled as a ball of a certain radius, such that the volume of the ball matches the volume of the lattice. Containment of the reduced vector $\boldsymbol{t}' \in \boldsymbol{t} + \mathcal{L}$ in $\mathcal{V}$ was then estimated to be equivalent to the condition $\|\boldsymbol{t}'\| \leq \lambda_1(\mathcal{L})$.

Then, to analyze the success probability of the slicing routine, first it was observed that if $\boldsymbol{t}'$ has a rather large norm, then it is likely that L contains a vector $\boldsymbol{v}$ such that $\boldsymbol{t}' - \boldsymbol{v}$ is shorter than $\boldsymbol{t}'$; progress can then still be made with ease. There is a phase transition at a certain value β such that

- If $\|\boldsymbol{t}'\| > \beta$, then with probability at least $d^{-\Theta(1)}$ there exists a vector $\boldsymbol{v} \in L$ such that $\|\boldsymbol{t}' - \boldsymbol{v}\| \leq \|\boldsymbol{t}'\|$;
- If $\|\boldsymbol{t}'\| < \beta$, then with probability at most $2^{-\Theta(d)}$ there exists a vector $\boldsymbol{v} \in L$ such that $\|\boldsymbol{t}' - \boldsymbol{v}\| \leq \|\boldsymbol{t}'\|$.

After reaching norm β, the algorithm may still find a solution, but each additional reduction step is exponentially small to occur. To obtain a bound on the overall success probability of the algorithm, the authors studied the probability that after *exactly one more reduction* with the list L, we reach the desired norm $\lambda_1(\mathcal{L})$, so that $\boldsymbol{t}'$ is expected to be contained in $\mathcal{V}$. This is of course only one way for the algorithm to "reach" the Voronoi cell, and it may also happen that after two, three, and any number of additional reductions we still reach the solution, albeit with exponentially small probability. The analysis based on finding the solution in exactly one step, jumping from norm β to $\lambda_1(\mathcal{L})$, is therefore only a lower bound on the overall success probability of the algorithm. This directly leads to the bound on the success probability stated in Equation (5).

Then, given this analysis of the algorithm, the authors obtained a lower bound on the success probability p of a single run of the (randomized) iterative slicer. If one then makes the additional assumption that the success probability of the algorithm is *equal* to the ratio of the volume of the exact cell over the volume of the approximate Voronoi cell, then this would immediately yield a lower bound on the ratio of these volumes as well. This would then lead to the conjectured lower bound on the ratio of the volumes given in Equation (4).

As shown in this paper, the latter step is incorrect, as we give tight bounds on the ratio of these volumes, and show that the inverse of the expression from (4) is *not* a lower bound on the ratio of the volumes.