An $n^{n+o(n)}$-Time Algorithm for the Lattice Isomorphism Problem

Divesh Aggarwal* Kaijie Jiang† Zihan Li‡ Yinchen Liu§

Abstract

The *Lattice Isomorphism Problem* asks whether two given lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are related by an orthogonal linear transformation. Haviv and Regev gave a seminal $n^{O(n)}$-time algorithm for this problem based on an isolation lemma (SODA 2014).

We give randomized algorithms for the decision, search, and all-isomorphisms versions of the problem running in $O(n\sqrt{\log n})^n \leq n^{n+o(n)}$ time, up to polynomial factors in the input size. We first reduce general instances to the well-rounded case. In this setting, our main new ingredient is a Gaussian heat-comparison argument on polytopes generated by shortest vectors: for a suitable parameter s, a sample $w \sim D_{\mathcal{L}^*,s}$ canonically determines $n - o(n)$ independent shortest vectors with constant probability, leaving a residual instance of rank $o(n)$. We handle this residual instance using an $n^{o(n)}$-time canonicalizer obtained by adapting the Haviv–Regev algorithm. We then combine this canonicalizer with a birthday collision argument to recover all isomorphisms.

In the QRAM model, we also give a quantum variant running in time $O(n^2 \log n)^{\frac{n}{3}}$. It outputs one isomorphism together with generators for the automorphism group, giving a compact description of the entire isomorphism coset.

*National University of Singapore, Singapore. dcsdiva@nus.edu.sg.

†Institute for Advanced Study, Tsinghua University, Beijing, China. kjj101110@gmail.com.

‡National University of Singapore, Singapore. zihan_li_05@u.nus.edu.

§Institute for Interdisciplinary Information Sciences, Tsinghua University. liuyinch23@mails.tsinghua.edu.cn.

Contents

1 Introduction **1**
- 1.1 Our results . . . 2
- 1.2 Proof overview and technical ideas . . . 2
  - 1.2.1 The Haviv–Regev algorithm and its enumeration bottleneck . . . 3
  - 1.2.2 Markers, the shortest-vector matroid, and the heat comparison . . . 4
  - 1.2.3 The exact marker label and a canonicalizer . . . 6
  - 1.2.4 The birthday search, the recursion, and the quantum speedup . . . 7
- 1.3 Open problems . . . 8
- 1.4 Organization . . . 8

2 Preliminaries **8**
- 2.1 Basic notation . . . 8
- 2.2 Lattice isomorphism and hard problems . . . 10
- 2.3 Discrete Gaussians and smoothing . . . 13
- 2.4 Shortest-vector enumeration and sampling . . . 15
- 2.5 Matroids . . . 16

3 Projected Rank Polytopes and Ambiguous Ranks **17**
- 3.1 Properties of polytopes . . . 18
- 3.2 Certifying ambiguity and bounding the residual rank . . . 19

4 Gaussian Heat Comparison for Dual Markers **20**
- 4.1 Local lower bound . . . 21
- 4.2 Global upper bound . . . 22
- 4.3 Ambiguous-rank bound . . . 24

5 Small-Rank Canonical Forms for Residual-Glue Instances **25**
- 5.1 Basic tools . . . 25
- 5.2 Residual-glue canonical forms . . . 28

6 Exact Marker Records and Birthday Sampling **30**
- 6.1 Marker labels . . . 30
- 6.2 Exactness of the labels . . . 32
- 6.3 Reciprocal collision probability of the marker distribution . . . 34
- 6.4 The birthday algorithm . . . 35

7 The Main Algorithm **37**
- 7.1 The well-rounded case . . . 38
- 7.2 General instances . . . 38

8 A Quantum Speedup for the Isomorphism Coset **42**
8.1 Quantum notation and coherent computation 42
8.2 QRAM and the BHT template 43
8.3 Probability of matching a table label 44
8.4 Quantum birthday search for well-rounded lattices 45
8.5 BSGS and automorphism group generation 47
8.6 Recursion for general lattices 49

Acknowledgments **52**

AI Statement **52**

A The discriminant normal form **52**

B The Haviv–Regev canonicalizer for lattices **54**

C Auxiliary probabilistic results **59**

1 Introduction

A lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is a discrete additive subgroup generated by n linearly independent vectors. Two full-rank lattices $\mathcal{L}_1, \mathcal{L}_2 \subseteq \mathbb{R}^n$ are *isomorphic* if there exists an orthogonal transformation $O \in \mathrm{O}_n(\mathbb{R})$ such that $O\mathcal{L}_1 = \mathcal{L}_2$, i.e., $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) := \{O \in \mathrm{O}_n(\mathbb{R}) : O\mathcal{L}_1 = \mathcal{L}_2\} \neq \varnothing$. The *Lattice Isomorphism Problem* (LIP) asks whether such an O exists. We also consider the search version, which asks to find one such isomorphism, and the all-isomorphisms version, which asks to enumerate the full set $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$.

The problem is closely related to equivalence of positive definite integral quadratic forms and belongs to the same family of isomorphism problems as the *Graph Isomorphism Problem* (GIP) and the *Code Equivalence Problem* (CEP). Unlike the extensively studied SVP and CVP [HR07, Ste16, BGS17, ABGS21, AS18], which seek individual lattice vectors, LIP studies the global symmetry between lattices. It also appears to lie outside the reduction framework that makes CVP "lattice complete": nearly all well-studied lattice problems admit dimension- and approximation-factor-preserving reductions to CVP [Ste16].

On the algorithmic side, the study of LIP goes back to work of Plesken and Pohst [PP85] and of Plesken and Souvignier [PS97], who developed practical algorithms for low-dimensional instances. Haviv and Regev [HR14] later gave an $n^{O(n)}$-time algorithm for enumerating all isomorphisms using a generalized isolation lemma [VV85] and an SVP oracle. Another algorithmic connection was given by Dutour Sikirić, Haensch, Voight, and van Woerden [DSHVvW20], who showed that, given access to an SVP oracle, a search-LIP instance can be reduced to a GIP instance. Although GIP admits a quasi-polynomial-time algorithm by Babai [Bab16], their construction produces a graph with at least as many vertices as the number of shortest nonzero vectors in the lattice. This number can be as large as $2^{\Omega(\sqrt{n})}$ [LK26, BHLS26], while the best known upper bound is $2^{O(n)}$ [KL78]. Prior to the present work, the fastest known worst-case asymptotic algorithm for arbitrary rank-n instances of LIP remained the $n^{O(n)}$ framework of Haviv and Regev. For a special class of self-dual lattices, Bennett and Fridberg recently gave a $2^{n/2+o(n)}$ randomized isomorphism algorithm [BF26].

A separate line of work concerns the complexity-theoretic and structural status of LIP. Dutour Sikirić, Schürmann, and Vallentin [DSSV09] proved that the decision version of LIP is at least as hard as GIP. Haviv and Regev [HR14] further proved that the decision version lies in SZK; that is, it admits a statistical zero-knowledge proof system. More recently, Jiang and Liu [JL26] investigated the complexity of search LIP by studying the irreducible decomposition of lattices.

On the applications side, LIP and its variants have recently been used to construct a wide range of cryptographic primitives, including public-key encryption [BGPS23, ARLW24], fully homomorphic encryption [BMM25, LRvW25, BLS26], signatures [DPPvW22, DvW22], advanced signature primitives [KS26, vNdP26, Pha25], key encapsulation mechanisms [DvW22], identification schemes [DvW22], and commitment schemes [JWL+25a, LJJ+26].

Our starting point is the $n^{O(n)}$-time algorithm of Haviv and Regev [HR14]. It first reduces the problem to the well-rounded case, in which the shortest vectors span the ambient space. A suitably chosen dual vector then orders the shortest vectors into a canonical independent chain. Finding the image of this dual vector in the second lattice determines the image of the chain and hence the isomorphism. The main cost is enumerating candidate images of that dual vector, leading to running times of the form $n^{cn+o(n)}$ for constants $c > 1$. We note, however, that the Haviv–Regev

algorithm has one feature that ours does not: it can be implemented in polynomial space.[1] As discussed in Section 1.2.1, their algorithms give $n^{10n+o(n)}$ time without a space restriction.

1.1 Our results

In this paper, we give an $O(n\sqrt{\log n})^n$-time algorithm for the decision, search, and all-isomorphisms versions of LIP.[2]

Theorem 1.1 (Main theorem, see Theorem 7.3)**.** *There is a randomized algorithm that, given two rank-n integral lattices $\mathcal{L}_1, \mathcal{L}_2$, runs in time $O(n\sqrt{\log n})^n$, up to polynomial factors in the input encoding length, and with probability at least $1 - 2^{-\Omega(n \log n)}$ outputs the full set $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$; in particular, it outputs $\varnothing$ if the lattices are not isomorphic.*

The output size of LIP gives a nearly matching lower bound for enumerating all isomorphisms. Feit's theorem [Fei96] implies that, for every $n > 10$, the automorphism group of any rank-n lattice has size at most $2^n\, n! \le O(n)^n$, and the lattice $\mathbb{Z}^n$ attains this order of magnitude (Theorem 2.11). Thus our running time is optimal in the worst case up to an $O(\sqrt{\log n})^n$ factor.

This output-size barrier does not apply to compact descriptions. Whenever it is nonempty, $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ is a coset of $\mathrm{Aut}(\mathcal{L}_1)$. Every finite group G has a generating set of size $O(\log |G|)$ [Bab91], so one isomorphism and polynomially many generators of $\mathrm{Aut}(\mathcal{L}_1)$ suffice to describe the full isomorphism set. In the QRAM model, we compute such a compact description in $O(n^2 \log n)^{n/3}$ time, improving on the classical algorithm. This also solves the decision and search versions of LIP.

Theorem 1.2 (Quantum speedup, see Theorem 8.15)**.** *In the QRAM model, there is a quantum algorithm that, given rank-n integral lattices $\mathcal{L}_1, \mathcal{L}_2$, runs in time $O(n^2 \log n)^{n/3}$ with QRAM of size $O(n^2 \log n)^{n/3}$, up to polynomial factors in the input encoding length, and with probability at least $1 - 2^{-\Omega(n \log n)}$ outputs a representative isomorphism and generators for $\mathrm{Aut}(\mathcal{L}_1)$, or $\varnothing$ if the lattices are not isomorphic.*

Throughout, each input lattice is specified by its Gram matrix and each isomorphism by an integral coordinate-change matrix, following the standard quadratic-form model of lattice isomorphism; see Section 2.2 for the precise input conventions.

1.2 Proof overview and technical ideas

We first recall the Haviv–Regev algorithm [HR14], and study the main bottleneck that makes it expensive. After this, we outline the new ingredients of our algorithm. These include a matroid-polytope analysis, and a Gaussian heat comparison showing that a random marker sampled from a discrete Gaussian determines all but $O(n/\log n)$ ranks with constant probability. Then we introduce

[1] Haviv–Regev's shortest-vector subroutines can use Kannan's algorithm [Kan87], which runs in $n^{O(n)}$ time and polynomial space. Our algorithm instead relies on the $2^{n+o(n)}$-time discrete-Gaussian sampler of [ADRS15], which uses $2^{n+o(n)}$ space. In addition, our algorithm uses a birthday argument to give a quadratic speedup, hence blowing up the space usage to $n^{n+o(n)}$.

[2] Throughout the paper, $O(f(n))^n$ stands for $\big(Cf(n)\big)^n$, where $C > 0$ is a universal constant; in particular, $O(n)^n = (Cn)^n$, and $O(n\sqrt{\log n})^n = n^n \cdot O(\sqrt{\log n})^n \le n^{n+o(n)}$.

a birthday search that finds the matching markers on the two sides. Finally, we give a quantum speedup of the birthday search using the standard Brassard–Høyer–Tapp collision finding algorithm [BHT98].

Recall that a rank-n lattice $\mathcal{L}$ is a discrete subgroup of Euclidean space generated by n independent vectors. For simplicity, in the overview we ignore the bit complexity issues and assume all lattices are given by rational bases. It's worth noting that for two isomorphic lattices $\mathcal{L}_1$ and $\mathcal{L}_2$, every isomorphism O maps the shortest vector shell $\mathrm{Min}(\mathcal{L}_1) = \{x \in \mathcal{L}_1 : \|x\| = \lambda_1(\mathcal{L}_1)\}$ onto $\mathrm{Min}(\mathcal{L}_2)$. Moreover, the shortest-vector shell of a rank-n lattice can be computed in time $2^{O(n)}$; see, for example, [MV13]. This cost is negligible compared with $n^{n+o(n)}$.

1.2.1 The Haviv–Regev algorithm and its enumeration bottleneck

First suppose the shortest nonzero vectors of $\mathcal{L}$ do not span the ambient space. Write $V_i = \mathrm{span}(\mathrm{Min}(\mathcal{L}_i))$ for the two input lattices. Every isomorphism maps V_1 to V_2, so the instance decomposes into sections $A_i = \mathcal{L}_i \cap V_i$, quotients $Q_i = \pi_{V_i^\perp}(\mathcal{L}_i)$, and gluing data. Both subproblems have smaller rank, so solving them recursively gives a time bound

$$k^{Ck}(n-k)^{C(n-k)} \le n^{Cn}.$$

The n^{Cn} running-time is preserved as long as a single universal constant C serves in the exponent at every node.

It therefore remains to handle the *well-rounded* case, in which the shortest nonzero vectors span the whole ambient space. We assume well-roundedness for the rest of this overview.

For a well-rounded lattice, Haviv and Regev use an isolation lemma to choose a dual vector $v \in \mathcal{L}^*$. It assigns integer scores $\langle v, x\rangle$ to each $x \in \mathrm{Min}(\mathcal{L})$. They then build an ordered chain greedily from these scores. Take x_1 to be the shortest nonzero vector maximizing score $\langle v, x_1\rangle$. And given $x_1, \dots, x_{j-1}$, the procedure chooses x_j to be the shortest nonzero vector outside their span with maximum score. After n steps, it produces an ordered real basis $x_1, \dots, x_n$ of shortest nonzero vectors. The danger is a tie: if at some step two candidate vectors share the same maximum score, the choice is not unique and the chain is not canonical. The isolation lemma guarantees a vector $v \in \mathcal{L}^*$ with $\|v\| \le 5n^{8.5}\,\lambda_1(\mathcal{L}^*)$, such that the maximizer is unique at every step. The resulting chain is determined uniquely by $(\mathcal{L}, v)$, and an isomorphism carrying v to another marker carries this chain to the corresponding chain.

Once such v has been chosen for $\mathcal{L}_1$, the algorithm searches for an image of v under isomorphism, by trying every candidate in $\mathcal{L}_2^*$ up to the same length bound. A packing bound shows that the number of such candidate dual vectors is at most $(10n^{8.5}+1)^n \le O(n^{8.5})^n$. The coordinate enumeration used by Haviv–Regev is larger than this packing bound: for a vector of length at most $t\lambda_1$, a dual Korkine–Zolotarev basis bounds every integral coefficient by $tn^{3/2}$ [HR14, Lemma 2.7]. Substituting $t = 5n^{8.5}$, their procedure enumerates $(2tn^{3/2}+1)^n = (10n^{10}+1)^n \le O(n^{10})^n$ coefficient vectors, yielding the running time in [HR14, Corollary 2.16]. This is the main bottleneck of the algorithm.

In fact, this construction is the greedy algorithm on the matroid with ground set $A = \mathrm{Min}(\mathcal{L})$ and independent sets given by linearly independent subsets of A. A dual vector $w \in \mathcal{L}^*$, which we

call a *marker*, gives weights

$$A \to \mathbb{R}, \qquad x \mapsto \langle w, x \rangle.$$

Haviv and Regev require these weights to give a unique choice at all n greedy steps. We relax this requirement and allow for a small number of steps to be not unique.

1.2.2 Markers, the shortest-vector matroid, and the heat comparison

The shortest-vector matroid. Fix a well-rounded $\mathcal{L}$ and let $A = \mathrm{Min}(\mathcal{L})$ be its shortest shell, with $\lambda = \lambda_1(\mathcal{L}) = \lambda_n(\mathcal{L})$. We consider the matroid $\mathcal{M} = (A, \mathcal{I})$ on A, where a subset of shortest vectors is in $\mathcal{I}$ if it is linearly independent over $\mathbb{R}$. A marker interacts with this matroid through a family of polytopes. For each rank j and each independent j-set $B \subseteq A$, form the sum $p_B = \sum_{b \in B} b \in \mathcal{L}$, and let the *rank-$j$ projected polytope* be

$$P_j = \mathrm{conv}\{\, p_B : B \in \mathcal{I},\ |B| = j \,\}$$

For a marker w, the greedy choice amounts to maximizing $\langle w, \cdot \rangle$ over P_j. Write the support function and the exposed face of P_j as

$$h_j(w) = \max_{p \in P_j} \langle w, p \rangle, \qquad F_j(w) = \arg\max_{p \in P_j} \langle w, p \rangle.$$

Since $A = -A$, each P_j is centrally symmetric and $h_j \geq 0$; as a pointwise maximum of linear maps, h_j is convex. Call rank j *resolved* at w if $F_j(w)$ is a single vertex, and *ambiguous* if two or more vertices tie.

We use the following observation about the polytopes, which is a simple consequence of matroid basis exchange.

Lemma 1.3 (See Lemma 3.4)**.** *Every vertex p of P_j is p_B for a* unique *independent j-set $B \subseteq A$.*

Thus each vertex of P_j corresponds to a unique independent j-set. Given this, we simply run the matroid greedy algorithm against the scores $x \mapsto \langle w, x \rangle$ and obtain a nested sequence $\emptyset = B_0 \subseteq B_1 \subseteq \cdots \subseteq B_n$, in which $p_{B_j} \in F_j(w)$.

A zero-slack exchange test (see Lemmas 3.5 and 3.8) then identifies all ambiguous ranks in $2^{O(n)}$ time.

Now suppose ranks $j-1$ and j are both resolved, with exposed vertices $p_{j-1}(w)$ and $p_j(w)$. Their unique independent sets B_{j-1}, B_j satisfy $B_{j-1} \subseteq B_j$ (Lemma 3.6), so

$$u_j(w) := p_j(w) - p_{j-1}(w)$$

is a shortest nonzero vector uniquely determined by $(\mathcal{L}, w)$. We call $u_j(w)$ a *fixed vector*. Iterating over all consecutive resolved ranks, the fixed vectors form an independent chain with length at least $n - 2\mathrm{Amb}(w)$, where $\mathrm{Amb}(w)$ is the number of ambiguous ranks.

We next prove that, with at least 0.99 probability, the number of ambiguous ranks is small when the marker w is drawn from the discrete Gaussian $D_{\mathcal{L}^*, s}$, with s chosen below.

The Gaussian heat comparison. Ambiguity at rank j means two distinct vertices $p \neq q$ of P_j attain the maximum inner product simultaneously with marker w. We bound the ambiguity by looking at the growth of the support function h_j when w is smoothed by a continuous Gaussian. Writing $\lambda = \lambda_1(\mathcal{L})$, and $G_a \sim \gamma_a$ for an isotropic continuous Gaussian of width a, we will consider the *heat increment*

$$\Delta_j(w) = \mathbb{E}\big[h_j(w + G_t)\big] - \mathbb{E}\big[h_j(w + G_{t/2})\big], \qquad t = \Theta(\sqrt{n}/\lambda).$$

If rank j is ambiguous at w, the tied vertices are lattice vectors, so $p - q \in \mathcal{L}$ is nonzero and $\|p - q\| \geq \lambda$. Consequently, we derive

$$\mathbb{E}\big[h_j(w + G_t)\big] - h_j(w) \geq \tfrac{1}{2}\,\mathbb{E}\big|\langle G_t, p - q\rangle\big| = \Theta\big(t\,\|p - q\|\big) = \Omega(t\lambda).$$

The function $a \mapsto \mathbb{E}[h_j(w + aG_1)] - h_j(w)$ is convex and vanishes at 0, its value at $t/2$ is at most half its value at t. Therefore, we obtain $\Delta_j(w) \geq c\,t\lambda$ for a universal constant c, while $\Delta_j \geq 0$ for every w by Jensen's inequality.

For the upper bound, average over $W \sim D_{\mathcal{L}^*,s}$. With the parameters chosen to satisfy the smoothing condition of Lemma 2.16, the distributions of $W + G_t, W + G_{t/2}$ are pointwise close to *continuous* Gaussians of widths $r_1 = \sqrt{s^2 + t^2}$ and $r_0 = \sqrt{s^2 + t^2/4}$. The support function h_j is homogeneous of degree one and satisfies $h_j(z) \leq j\lambda\|z\|$. Since $r_1 - r_0 = O(t^2/s)$, Section 4 shows that $\mathbb{E}[\Delta_j(W)] = O(t^2 j\lambda\sqrt{n}/s)$.

Comparing the two bounds and choosing $s = \Theta\big(n^2 \log n/\lambda\big)$ gives

$$\Pr[j \text{ ambiguous}] = O\left(\frac{t^2 j\lambda\sqrt{n}/s}{t\lambda}\right) = O\left(\frac{j}{n \log n}\right).$$

Summing over j yields $\mathbb{E}[\mathrm{Amb}(W)] = O(n/\log n)$. Markov's inequality then shows a random marker w gives an ordered independent chain of length $n - O(n/\log n)$ with probability at least 0.99. We call such a marker *resolving*.

An $O(n^2 \log n)^n$-time search. These bounds already suggest an $O(n^2 \log n)^n$-time algorithm. Assume $\mathcal{L}_1 \cong \mathcal{L}_2$, and sample $w_1 \sim D_{\mathcal{L}_1^*,s}$. With probability 0.99, the marker is resolving: it determines an ordered chain of $n - O(n/\log n)$ shortest nonzero vectors of $\mathcal{L}_1$ and leaves a residual of rank $r \leq O(n/\log n)$. Every isomorphism O carries w_1 to a marker in $\mathcal{L}_2^*$. Once this image is found, the fixed chains must match in order, and the residual instance can be handled by adapting the Haviv–Regev algorithm in $r^{O(r)} \leq 2^{O(n)}$ time.

The main cost is therefore finding the image of w_1. Recall that $s = \Theta(n^2 \log n/\lambda)$, and the Gaussian lies in a ball of radius $O(s\sqrt{n})$ except for $2^{-\Omega(n)}$ of its mass. A volume bound gives $2^{O(n)} s^n \det(\mathcal{L}_2)$ dual lattice points in this ball, and $\det(\mathcal{L}_2) \leq \lambda^n$ because $\mathcal{L}_2$ is well-rounded. So there are at most $O(s\lambda)^n = O(n^2 \log n)^n$ candidate vectors, and combining the enumeration algorithm of [Dad12, Theorem 5.2.6] gives an $O(n^2 \log n)^n$ running time.

1.2.3 The exact marker label and a canonicalizer

To obtain further speedup, we use a birthday (meet-in-the-middle) approach. Instead of enumerating the image of w_1, we attach to every resolving marker a canonical *label* with two properties:

- *Completeness*: whenever some isomorphism carries w_1 to w_2, the two labels agree.
- *Soundness*: whenever the two labels agree, the isomorphisms mapping w_1 to w_2 form a nonempty set, and *all* these isomorphisms can be recovered in $2^{O(n)}$ time per isomorphism.

We can then sample markers from both lattices and search for equal labels across the two sides. This takes only $O(n\sqrt{\log n})^n$ samples, roughly the square root of the enumeration bound. We next show how to construct the labels.

Fix a resolving marker w, and let $(u_j(w))_{j\in J(w)}$ be its ordered fixed vectors ($J(w)$ is the set of indices for fixed vectors). They span a rational subspace $U = U(w)$. Define

$$C = \mathcal{L} \cap U, \qquad R = \mathcal{L} \cap U^{\perp}, \qquad w = w_U + w_R \ \ (w_R \in R^*),$$

where we call C the *fixed lattice*, and R the *residual lattice*, which has rank $r = O(n/\log n)$.

The fixed vectors generate a finite-index sublattice $F = \sum_{j\in J(w)} \mathbb{Z}\, u_j(w) \subseteq C$. We define the *fixed data* $\mathcal{F}$, which records the Gram matrix of the fixed vectors, the fixed marker scores $(\langle w, u_j(w)\rangle)_j$, and the finite quotient C/F. An isomorphism matching two markers must match their fixed vectors in order, so its restriction to U is uniquely determined and the fixed data $\mathcal{F}$ is preserved. Thus this part requires comparison but no search.

The next piece of data is called the *glue*. In general $S := C \oplus R$ is only a finite-index sublattice of $\mathcal{L}$. The glue subgroup $H = \mathcal{L}/S \le D(S) = D(C) \oplus D(R)$ specifies how to get back $\mathcal{L}$ from S. Together, C, R and H determine $\mathcal{L}$.

Suppressing details about scale data and encoding conventions, the *residual-glue* instance is

$$\mathcal{G}(\mathcal{L}, w) = \big(R,\ \mathcal{F},\ H,\ w_R\big).$$

The fixed data $\mathcal{F}$ is compared directly, while the residual lattice R needs to be canonicalized. Adapting the Haviv–Regev algorithm gives a canonical representative $\mathrm{Can}_0(R)$, and the set $\Phi_0(R)$ of isometries from R to this representative. The last two fields w_R and H are defined only relative to a choice of such an isomorphism, so choosing the lexicographical minimum gives

$$\mathsf{Lab}(\mathcal{L}, w) = \Big(\mathrm{str}(\mathcal{F}),\ \mathrm{Can}_0(R),\ \min\nolimits^{\mathrm{lex}}_{\phi\in\Phi_0(R)}\big(\phi\, w_R,\ (\mathrm{id}\oplus\overline{\phi})H\big)\Big).\ ^{3}$$

Here $\overline{\phi}$ denotes the action of ϕ on the discriminant group $D(R) = R^*/R$ (in fact, H lives in the quotient $(C^* \oplus R^*)/(C \oplus R)$, assuming $\mathcal{L}$ is integral). The resulting label is sound and complete (Theorem 6.5), and can be computed in $2^{O(n)}$ time (Lemma 6.6).

[3] The overview ignores how these objects are actually written down. In the real algorithm the fields are rational data, because C, R and w_R come from intersecting $\mathcal{L}$ with U or $U^{\perp}$. We clear all denominators by one common scalar, which does not change the isomorphism set, and record that scalar in the label.

1.2.4 The birthday search, the recursion, and the quantum speedup

The *labels* turn the search for the image of w_1 into a collision problem. Sample markers independently from $D_{\mathcal{L}_1^*,s}$ and $D_{\mathcal{L}_2^*,s}$, discard non-resolving markers, and group the samples by label. Each collision (w_1, w_2) yields all isomorphisms mapping w_1 to w_2. Here sampling and labeling a marker costs $2^{O(n)}$ time with the discrete-Gaussian sampler of [ADRS15].

Since we need to output the whole isomorphism set, a single collision pair (w_1, w_2) may not suffice. To enumerate the full set, we show that each $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ is witnessed by a sampled pair (w, Ow). Write $\mu_i = D_{\mathcal{L}_i^*,s}$. Since $\mu_2(Ow) = \mu_1(w)$, an independent pair equals (w, Ow) with probability $\mu_1(w)^2$. Summing over resolving markers gives a collision probability $\Theta(1/S_2(\mu_1))$, where

$$S_2(\mu) := \Big(\sum_w \mu(w)^2\Big)^{-1} = \frac{\rho_s(\mathcal{L}^*)^2}{\rho_{s/\sqrt{2}}(\mathcal{L}^*)}.$$

This collision probability is independent of O. For $s = \Theta(n^2 \log n/\lambda)$, both s and $s/\sqrt{2}$ are above the smoothing parameter of $\mathcal{L}^*$. Hence $S_2(\mu) = (1 \pm o(1))\, 2^{n/2} s^n \det(\mathcal{L})$. Bounding $\det(\mathcal{L}) \le \lambda^n$ as before, we obtain $S_2(\mu) \le O(n^2 \log n)^n$.

Taking $N = \sqrt{S_2(\mu)} \cdot \mathrm{poly}(n)$ samples makes the failure probability for each O small enough to survive a union bound over the at most $O(n)^n$ isomorphisms. In the algorithm, we use a hash table to check for equal labels, and recover all isomorphisms compatible with each collision.

Running Time. Processing the collisions costs $2^{O(n)}$ per sampled triple (w_1, w_2, O) with $Ow_1 = w_2$. For a fixed O, the expected number of such triples is $N^2/S_2(\mu) = \mathrm{poly}(n)$, so there are at most $|\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)| \cdot \mathrm{poly}(n) \le O(n)^n$ triples in expectation. Combining sampling, labeling and recovery, the running time is at most

$$O(n^2 \log n)^{n/2} + O(n)^n \cdot 2^{O(n)} \le n^n \cdot O(\sqrt{\log n})^n = O(n\sqrt{\log n})^n.$$

Quantum Speedup. In the QRAM model, a quantum algorithm may query a stored classical table in superposition at unit cost. This allows us to use Brassard–Høyer–Tapp collision finding [BHT98] to replace birthday search. The markers are still drawn from $D_{\mathcal{L}_1^*,s}$ and $D_{\mathcal{L}_2^*,s}$ at the same width $s = \Theta(n^2 \log n/\lambda)$ as before. We first store the labels of t markers of $\mathcal{L}_1^*$ in QRAM. Then apply amplitude amplification to find a marker from $\mathcal{L}_2^*$ whose label occurs in the table. For most choices of tables, this happens with probability $\Omega(t/S_2(\mu))$. Amplitude amplification therefore needs $O(\sqrt{S_2(\mu)/t})$ steps, and balancing this against table size t gives $t = S_2(\mu)^{1/3}$ and a total cost $S_2(\mu)^{1/3} \le O(n^2 \log n)^{n/3}$.

To obtain a compact representation of the full isomorphism set, we apply the routine to $(\mathcal{L}_1, \mathcal{L}_1)$ and choose uniformly among the compatible automorphisms of each collision. The resulting automorphism is within $2^{-\Omega(n)}$ of uniform on $\mathrm{Aut}(\mathcal{L}_1)$. Polynomially many independent samples hence generate the group with high probability. Together with one isomorphism $O_0 \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$, these generators describe $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = O_0\,\mathrm{Aut}(\mathcal{L}_1)$. We then extend this to general lattices through the section–quotient recursion, preserving the $O(n^2 \log n)^{n/3}$ running-time bound.

1.3 Open problems

Since the all-isomorphisms bound is optimal up to an $O(\sqrt{\log n})^n$ factor, a natural question is whether decision or search LIP admits a faster algorithm or a nontrivial time–space tradeoff. It is also open whether using higher norm shells, rather than only $\mathrm{Min}(\mathcal{L})$, can reduce the collision scale or the residual rank. Finally, another future direction is to extend our marker-and-label framework to the *Lattice Distortion Problem* (LDP), introduced in [BDS16]. Given two rank-n lattices $\mathcal{L}_1$ and $\mathcal{L}_2$, LDP asks for a bijective linear transformation T satisfying $T\mathcal{L}_1 = \mathcal{L}_2$ while minimizing its condition number $\|T\| \cdot \|T^{-1}\|$, where $\|T\| = \sup_{\|x\|=1} \|Tx\|$. LDP is therefore a natural generalization of LIP in which arbitrary linear bijections are allowed instead of orthogonal isometries.

1.4 Organization

Section 2 introduces notation and collects the lattice facts we use. Sections 3 and 4 show how resolved ranks determine shortest vectors and bound the number of ambiguous ranks. Section 5 develops the residual canonicalizer, and Sections 6 and 7 construct exact labels and prove the classical algorithm. Section 8 gives the quantum speedup. The appendices collect auxiliary tools and deferred proofs.

2 Preliminaries

2.1 Basic notation

- For a positive integer n, write $[n] = \{1, \ldots, n\}$. All vector spaces and all norms are Euclidean. For a finite set S, $|S|$ denotes its cardinality.
- Let $\mathcal{S}_n^{>0}(\mathbb{Z})$ denote the set of $n \times n$ symmetric positive definite matrices with integer entries. Let $\mathrm{GL}_n(\mathbb{R})$ and $\mathrm{GL}_n(\mathbb{Z})$ be the general linear group of rank n over $\mathbb{R}$ and $\mathbb{Z}$, respectively. We use $\mathrm{O}_n(\mathbb{R})$ to represent the group of orthogonal matrices $O \in \mathrm{GL}_n(\mathbb{R})$ such that $O^\top O = I_n$, where I_n is the identity matrix.
- We use $x \sim \mathcal{D}$ to denote that x is sampled from a distribution $\mathcal{D}$. For a finite set S, we write $s \sim S$ to denote that s is drawn uniformly from S.
- We use capital letters, such as W, X, Y, to denote random variables, and the corresponding lowercase letters, such as w, x, y, to denote concrete values. Thus, for example, $W \sim D_{\mathcal{L}^*,s}$ denotes a random marker sampled from the discrete Gaussian, while $w \in \mathcal{L}^*$ denotes a fixed marker value.
- $\mathrm{TV}(\mu, \nu)$ denotes the total variation distance between probability measures: $\mathrm{TV}(\mu, \nu) := \sup_{A \in \mathcal{B}(\mathbb{R}^n)} |\mu(A) - \nu(A)|$, where $\mathcal{B}(\mathbb{R}^n)$ denotes the Borel sigma-algebra on $\mathbb{R}^n$.
- All asymptotic statements are with respect to the ambient rank n and hold for sufficiently large n. Following Section 1.2, we write $O(f(n))^n$ for $\big(Cf(n)\big)^n$ with a universal constant $C > 0$; in particular, $O(n)^n$ denotes $(Cn)^n$, and any factor $2^{O(n)}$ is absorbed by such a bound. Note also that when $r \le r_*(n) := \lceil \frac{2n}{\log n} \rceil$, we have $r^{O(r)} \le n^{O(r)} \le 2^{O(n)}$.

For a set $X \subseteq \mathbb{R}^n$, $\mathrm{span}(X)$ denotes its real span and $\mathrm{conv}(X)$ its convex hull.

Lattices A full-rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is represented either by a basis $B \in \mathbb{R}^{n\times n}$ with $\mathcal{L} = B\mathbb{Z}^n$ or by its Gram matrix $G = B^\top B$. We work with integral lattices: $\langle x, y\rangle \in \mathbb{Z}$ for all $x, y \in \mathcal{L}$. The determinant is $\det(\mathcal{L}) = \sqrt{\det(B^\top B)}$.

The i-th successive minimum is

$$\lambda_i(\mathcal{L}) = \inf\{r : \dim(\mathrm{span}(\mathcal{L} \cap rB_2^n)) \geq i\},$$

where $rB_2^n = \{x \in \mathbb{R}^n : \|x\| \leq r\}$ is the closed Euclidean ball of radius r around 0. The shortest-vector shell is

$$\mathrm{Min}(\mathcal{L}) = \{x \in \mathcal{L} : \|x\| = \lambda_1(\mathcal{L})\}.$$

A lattice is *well-rounded* if $\mathrm{span}(\mathrm{Min}(\mathcal{L})) = \mathrm{span}(\mathcal{L})$. We write $\lambda = \lambda_1(\mathcal{L}) = \lambda_n(\mathcal{L})$ when the lattice $\mathcal{L}$ is well-rounded and clear from context.

When a lattice $\mathcal{L}$ is represented by a Gram matrix $G \in \mathcal{S}_n^{>0}(\mathbb{Z})$, we use the norm induced by G on coordinate vectors. Namely, for $z \in \mathbb{Z}^n$, we write $\|z\|_G = \sqrt{z^\top G z}$. Equivalently, if $B^\top B = G$ and $\mathcal{L} = B\mathbb{Z}^n$, then the coordinate vector z represents the lattice vector Bz, and $\|z\|_G = \|Bz\|_2$. Accordingly, lattice parameters defined via Euclidean lengths can be written equivalently in terms of G. For example,

$$\lambda_1(\mathcal{L}) := \min_{z\in\mathbb{Z}^n\setminus\{0\}} \|z\|_G.$$

Several intermediate lattices are represented by rational Gram matrices. The following lemma lets us clear denominators on both sides of an instance at once.

Lemma 2.1. *Let G_1, G_2 be positive definite Gram matrices and let $\alpha > 0$. Then the set of rational or integral transporter matrices is unchanged by multiplying both forms by α:*

$$G_1 = U^\top G_2 U \quad \Longleftrightarrow \quad \alpha G_1 = U^\top (\alpha G_2) U.$$

Proof. The displayed equivalence follows by multiplying or dividing the equality by the same positive scalar α. The corresponding Euclidean maps are unchanged because both forms are rescaled by the same homothety. □

Next, we define the dual lattice,

$$\mathcal{L}^* = \{y \in \mathrm{span}(\mathcal{L}) : \langle y, x\rangle \in \mathbb{Z} \text{ for all } x \in \mathcal{L}\}.$$

Then $(\mathcal{L}^*)^* = \mathcal{L}$ and $\det(\mathcal{L}^*)\det(\mathcal{L}) = 1$.

For an integral lattice $\mathcal{L} \subseteq \mathcal{L}^*$, its discriminant group is the finite abelian group

$$D(\mathcal{L}) = \mathcal{L}^*/\mathcal{L}.$$

For a single rational lattice K, we use the following canonical integral model. Let

$$\mathfrak{a}(K) = \big\langle \langle x, y\rangle : x, y \in K \big\rangle_{\mathbb{Z}} \subseteq \mathbb{Q}$$

be the additive subgroup generated by all pairings in K. Since K has finite rank, $\mathfrak{a}(K) = \alpha(K) \cdot \mathbb{Z}$ for a unique $\alpha(K) > 0$ when $K \neq 0$; for the rank-zero lattice we set $\alpha(K) = 1$. The *primitive integral model* of K is the same underlying $\mathbb{Z}$-module with bilinear form $\alpha(K)^{-1}\langle\cdot,\cdot\rangle$. The scalar $\alpha(K)$ is part of the canonical encoding.

More generally, if $S \subseteq K$ is a finite-index full-rank sublattice and K may be rational, the clearing scalar is always chosen from the full overlattice K, not from S alone. After replacing the form by $\alpha(K)^{-1}\langle\cdot,\cdot\rangle$, both K and S are integral and

$$K \subseteq S^*, \qquad K/S \leq D(S).$$

This is the convention used for glue (see Section 5).

Remark 2.2 (Bit complexity and timing convention)**.** *When comparing two lattices, we check if their scales $\alpha(\cdot)$ agree. If so, we rescale by one common positive scalar into the primitive integral model. This preserves the isomorphism set (Lemma 2.1). The canonical encoding also records this scale.*

At each recursive call, we use an LLL-reduced basis [LLL82] to control the bit length of the Gram matrix entries. The norm cutoff in Lemma 2.22 gives the sampled markers polynomial bit length in the working basis. The HNF, SNF, and discriminant-group computations of Proposition 5.3 run in polynomial time in the bit length of their inputs.

Convention. In view of the above, all running times suppress a multiplicative $\mathrm{poly}(n, \mathrm{bits})$ factor, where bits is the encoding length of the input Gram matrices. The scaling convention and the polynomial-bit accounting are applied in the proof of Theorem 7.3.

2.2 Lattice isomorphism and hard problems

Definition 2.3 (Lattice isomorphisms and automorphisms)**.** *For full-rank lattices $\mathcal{L}_1, \mathcal{L}_2 \subseteq \mathbb{R}^n$, define*

$$\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \{O \in O_n(\mathbb{R}) : O\mathcal{L}_1 = \mathcal{L}_2\}.$$

The automorphism group of $\mathcal{L}$ is $\mathrm{Aut}(\mathcal{L}) = \mathrm{Iso}(\mathcal{L}, \mathcal{L})$. If $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ is nonempty, it is a coset of $\mathrm{Aut}(\mathcal{L}_1)$.

All algorithmic problems below use Gram matrices as inputs. Thus an input lattice is given by an integral Gram matrix $G \in \mathcal{S}_n^{>0}(\mathbb{Z})$, and isomorphisms are represented by coordinate change matrices $U \in \mathrm{GL}_n(\mathbb{Z})$. This convention avoids bit-complexity issues caused by arbitrary real bases and includes the usual integral-lattice model.[4]

For Gram matrices, we use a different symbol for coordinate stabilizers. We write

$$\mathrm{Stab}(G) = \{U \in \mathrm{GL}_n(\mathbb{Z}) : G = U^\top G U\}$$

[4]Every operation we use is intrinsic to the Gram matrix; no ambient real basis is needed: shortest-vector enumeration; discrete-Gaussian sampling; duals, orthogonal projections, section, quotient, and residual forms; discriminant groups and glue (overlattice) compatibility; HNF/SNF and canonical forms; and coordinate isomorphism verification. This representation is standard in the quadratic-form treatment of lattice isomorphism; see [HR14, DvW22, vW24, Hun24].

for the stabilizer of the quadratic form G, and for $G_1, G_2 \in \mathcal{S}_n^{>0}(\mathbb{Z})$,

$$\mathrm{Iso}(G_1, G_2) = \{U \in \mathrm{GL}_n(\mathbb{Z}) : G_1 = U^\top G_2 U\}$$

for the corresponding coordinate isomorphisms. We have the following connections between the basis form and Gram matrix form in isomorphism:

Lemma 2.4 ([JWL+25b, Lemma 2.9])**.** *Let $\mathcal{L}$ be a lattice with basis B, and let $G = B^\top B$. Then* $\mathrm{Stab}(G) \cong \mathrm{Aut}(\mathcal{L})$*. The map ϕ defined by $\phi(U) = BUB^{-1}$, for all $U \in \mathrm{Stab}(G)$, is an isomorphism from* $\mathrm{Stab}(G)$ *to* $\mathrm{Aut}(\mathcal{L})$*.*

Next, we define the decision and search versions of the Lattice Isomorphism Problem.

Definition 2.5 (Decision Lattice Isomorphism Problem (DLIP))**.** *Given $G_1, G_2 \in \mathcal{S}_n^{>0}(\mathbb{Z})$,* DLIP *is to decide whether* $\mathrm{Iso}(G_1, G_2) \neq \varnothing$*.*

Definition 2.6 (Search Lattice Isomorphism Problem (SLIP))**.** *Given $G_1, G_2 \in \mathcal{S}_n^{>0}(\mathbb{Z})$ with* $\mathrm{Iso}(G_1, G_2) \neq \varnothing$*,* SLIP *is to output a matrix $U \in \mathrm{Iso}(G_1, G_2)$.*

Next, we define the search version of the Shortest Vector Problem.

Definition 2.7 (Search Shortest Vector Problem (SVP))**.** *Given as input a Gram matrix $G \in \mathcal{S}_n^{>0}(\mathbb{Z})$ of an integral lattice $\mathcal{L}$, the goal is to output a nonzero vector $v \in \mathcal{L}$ such that $\|v\| = \lambda_1(\mathcal{L})$.*

In our paper, we use the following standard facts.

Lemma 2.8 (Dual invariance)**.** *For any full-rank lattices $\mathcal{L}_1, \mathcal{L}_2$,*

$$\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \mathrm{Iso}(\mathcal{L}_1^*, \mathcal{L}_2^*).$$

Proof. If $O\mathcal{L}_1 = \mathcal{L}_2$ and $y \in \mathcal{L}_1^*$, then for every $x \in \mathcal{L}_2$ we may write $x = Oz$ with $z \in \mathcal{L}_1$, and

$$\langle Oy, x\rangle = \langle y, z\rangle \in \mathbb{Z}.$$

Thus $O\mathcal{L}_1^* \subseteq \mathcal{L}_2^*$. Applying the same argument to O^{-1} gives equality. Hence $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) \subseteq \mathrm{Iso}(\mathcal{L}_1^*, \mathcal{L}_2^*)$. The reverse inclusion follows by dualizing again, since $(\mathcal{L}_i^*)^* = \mathcal{L}_i$. □

Lemma 2.9 (Section/quotient recursion, adapted from [HR14, Lemma 4.3])**.** *Let $\mathcal{L}$ be an integral lattice that is not well-rounded, and set $V = \mathrm{span}(\mathrm{Min}(\mathcal{L}))$, $A = \mathcal{L} \cap V$, $Q = \pi_{V^\perp}(\mathcal{L})$. Then A, Q are lattices of ranks* $\dim(V)$ *and* $\mathrm{rank}(\mathcal{L}) - \dim(V)$*, and:*

(i) Reconstruction: *The map*

$$\tau_{\mathcal{L}} : Q \to D(A) = A^*/A, \qquad \tau_{\mathcal{L}}(q) = \pi_V(x) + A \text{ for any lift } x \in \mathcal{L} \text{ of } q \in Q,$$

is a well-defined group homomorphism, and

$$\mathcal{L} = \{\, a + u(q) + q : a \in A,\ q \in Q \,\},$$

where $u(q) \in A^*$ is any representative of $\tau_{\mathcal{L}}(q)$.

(ii) Gluing: *If $\mathcal{L}_1, \mathcal{L}_2$ are both not well-rounded, every isomorphism between them maps V_1 to V_2, and*

$$\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \{\, O_A \oplus O_Q : O_A \in \mathrm{Iso}(A_1, A_2),\ O_Q \in \mathrm{Iso}(Q_1, Q_2),\ \overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q \,\},$$

where $O_A \oplus O_Q$ acts as O_A on V_1 and O_Q on $V_1^\perp$, and $\overline{O}_A : D(A_1) \to D(A_2), a + A_1 \mapsto O_A(a) + A_2$ is the induced map on discriminant groups.

Proof. Since $\mathrm{Min}(\mathcal{L})$ spans V and $\mathrm{Min}(\mathcal{L}) \subseteq A = \mathcal{L} \cap V$, A has rank $k := \dim V$, and $k < \mathrm{rank}(\mathcal{L})$ because $\mathcal{L}$ is not well-rounded. As V is spanned by lattice vectors, it is a rational subspace of $\mathcal{L}$, so $V^\perp$ is rational and $Q = \pi_{V^\perp}(\mathcal{L})$ is a lattice of rank $\mathrm{rank}(\mathcal{L}) - k$ in $V^\perp$.

i. Reconstruction. For $q \in Q$ choose a lift $x \in \mathcal{L}$ with $\pi_{V^\perp}(x) = q$, and write $x = x_V + q$. For every $a \in A$, using $a \in V$, $q \in V^\perp$, and integrality of $\mathcal{L}$,

$$\langle x_V, a\rangle = \langle x - q, a\rangle = \langle x, a\rangle \in \mathbb{Z},$$

so $x_V \in A^*$. Note that two lifts of q differ by an element of $\mathcal{L} \cap V = A$, so $x_V + A$ is independent of the lift. The addition of lifts shows $\tau_{\mathcal{L}}(q) = x_V + A$ is a well-defined homomorphism $Q \to D(A)$.

For any representative $u(q) \in A^*$ of $\tau_{\mathcal{L}}(q)$, the vectors in $\mathcal{L}$ projecting to q form the set

$$u(q) + q + A\ .$$

Note that this set is independent of the choice of $u(q)$. Taking the union over q gives the reconstruction $\mathcal{L} = \{\, a + u(q) + q : a \in A,\ q \in Q \,\}$.

ii. Gluing. Let $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$. As an isometry O preserves norms, so $O(\mathrm{Min}(\mathcal{L}_1)) = \mathrm{Min}(\mathcal{L}_2)$, $O(V_1) = V_2$ and $O(V_1^\perp) = V_2^\perp$. Thus O decomposes into $O_A \oplus O_Q$ on $\mathbb{R}^n = V_1 \oplus V_1^\perp$, and $O_A := O|_{V_1} : A_1 \to A_2$ is an isometry. On the other hand, for all $x_i \in \mathcal{L}_i$, we have $O_Q \pi_{V_1^\perp}(x_1) = \pi_{V_2^\perp}(Ox_1) \in Q_2$, $\pi_{V_2^\perp}(x_2) = O_Q \pi_{V_1^\perp}(O^{-1}x_2) \in O_Q(Q_1)$. So O_Q is an isometry between Q_1, Q_2. For any $q \in Q_1$ and a lift $x = x_V + q$, the point $Ox = O_A x_V + O_Q q$ lifts $O_Q q$, so

$$\tau_{\mathcal{L}_2}(O_Q q) = O_A x_V + A_2 = \overline{O}_A\big(\tau_{\mathcal{L}_1}(q)\big),$$

i.e. $\overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q$. Hence every isometry has the stated form.

Conversely, let $O_A \in \mathrm{Iso}(A_1, A_2)$ and $O_Q \in \mathrm{Iso}(Q_1, Q_2)$ satisfy $\overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q$, and put $O = O_A \oplus O_Q$, an isometry of $\mathbb{R}^n$ since $V_1 \perp V_1^\perp$. For $a + u(q) + q \in \mathcal{L}_1$ as in (i),

$$O(a + u(q) + q) = O_A a + O_A u(q) + O_Q q.$$

Here $O_A a \in A_2$, and $O_A u(q) \in A_2^*$ (Lemma 2.8) represents $\overline{O}_A(\tau_{\mathcal{L}_1}(q)) = \tau_{\mathcal{L}_2}(O_Q q)$, hence is a valid representative of $\tau_{\mathcal{L}_2}(O_Q q)$. By the reconstruction of $\mathcal{L}_2$, $O_A a + O_A u(q) + O_Q q$ lies in $\mathcal{L}_2$, so $O\mathcal{L}_1 \subseteq \mathcal{L}_2$; the same argument for $O^{-1} = O_A^{-1} \oplus O_Q^{-1}$ gives equality. Therefore $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$. □

Remark 2.10 (Coordinate form of the reconstruction)**.** *Item* (i) *determines $\mathcal{L}$ from $(A, Q, \tau_{\mathcal{L}})$ with no reference to the ambient form, so it applies verbatim to Gram matrices. Concretely, fix $\mathbb{Z}$-bases $a_1, \ldots, a_k$ of A and $q_1, \ldots, q_{m-k}$ of Q (here $m = \mathrm{rank}(\mathcal{L})$), with Gram matrices $G_A = (\langle a_i, a_j\rangle)$,*

$G_Q = (\langle q_i, q_j \rangle)$, *and put* $u_j := u(q_j) \in A^*$. *Item (i) implies the vectors*

$$a_1, \ldots, a_k, \;\; u_1 + q_1, \ldots, u_{m-k} + q_{m-k}$$

form a $\mathbb{Z}$*-basis of* $\mathcal{L}$*, in which* $\mathcal{L}$ *has the block Gram matrix*

$$G_{\mathcal{L}} = \begin{pmatrix} G_A & U \\ U^\top & G_Q + U^\top G_A^{-1} U \end{pmatrix}, \qquad U = \big(\langle a_i, u_j \rangle\big) \in \mathbb{Z}^{k \times (m-k)}.$$

For the cross block, $\langle a_i, u_j + q_j \rangle = \langle a_i, u_j \rangle = U_{ij}$*, since* $a_i \in V \perp q_j$*. For the bottom-right corner,* $\langle u_i + q_i, u_j + q_j \rangle = \langle u_i, u_j \rangle + \langle q_i, q_j \rangle$*. Writing* $u_j = \sum_i U_{ij}\, a_i^*$ *in the dual basis, whose Gram is* G_A^{-1}*, gives* $\langle u_i, u_j \rangle = (U^\top G_A^{-1} U)_{ij}$*, hence* $G_Q + U^\top G_A^{-1} U$*.*

Reducing U *modulo* $G_A \mathbb{Z}^k$ *recovers* τ_L *on the chosen bases, so, with a fixed deterministic choice of lifts,* G_L *depends only on* (G_A, G_Q, τ_L)*.*

Theorem 2.11 ([Fei96])**.** *For* $n > 10$*, the integer lattice* $\mathbb{Z}^n$ *possesses the largest automorphism group among all* n*-dimensional lattices, with* $|\mathrm{Aut}(\mathbb{Z}^n)| = 2^n \cdot n!$*.*

Corollary 2.12. *Let* $B_{\mathrm{aut}}(n) = \max\{2^{11} \cdot 11!,\; 2^n \cdot n!\} \le O(n)^n$*. Then, for all rank-*$n$ *lattices* $\mathcal{L}_1, \mathcal{L}_2$*,* $|\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)| \le B_{\mathrm{aut}}(n) \le O(n)^n$*.*

Proof. If $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \varnothing$ the bound is trivial. Otherwise it is a coset of $\mathrm{Aut}(\mathcal{L}_1)$, so $|\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)| = |\mathrm{Aut}(\mathcal{L}_1)|$. Now if $n \ge 11$, Theorem 2.11 gives $|\mathrm{Aut}(\mathcal{L}_1)| \le 2^n n! \le B_{\mathrm{aut}}(n)$.

It remains to handle $1 \le n \le 10$. Let $m = 11 - n$ and form the orthogonal direct sum $\mathcal{L}_1^+ := \mathcal{L}_1 \oplus \mathbb{Z}^m$. The map $\mathrm{Aut}(\mathcal{L}_1) \longrightarrow \mathrm{Aut}(\mathcal{L}_1^+) : A \longmapsto A \oplus I_m$ is injective. Since $\mathcal{L}_1^+$ has rank 11, Theorem 2.11 implies $|\mathrm{Aut}(\mathcal{L}_1)| \le |\mathrm{Aut}(\mathcal{L}_1^+)| \le 2^{11} 11! = B_{\mathrm{aut}}(n)$. □

2.3 Discrete Gaussians and smoothing

For $s > 0$ and $x \in \mathbb{R}^n$, define

$$\rho_s(x) = e^{-\pi \|x\|^2 / s^2}.$$

For a countable set A, let $\rho_s(A) = \sum_{x \in A} \rho_s(x)$.

Definition 2.13 (Discrete Gaussian and smoothing parameter)**.** *For a lattice* $\mathcal{L}$ *and parameter* $s > 0$*, the discrete Gaussian* $D_{\mathcal{L},s}$ *is the distribution on* $\mathcal{L}$ *given by*

$$\Pr_{X \sim D_{\mathcal{L},s}}[X = x] = \frac{\rho_s(x)}{\rho_s(\mathcal{L})}.$$

The smoothing parameter $\eta_\varepsilon(\mathcal{L})$ *is defined as the smallest* $s > 0$ *such that*

$$\rho_{1/s}(\mathcal{L}^* \setminus \{0\}) \le \varepsilon.$$

Let γ_s denote the continuous Gaussian distribution on $\mathbb{R}^n$ with density $s^{-n}\rho_s(x)$. This is the Gaussian of covariance $s^2(2\pi)^{-1} I$.

We use the following smoothing bound from the shortest nonzero vector of the dual lattice.

Theorem 2.14 (Smoothing bound [MR07, Lemma 3.2]). *For every rank-n lattice $\mathcal{L}$ and $\varepsilon = 2^{-n}$,*

$$\eta_{2^{-n}}(\mathcal{L}) \le \frac{\sqrt{n}}{\lambda_1(\mathcal{L}^*)}.$$

We also use the following two elementary consequences of smoothing.

Lemma 2.15 (Adapted from [MR07, Lemma 4.1]). *Let $\mathcal{L}$ be a rank-n lattice and let $a \ge \eta_\varepsilon(\mathcal{L})$. Then for every shift $c \in \mathbb{R}^n$,*

$$\rho_a(\mathcal{L} + c) = (1 + \delta_c)\frac{a^n}{\det(\mathcal{L})}, \qquad |\delta_c| \le \varepsilon.$$

In particular, if $b \ge a$, then

$$\rho_b(\mathcal{L}) \in \frac{b^n}{\det(\mathcal{L})} \cdot (1 - \varepsilon, 1 + \varepsilon).$$

Proof. By Poisson summation,

$$\rho_a(\mathcal{L} + c) = \frac{a^n}{\det(\mathcal{L})} \sum_{y \in \mathcal{L}^*} \rho_{1/a}(y) e^{2\pi i \langle y, c \rangle}.$$

The $y = 0$ term is 1. Since $a \ge \eta_\varepsilon(\mathcal{L})$, the sum of the absolute values of all nonzero Fourier terms is at most ε. This gives the uniform shifted estimate. The last assertion is the special case $c = 0$, using monotonicity of the smoothing condition in the parameter. □

The following result is implicit in the proof of [Reg09, Claim 3.9].

Lemma 2.16 (Discrete Gaussian plus continuous Gaussian [Reg09]). *Let $\mathcal{L}$ be a rank-n lattice. Let $X \sim D_{\mathcal{L},s}$ and $Y \sim \gamma_t$ be independent. Put*

$$r = \sqrt{s^2 + t^2}, \qquad u = \frac{st}{\sqrt{s^2 + t^2}}.$$

If $\varepsilon < \frac{1}{2}$ and $u \ge \eta_\varepsilon(\mathcal{L})$, then the density of $X + Y$ is pointwise within a factor $1 \pm O(\varepsilon)$ of the density of γ_r. In particular, $\mathrm{TV}(X + Y, \gamma_r) = O(\varepsilon)$.

And in Section 4, we need the following analytical result.

Lemma 2.17 (Integration error for linear-growth test functions). *In the setting of Lemma 2.16, let $F : \mathbb{R}^n \to \mathbb{R}_{\ge 0}$ satisfy*

$$0 \le F(z) \le L_F \cdot \|z\| \qquad \text{for all } z.$$

Then

$$|\mathbb{E}[F(X + Y)] - \mathbb{E}_{Z \sim \gamma_r}[F(Z)]| \le O(\varepsilon L_F r \sqrt{n}).$$

Proof. Let $p_{s,t}$ be the density of $X + Y$ and let $q_r(z) = r^{-n}\rho_r(z)$ be the density of γ_r. Lemma 2.16 gives

$$p_{s,t}(z) = q_r(z)(1 + \Delta(z)), \qquad |\Delta(z)| \le C\varepsilon$$

pointwise. Therefore the error can be integrated directly, without any truncation:

$$\begin{aligned}|\mathbb{E}[F(X+Y)] - \mathbb{E}_{Z\sim\gamma_r}[F(Z)]| &= \left|\int_{\mathbb{R}^n} F(z)(p_{s,t}(z) - q_r(z))\,dz\right| \\ &\le C\varepsilon \int_{\mathbb{R}^n} F(z) q_r(z)\,dz \\ &\le C\varepsilon L_F \mathbb{E}_{Z\sim\gamma_r}\|Z\|\,.\end{aligned}$$

For our normalization of γ_r,

$$\mathbb{E}\,\|Z\| \le (\mathbb{E}[\|Z\|^2])^{1/2} = r\sqrt{n/(2\pi)}.$$

This proves the displayed bound. □

2.4 Shortest-vector enumeration and sampling

We use the following standard algorithmic facts.

Theorem 2.18 (Adapted from [MV13, Corollaries 5.5 and 5.8])**.** *Let $\mathcal{L}$ be a rational n-dimensional lattice. There is a deterministic $2^{O(n)}$ time and space algorithm that outputs the set* $\mathrm{Min}(\mathcal{L})$.

Proof. One call of exact SVP takes deterministic $2^{O(n)}$ time and space, producing $\lambda := \lambda_1(\mathcal{L})$ [MV13, Corollary 5.5]. Then, the Voronoi-cell algorithm [MV13, Corollary 5.8] with target 0 and parameter $k = 2$, running in deterministic $2^{O(n)}$ time and space, lists all points of $\mathcal{L} \cap 2\lambda B_2^n$, a set of size at most 5^n by Lemma 2.19. Filtering by the test $\|v\|^2 = \lambda^2$ retains $\mathrm{Min}(\mathcal{L})$. □

Lemma 2.19 ([HR14, Fact 2.2])**.** *For every lattice $\mathcal{L}$ of rank n and for every $a > 0$, the number of vectors in $\mathcal{L}$ of norm at most $a \cdot \lambda_1(\mathcal{L})$ is at most $(2a+1)^n$.*

We also need the following result on discrete-Gaussian sampling. The original statement in [ADRS15, Theorem 3.7] is stated in a batched version, while we only use the first sample from the batch in our application.

Theorem 2.20 ([ADRS15, Theorem 3.7])**.** *For every rank-n rational lattice $\mathcal{L}$, every $s > 0$ and every confidence parameter $\kappa \ge n$, there exists a sampling algorithm that produces a sample at total variation distance $\exp(-\Omega(\kappa))$ to $D_{\mathcal{L},s}$, in randomized time $2^{O(n)+\mathrm{poly}\log(\kappa)}$.*

The next implementation lemma gives a sampler on finitely many uniformly random bits; the proof is given in Appendix C.

Lemma 2.21 (Finite-tape ADRS realization)**.** *Let $\mathcal{L}$ be a rank-n rational lattice, let $s > 0$ be efficiently computable, and let $\kappa \ge Cn$ for a sufficiently large absolute constant C. There exist integers b, G and a uniformly constructible deterministic Boolean circuit*

$$F_{\mathcal{L},s,\kappa} : \{0,1\}^b \longrightarrow \mathcal{L} \cup \{\perp\}$$

of size at most G, where the output lattice vector is represented by its integer coordinates in the displayed basis, such that

$$b, G \le 2^{O(n)+\mathrm{poly}\log(\kappa)}\mathrm{poly}(\mathrm{bits}(\mathcal{L}, s)).$$

For a uniform seed $R \in \{0,1\}^b$, the law of $F_{\mathcal{L},s,\kappa}(R)$ is within total variation distance $\exp(-\Omega(\kappa))$ of $D_{\mathcal{L},s}$, where the latter assigns mass zero to $\perp$. Every nonfailure output also satisfies $\|F_{\mathcal{L},s,\kappa}(R)\| \le s\sqrt{\kappa}$.

We also use the following tail bound for discrete Gaussian samples.

Lemma 2.22 (Adapted from [Ban93, Lemma 1.5])**.** *For every rank-n lattice $\mathcal{L}$, every $s > 0$, and every $\kappa \ge Cn$, a sample $X \sim D_{\mathcal{L},s}$ satisfies*

$$\Pr[\|X\| > s\sqrt{\kappa}] \le \exp(-\Omega(\kappa)).$$

Proof. In [Ban93, Lemma 1.5], it is shown that for every rank-n lattice $\mathcal{L}$ and every $c \ge 1/\sqrt{2\pi}$,

$$\rho_1\big(\mathcal{L} \setminus c\sqrt{n}\, B_2^n\big) \le \big(c\sqrt{2\pi e}\, e^{-\pi c^2}\big)^n \rho_1(\mathcal{L}),$$

where B_2^n is the Euclidean unit ball. Hence, by scaling $\mathcal{L}$ to $\mathcal{L}/s$ and taking $c = \sqrt{\kappa/n}$, we obtain the desired bound for $X \sim D_{\mathcal{L},s}$ when the absolute constant C is sufficiently large. □

We will use the discrete-Gaussian sampler algorithm in Theorem 2.20, and the norm cutoff from Lemma 2.22 with $\kappa = C_0\, n \log n$ for a sufficiently large absolute constant C_0, where n is the dimension of the input lattice. We will pass κ unchanged to every recursive call, with possibly lower dimensional lattices. If an algorithm makes T calls to the discrete-Gaussian sampler, then a standard hybrid argument and a union bound show that replacing ideal samples by implemented samples, and aborting whenever some sample has norm larger than $s\sqrt{\kappa}$, changes any success probability by at most $T \cdot \exp(-\Omega(\kappa))$. In our algorithms $T \le O(n\sqrt{\log n})^n \mathrm{poly}(\kappa)$, so this is $2^{-\Omega(n \log n)}$ for sufficiently large C_0. On every non-aborting run, all sampled vectors have norm at most $s\sqrt{\kappa}$, and hence have coordinate bit length polynomial in n and in the input bit length.

2.5 Matroids

For a set S and element s, we write $S \cup \{s\}$ as $S + s$ and $S \setminus \{s\}$ as $S - s$ if there is no ambiguity. The concept of a matroid generalizes linear independence; it is defined as follows.

Definition 2.23 (Matroid)**.** A (finite) matroid $\mathcal{M}$ is a pair $(E, \mathcal{I})$, where E is called the ground set, and $\mathcal{I} \subseteq \mathcal{P}(E)$. Sets in $\mathcal{I}$ are called independent sets. They satisfy the following properties:

1. $\varnothing \in \mathcal{I}$.
2. For any independent set $S \in \mathcal{I}$, and subset $S' \subseteq S$, we have $S' \in \mathcal{I}$ as well.
3. For $S, T \in \mathcal{I}$ and $|S| > |T|$, there exists an element $x \in S \setminus T$ such that $T + x \in \mathcal{I}$.

A basis of $\mathcal{M}$ is a maximal independent set $B \in \mathcal{I}$, i.e., for any $x \in E \setminus B$, $B + x \notin \mathcal{I}$.

By the third property of a matroid, all bases have equal size, which is compatible with the usual notion of a basis of a finite-dimensional linear space. We use only the matroid of lattice vectors. The ground set will be the shortest-vector shell $A = \mathrm{Min}(\mathcal{L})$, and independence means linear independence over $\mathbb{R}$.

Definition 2.24 (Vector matroid)**.** *For a finite set $A \subseteq \mathbb{R}^n$, the vector matroid on A has independent sets*

$$\mathcal{I} = \{B \subseteq A : B \text{ is linearly independent}\} .$$

For $j \in [n]$ consider the rank-j truncation $\mathcal{M}_j = (A, \mathcal{I}_j)$, where

$$\mathcal{I}_j = \{B \subseteq A : |B| \leq j, B \text{ is linearly independent}\} .$$

This again gives a matroid.

The greedy algorithm is efficient and useful for finding a maximum-weight independent set in a matroid.

Theorem 2.25 (Matroid greedy [Edm71])**.** *For a matroid $\mathcal{M} = (E, \mathcal{I})$ with weights $w : E \to \mathbb{R}$, one run of the greedy algorithm, which repeatedly adds the heaviest-weight element in E that preserves independence, returns a nested sequence $G_0 \subset G_1 \subset \cdots \subset G_\rho$ such that G_j is a maximum-weight independent set among all independent sets of cardinality j. The statement holds for any deterministic tie-breaking rule.*

We use the following strong basis exchange property implicit in [Bru69]. Note that even if the original theorem only proved the exchange between bases, the argument can be extended to any independent sets of equal size j by considering the matroid $\mathcal{M}_j$.

Theorem 2.26 ([Bru69])**.** *For a matroid $\mathcal{M} = (A, \mathcal{I})$ of lattice vectors as defined above, if B, C are distinct independent sets in $\mathcal{I}$ of the same size, and $b \in B \setminus C$, then there exists $c \in C \setminus B$ such that both $B - b + c$ and $C - c + b$ are independent sets.*

3 Projected Rank Polytopes and Ambiguous Ranks

In our application, a polytope P is the convex hull of a finite set of points S in a real vector space. We say that a point $p \in P$ is a vertex if it is not a nontrivial convex combination of other points in P. For $P = \mathrm{conv}(S)$, its vertices form a subset of S.

Throughout this section, $\mathcal{L}$ is a rank-n well-rounded lattice,

$$A = \mathrm{Min}(\mathcal{L}), \qquad \lambda = \lambda_1(\mathcal{L}) = \lambda_n(\mathcal{L}),$$

and $\mathcal{M} = (A, \mathcal{I})$ is the vector matroid on A.

For an independent set $B \in \mathcal{I}$ we write

$$p_B = \sum_{b \in B} b.$$

Definition 3.1 (Projected polytope and support function)**.** *The rank-j* projected polytope *of $\mathcal{L}$ is*

$$P_j = \operatorname{conv}\{p_B : B \in \mathcal{I}, |B| = j\}.$$

Let its support function be

$$h_j(w) = h_{P_j}(w) = \max_{p \in P_j} \langle w, p \rangle.$$

Since $A = -A$, the polytope P_j is centrally symmetric, so $h_j(w) \geq 0$ for all $w \in \operatorname{span}(\mathcal{L})$. Also h_j is the pointwise maximum of linear maps, so it is a convex function.

Definition 3.2 (Resolved and ambiguous ranks)**.** *For $w \in \mathcal{L}^*$, rank j is called* resolved *at w if the maximum-achieving set*

$$F_j(w) = \arg\max_{p \in P_j} \langle w, p \rangle$$

is a single vertex of P_j. Otherwise rank j is ambiguous *at w.*

3.1 Properties of polytopes

Lemma 3.3 (Exposed faces of P_j)**.** *Let $V_j = \operatorname{Vert}(P_j)$ be the set of vertices. For every $w \in \mathbb{R}^n$,*

$$F_j(w) = \{p \in P_j : \langle w, p \rangle = h_j(w)\} = \operatorname{conv}\{v \in V_j : \langle w, v \rangle = h_j(w)\}.$$

In particular, $F_j(w)$ is nonempty. If $F_j(w)$ is not a single vertex, then it contains two distinct vertices of P_j.

Proof. The polytope P_j is compact and nonempty, the linear functional $p \mapsto \langle w, p \rangle$ attains its maximum on P_j. Every point $x \in P_j$ can be written as a convex combination of vertices, and each vertex has score at most $h_j(w)$. So their convex combination has score $h_j(w)$ if and only if every vertex v with nonzero coefficient in the combination attains score $h_j(w)$. Hence

$$F_j(w) = \operatorname{conv}\{v \in V_j : \langle w, v \rangle = h_j(w)\}.$$

If $F_j(w)$ is not a single vertex, then it must contain at least two vertices. □

Lemma 3.4 (Projected vertices are uniquely represented)**.** *Let p be a vertex of P_j. Then there is a unique independent set $B \subseteq A$ of size j with*

$$p = p_B = \sum_{b \in B} b.$$

We call B the basis representation of the vertex p.

Proof. The existence of B with $p_B = p$ is guaranteed since a vertex of a polytope must correspond to some independent set. Suppose $B \neq C$ are independent sets of size j and $p_B = p_C = p$. Regard B and C as bases of the vector matroid $\mathcal{M}_j$. By Theorem 2.26, there exist $b \in B \setminus C$ and $c \in C \setminus B$ such that

$$B' = B - b + c, \qquad C' = C - c + b$$

are independent sets of size j. Then

$$p_{B'} = p + c - b, \qquad p_{C'} = p - (c - b).$$

Since $b \neq c$ as vectors of the ground set, $c - b \neq 0$. Hence $p_{B'}$ and $p_{C'}$ are distinct points of P_j and

$$p = \frac{1}{2}p_{B'} + \frac{1}{2}p_{C'}.$$

Therefore p is not a vertex. This proves the lemma. □

3.2 Certifying ambiguity and bounding the residual rank

In the following lemma, we study the property that certifies an ambiguous rank.

Lemma 3.5 (Zero-slack exchange)**.** *If rank j is ambiguous at w, and B is any independent j-set with $p_B \in F_j(w)$, then there exist $b \in B$ and $x \in A \setminus B$ such that $B - b + x$ is independent and*

$$\langle w, b - x \rangle = 0.$$

We call this a zero-slack exchange.

Proof. Since $F_j(w)$ is not a single vertex, it contains at least two distinct vertices of P_j by Lemma 3.3. If p_B is not a vertex, choose any vertex p_C of $F_j(w)$; otherwise choose a vertex $p_C \neq p_B$ in the face. In either case $p_C \neq p_B$. By Lemma 3.4, let C be the unique independent j-set with p_C as its projected sum. Then $B \neq C$.

By Theorem 2.26, there exist $b \in B \setminus C$ and $c \in C \setminus B$ such that $B' := B - b + c$ and $C' := C - c + b$ are independent j-sets. The two exchanged sets have support values

$$\langle w, p_{B'} \rangle = h_j(w) + \langle w, c - b \rangle, \qquad \langle w, p_{C'} \rangle = h_j(w) - \langle w, c - b \rangle.$$

If $\langle w, c - b \rangle \neq 0$, one of these values is strictly larger than $h_j(w)$, contradicting maximality. Hence $\langle w, b - c \rangle = 0$. Taking $x = c$ gives a zero-slack feasible exchange. □

Next we give a bound on residual rank based on the number of ambiguous ranks.

Lemma 3.6 (Nesting of resolved greedy prefixes)**.** *Let $a < b$ be resolved ranks at the same w. Let B_a and B_b be the unique independent-set representations of the exposed vertices $p_a \in F_a(w)$ and $p_b \in F_b(w)$. Then $B_a \subseteq B_b$.*

Proof. Run the matroid greedy algorithm with weights $x \mapsto \langle w, x \rangle$ until it has selected b elements, using any deterministic tie-breaking rule. By Theorem 2.25, its k-element prefix is a maximum-weight independent k-set for every $k \leq b$.

Since rank b is resolved, every maximum-weight independent b-set has projected sum in the singleton face $F_b(w) = \{p_b\}$. Lemma 3.4 then forces the greedy b-set to be B_b. Its a-element prefix is a maximum-weight independent a-set, so its projected sum lies in $F_a(w) = \{p_a\}$; applying Lemma 3.4 again, the prefix is B_a. Hence B_a is a prefix of, and therefore a subset of, B_b. □

Lemma 3.7 (Residual rank bound)**.** *Let*

$$\mathrm{Amb}(w) = |\{j \in [n] : j \text{ is ambiguous at } w\}|.$$

Let $J(w) = \{j \in [n] : j-1$ *and* j *are resolved at* $w\}$*, where rank* 0 *is declared resolved and* $B_0 = \varnothing$*. We also say that the step* j *is* fixed*, and for* $j \in J(w)$ *define* fixed vectors

$$u_j(w) = p_j(w) - p_{j-1}(w).$$

Then the vectors $\{u_j(w) : j \in J(w)\}$ *are linearly independent vectors in* A*, and*

$$|J(w)| \geq n - 2\mathrm{Amb}(w).$$

Consequently the number of steps left unfixed is at most $2\mathrm{Amb}(w)$*.*

Proof. For each resolved rank j, the set $F_j(w)$ is a single vertex p_j and, by Lemma 3.4, has a unique independent-set representation B_j.

By Lemma 3.6, the unique independent sets B_j at the resolved ranks are nested. It follows that for every fixed step $j \in J(w)$,

$$B_j = B_{j-1} \cup \{u_j(w)\},$$

where we identify $u_j(w)$ with the unique shortest vector in the set difference. All fixed vectors lie in B_q for the largest resolved rank q, so they are linearly independent.

A step j fails to be fixed only if rank j is ambiguous or rank $j-1$ is ambiguous, and rank 0 is never ambiguous; hence each ambiguous rank accounts for at most two non-fixed steps, so the number of non-fixed steps is at most $2\mathrm{Amb}(w)$. Since the fixed vectors are independent, their span has dimension $|J(w)| \geq n - 2\mathrm{Amb}(w)$. □

Together with Lemma 3.5, we have the following algorithmic result.

Lemma 3.8. *Given the shortest-vector shell* $A = \mathrm{Min}(\mathcal{L})$ *with* $|A| \leq 2^{O(n)}$*, and* $w \in \mathcal{L}^*$*, the resolved ranks and fixed vectors can be computed in* $2^{O(n)}$ *time.*

Proof. Run the matroid greedy algorithm with weights $x \mapsto \langle w, x\rangle$ to obtain a maximum-weight independent j-set B_j for $1 \leq j \leq n$. For each j, check all pairs $b \in B_j, x \in A \setminus B_j$ whether $B_j - b + x$ is independent and $\langle w, b - x\rangle = 0$. By Lemma 3.5, ambiguous ranks admit such an exchange. Conversely, any such exchange also gives two distinct maximizing points of P_j, so the rank is ambiguous.

The greedy algorithm and exchange checks take $\mathrm{poly}(n, |A|) \leq 2^{O(n)}$ time. At each resolved rank the unique exposed vertex is p_{B_j}, and we can then compute fixed vectors by subtracting these sums at consecutive resolved ranks. □

4 Gaussian Heat Comparison for Dual Markers

This section proves the core probabilistic estimate that a dual marker sampled from a discrete Gaussian will leave a small number of ambiguous ranks with constant probability. Let $\lambda = \lambda_1(\mathcal{L}) =$

$\lambda_n(\mathcal{L})$, we sample $W \sim D_{\mathcal{L}^*,s}$ for

$$s = \frac{Cn^2 \log n}{\lambda},$$

where C is a sufficiently large absolute constant. For the actual algorithm one may take $\lceil \log n \rceil$ in place of $\log n$.

4.1 Local lower bound

The following simple fact about Gaussian variables will be used in our analysis.

Fact 4.1. *Let $X \sim \gamma_a$ and fix $v \in \mathbb{R}^n$. Then $\langle X, v\rangle$ is a centered Gaussian of variance $\frac{a^2}{2\pi}\|v\|^2$, and*

$$\mathbb{E}|\langle X, v\rangle| = \frac{a\,\|v\|}{\pi}.$$

Proof. If $v = 0$ the claim is trivial. Otherwise, as a linear functional of the Gaussian vector X, the variable $Z := \langle X, v\rangle$ is Gaussian with mean $\langle \mathbb{E}[X], v\rangle = 0$ and

$$\mathrm{Var}[Z] = \sum_i v_i^2\ \mathrm{Var}[X_i] = \frac{a^2}{2\pi}\|v\|^2,$$

so $\sigma := \sqrt{\mathrm{Var}[Z]} = \frac{a}{\sqrt{2\pi}}\|v\|$. For $Z \sim N(0, \sigma^2)$,

$$\mathbb{E}|Z| = 2\int_0^\infty \frac{z}{\sqrt{2\pi}\,\sigma}\, e^{-z^2/2\sigma^2}\, dz = \sigma\sqrt{\tfrac{2}{\pi}}.$$

Substituting σ gives $\mathbb{E}|Z| = a\|v\|/\pi$. □

We use the following heat increment to bound the probability that rank j is ambiguous at w.

Definition 4.2 (Gaussian heat increment)**.** *Fix $t > 0$. The* Gaussian heat increment *of h_j at w is*

$$\Delta_j(w) := \mathbb{E}_{G_t \sim \gamma_t}[h_j(w + G_t)] - \mathbb{E}_{G_{t/2} \sim \gamma_{t/2}}[h_j(w + G_{t/2})].$$

The next lemma shows this increment is nonnegative at every w and $t > 0$.

Lemma 4.3 (Gaussian heat monotonicity)**.** *For the rank-j projected polytope P_j, $w \in \mathbb{R}^n$ and $0 < a \le b$,*

$$\mathbb{E}_{G_b \sim \gamma_b}[h_j(w + G_b)] \ge \mathbb{E}_{G_a \sim \gamma_a}[h_j(w + G_a)].$$

In particular, for fixed $t > 0$, the Gaussian heat increment $\Delta_j(w)$ is nonnegative for every w.

Proof. Write $G_b = G_a + G'$ with independent $G_a \sim \gamma_a$ and $G' \sim \gamma_{\sqrt{b^2 - a^2}}$. Since h_j is convex, fix G_a and use Jensen's inequality, we have

$$\mathbb{E}[h_j(w + G_a + G') \mid G_a] \ge h_j\big(w + G_a + \mathbb{E}[G']\big) = h_j(w + G_a).$$

Taking expectation over G_a proves the claim. □

Lemma 4.4 (Local heat lower bound)**.** *Let $w \in \mathcal{L}^*$. If rank j is ambiguous at w, then for every $t > 0$,*

$$\Delta_j(w) = \mathbb{E}_{G_t \sim \gamma_t}[h_j(w + G_t)] - \mathbb{E}_{G_{t/2} \sim \gamma_{t/2}}[h_j(w + G_{t/2})] \geq ct\lambda$$

for an absolute constant $c > 0$.

Proof. Since rank j is ambiguous at w, $F_j(w)$ contains two distinct vertices $p, q \in P_j$ with

$$\langle w, p\rangle = \langle w, q\rangle = h_j(w).$$

Because p and q are sums of lattice vectors in $\mathcal{L}$, $p - q \in \mathcal{L}$. Since they are distinct, $\|p - q\| \geq \lambda$. For $G_t \sim \gamma_t$,

$$h_j(w + G_t) \geq h_j(w) + \max\{\langle G_t, p\rangle, \langle G_t, q\rangle\}.$$

By Fact 4.1, the random variables $\langle G_t, p\rangle$ and $\langle G_t, q\rangle$ are centered Gaussians, so $\mathbb{E}[\langle G_t, p\rangle] = \mathbb{E}[\langle G_t, q\rangle] = 0$. Since for real numbers a, b, $\max\{a, b\} = \frac{a+b+|a-b|}{2}$,

$$\begin{aligned}\mathbb{E}[\max\{\langle G_t, p\rangle, \langle G_t, q\rangle\}] &= \frac{1}{2}\left(\mathbb{E}|\langle G_t, p - q\rangle| + \mathbb{E}[\langle G_t, p\rangle] + \mathbb{E}[\langle G_t, q\rangle]\right) \\ &= \frac{1}{2}\mathbb{E}|\langle G_t, p - q\rangle| \geq c_0 t \|p - q\| \geq c_0 t\lambda\ ,\end{aligned}$$

for some absolute constant $c_0 > 0$ (Fact 4.1). Let

$$f(t) = \mathbb{E}_{G_t \sim \gamma_t}[h_j(w + G_t)] - h_j(w).$$

Equivalently, with $G_1 \sim \gamma_1$, $f(t) = \mathbb{E}[h_j(w + tG_1)] - h_j(w)$. For every fixed G_1, the function $t \mapsto h_j(w + tG_1)$ is convex, hence f is convex (by linearity of $\mathbb{E}$) and $f(0) = 0$. Therefore $f(t/2) \leq \frac{1}{2}f(t)$. Since $f(t) \geq c_0 t\lambda$, we get

$$\Delta_j(w) = f(t) - f(t/2) \geq \frac{1}{2}f(t) \geq \frac{c_0}{2}t\lambda,$$

as required. □

4.2 Global upper bound

Choose $\varepsilon = 2^{-n}$, recall $\Delta_j(w) = \mathbb{E}_{G_t \sim \gamma_t}[h_j(w + G_t)] - \mathbb{E}_{G_{t/2} \sim \gamma_{t/2}}[h_j(w + G_{t/2})]$. In this subsection fix

$$t = 8\frac{\sqrt{n}}{\lambda}.$$

Then $t\lambda = \Theta(\sqrt{n})$ and, by Theorem 2.14, $\eta_\varepsilon(\mathcal{L}^*) \leq \frac{\sqrt{n}}{\lambda} = t/8$, because $(\mathcal{L}^*)^* = \mathcal{L}$. Since $s\lambda = Cn^2 \log n$ and C is chosen large, $s \gg t$; in particular, for all sufficiently large n we have $s \geq 2t$.

Lemma 4.5 (Global heat upper bound)**.** *For $W \sim D_{\mathcal{L}^*, s}$,*

$$\mathbb{E}_W[\Delta_j(W)] \leq O\left(\frac{t^2 j\lambda\sqrt{n}}{s}\right).$$

Proof. Let

$$r_1 = \sqrt{s^2 + t^2}, \qquad r_0 = \sqrt{s^2 + t^2/4}.$$

Observe that $0 \le h_j(z) \le j\lambda\|z\|$ for any z. To apply Lemma 2.17 with parameters (s,t) and $(s,t/2)$, we must check its smoothing hypothesis $u \ge \eta_\varepsilon(\mathcal{L}^*)$ for the corresponding

$$u_1 = \frac{st}{\sqrt{s^2 + t^2}}, \qquad u_0 = \frac{s(t/2)}{\sqrt{s^2 + t^2/4}}.$$

Since $s \ge 2t$, both $u_1 \ge t/\sqrt{5}$ and $u_0 \ge (t/\sqrt{5}) \cdot \frac{1}{2}$, so by the smoothing bound $\eta_\varepsilon(\mathcal{L}^*) \le t/8$ above we have $u_1, u_0 \ge \eta_\varepsilon(\mathcal{L}^*)$. Writing $G_a \sim \gamma_a$ for $a > 0$, we have

$$\mathbb{E}[h_j(W + G_t)] = \mathbb{E}_{Z\sim\gamma_{r_1}}[h_j(Z)] + O\big(\varepsilon j\lambda r_1\sqrt{n}\big),$$
$$\mathbb{E}[h_j(W + G_{t/2})] = \mathbb{E}_{Z\sim\gamma_{r_0}}[h_j(Z)] + O\big(\varepsilon j\lambda r_0\sqrt{n}\big).$$

Since $s \gg t$, both r_0 and r_1 are $s(1 + o(1))$, so the total integration error is

$$O(\varepsilon j\lambda s\sqrt{n}). \tag{4.1}$$

The ratio of this error to the claimed upper bound is

$$\frac{\varepsilon j\lambda s\sqrt{n}}{t^2 j\lambda\sqrt{n}/s} = \varepsilon\,\frac{s^2}{t^2} \le 2^{-n} \cdot \mathrm{poly}(n).$$

Thus the ratio is exponentially small and (4.1) is absorbed by the final $O(t^2 j\lambda\sqrt{n}/s)$ bound.

It remains to compare the two continuous Gaussian expectations. Since h_j is homogeneous of degree one,

$$\mathbb{E}_{Z\sim\gamma_r}[h_j(Z)] = r\,\mathbb{E}_{Z\sim\gamma_1}[h_j(Z)].$$

Since $\mathbb{E}_{Z\sim\gamma_1}\|Z\| \le C'\sqrt{n}$ for some constant C',

$$\mathbb{E}_{Z\sim\gamma_1}[h_j(Z)] \le C' j\lambda\sqrt{n}.$$

Therefore

$$\mathbb{E}_{Z\sim\gamma_{r_1}}[h_j(Z)] - \mathbb{E}_{Z\sim\gamma_{r_0}}[h_j(Z)] = (r_1 - r_0)\mathbb{E}_{Z\sim\gamma_1}[h_j(Z)] \le C'(r_1 - r_0) j\lambda\sqrt{n}.$$

Finally,

$$r_1 - r_0 = \frac{3t^2/4}{\sqrt{s^2 + t^2} + \sqrt{s^2 + t^2/4}} = O\big(\frac{t^2}{s}\big),$$

because $s \gg t$. Combining this with the absorbed error term proves

$$\mathbb{E}[h_j(W + G_t)] - \mathbb{E}[h_j(W + G_{t/2})] \le O\left(\frac{t^2 j\lambda\sqrt{n}}{s}\right).$$

□

4.3 Ambiguous-rank bound

Theorem 4.6 (Probability that a rank is ambiguous)**.** *Let $\mathcal{L}$ be rank-n well-rounded, $\lambda = \lambda_1(\mathcal{L})$, and*

$$s = \frac{Cn^2 \log n}{\lambda}.$$

There is an absolute constant $c_0 > 0$, independent of C, such that if $W \sim D_{\mathcal{L}^*,s}$, then for every $j \in [n]$,

$$\Pr[j \text{ is ambiguous at } W] \le \frac{c_0\, j}{C\, n \log n}.$$

Consequently,

$$\mathbb{E}[\mathrm{Amb}(W)] \le \frac{c_0\, n}{C \log n}.$$

Proof. Let $\mathcal{B}_j$ be the event that rank j is ambiguous at W, $G_a \sim \gamma_a$, and put

$$\Delta_j(W) = \mathbb{E}_{G_t \sim \gamma_t}[h_j(W + G_t)] - \mathbb{E}_{G_{t/2} \sim \gamma_{t/2}}[h_j(W + G_{t/2})].$$

By Lemma 4.3, $\Delta_j(W) \ge 0$ for every W. By Lemma 4.4, $\Delta_j(W) \ge ct\lambda$ on $\mathcal{B}_j$. Hence

$$\mathbf{1}_{\mathcal{B}_j} \le \frac{\Delta_j(W)}{ct\lambda} \qquad \text{for every } W \in \mathcal{L}^*.$$

Taking expectation over $W \sim D_{\mathcal{L}^*,s}$ and applying Lemma 4.5,

$$\Pr[\mathcal{B}_j] \le O\left(\frac{t^2 j \lambda \sqrt{n}/s}{t\lambda}\right) = O\left(\frac{tj\sqrt{n}}{s}\right) = \frac{c_0\, j}{C\, n \log n},$$

where the last step uses $t = \Theta(\sqrt{n}/\lambda)$ and $s\lambda = Cn^2 \log n$; the implied constants in Lemma 4.4 and Lemma 4.5 and in the choice of t do not depend on C, so neither does c_0. Summing over j gives

$$\mathbb{E}[\mathrm{Amb}(W)] \le \sum_{j=1}^{n} \frac{c_0\, j}{C\, n \log n} \le \frac{c_0\, n}{C \log n}.$$

□

Corollary 4.7 (Resolving markers)**.** *Once the absolute constant C in $s = Cn^2 \log n/\lambda$ satisfies $C \ge 100\, c_0$, with probability at least 0.99 over $W \sim D_{\mathcal{L}^*,s}$ the residual rank left by W is at most $\frac{2n}{\log n}$. The same conclusion holds for $W \sim D_{\mathcal{L}^*,s/\sqrt{2}}$ after increasing the absolute constant in the definition of s.*

Proof. By Theorem 4.6, $\mathbb{E}[\mathrm{Amb}(W)] \le c_0 n/(C \log n)$. Hence, by Markov's inequality,

$$\Pr\left[\mathrm{Amb}(W) > \frac{n}{\log n}\right] \le \frac{\log n}{n} \cdot \frac{c_0\, n}{C \log n} = \frac{c_0}{C} \le \frac{1}{100}$$

whenever $C \ge 100\, c_0$. Lemma 3.7 then gives residual rank at most $\frac{2n}{\log n}$. Replacing s by $s/\sqrt{2}$ only

changes the absolute constant C in $s = Cn^2 \log n/\lambda$. □

Call a marker w *resolving* if its residual rank is at most

$$r_*(n) := \lceil \frac{2n}{\log n} \rceil .$$

By Corollary 4.7, a sample from either $D_{\mathcal{L}^*,s}$ or $D_{\mathcal{L}^*,s/\sqrt{2}}$ is resolving with probability at least 0.99, provided the absolute constant in s is sufficiently large. We only use this constant success probability in the final algorithm, and the choice of $s = \Theta(n^2 \log n/\lambda)$ yields the collision scale $O(n^2 \log n)^n$ and the final running time $O(n\sqrt{\log n})^n$.

5 Small-Rank Canonical Forms for Residual-Glue Instances

We first collect the required algebraic conventions in Section 5.1, and give a lattice canonicalizer from the Haviv–Regev algorithm (construction deferred to Appendix B). Section 5.2 extends this construction to residual-glue instances: we canonicalize the residual sublattice, account for the residual marker and glue, and keep the fixed sublattice unchanged. On a first reading, readers may take the target primitive below as given and return to the constructions in this section as needed.

Target primitive. A residual-glue instance (Definition 5.13) is a tuple

$$\mathcal{G} = (\sigma, R, \mathcal{F}, H, \omega),$$

where σ is the scale factor, R is the residual lattice of rank r, $\mathcal{F}$ is the fixed data pinned down by the marker, H is the glue subgroup, and $\omega \in \mathrm{span}(R)$ is the residual marker.

We construct a canonical string $\mathrm{Can}(\mathcal{G})$. Two instances have the same canonical string if and only if their scales and fixed data agree and there is an isometry $O_R : R_1 \to R_2$ preserving the residual marker and glue. For matching instances, a lifting algorithm enumerates all such residual isometries in $r^{O(r)}$ time. In our application, $r = O(n/\log n)$, so this cost is $2^{O(n)}$.

5.1 Basic tools

We use *column* Hermite form unless otherwise stated, and fix a deterministic Smith normal form algorithm, including its transformation matrices. The existence and uniqueness of these normal forms are standard; see, e.g., [Coh93, Section 2.4]. Their polynomial-time computation, including transformation matrices, follows from [KB79]; see also [Sch86, Section 5.3].

Definition 5.1 (Hermite Normal Form)**.** *For $M \in \mathbb{Z}^{m\times n}$ with full column rank, let* $\mathrm{HNF}(M)$ *denote its unique* (column) Hermite normal form. *It satisfies* $\mathrm{HNF}(M) = MU$ *for some unimodular* $U \in \mathrm{GL}_n(\mathbb{Z})$. *For a full-rank sublattice $\mathcal{L} \subseteq \mathbb{Z}^d$ we write* $\mathrm{HNF}(\mathcal{L})$ *for the HNF of any basis matrix of $\mathcal{L}$.*

For the full row rank case, let $M \in \mathbb{Z}^{m\times n}$ have full row rank m. There is a unimodular $U \in \mathrm{GL}_n(\mathbb{Z})$ such that $MU = [\,H \mid 0\,]$, where the nonsingular column-Hermite block $H \in \mathbb{Z}^{m\times m}$ is unique; we define $\mathrm{HNF}(M) := H$.

Definition 5.2 (Smith Normal Form). *Let $G \in \mathcal{S}_d^{>0}(\mathbb{Z})$ be nonsingular. There exist $P, Q \in \mathrm{GL}_d(\mathbb{Z})$ with*

$$P\,G\,Q = \mathrm{diag}(s_1, \ldots, s_d), \qquad 1 \le s_1 \mid s_2 \mid \cdots \mid s_d.$$

The diagonal entries $s_1, \ldots, s_d$ are invariant factors *of G, and P, Q are transformation matrices.*

Proposition 5.3 (Existence, uniqueness, computability). *For every full column rank $M, N \in \mathbb{Z}^{m\times n}$, the matrix $\mathrm{HNF}(M)$ of Definition 5.1 exists and is unique. Also*

$$M\mathbb{Z}^n = N\mathbb{Z}^n \quad \Longleftrightarrow \quad \mathrm{HNF}(M) = \mathrm{HNF}(N).$$

Hence for full rank sublattices $\mathcal{L}_1, \mathcal{L}_2 \subseteq \mathbb{Z}^n$, $\mathrm{HNF}(\mathcal{L}_i)$ is independent of the chosen basis. And $\mathrm{HNF}(\mathcal{L}_1) = \mathrm{HNF}(\mathcal{L}_2) \Leftrightarrow \mathcal{L}_1 = \mathcal{L}_2$. All of the above holds in the full row rank case as well.

For every $G \in \mathcal{S}_d^{>0}(\mathbb{Z})$, the diagonal $(s_1, \ldots, s_d)$ of Definition 5.2 exists and is unique. The transformation matrices P, Q are not unique, but they can be computed as a deterministic function of G; we fix one such procedure throughout.

Both forms, together with the corresponding unimodular transforms, are computable in time polynomial in the bit-size of the input.

We now define a few other normal-form conventions.

Definition 5.4 (Full-rank rational module normal form). *Let $T \subseteq \mathbb{Q}^d$ be a full-rank $\mathbb{Z}$-module. Define $d_T = \min\{m \ge 1 : mT \subseteq \mathbb{Z}^d\}$. Then $d_T T$ is a full-rank sublattice of $\mathbb{Z}^d$, and we set*

$$\mathrm{NF}_{\mathrm{mod}}(T) = \big(d_T, \mathrm{HNF}(d_T T)\big).$$

By Proposition 5.3, $\mathrm{NF}_{\mathrm{mod}}(T_1) = \mathrm{NF}_{\mathrm{mod}}(T_2)$ if and only if $T_1 = T_2$ as submodules of $\mathbb{Q}^d$. The normal form is computable in polynomial time from any rational basis of T.

Definition 5.5 (Reduced rational coordinates). *A vector $v \in \mathbb{Q}^n$ is written in reduced form $v = \frac{1}{d}(a_1, \ldots, a_n)^\top$ with $a_1, \ldots, a_n \in \mathbb{Z}, d \in \mathbb{Z}_{>0}$ and $\gcd(d, a_1, \ldots, a_n) = 1$* (take $d = 1$ if v is the zero vector); $\mathrm{NF}_{\mathbb{Q}}(v)$ denotes this string. Equal $\mathrm{NF}_{\mathbb{Q}}$ is equivalent to being equal as vectors.

Definition 5.6 (Discriminant subgroup normal form). *Let $\mathcal{L}$ be integral with the Gram matrix G, and fix the Smith presentation $D(\mathcal{L}) \cong \bigoplus_{i:s_i>1} \mathbb{Z}/s_i\mathbb{Z}$. For $H \le D(\mathcal{L})$, $\mathrm{NF}_{D(\mathcal{L})}(H)$ is the canonical Hermite normal form of H inside this fixed Smith presentation* (construction in Appendix A).

Remark 5.7 (Notation for NF_D). *The subscript in $\mathrm{NF}_{D(\mathcal{L})}$ specifies the ambient discriminant group: $\mathrm{NF}_{D(\mathcal{L})}$ sends a subgroup $H \le D(\mathcal{L})$ to its canonical string in the fixed Smith presentation of $D(\mathcal{L})$. We write NF_D for $\mathrm{NF}_{D(\mathcal{L})}$ when $\mathcal{L}$ is clear from context.*

Proposition 5.8 (Faithfulness and computability of NF_D). *$\mathrm{NF}_{D(\mathcal{L})}$ is computable in polynomial time and, for $H_1, H_2 \le D(\mathcal{L})$,*

$$\mathrm{NF}_{D(\mathcal{L})}(H_1) = \mathrm{NF}_{D(\mathcal{L})}(H_2) \iff H_1 = H_2\ .$$

The proof is deferred to Appendix A.

Definition 5.9 (Overlattice from a glue subgroup)**.** *Let S be an integral lattice and $H \le D(S) = S^*/S$. The associated* overlattice *is*

$$K_H = \{x \in S^* : \ x + S \in H\}, \qquad so \qquad S \subseteq K_H \subseteq S^*, \quad K_H/S = H.$$

Lemma 5.10 (Overlattice isomorphism)**.** *Let S_1, S_2 be integral lattices, $H_i \le D(S_i)$, and $K_i = K_{H_i}$. For every isometry $O \in \mathrm{Iso}(S_1, S_2)$, writing $\overline{O}\colon D(S_1) \to D(S_2)$ for the induced map on discriminant groups,*

$$\overline{O}(H_1) = H_2 \quad \iff \quad O(K_1) = K_2.$$

Equivalently,

$$\{O \in \mathrm{Iso}(S_1, S_2) : \overline{O}(H_1) = H_2\} = \{O \in \mathrm{Iso}(K_1, K_2) : O(S_1) = S_2\}.$$

For rational chains the identity holds after clearing each chain $S_i \subseteq K_i$ by the scalar $\alpha(K_i)$ and comparing the recorded scalars; differing scalars give no isomorphism.

Proof. An isometry O between S_1, S_2 extends to spans preserving inner products, hence $O(S_1^*) = S_2^*$ (Lemma 2.8) and induces $\overline{O}$ on $D(S_i) = S_i^*/S_i$. If $\overline{O}$ maps H_1 to H_2, using Definition 5.9 and $x + S_1 \in H_1 \iff Ox + S_2 = \overline{O}(x + S_1) \in H_2$,

$$OK_1 = O\{x \in S_1^* : x + S_1 \in H_1\} = \{y \in S_2^* : y + S_2 \in H_2\} = K_2.$$

Conversely, if $O(K_1) = K_2$ and $O(S_1) = S_2$, then O carries the chain $S_1 \subseteq K_1$ to $S_2 \subseteq K_2$ and induces an isomorphism $K_1/S_1 \to K_2/S_2$, i.e. $\overline{O}(H_1) = H_2$. The rational case follows by the same proof after clearing the denominators of each chain. □

Remark 5.11 (Residual-glue specialization)**.** *In the setting of Definition 5.13, $S_i = \mathcal{F} \oplus R_i$ and the admissible maps are $O = \mathrm{id}_{\mathcal{F}} \oplus O_R$, which send $S_1 \to S_2$ by construction. Hence the side condition $O(S_1) = S_2$ is automatic, and Lemma 5.10 gives*

$$\overline{(\mathrm{id}_{\mathcal{F}} \oplus O_R)}(H_1) = H_2 \quad \iff \quad (\mathrm{id}_{\mathcal{F}} \oplus O_R)(K_{H_1}) = K_{H_2}.$$

We next give a canonicalizer for lattices based on the Haviv–Regev algorithm. The construction and proof are given in Appendix B. It inherits the $r^{O(r)}$-running time from [HR14], but we apply it only on lattices with rank $r = O(n/\log n)$.

Theorem 5.12 (Haviv–Regev-derived lattice canonicalizer)**.** *There is a deterministic algorithm that, given a rank-r positive definite rational lattice $\mathcal{L}$, computes a canonical string $\mathrm{Can}_0(\mathcal{L})$ and a complete* transporter list

$$\Phi_0(\mathcal{L}) = \{\phi : \mathcal{L} \to \mathcal{L}^{\mathrm{can}}\}$$

of isometries from $\mathcal{L}$ to the canonical representative $\mathcal{L}^{\mathrm{can}}$, in time $r^{O(r)}$ plus output. Hence, for two lattices $\mathcal{L}_1, \mathcal{L}_2$,

$$\mathrm{Can}_0(\mathcal{L}_1) = \mathrm{Can}_0(\mathcal{L}_2) \iff \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) \neq \varnothing,$$

and in the equal case

$$\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \{\phi_2^{-1}\phi_1 : \phi_i \in \Phi_0(\mathcal{L}_i)\},$$

after duplicate removal.

5.2 Residual-glue canonical forms

A resolving marker determines an ordered chain of vectors spanning all but $O(n/\log n)$ dimensions. We treat the sublattice in those dimensions as fixed data; only the residual lattice R is canonicalized. The original lattice is an overlattice of the orthogonal sum of the fixed and residual lattices, so we also record its glue subgroup. An admissible residual isometry thus must preserve the glue and the residual marker as well.

We work in an integral model and record the original scale as σ. Two instances can be isomorphic only if their scale factors and fixed-data strings agree.

Definition 5.13 (Residual-glue instance)**.** *A* residual-glue instance *of residual rank r is a tuple*

$$\mathcal{G} = (\sigma, R, \mathcal{F}, H, \omega),$$

presented in the primitive integral model*: $\sigma \in \mathbb{Q}_{>0}$ is the recorded scale factor, and R is a rank-r integral lattice (the* residual *sublattice). The fixed data $\mathcal{F}$ is recorded as the* string $\mathrm{str}(\mathcal{F})$*. Its underlying integral lattice is represented by a rational module lattice T_F with Gram matrix G_F, encoded by the pair $(G_F, \mathrm{NF}_{\mathrm{mod}}(T_F))$; see Section 6 for more details on the construction of $\mathcal{F}$.*

The canonicalizer compares $\mathrm{str}(\mathcal{F})$ *directly and does not transform its coordinates. When writing $\mathcal{F} \oplus R$, we identify $\mathcal{F}$ with its underlying integral lattice. If $\mathrm{NF}_{\mathrm{mod}}(T_F) = (d_F, H_F)$, we use $B_F = d_F^{-1} H_F$ as its canonical $\mathbb{Z}$-basis. Direct sum and discriminant group computations use the Gram matrix $B_F^{\top} G_F B_F$, with all glue generators expressed in the same basis. The orthogonal direct sum $S = \mathcal{F} \oplus R$ is therefore integral with*

$$S^* = \mathcal{F}^* \oplus R^*, \qquad D(S) = D(\mathcal{F}) \oplus D(R).$$

The glue subgroup $H \le D(S)$ is presented by explicit cosets. We require that its associated overlattice

$$K_H = \{x \in S^* : \ x + S \in H\}$$

is integral. Then

$$S \subseteq K_H \subseteq S^*, \quad K_H/S = H.$$

Finally, $\omega \in \mathrm{span}(R)$ is a rational vector, called the residual marker. The lattice represented by $\mathcal{G}$ is K_H with the form $\sigma \langle \cdot, \cdot \rangle_{\mathrm{int}}$.

From a rational lattice. *Let $S = \mathcal{F} \oplus R \subseteq K$ be a chain of rational lattices. Take $\sigma = \alpha(K)$, and multiply the entire chain by σ^{-1}. Since $\mathcal{F}, R \subseteq K$, every pairing of K lies in $\alpha(K)\mathbb{Z}$, this rescaling makes $\mathcal{F}, R, S, K$ integral* simultaneously*. Using the same symbols for the resulting integral lattices, set*

$$H = K/S \le D(S) = D(\mathcal{F}) \oplus D(R), \qquad K = K_H,$$

then the recorded factor σ recovers the original lattice.

Isomorphisms. *A* residual-glue *isomorphism $\mathcal{G}_1 \to \mathcal{G}_2$ exists only if $\sigma_1 = \sigma_2$ and the $\mathcal{F}$-strings agree; it is then a residual isometry $O_R\colon R_1 \to R_2$ with*

$$O_R\,\omega_1 = \omega_2, \qquad (\mathrm{id}_{D(\mathcal{F})} \oplus \overline{O_R})(H_1) = H_2, \tag{5.1}$$

with $\mathcal{F}$ fixed pointwise. By Remark 5.11 the glue condition is equivalent to $(\mathrm{id}_{\mathcal{F}} \oplus O_R)(K_{H_1}) = K_{H_2}$, i.e. it sends the whole glued lattice K_{H_1} to K_{H_2}.

Theorem 5.14 (Small-rank residual-glue canonical forms)**.** *There is a deterministic algorithm that, given a residual-glue instance $\mathcal{G} = (\sigma, R, \mathcal{F}, H, \omega)$ with residual rank r, computes a canonical string $\mathrm{Can}(\mathcal{G})$. Given two instances with equal canonical strings, it enumerates all residual-glue isomorphisms. Both take time $r^{O(r)}$ plus output.*

Proof. By Definition 5.13 the instance is already in the integral model: $S = \mathcal{F} \oplus R$ is integral, $H \le D(S) = D(\mathcal{F}) \oplus D(R)$, and the scale $\sigma = \alpha(K)$ is recorded. Run the lattice canonicalizer (Theorem 5.12) on R to obtain $\mathrm{Can}_0(R)$ and the complete transporter list $\Phi_0(R)$, with $|\Phi_0(R)| \le r^{O(r)}$. Writing $\overline{\phi}$ for the action induced by $\phi \in \Phi_0(R)$ on $D(R)$, set

$$\mathrm{Can}(\mathcal{G}) = \Big(\sigma,\ \mathrm{str}(\mathcal{F}),\ \mathrm{Can}_0(R),\ \operatorname*{lex\,min}_{\phi \in \Phi_0(R)} \big(\mathrm{NF}_{\mathbb{Q}}(\phi\omega),\ \mathrm{NF}_D\big((\mathrm{id}_{D(\mathcal{F})} \oplus \overline{\phi})\,H\big)\big)\Big),$$

where $\mathrm{NF}_{\mathbb{Q}}, \mathrm{NF}_D$ are defined in Definition 5.5, Definition 5.6 respectively (D refers to $D(\mathcal{F}) \oplus D(R^{\mathrm{can}})$), and $\mathrm{str}(\mathcal{F})$ is the recorded fixed-side comparison string of Definition 5.13.

Let $O_R\colon R_1 \to R_2$ be a residual-glue isomorphism: $\sigma_1 = \sigma_2$, $\mathrm{str}(\mathcal{F}_1) = \mathrm{str}(\mathcal{F}_2)$, $O_R\omega_1 = \omega_2$, and $(\mathrm{id}_{D(\mathcal{F})} \oplus \overline{O_R})(H_1) = H_2$, where $\overline{O_R}$ is the map induced on $D(R)$. By Theorem 5.12, $\mathrm{Can}_0(R_1) = \mathrm{Can}_0(R_2)$, R_1 and R_2 share a canonical representative R^{can}, and post-composition $\phi \mapsto \phi\,O_R^{-1}$ is a bijection $\Phi_0(R_1) \to \Phi_0(R_2)$. For $\phi_1 \in \Phi_0(R_1)$ and $\phi_2 := \phi_1 O_R^{-1}$,

$$\phi_2\omega_2 = \phi_1 O_R^{-1}\omega_2 = \phi_1\omega_1, \qquad (\mathrm{id}_{D(\mathcal{F})} \oplus \overline{\phi}_2)H_2 = (\mathrm{id}_{D(\mathcal{F})} \oplus \overline{\phi}_1)(\mathrm{id}_{D(\mathcal{F})} \oplus \overline{O_R}^{-1})H_2 = (\mathrm{id}_{D(\mathcal{F})} \oplus \overline{\phi}_1)H_1.$$

Thus the candidate strings in the fourth slot of $\mathrm{Can}(\mathcal{G})$ agree for $\mathcal{G}_1$ and $\mathcal{G}_2$. This bijection hence implies the lexicographic minima are the same for the two residual-glue instances. With σ, $\mathrm{str}(\mathcal{F})$, and $\mathrm{Can}_0(R)$ already equal, $\mathrm{Can}(\mathcal{G}_1) = \mathrm{Can}(\mathcal{G}_2)$.

Conversely suppose $\mathrm{Can}(\mathcal{G}_1) = \mathrm{Can}(\mathcal{G}_2)$. Then $\sigma_1 = \sigma_2$ and the labeled $\mathcal{F}$-strings agree; $\mathrm{Can}_0(R_1) = \mathrm{Can}_0(R_2)$ gives a common representative R^{can} with $\Phi_0(R_i) = \mathrm{Iso}(R_i, R^{\mathrm{can}})$. Let $\phi_i \in \Phi_0(R_i)$ attain the minimum on side i. Since $\mathrm{NF}_{\mathbb{Q}}$ and $\mathrm{NF}_{D(\mathcal{F}) \oplus D(R^{\mathrm{can}})}$ are faithful in the fixed coordinates of $D(\mathcal{F}) \oplus D(R^{\mathrm{can}})$ (Proposition 5.8), this implies literal equalities

$$\phi_1\omega_1 = \phi_2\omega_2, \qquad (\mathrm{id}_{D(\mathcal{F})} \oplus \overline{\phi}_1)H_1 = (\mathrm{id}_{D(\mathcal{F})} \oplus \overline{\phi}_2)H_2.$$

Put $O_R := \phi_2^{-1}\phi_1 \in \mathrm{Iso}(R_1, R_2)$. Then $O_R\omega_1 = \omega_2$ and $(\mathrm{id}_{D(\mathcal{F})} \oplus \overline{O_R})H_1 = H_2$, which are conditions (5.1). By Theorem 5.12, equality of $\mathrm{Can}(\mathcal{G}_1)$ and $\mathrm{Can}(\mathcal{G}_2)$ is therefore equivalent to the existence of a residual isometry $O_R\colon R_1 \to R_2$ satisfying (5.1), which gives an isomorphism between the residual-glue instances.

The routine outputs $\phi_2^{-1}\phi_1$ for every pair $(\phi_1, \phi_2) \in \Phi_0(R_1) \times \Phi_0(R_2)$ attaining the two minima

and passing the checks (5.1). Each such map is a residual-glue isomorphism by the soundness computation, and the completeness computation shows every residual-glue isomorphism arises from such a pair. Duplicates are removed by comparing the rational matrices.

By Theorem 5.12, the data $\mathrm{Can}_0(R)$, R^{can}, $\Phi_0(R)$ are computed in $r^{O(r)}$ time, with $|\Phi_0(R)| \le r^{O(r)}$ by Corollary 2.12. Each $\mathrm{NF}_{\mathbb{Q}}$ and NF_D is polynomial (Proposition 5.8), so the lexicographic minimum over $\Phi_0(R)$ costs $r^{O(r)}$. Enumeration ranges over $|\Phi_0(R_1)|\,|\Phi_0(R_2)| \le r^{O(r)}$ pairs, each checked in polynomial time. The total is $r^{O(r)}$ plus output. □

Corollary 5.15 (Cost in residual rank)**.** *If $r \le r_*(n) = \lceil \frac{2n}{\log n} \rceil$, then the residual-glue canonicalizer costs time $r^{O(r)} \le 2^{O(n)}$.*

The main algorithm (Theorem 1.1) applies Theorem 5.14 only to instances induced by resolving markers. Their residual ranks satisfy $r \le r_*(n) = \lceil \frac{2n}{\log n} \rceil$. Hence each residual canonicalization and each residual lift costs $2^{O(n)}$ by Corollary 5.15.

6 Exact Marker Records and Birthday Sampling

This section turns resolving markers into an all-isomorphism algorithm, in two steps. First, to each resolving marker w we attach an exact canonical *label* $\mathsf{Lab}(\mathcal{L}, w)$. The label is designed so that $\mathsf{Lab}(\mathcal{L}_1, w_1) = \mathsf{Lab}(\mathcal{L}_2, w_2)$ holds exactly when some $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ sends w_1 to w_2, and in that case all such O can be recovered from the two lattices.

Second, we sample $O(n\sqrt{\log n})^n$ markers from each dual lattice. For a fixed isomorphism O, a sampled pair (w, Ow) witnesses O. The collision distribution is the square-biased law $D_{\mathcal{L}^*, s/\sqrt{2}}$, which by Section 4 yields resolving markers with probability at least 0.99. So every isomorphism is witnessed with high probability, and we show that the total work is still bounded by $O(n\sqrt{\log n})^n$.

6.1 Marker labels

A resolving marker w fixes all but $O(n/\log n)$ dimensions of $\mathcal{L}$ and leaves a small residual problem. That problem is a residual-glue instance in Definition 5.13, and the label will be its canonical string. We now describe the map $(\mathcal{L}, w) \mapsto \mathcal{G}(\mathcal{L}, w)$ that constructs the residual-glue instance from the lattice and resolving marker.

Lemma 6.1 (Decomposing after a fixed subspace)**.** *Let $\mathcal{L}$ be a full rank integral lattice and let $U \subseteq \mathrm{span}(\mathcal{L})$ be a rational subspace spanned by vectors of $\mathcal{L}$. Set $C = \mathcal{L} \cap U$ and $R = \mathcal{L} \cap U^{\perp}$. Then $C \oplus R$ is a full-rank sublattice of $\mathcal{L}$ with $\mathcal{L} \subseteq (C \oplus R)^* = C^* \oplus R^*$, so $\mathcal{L}/(C \oplus R)$ is a finite glue subgroup. If $w \in \mathcal{L}^*$ and $w = w_U + w_R$ is its orthogonal decomposition relative to $U \oplus U^{\perp}$, then $w_R \in R^*$.*

Proof. As U is spanned by lattice vectors it is rational in any rational basis of $\mathcal{L}$, so $U^{\perp}$ is cut out by rational equations and C, R have ranks $\dim U$, $\dim U^{\perp}$. They are orthogonal sublattices, so $C \oplus R \subseteq \mathcal{L}$ has full rank and finite index. For $m \in \mathcal{L}$ and $c + r \in C \oplus R \subseteq \mathcal{L}$, integrality gives $\langle m, c + r \rangle \in \mathbb{Z}$, hence $m \in (C \oplus R)^*$ and $\mathcal{L} \subseteq (C \oplus R)^*$. Thus $H := \mathcal{L}/(C \oplus R)$ is a finite subgroup of $D(C \oplus R)$. Finally, for $r \in R$ we have $\langle w_R, r \rangle = \langle w, r \rangle \in \mathbb{Z}$ since $r \perp U$ and $w \in \mathcal{L}^*$, so $w_R \in R^*$. □

We apply this to the fixed subspace of w. Let $\mathcal{L}$ be well-rounded and $w \in \mathcal{L}^*$. The resolved ranks, the fixed steps $J(w)$, and the fixed vectors $u_j(w) = p_j(w) - p_{j-1}(w) \in \mathrm{Min}(\mathcal{L})$ are those of Section 3 (Lemma 3.7); write $U(w) = \mathrm{span}\{u_j(w) : j \in J(w)\}$ for their span. If w is resolving then $\dim(U(w)^\perp) \le r_*(n) = \lceil \frac{2n}{\log n} \rceil$. Apply Lemma 6.1 with $U = U(w)$ and set

$$C(\mathcal{L}, w) = \mathcal{L} \cap U(w), \qquad R(\mathcal{L}, w) = \mathcal{L} \cap U(w)^\perp, \qquad w = w_U + w_R.$$

The ordered basis $(u_j(w))_{j \in J(w)}$ gives coordinates on the fixed space $U(w)$, but it need not be a $\mathbb{Z}$-basis of the fixed lattice $C(\mathcal{L}, w) = \mathcal{L} \cap U(w)$. Let $X_w : \mathbb{R}^{J(w)} \to U(w)$ be the map $X_w e_j = u_j(w)$ (indices ordered increasingly), and put $T_C(\mathcal{L}, w) = X_w^{-1} C(\mathcal{L}, w) \subseteq \mathbb{Q}^{J(w)}$. The fixed lattice is recorded by

$$\mathrm{NF}_{\mathrm{mod}}\big(T_C(\mathcal{L}, w)\big) = \big(d_C, \mathrm{HNF}(d_C T_C(\mathcal{L}, w))\big)$$

of Definition 5.4. Thus the *fixed data* $\mathcal{F}(\mathcal{L}, w)$ contains:

1. the ordered set $J(w)$;
2. the Gram matrix of $(u_j(w))_{j \in J(w)}$;
3. the rational-module normal form $\mathrm{NF}_{\mathrm{mod}}(T_C(\mathcal{L}, w))$;
4. the marker coordinates $(\langle w, u_j(w) \rangle)_{j \in J(w)}$.

Equivalently, we write

$$\mathrm{str}(\mathcal{F}(\mathcal{L}, w)) = (J(w), G_F, \mathrm{NF}_{\mathrm{mod}}(T_C(\mathcal{L}, w)),\ (\langle w, u_j(w) \rangle)_{j \in J(w)}).$$

The residual-glue instance attached to $(\mathcal{L}, w)$ is then

$$\mathcal{G}(\mathcal{L}, w) = \big(\sigma, R(\mathcal{L}, w),\ \mathcal{F}(\mathcal{L}, w),\ H(\mathcal{L}, w),\ w_R\big), \qquad H(\mathcal{L}, w) = \mathcal{L} / \big(C(\mathcal{L}, w) \oplus R(\mathcal{L}, w)\big),$$

where the chain $C(\mathcal{L}, w) \oplus R(\mathcal{L}, w) \subseteq \mathcal{L}$ is scaled by the convention of Definition 5.13, recording $\sigma = \alpha(\mathcal{L})$. All fields of $\mathrm{str}(\mathcal{F}(\mathcal{L}, w))$, $R(\mathcal{L}, w)$, $H(\mathcal{L}, w)$, and w_R are written in this same primitive model. By Lemma 6.1 the overlattice of $H(\mathcal{L}, w)$ is $\mathcal{L}$, so the instance reconstructs $\mathcal{L}$. The residual rank is $\mathrm{rank}(R(\mathcal{L}, w)) = \dim(U(w)^\perp) \le r_*(n)$ for a resolving w.

The glue subgroup $H(\mathcal{L}, w)$ must be part of the label. Even when a residual isometry preserves the residual marker, it can move the glue and therefore does not preserve the original lattice. The following example shows how the glue detects this obstruction.

Example 6.2. *Let $C = \mathbb{Z}c$ and $R = \mathbb{Z}r_1 \oplus \mathbb{Z}r_2$, where c, r_1, r_2 are mutually orthogonal of norm 2, and take the glue*

$$H_1 = \big\langle \big(\tfrac{c}{2} + C,\ \tfrac{r_1}{2} + R\big) \big\rangle \ \le\ D(C) \oplus D(R).$$

The residual isometry swapping $r_1 \leftrightarrow r_2$ preserves both C and R, but carries H_1 to the distinct subgroup $H_2 = \big\langle \big(\frac{c}{2} + C,\ \frac{r_2}{2} + R\big) \big\rangle$. It is therefore not an automorphism of the overlattice $K_{H_1} = (C \oplus R) + \mathbb{Z}\frac{c + r_1}{2}$, although it is an isometry from K_{H_1} onto the different overlattice $K_{H_2} = (C \oplus R) + \mathbb{Z}\frac{c + r_2}{2}$.

Definition 6.3 (Exact marker label)**.** *If w is resolving, the* exact marker label *is*

$$\mathsf{Lab}(\mathcal{L}, w) = \mathrm{Can}\big(\mathcal{G}(\mathcal{L}, w)\big),$$

the canonical residual-glue string of Theorem 5.14. Markers that are not resolving are discarded and receive no label.

6.2 Exactness of the labels

We prove that equal labels characterize isomorphisms matching the two markers, and that a lifting algorithm recovers all such isomorphisms. First, we consider isometries on the fixed lattice.

Lemma 6.4 (The fixed coordinates determine one isometry)**.** *Let U_1, U_2 be Euclidean spaces with ordered bases $u_1 = (u_{1,1}, \dots, u_{1,d})$ and $u_2 = (u_{2,1}, \dots, u_{2,d})$ sharing the same Gram matrix $G_i = (\langle u_{i,a}, u_{i,b}\rangle)_{a,b}$.*

1. *There is a unique isometry $O_U : U_1 \to U_2$ with $O_U u_{1,a} = u_{2,a}$ for every a.*
2. *Let $C_1 \subseteq U_1$ and $C_2 \subseteq U_2$ be sublattices, and write $T_i = \phi_i(C_i) \subseteq \mathbb{Q}^d$ in coordinates relative to the ordered basis u_i. If $\mathrm{NF}_{\mathrm{mod}}(T_1) = \mathrm{NF}_{\mathrm{mod}}(T_2)$, then $O_U(C_1) = C_2$.*
3. *Let $w_1 \in U_1$ and $w_2 \in U_2$. If $\langle w_1, u_{1,a}\rangle = \langle w_2, u_{2,a}\rangle$ for every a, then $O_U(w_1) = w_2$.*

Proof. Write $\phi_i : U_i \to \mathbb{R}^d$ for the coordinate map of u_i, so $\phi_i(u_{i,a}) = e_a$.

(1) A linear map is fixed by its values on a basis, so $u_{1,a} \mapsto u_{2,a}$ defines a unique $O_U : U_1 \to U_2$, namely $O_U = \phi_2^{-1}\phi_1$. By bilinearity, equality of the Gram matrices makes O_U inner-product preserving, hence an isometry. By construction $\phi_2 \circ O_U = \phi_1$, i.e. O_U carries u_1-coordinates to identical u_2-coordinates; the remaining claims follow from this.

(2) Equal rational-module normal forms mean $T_1 = T_2$ by Definition 5.4. Since $\phi_2 \circ O_U = \phi_1$, we have $\phi_2(O_U C_1) = \phi_1(C_1) = T_1 = T_2 = \phi_2(C_2)$, so $O_U(C_1) = C_2$.

(3) For $w_i \in U_i$ the invertibility of G_i gives $\phi_i(w_i) = G_i^{-1}\big(\langle w_i, u_{i,a}\rangle\big)_a$, so equal inner-product vectors force $\phi_1(w_1) = \phi_2(w_2)$ and thus $O_U w_1 = w_2$. □

Theorem 6.5 (Exact marker label)**.** *Let $\mathcal{L}_1, \mathcal{L}_2$ be well-rounded lattices and let $w_i \in \mathcal{L}_i^*$ be resolving markers.*

(a) If there exists an $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ with $Ow_1 = w_2$, then

$$\mathsf{Lab}(\mathcal{L}_1, w_1) = \mathsf{Lab}(\mathcal{L}_2, w_2).$$

(b) If the two labels are equal, then the lifting routine, denoted $\textsc{Lift}(\mathcal{L}_1, \mathcal{L}_2, w_1, w_2)$, outputs the set

$$\mathcal{S}(w_1, w_2) = \{O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) : Ow_1 = w_2\}$$

in $2^{O(n)}$ time. In particular, equal labels occur if and only if $\mathcal{S}(w_1, w_2)$ is nonempty.

Proof. **For (a).** An isomorphism preserves shortest vectors, linear independence, sums of independent sets, and support functions (see Definition 3.1). Therefore it sends the rank-j projected polytope of $\mathcal{L}_1$ to the rank-j projected polytope of $\mathcal{L}_2$, and sends the maximum-achieving set F_j in $\mathcal{L}_1$ (Definition 3.2) to the maximum-achieving set in $\mathcal{L}_2$. Resolved ranks, fixed steps, and fixed vectors are consequently preserved: $Ou_j(w_1) = u_j(w_2)$ for every $j \in J(w_1)$. Thus the fixed header data $\mathcal{F}$ agree, since the Gram matrix, rational-module normal form, and inner products are preserved by isometry. The same map keeps $\alpha(\mathcal{L}_1) = \alpha(\mathcal{L}_2)$, sends $U(w_1)^\perp$ to $U(w_2)^\perp$, $R(\mathcal{L}_1, w_1)$ to $R(\mathcal{L}_2, w_2)$, $w_{1,R}$ to $w_{2,R}$, and carries the finite quotient $\mathcal{L}_1/(C(\mathcal{L}_1, w_1) \oplus R(\mathcal{L}_1, w_1))$ to the corresponding quotient for $\mathcal{L}_2$. Hence the residual-glue instances are isomorphic, so their canonical strings agree by the completeness of Theorem 5.14.

For (b). Suppose the labels agree. The fixed header data have the same ordered Gram matrix, the same rational-module normal form for the fixed lattice, and the same fixed marker coordinates. Lemma 6.4 gives a unique fixed-space isometry $O_U : U(w_1) \to U(w_2)$ sending every fixed vector to the corresponding fixed vector, sending $C(\mathcal{L}_1, w_1)$ to $C(\mathcal{L}_2, w_2)$, and sending $w_{1,U}$ to $w_{2,U}$.

By Theorem 5.14, equal residual canonical strings allow us to enumerate residual isometries $O_R : R(\mathcal{L}_1, w_1) \to R(\mathcal{L}_2, w_2)$ that send $w_{1,R}$ to $w_{2,R}$ and preserve the recorded glue.

In the same fixed coordinates (given by the equal Gram matrix), O_U is the identity (Lemma 6.4). Thus the residual-glue map $\mathrm{id}_{\mathcal{F}} \oplus O_R$ of Definition 5.13 is $O_U \oplus O_R$ as a map $\mathcal{L}_1 \to \mathcal{L}_2$. By Remark 5.11, $O_U \oplus O_R$ sends the glued lattice $\mathcal{L}_1$ to $\mathcal{L}_2$. Since O_U, O_R preserve w_U, w_R respectively, $O_U \oplus O_R$ is an element of $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ sending w_1 to w_2.

Conversely, every isomorphism sending w_1 to w_2 was shown in part (a) to preserve the fixed sublattice data $\mathcal{F}$, and it also restricts to one of these residual-glue instance isomorphisms. The lifting routine therefore outputs $\mathcal{S}(w_1, w_2)$.

The enumeration underlying Lift is that of Theorem 5.14 on the two residual-glue instances, so it runs in time $r^{O(r)}$ plus output in the residual rank $r = \mathrm{rank}(R(\mathcal{L}_1, w_1))$, each returned map being verified in polynomial time. For resolving markers $r \le r_*(n)$, hence Lift costs at most $r^{O(r)} \le 2^{O(n)}$ plus output per pair by Corollary 5.15. □

Lemma 6.6 (Computing marker labels)**.** *Assume that the shortest-vector shell $A = \mathrm{Min}(\mathcal{L})$ has been enumerated and that $|A| \le 2^{O(n)}$. Given $w \in \mathcal{L}^*$, one can compute the resolved ranks and fixed vectors, decide whether w is resolving, and, if it is, compute its residual-glue instance and exact label in time $2^{O(n)} + r^{O(r)}$, where $r = \mathrm{rank}(R(\mathcal{L}, w))$.*

Proof. By Lemma 3.8, the resolved ranks and fixed vectors can be determined in $2^{O(n)}$ time. Set $r = n - |J(w)| = \mathrm{rank}(R(\mathcal{L}, w))$. If $r > r_*(n)$, report that w is non-resolving and stop. Hence assume below that $r \le r_*(n)$.

Once the fixed space $U(w)$ is known, the lattices $C(\mathcal{L}, w)$ and $R(\mathcal{L}, w)$, the projection w_R, and the glue quotient $\mathcal{L}/(C \oplus R)$ are computed by rational linear algebra, Hermite normal form, and Smith normal form. The fixed sublattice is written as the rational coordinate module $T_C(\mathcal{L}, w) = X_w^{-1} C(\mathcal{L}, w)$ and encoded by $\mathrm{NF}_{\mathrm{mod}}(T_C(\mathcal{L}, w))$, not by assuming that the fixed vectors form a $\mathbb{Z}$-basis. The finite quotient is encoded in the primitive integral model of the full glued chain $C \oplus R \subseteq \mathcal{L} \subseteq (C \oplus R)^*$, so the clearing scalar $\alpha(\mathcal{L})$ is intrinsic and recorded. Finally we call the small-rank canonical-form primitive (Theorem 5.14) on the residual-glue instance. Since $|A| \le 2^{O(n)}$ and the residual rank is r, the stated bound follows. □

6.3 Reciprocal collision probability of the marker distribution

For a distribution μ, write $S_2(\mu) := (\sum_x \mu(x)^2)^{-1}$ for its *collision scale*: the reciprocal of the collision probability, equivalently the exponential of its Rényi order-2 entropy.

Lemma 6.7 (Collision scale)**.** *Let $\mathcal{L}$ be rank-n well-rounded, let $\lambda = \lambda_1(\mathcal{L})$, and set*

$$s = \frac{Cn^2 \log n}{\lambda}.$$

For C sufficiently large,

$$S_2(D_{\mathcal{L}^*,s}) := \left(\sum_{x \in \mathcal{L}^*} \Pr[X = x]^2 \right)^{-1} = \frac{\rho_s(\mathcal{L}^*)^2}{\rho_{s/\sqrt{2}}(\mathcal{L}^*)} = (1 \pm o(1)) 2^{n/2} \frac{s^n}{\det \mathcal{L}^*}.$$

In particular, $2^{n/2} s^n / \det \mathcal{L}^*$ is computable from the input and approximates $S_2(D_{\mathcal{L}^*,s})$ within a $1 + o(1)$ factor, and for constant C' dependent on C

$$S_2(D_{\mathcal{L}^*,s}) \le (C' n^2 \log n)^n \le O(n^2 \log n)^n,$$

where C is the absolute constant fixed in Section 4.

Proof. By Theorem 2.14 applied to $\mathcal{L}^*$ we have $\eta_{2^{-n}}(\mathcal{L}^*) \le \sqrt{n}/\lambda_1(\mathcal{L}) \le s/\sqrt{2}$. Hence the theta estimate above smoothing (Lemma 2.15) gives

$$\rho_s(\mathcal{L}^*) = (1 \pm o(1)) \frac{s^n}{\det \mathcal{L}^*}, \qquad \rho_{s/\sqrt{2}}(\mathcal{L}^*) = (1 \pm o(1)) \frac{(s/\sqrt{2})^n}{\det \mathcal{L}^*}.$$

Substituting these two estimates proves the displayed asymptotic formula. Because $\mathcal{L}$ is well-rounded, it contains n linearly independent vectors of length λ. They span a sublattice of determinant at most λ^n, so $\det \mathcal{L} \le \lambda^n$, and hence $1/\det \mathcal{L}^* = \det \mathcal{L} \le \lambda^n$. Therefore

$$S_2(D_{\mathcal{L}^*,s}) \le (1 + o(1)) 2^{n/2} (s\lambda)^n \le (C' n^2 \log n)^n \le O(n^2 \log n)^n$$

after increasing C by an absolute factor into C'. □

Lemma 6.8 (A birthday collision sees the square-biased law)**.** *Let $\mu = D_{\mathcal{L}^*,s}$ and $S_2(\mu) = (\sum_x \mu(x)^2)^{-1}$. Weighting x proportionally to $\mu(x)^2$ gives exactly $D_{\mathcal{L}^*,s/\sqrt{2}}$. Equivalently, for every $K \subseteq \mathcal{L}^*$,*

$$\sum_{x \in K} S_2(\mu) \mu(x)^2 = \Pr_{X \sim D_{\mathcal{L}^*,s/\sqrt{2}}} [X \in K].$$

Proof. For every $x \in \mathcal{L}^*$,

$$\mu(x)^2 = \frac{\rho_s(x)^2}{\rho_s(\mathcal{L}^*)^2} = \frac{\rho_{s/\sqrt{2}}(x)}{\rho_s(\mathcal{L}^*)^2}.$$

Normalizing over x gives the discrete Gaussian of parameter $s/\sqrt{2}$. □

6.4 The birthday algorithm

This subsection assembles the exact labels and the collision-entropy bound into the birthday sampling algorithm and proves that it computes $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ in time $O(n\sqrt{\log n})^n$.

Theorem 6.9 (Birthday sampling with exact marker labels)**.** *Let $\mathcal{L}_1, \mathcal{L}_2$ be well-rounded rank-n lattices. There is an absolute constant κ_0 such that the following holds for $\kappa \geq \kappa_0 n \log n$*[5]*. Set $s = Cn^2 \log n/\lambda_1(\mathcal{L}_1)$. Assume that for $i = 1, 2$ a sample from $D_{\mathcal{L}_i^*, s/\sqrt{2}}$ is resolving with probability at least 0.99. Then $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ can be computed in time $O(n\sqrt{\log n})^n \, 2^{\mathrm{polylog}(\kappa)}$ with probability at least $1 - \exp(-\Omega(\kappa))$.*

Proof. If $\det(\mathcal{L}_1) \neq \det(\mathcal{L}_2)$ or $\lambda_1(\mathcal{L}_1) \neq \lambda_1(\mathcal{L}_2)$, no isomorphism exists and we return $\varnothing$. Assume from now on that both agree, so that s is well defined by either minimum, and write $\mu_i = D_{\mathcal{L}_i^*, s}$. We first analyze exact samples; the implementable sampler is coupled to this idealized process at the end.

The algorithm. Let

$$\widehat{S}_i = 2^{n/2} \frac{s^n}{\det(\mathcal{L}_i^*)}.$$

By Lemma 6.7, $\widehat{S}_i = (1 \pm o(1)) S_2(\mu_i)$ and $\widehat{S}_i \leq O(n^2 \log n)^n$. Since the determinants agree, $\widehat{S}_1 = \widehat{S}_2 =: \widehat{S}$. Choose

$$N = \left\lceil C_1 \kappa \sqrt{\widehat{S}} \right\rceil \leq O(n\sqrt{\log n})^n \mathrm{poly}(\kappa)$$

for a sufficiently large absolute constant C_1.

We use the Poissonized sampling: draw $K_i \sim \mathrm{Pois}(N)$ independently on each side, and abort the run if either count exceeds $2N$. Draw K_i markers from μ_i, aborting also if a sampled marker exceeds the norm cap $s\sqrt{\kappa}$ of Lemma 2.22. Outside these negligible abort events, the table has size $O(n\sqrt{\log n})^n \mathrm{poly}(\kappa)$. We describe an implementation using a bounded number of random bits at the end of the proof.

For every sampled marker, compute its fixed sublattice and discard it unless the residual rank is at most $r_*(n) = \lceil \frac{2n}{\log n} \rceil$. For retained markers compute the exact labels of Definition 6.3. Hash retained samples by their label string, but label equality is checked by comparing the entire strings. For every pair of markers from opposite sides with equal labels, run the lifting procedure from Theorem 6.5(b), verify that each returned map preserves inner products, maps $\mathcal{L}_1$ to $\mathcal{L}_2$, and sends w_1 to w_2; then remove duplicates.

Every isomorphism is seen with high probability. Fix an isomorphism $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$. If $w \sim \mu_1$, then $Ow \sim \mu_2$. A birthday witness for O is a sampled pair (w, Ow) with w resolving. Let $p_w = \mu_1(w)$ and let $S = S_2(\mu_1)$. In the Poissonized model, set $a_w = Np_w$. The counts of w on side 1 and of Ow on side 2 are then independent $\mathrm{Pois}(a_w)$ variables, jointly independent across all w. This is Poisson thinning: splitting a $\mathrm{Pois}(N)$-sized i.i.d. sample by value yields independent

[5] In a recursive node, the dimension n may drop, but κ is chosen at the root, hence κ may be larger than $n \log n$ itself.

Poisson multiplicities across distinct values; the two sides are drawn independently; and $w \mapsto Ow$ is a bijection.

Let M be the set of resolving markers. Since a witness requires both w on side 1 and Ow on side 2, the probability that w yields no witness is $1-(1-e^{-a_w})^2 = 2e^{-a_w} - e^{-2a_w}$, and these events are independent across $w \in M$. Hence the probability that O has no resolving witness is

$$\prod_{w \in M} \left(2e^{-a_w} - e^{-2a_w}\right). \tag{6.1}$$

For all $a \geq 0$,

$$2e^{-a} - e^{-2a} \leq \exp(-c \min\{a, a^2\}) \tag{6.2}$$

for an absolute constant $c > 0$. Put

$$q_w = p_w^2 \cdot S.$$

Then $\sum_{w \in \mathcal{L}_1^*} q_w = 1$, and by Lemma 6.8 the weights q_w are exactly the law $D_{\mathcal{L}_1^*, s/\sqrt{2}}$. Since this square-biased law is resolving with probability at least 0.99,

$$\sum_{w \in M} q_w \geq 0.99. \tag{6.3}$$

Using $N = C_1 \kappa \sqrt{\widehat{S}}$ and $\widehat{S} = (1 \pm o(1))S$, we have for all large n

$$a_w \geq \frac{C_1 \kappa}{2} \sqrt{q_w}.$$

Since $\min\{T\sqrt{q}, T^2 q\} \geq Tq$ for $T \geq 1$ and $0 \leq q \leq 1$, taking $T = C_1\kappa/2$ in (6.1)–(6.3) gives

$$\begin{aligned} \Pr[O \text{ has no resolving witness}] &\leq \exp\Big(-c \sum_{w \in M} \min\{T\sqrt{q_w}, T^2 q_w\}\Big) \\ &\leq \exp\big(-cT \sum_{w \in M} q_w\big) \leq \exp(-\Omega(C_1\kappa)). \end{aligned}$$

The number of isomorphisms is at most $O(n)^n$ by Corollary 2.12. Choosing C_1 large enough and using $\kappa = \Omega(n \log n)$ with a sufficiently large implied constant, a union bound over all O shows that, with probability $1 - \exp(-\Omega(\kappa))$, every isomorphism has at least one resolving witness. For such a witness (w, Ow), Theorem 6.5(a) gives equal labels, and Theorem 6.5(b) ensures that the output of the lifting step includes that same isomorphism.

Bounding the amount of work. For a processed equal-label pair (w_1, w_2), recall $\mathcal{S}(w_1, w_2) = \{O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) : Ow_1 = w_2\}$. By Theorem 6.5, the lifting routine outputs this set, and equal labels imply that it is nonempty. Let

$$L = \sum_{\text{processed } (w_1, w_2)} |\mathcal{S}(w_1, w_2)|$$

be the total number of lifted maps before duplicate removal.

For each fixed isomorphism O, let Y_O be the number of sampled pairs of the form (w, Ow), without requiring w to be resolving. Every lifted triple (w_1, w_2, O) is counted by exactly one such Y_O, so

$$L \le \sum_{O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)} Y_O.$$

Let A_w, B_{Ow} be the number of copies of w on side 1 and of Ow on side 2. Using independence,

$$\mathbb{E}[Y_O] = \sum_{w \in \mathcal{L}_1^*} \mathbb{E}[A_w]\mathbb{E}[B_{Ow}] = \sum_{w \in \mathcal{L}_1^*} p_w^2 \cdot N^2 = \frac{N^2}{S_2(\mu_1)} \le O(\kappa^2).$$

Therefore $\mathbb{E}[L] \le |\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)|\, C_2\kappa^2$ for an absolute constant C_2. The implementation caps the number of generated maps at

$$L_{\max} = 8B_{\mathrm{aut}}(n)C_2\kappa^2,$$

where $B_{\mathrm{aut}}(n) \le O(n)^n$ is the bound from Corollary 2.12. If the cap is exceeded, the run is discarded. Markov's inequality gives a constant lower bound on the probability that the cap is not exceeded. Thus one capped run has constant success probability, and $\Theta(\kappa)$ independent capped runs boost the probability to $1 - \exp(-\Omega(\kappa))$. The extra factor is polynomial in κ.

Every retained label and every lift uses only residual rank at most $r_*(n) = \lceil \frac{2n}{\log n} \rceil$, so Theorem 6.5 contributes $2^{O(n)}$ per retained sample or processed pair (Lemma 6.6, Corollary 5.15). The table size, the cap $L_{\max}$, duplicate removal, and final verified output give total time $\big(O(n\sqrt{\log n})^n + |\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)|\, 2^{O(n)}\big)2^{\mathrm{polylog}(\kappa)}\mathrm{poly}(\kappa) \le O(n\sqrt{\log n})^n 2^{\mathrm{polylog}(\kappa)}$.

Approximate sampling. We first implement the Poisson sampling with finite random bits. Merge the event $K_i > 2N$ into one abort symbol and approximate the resulting distribution on $\{0, \dots, 2N\} \cup \{\mathsf{abort}\}$ by a dyadic distribution with $q = \Theta(\kappa + \log N)$ bits. Using the recurrence for the Poisson probabilities, this costs $O(N\mathrm{poly}(q, \log N))$ time per repetition and changes the joint law over all $\Theta(\kappa)$ repetitions by $\exp(-\Omega(\kappa))$.

Then replace exact marker samples by the finite-tape sampler of Lemma 2.21, treating $\perp$ as an abort for that run. The algorithm samples only $D_{\mathcal{L}^*,s}$ ($D_{\mathcal{L}^*,s/\sqrt{2}}$ appears only in the square-biased analysis). Across all capped repetitions there are at most $O(n\sqrt{\log n})^n\mathrm{poly}(\kappa)$ sampler calls, so by the hybrid argument at the end of Section 2.4, the finite-tape sampler and the norm cap change the joint law by at most $\exp(-\Omega(\kappa))$. Together with the Poisson-count approximation above, the total implementation error remains $\exp(-\Omega(\kappa))$. The per-call time $2^{O(n)+\mathrm{polylog}(\kappa)}$ is absorbed in the $2^{O(n)}2^{\mathrm{polylog}(\kappa)}$ bound. Since every returned map is verified, no false isomorphism is output. □

7 The Main Algorithm

We present our main lattice isomorphism algorithm in this section, which uses the birthday collision routine to deal with the well-rounded lattices and handles general lattices through the section-quotient recursion.

7.1 The well-rounded case

Theorem 7.1 (Well-rounded all-isomorphism algorithm)**.** *There is an absolute constant κ_0 such that, for rank-n well-rounded integral lattices $\mathcal{L}_1, \mathcal{L}_2$ and $\kappa \geq \kappa_0 n \log n$, Algorithm 1 computes $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ with probability at least $1-\exp(-\Omega(\kappa))$ in randomized time $O(n\sqrt{\log n})^n 2^{\mathrm{polylog}(\kappa)}\mathrm{poly}(\kappa)$.*

Proof. Fix once and for all a constant n_0 so large that the resolving-marker threshold $r_*(n) = \lceil \frac{2n}{\log n} \rceil$ is smaller than n and all asymptotic estimates used in Sections 4 to 6 hold for $n \geq n_0$. If $n < n_0$, we use the deterministic small-rank canonicalizer of Theorem 5.12, and its cost is absorbed into the polynomial factor for these finitely many ranks. Hence assume $n \geq n_0$.

If $\det(\mathcal{L}_1) \neq \det(\mathcal{L}_2)$ or $\lambda_1(\mathcal{L}_1) \neq \lambda_1(\mathcal{L}_2)$, then $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \varnothing$. Otherwise let

$$\lambda = \lambda_1(\mathcal{L}_1) = \lambda_1(\mathcal{L}_2), \qquad s = \frac{Cn^2 \log n}{\lambda}.$$

By Corollary 4.7, applied at scale $s/\sqrt{2}$ after increasing C by a constant, a sample from $D_{\mathcal{L}_i^*, s/\sqrt{2}}$ is resolving with probability at least 0.99 for $i = 1, 2$. This is the hypothesis of Theorem 6.9, whose algorithm is Algorithm 1. The theorem gives the claimed running time and success probability. □

7.2 General instances

The general algorithm is shown in Algorithm 2. It follows the same section–quotient recursion as the algorithm of Haviv and Regev [HR14]; we include the full description here to fix notation and to make explicit the residual-glue canonicalizer and well-rounded birthday subroutine used in our algorithm.

For a full rank lattice $\mathcal{L}$ write $V(\mathcal{L}) = \mathrm{span}(\mathrm{Min}(\mathcal{L}))$. Suppose $V(\mathcal{L}_i)$ has rank $k < n$. Every isomorphism preserves shortest vectors, hence sends $V(\mathcal{L}_1)$ to $V(\mathcal{L}_2)$, so the instance splits into the section $A_i = \mathcal{L}_i \cap V(\mathcal{L}_i)$ (rank k) and the projected quotient $Q_i = \pi_{V(\mathcal{L}_i)^\perp}(\mathcal{L}_i)$ (rank $n-k$). By the section–quotient gluing lemma (Lemma 2.9),

$$\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \left\{ O_A \oplus O_Q : \ O_A \in \mathrm{Iso}(A_1, A_2),\ O_Q \in \mathrm{Iso}(Q_1, Q_2),\ \overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q \right\},$$

with the glue homomorphism $\tau_{\mathcal{L}_i} : Q_i \to D(A_i)$ of Lemma 2.9. Equivalently, after clearing denominators the compatibility is the overlattice isomorphism condition of Lemma 5.10. All child maps are transported back to the parent, with the parent's denominators cleared, before comparing

$$\overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q.$$

Both subproblem ranks are strictly below n, so the recursion descends.

Lemma 7.2 (Cost over the recursion tree)**.** *Let T be a binary recursion tree in which every node v has an integer rank $m_v \geq 1$, every internal node has two children of ranks $a, b \geq 1$ with $a + b = m_v$, and the root has rank n. Suppose a node of rank m incurs local non-output cost at most $O(m\sqrt{\log m})^m$, and an internal node also filters at most $O(a\sqrt{\log a})^a O(b\sqrt{\log b})^b$ candidate pairs. Then the total non-output cost over the tree is $O(n\sqrt{\log n})^n$.*

Algorithm 1 WellRounded$(\mathcal{L}_1, \mathcal{L}_2; \kappa)$ (Theorem 7.1)

Input: rank-n full-rank well-rounded integral lattices $\mathcal{L}_i$ with Gram matrices G_i, and the global confidence parameter κ

Output: the full set $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$

if $n < n_0$ **then**
  Compute $\mathcal{I} = \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ using Theorem 5.12.
  return $\mathcal{I}$
end if
if $\det(\mathcal{L}_1) \neq \det(\mathcal{L}_2)$ **or** $\lambda_1(\mathcal{L}_1) \neq \lambda_1(\mathcal{L}_2)$ **then**
  return $\varnothing$
end if
$\lambda \leftarrow \lambda_1(\mathcal{L}_1)$; $s \leftarrow Cn^2 \log n/\lambda$; $r_* \leftarrow \lceil \frac{2n}{\log n} \rceil$
$\widehat{S} \leftarrow 2^{n/2}\, s^{\,n} / \det(\mathcal{L}_1^*)$; $N \leftarrow \lceil C_1\, \kappa\, \sqrt{\widehat{S}}\, \rceil$ ▷ $N \le O(n\sqrt{\log n})^n \mathrm{poly}(\kappa)$
$L_{\max} \leftarrow 8B_{\mathrm{aut}}(n) C_2 \kappa^2$ ▷ cap from Theorem 6.9
$\mathcal{R}_{\mathrm{all}} \leftarrow \varnothing$
for $\rho = 1, \ldots, C_2\kappa$ **do**
  $T_1, T_2 \leftarrow$ empty hash tables; $\mathcal{R} \leftarrow \varnothing$; $L \leftarrow 0$
  run one Poissonized trial below; discard this repetition if any abort occurs or if $L > L_{\max}$
  for $i \in \{1, 2\}$ **do**
    draw $\mathrm{Pois}(N)$ samples $w \sim D_{\mathcal{L}_i^*, s}$; **abort trial** if count $> 2N$
    for all samples w **do**
      abort trial if $\|w\| > s\sqrt{\kappa}$ ▷ norm cap, Lemma 2.22
      compute resolved ranks / fixed sublattice of w
      discard w if residual rank $> r_*$
      insert w into T_i under key $\mathsf{Lab}(\mathcal{L}_i, w)$
    end for
  end for
  for all $(w_1, w_2) \in T_1 \times T_2$ with $\mathsf{Lab}(\mathcal{L}_1, w_1) = \mathsf{Lab}(\mathcal{L}_2, w_2)$ **do**
    $\mathcal{O} \leftarrow$ Lift$(\mathcal{L}_1, \mathcal{L}_2, w_1, w_2)$ ▷ Theorem 6.5(b)
    $L \leftarrow L + |\mathcal{O}|$; **abort trial** if $L > L_{\max}$
    for all $O \in \mathcal{O}$ **do**
      if the source-to-target matrix U_O lies in $\mathrm{GL}_n(\mathbb{Z})$, satisfies $U_O^{\top} G_2 U_O = G_1$, and has $Ow_1 = w_2$ **then**
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{O\}$
      end if
    end for
  end for
  if the repetition was not discarded **then**
    $\mathcal{R}_{\mathrm{all}} \leftarrow \mathcal{R}_{\mathrm{all}} \cup \mathcal{R}$
  end if
end for
return $\mathcal{R}_{\mathrm{all}}$ ▷ remove duplicates

Algorithm 2 LatticeIso$(\mathcal{L}_1, \mathcal{L}_2; \kappa)$—all isomorphisms $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ (Theorem 1.1)

Input: rank-n full-rank integral lattices $\mathcal{L}_1, \mathcal{L}_2$ and the global confidence parameter κ
Output: the full set $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$

if $n < n_0$ **then**
  Compute $\mathcal{I} = \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ using Theorem 5.12.
  return $\mathcal{I}$
end if
$V(\mathcal{L}_i) \leftarrow \mathrm{span}(\mathrm{Min}(\mathcal{L}_i))$; $k_i \leftarrow \mathrm{rank}(V(\mathcal{L}_i))$
if $\det(\mathcal{L}_1) \neq \det(\mathcal{L}_2)$ **or** $\lambda_1(\mathcal{L}_1) \neq \lambda_1(\mathcal{L}_2)$ **or** $k_1 \neq k_2$ **then**
  return $\varnothing$ ▷ invariant mismatch
end if
if $k_1 = n$ **then** ▷ well-rounded: shortest vectors span $\mathbb{R}^n$
  return WellRounded$(\mathcal{L}_1, \mathcal{L}_2; \kappa)$
end if
$k \leftarrow k_1$
  // Section–quotient split (Lemma 2.9); both ranks $< n$
$A_i \leftarrow \mathcal{L}_i \cap V(\mathcal{L}_i)$ ▷ section, rank k
$Q_i \leftarrow \pi_{V(\mathcal{L}_i)^\perp}(\mathcal{L}_i)$ ▷ projected quotient, rank $n-k$
compute $\alpha(A_i)$ and $\alpha(Q_i)$ for $i = 1, 2$
if $\alpha(A_1) \neq \alpha(A_2)$ **or** $\alpha(Q_1) \neq \alpha(Q_2)$ **then**
  return $\varnothing$
end if
replace each pair by its common primitive integral model and record the child-to-parent transport maps
$\mathcal{O}_A \leftarrow$ LatticeIso$(A_1, A_2; \kappa)$; $\mathcal{O}_Q \leftarrow$ LatticeIso$(Q_1, Q_2; \kappa)$
$\mathcal{R} \leftarrow \varnothing$
for all $(O_A, O_Q) \in \mathcal{O}_A \times \mathcal{O}_Q$ **do**
  if $\overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q$ after clearing the parent's denominators **then** ▷ glue compatibility
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{O_A \oplus O_Q\}$
  end if
end for
return $\mathcal{R}$

Proof. The cross-product filter at a node of rank $m = a + b$ satisfies $O(a\sqrt{\log a})^a O(b\sqrt{\log b})^b \le O(m\sqrt{\log m})^m$, so it remains to sum node costs. The rank of every proper descendant is at most $n-1$, the tree has at most $2n-1$ nodes because ranks strictly decrease and child ranks add, and the root contributes $O(n\sqrt{\log n})^n$. Hence

$$\sum_{v \in T} O(m_v \sqrt{\log m_v})^{m_v} \le O(n\sqrt{\log n})^n + O(n) \cdot O\big((n-1)\sqrt{\log n}\big)^{n-1} \le O(n\sqrt{\log n})^n.$$

Polynomial-time linear algebra and Smith-normal-form filtering at the $O(n)$ nodes are absorbed into the displayed bound. □

Theorem 7.3 (Main theorem)**.** *Given two rank-n integral lattices $\mathcal{L}_1, \mathcal{L}_2$, Algorithm 2 outputs the full set* $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ *in time $O(n\sqrt{\log n})^n$, up to polynomial factors in the input encoding length, with probability at least $1 - 2^{-\Omega(n \log n)}$. In particular decision and search* LIP *are solved with the same time bounds.*

Proof. Set $\kappa = C_\kappa n \log n$ at the root, for a sufficiently large absolute constant C_κ, and pass it unchanged to every recursive call. We prove the claim by strong induction on rank, allowing rational positive definite Gram matrices at recursive nodes. At each node, both lattices are multiplied by one common scalar into their primitive integral models. This common scaling preserves the isomorphism set by Lemma 2.1.

For ranks below the threshold n_0 fixed in Theorem 7.1, we use Theorem 5.12 to enumerate and return $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$. Otherwise, return the empty set if the determinants differ, $\lambda_1(\mathcal{L}_1) \neq \lambda_1(\mathcal{L}_2)$, or the two shortest-vector spans have different ranks.

If the shortest-vector span is full, the instance is well-rounded and we apply Theorem 7.1, at cost $O(n\sqrt{\log n})^n$ for this node. Otherwise the common shortest-vector span has rank $k < n$, so the section and quotient ranks are k and $n-k$, both strictly smaller than n. By the inductive hypothesis, enumerate the isomorphisms of the two lower-rank pieces. Combine the two lists and retain the pairs satisfying the glue compatibility condition of Lemma 2.9. This produces every parent isomorphism exactly once after duplicate-removal.

By Theorem 7.1, a well-rounded call of rank m takes $O(m\sqrt{\log m})^m 2^{\mathrm{polylog}(\kappa)}$ time. At a split $m = a + b$, the decomposition takes $2^{O(m)}$ time. The algorithm then tests the glue condition on at most $O(a\sqrt{\log a})^a O(b\sqrt{\log b})^b$ pairs, in polynomial time per pair. Lemma 7.2 then gives total non-output time $O(n\sqrt{\log n})^n$. Here the factor $2^{\mathrm{polylog}(\kappa)} = 2^{o(n)}$ is absorbed in the bound since $\kappa = \Theta(n \log n)$.

At every randomized node, the discrete Gaussian sampling error and the capped birthday repetitions contribute failure $\exp(-\Omega(\kappa))$. A union bound over the $O(n)$ nodes gives overall failure $\exp(-\Omega(\kappa)) = 2^{-\Omega(n \log n)}$. Beyond the non-output cost, each node writes a returned list of at most $O(m)^m$ maps (Corollary 2.12) at $2^{O(m)}\mathrm{poly}(\kappa)$ per map. The $\mathrm{poly}(\kappa)$ overhead is absorbed by $2^{O(n)}$ for the root choice above. Therefore the algorithm outputs $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ and runs in

$$O(n\sqrt{\log n})^n + |\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)|\, 2^{O(n)} \le O(n\sqrt{\log n})^n$$

using $|\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)| \le O(n)^n$ (Corollary 2.12). □

8 A Quantum Speedup for the Isomorphism Coset

The classical algorithm for LIP (Theorem 1.1) is a two-sided birthday search, where the sample complexity is governed by the collision scale (Lemma 6.7)

$$S_2 \;\le\; O(n^2 \log n)^n, \tag{8.1}$$

via $N \approx \sqrt{S_2}$. In this section we accelerate this birthday step with a quantum algorithm. In the well-rounded case, we use the collision-finding paradigm of Brassard, Høyer, and Tapp [BHT98], which roughly implies that a quantum algorithm solves the same search problem with only around $S_2^{1/3}$ cost. Since $S_2 \le O(n^2 \log n)^n$, this solves the well-rounded decision and search LIP in time $O(n^2 \log n)^{n/3}$.

To extend the speedup to general lattices, we combine the well-rounded subroutine with the section–quotient recursion of Lemma 2.9. The main difference from the classical algorithm is that we do not enumerate all isomorphisms during recursion. Instead, for each lattice $\mathcal{L}$ we maintain a compact description $\mathrm{Des}(\mathcal{L})$ of its automorphism group. At a recursive node, this allows the quantum algorithm to efficiently search for pairs of child automorphisms that satisfy the glue constraints, and hence obtain the parent isomorphisms. The total running time remains $O(n^2 \log n)^{n/3}$.

8.1 Quantum notation and coherent computation

We use standard Dirac notation. For a bit string $x \in \{0,1\}^m$, $|x\rangle$ denotes the corresponding computational basis state. Measuring a state $\sum_x \alpha_x |x\rangle$, where $\sum_x |\alpha_x|^2 = 1$, in the computational basis returns x with probability $|\alpha_x|^2$. We write $|x\rangle|y\rangle$ for $|x\rangle \otimes |y\rangle$ and use workspace (ancilla) registers for intermediate computations.

A quantum computation applies unitary gates to registers and measures at the end. We work in the standard quantum circuit model. The cost of an algorithm is the number of elementary gates, oracle calls, and QRAM (see Definition 8.3) calls.

Definition 8.1 (Coherent realization)**.** *Let $f : \{0,1\}^a \to \{0,1\}^b$. A unitary U_f coherently realizes f if, for some workspace size c,*

$$U_f|x\rangle|0^b\rangle|0^c\rangle = |x\rangle|f(x)\rangle|0^c\rangle$$

for every $x \in \{0,1\}^a$. The three registers are the input register, the output register, and the workspace register, of sizes a, b, c respectively.

Fact 8.2 (Bennett's simulation; circuit specialization [Ben73])**.** *Any deterministic computation given by a size-G Boolean circuit admits a coherent realization with $O(G)$ gates and $O(G)$ ancillae, using compute–copy–uncompute. In particular, every deterministic classical subroutine of cost $2^{O(n)}$ used below has a coherent implementation of cost $2^{O(n)}$.*

8.2 QRAM and the BHT template

Definition 8.3 (QRAM)**.** *A quantum random-access memory over classical cells $x_0, \dots, x_{T-1}$ supports queries*

$$|j\rangle|b\rangle \longmapsto |j\rangle|b \oplus x_j\rangle$$

at unit cost per call [GLM08], where $\oplus$ is the usual bitwise XOR operation. The cells are written classically and can then be read in superposition.

A *phase oracle* for a boolean function g is a unitary O_g with $O_g|x\rangle = (-1)^{g(x)}|x\rangle$: it records the value of g in a sign rather than in an output register. Any coherent realization of g (Definition 8.1) yields one by computing g into an ancilla, flipping the sign, and uncomputing.

We now introduce the BHT template; note item (ii) below follows from item (i) by truncating the search and applying Markov's inequality.

Lemma 8.4 (BHT template [Gro96, BHT98, BBHT98, BHMT02])**.** *Let $\mathcal{A}$ be a state-preparation unitary (possibly using ancilla registers), let g be a boolean function available as a coherent phase oracle O_g, and let p be the probability that measuring the seed register of $\mathcal{A}|0\rangle$ returns a seed ρ with $g(\rho) = 1$. There is a procedure $\mathsf{Search}(\mathcal{A}, O_g)$ with the following guarantees.*

(i) (Completeness.) *If $p > 0$, it returns an outcome x with $g(x) = 1$ using an expected $O(1/\sqrt{p})$ applications of $\mathcal{A}, \mathcal{A}^{-1}, O_g$. And, conditioned on success, x is distributed as $\mathcal{A}|0\rangle$ restricted to $\{g = 1\}$ (see Lemma C.1). In particular, from a uniform-amplitude search register, x is uniform over $\{g = 1\}$.*

(ii) (Promise gap and budget.) *Fix a threshold $p_0 > 0$ and truncate the schedule at $c/\sqrt{p_0}$ applications, for an absolute constant c. If $p = 0$ it returns "none." If $p \geq p_0$ it returns an outcome x with $g(x) = 1$ with probability $\geq 2/3$.*

Every returned outcome is verified by evaluating g, so if $g(x) = 0$ then x is never reported.

The quantum algorithm requires coherent implementations of marker sampling and label computation. Fix the global confidence parameter κ used throughout the recursion. For $w \in \mathcal{L}^*$ write $\ell(w) = \mathsf{Lab}(\mathcal{L}, w)$. We assign the rejection label $\perp$ to a sampler failure, to a marker with $\|w\| > s\sqrt{\kappa}$, or to a non-resolving marker; the label differs from every genuine label.

Lemma 8.5 (Coherent marker oracle)**.** *For $i \in \{1, 2\}$ there are $b, G \leq 2^{O(n)+\mathrm{polylog}(\kappa)}\mathrm{poly}(\mathrm{bits})$ and a reversible circuit $\mathcal{O}_i$ of size $O(G)$ such that with a b-bit seed ρ*

$$\mathcal{O}_i : |\rho\rangle|0\rangle|0\rangle \longmapsto |\rho\rangle|\widehat{W}_i(\rho)\rangle|\widehat{\ell}_i(\rho)\rangle.$$

Here $\widehat{W}_i(\rho) \in \mathcal{L}_i^* \cup \{\perp\}$ is the output of Lemma 2.21, and $\widehat{\ell}_i = \perp$ on any rejected output. Rejection by the norm or resolving test leaves $\widehat{W}_i(\rho)$ unchanged. For uniform ρ, the marker law $\widetilde{\mu}_i$ is within $\delta = \exp(-\Omega(\kappa))$ of $D_{\mathcal{L}_i^*, s}$ in total variation (here $D_{\mathcal{L}_i^*, s}$ assigns zero mass to $\perp$).

Proof. Lemma 2.21 supplies a boolean circuit mapping a fixed number of random bits to a marker or $\perp$. On a retained marker the norm test, resolving test, and exact label computation are fixed deterministic circuits of size $2^{O(n)}\mathrm{poly}(\mathrm{bits}, \kappa)$. Compose these circuits and apply Fact 8.2; compute–copy–uncompute restores every work register to zero and gives the stated oracle. □

8.3 Probability of matching a table label

The BHT template needs a lower bound on the probability that a fresh side-2 sample gives a label stored in the side-1 table. It is governed by the collision scale (8.1), through the square-biased identity of Lemma 6.8. We first prove the bound for the ideal μ_i, then transfer it to the implemented laws $\tilde{\mu}_i$ in Theorem 8.7.

Lemma 8.6 (Table preparation)**.** *Let $\mathcal{L}_1, \mathcal{L}_2$ be isomorphic well-rounded lattices of rank $n \geq n_0$, where the fixed cutoff n_0 is sufficiently large. Set $s = Cn^2 \log n / \lambda_1(\mathcal{L}_1)$. Let*

$$\mu = \mu_1 = D_{\mathcal{L}_1^*, s}, \qquad \mu_2 = D_{\mathcal{L}_2^*, s}, \qquad S_2 = S_2(\mu),$$

and let $t = \Theta(S_2^{1/3})$. Assume that the resolving-marker estimate of Corollary 4.7 applies to $D_{\mathcal{L}_1^*, s/\sqrt{2}}$, and that $s/\sqrt{2}$ is above the smoothing threshold needed in Lemma 2.15 for 2^{-n} error.

Let $T_1 = (W^{(1)}, \dots, W^{(t)})$ consist of i.i.d. samples from μ, and put

$$L(T_1) = \{\, \ell(W^{(j)}) : W^{(j)} \text{ is resolving} \,\}.$$

For this fixed table define its matching probability $p(T_1) := \Pr_{W' \sim \mu_2}[\ell(W') \in L(T_1)]$. *Then*

$$p(T_1) \geq \frac{2}{5} \cdot \frac{t}{S_2}$$

with probability $1 - 2^{-\Omega(n)}$ over the choice of T_1.

Proof. By Minkowski's first theorem, $\det(\mathcal{L}_1) \geq (\lambda_1(\mathcal{L}_1)/\sqrt{n})^n$. Since we take n sufficiently large, Lemma 6.7 gives

$$S_2(\mu_1) \geq \tfrac{1}{2}\, 2^{n/2} s^n \det(\mathcal{L}_1) \geq \tfrac{1}{2}\big(\sqrt{2}\, Cn^{3/2} \log n\big)^n \geq 3^n.$$

Let $R \subseteq \mathcal{L}_1^*$ be the set of resolving markers and set $q = \sum_{w \in R} \mu(w)^2$. By Lemma 6.8, the square-biased law $w \longmapsto \frac{\mu(w)^2}{\sum_v \mu(v)^2}$ is exactly $D_{\mathcal{L}_1^*, s/\sqrt{2}}$. By Corollary 4.7, this law lands in R with probability at least 0.99. Hence $qS_2 \geq 0.99$.

We next bound the largest atom of μ. Since μ is a centered discrete Gaussian, its largest atom is at 0, so $\mu_{\max} := \max_w \mu(w) = \frac{1}{\rho_s(\mathcal{L}_1^*)}$. Also

$$S_2 = \frac{1}{\sum_w \mu(w)^2} = \frac{\rho_s(\mathcal{L}_1^*)^2}{\rho_{s/\sqrt{2}}(\mathcal{L}_1^*)}.$$

Therefore $\mu_{\max} S_2 = \frac{\rho_s(\mathcal{L}_1^*)}{\rho_{s/\sqrt{2}}(\mathcal{L}_1^*)}$. Since $s/\sqrt{2}$ is above smoothing, Lemma 2.15 gives

$$\rho_s(\mathcal{L}_1^*) = (1 + o(1)) \frac{s^n}{\det(\mathcal{L}_1^*)}, \qquad \rho_{s/\sqrt{2}}(\mathcal{L}_1^*) = (1 + o(1)) \frac{(s/\sqrt{2})^n}{\det(\mathcal{L}_1^*)}.$$

Thus $\mu_{\max} S_2 \leq (1 + o(1)) 2^{n/2}$. Define $Y = \sum_{j=1}^{t} \mu(W^{(j)}) \mathbf{1}_{\{W^{(j)} \in R\}}$. The summands are independent

and $\mathbb{E}[Y] = tq \geq 0.99\frac{t}{S_2}$. Moreover,

$$\mathrm{Var}[Y] \leq t \sum_{w \in R} \mu(w)^3 \leq t\, \mu_{\max} q.$$

Therefore

$$\frac{\mathrm{Var}[Y]}{(\mathbb{E}[Y])^2} \leq \frac{\mu_{\max}}{tq} = \frac{\mu_{\max} S_2}{t\,(qS_2)} \leq \frac{2 \cdot 2^{n/2}}{t}.$$

Since $t = \Theta(S_2^{1/3})$ and $S_2 \geq 3^n$, the last quantity is $2^{-\Omega(n)}$. Hence, by Chebyshev's inequality,

$$Y \geq \frac{2}{5} \cdot \frac{t}{S_2}$$

with probability $1 - 2^{-\Omega(n)}$.

It remains to relate Y to $p(T_1)$. Let $\mathcal{E}$ be the event that $W^{(1)}, \ldots, W^{(t)}$ are all distinct. Then

$$\Pr[\mathcal{E}^c] \leq \binom{t}{2} \sum_{w} \mu(w)^2 = O(t^2/S_2) = O(S_2^{-1/3}) = 2^{-\Omega(n)}.$$

Fix $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$. On $\mathcal{E}$, the points $OW^{(j)}$ are distinct. For every resolving $W^{(j)}$, Theorem 6.5 gives $\ell(OW^{(j)}) = \ell(W^{(j)}) \in L(T_1)$, and the pushforward identity $\mu_2(Ow) = \mu(w)$ gives $\mu_2(OW^{(j)}) = \mu(W^{(j)})$. Thus, on $\mathcal{E}$,

$$p(T_1) \geq \sum_{j:\, W^{(j)} \in R} \mu_2(OW^{(j)}) = Y.$$

Combining this inequality with the high-probability lower bound on Y proves the claim. □

8.4 Quantum birthday search for well-rounded lattices

Theorem 8.7 (BHT speedup, well-rounded case)**.** *Let $\mathcal{L}_1, \mathcal{L}_2$ be rank-n well-rounded integral lattices and let $\kappa = \Omega(n \log n)$. In the QRAM model, there is a quantum algorithm running in time $O(n^2 \log n)^{n/3}\, 2^{\mathrm{polylog}(\kappa)}$ with QRAM of size $O(n^2 \log n)^{n/3}$ that, with probability $1 - \exp(-\Omega(\kappa))$, decides whether $\mathcal{L}_1, \mathcal{L}_2$ are isomorphic and, if so, outputs a single isomorphism $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$; otherwise it outputs $\varnothing$.*

Proof. Take the fixed cutoff n_0 as in Lemma 8.6. For $n < n_0$, use Theorem 5.12. We now assume $n \geq n_0$. Reject immediately if the determinants or minima differ. As in Theorem 6.9, set $s = Cn^2 \log n / \lambda_1(\mathcal{L}_1)$ and

$$\widehat{S} = 2^{n/2} \frac{s^n}{\det(\mathcal{L}_1^*)}, \qquad t = \lceil \widehat{S}^{1/3} \rceil.$$

By Lemma 6.7, $\widehat{S} = (1 \pm o(1)) S_2(\mu_1)$ and $\widehat{S} \leq O(n^2 \log n)^n$. Let $\mathcal{O}_1, \mathcal{O}_2$ be the reversible oracles of Lemma 8.5; each has cost $2^{O(n)+\mathrm{polylog}(\kappa)}\mathrm{poly}(\text{bits})$.

Phase 1 (classical table). Run $\mathcal{O}_1$ on t measured seeds, discard non-resolving markers (an expected fraction at most 0.01), and write the surviving pairs $(\ell(W), W)$ into QRAM, hashed by label. The cost is $t \cdot 2^{O(n)+\mathrm{polylog}(\kappa)} \le O(n^2 \log n)^{n/3}\ 2^{\mathrm{polylog}(\kappa)}$, and the table has $\le t$ entries, so its size is at most $O(n^2 \log n)^{n/3}$.

Phase 2 (amplitude amplification). Define the boolean function $g(\rho) = 1$ if and only if the marker W' produced coherently by $\mathcal{O}_2$ on seed ρ is resolving, and its label occurs in the table for some $\ell(W)$ (so Theorem 6.5 guarantees an isomorphism O with $O\mathcal{L}_1 = \mathcal{L}_2, OW = W'$). One evaluation of g costs one call to $\mathcal{O}_2$, one QRAM lookup, and other costs are in $2^{O(n)}$. The corresponding phase oracle O_g also has cost $2^{O(n)+\mathrm{polylog}(\kappa)}$.

Let $\mathcal{A}_2$ prepare the uniform seed superposition $\mathcal{A}_2|0\rangle = 2^{-b/2} \sum_{\rho \in \{0,1\}^b} |\rho\rangle$. Apply Lemma 8.4 with $\mathcal{A} = \mathcal{A}_2$, g as defined above, and threshold $p_0 = c_0 \widehat{S}^{-2/3}$ for a sufficiently small absolute constant c_0. By Lemma 8.6 and $\widehat{S} = (1 \pm o(1))S_2$, consider a table sampled from the ideal distribution. On a YES instance, except with negligible probability, it has matching probability at least p_0. Thus Lemma 8.4 returns a witness using $O(\widehat{S}^{1/3})$ oracle calls, for total time at most $O(n^2 \log n)^{n/3} 2^{\mathrm{polylog}(\kappa)}$; on a NO instance $g \equiv 0$, so it returns "none" within the same budget.

Implemented distribution. The actual sampler produces laws $\widetilde{\mu}_i$ satisfying

$$\mathrm{TV}(\widetilde{\mu}_i, \mu_i) \le \delta = \exp(-\Omega(\kappa))$$

by Lemma 2.21. The joint law of the t samples used to build T_1 differs from the ideal law by at most $t\delta$. Since $\log t = O(n \log n)$ and $\kappa = \Omega(n \log n)$ with a sufficiently large implied constant, we get $t\delta = \exp(-\Omega(\kappa))$. Thus Lemma 8.6 still holds for the implemented table.

Now fix a table T_1 with $p(T_1) \ge 2t/(5S_2)$. Replacing μ_2 by $\widetilde{\mu}_2$ changes the matching probability by at most δ. Since $\delta/(t/S_2) = \exp(-\Omega(\kappa))$, the implemented matching probability satisfies

$$\widetilde{p}(T_1) \ge p(T_1) - \delta = \Omega(t/S_2).$$

Witness and soundness. For any returned seed ρ^*, recompute $g(\rho^*)$ and verify exact label equality with the stored label. For decision LIP, an exact label match implies $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) \neq \varnothing$ (Theorem 6.5). For search LIP, we apply the lifting algorithm on the label collision to obtain the isomorphism O. On a NO instance, no cross-side markers share a label, so $g \equiv 0$ and no ρ^* is returned.

Error and amplification. A run can only fail on a YES instance. Preparing the table, sampler error, and amplitude amplification give constant success probability per run, so $\Theta(\kappa)$ independent capped repetitions reduce the failure probability to $\exp(-\Omega(\kappa))$, while the total cost remains $O(n^2 \log n)^{n/3}\ 2^{\mathrm{polylog}(\kappa)}$. □

8.5 BSGS and automorphism group generation

To extend the well-rounded quantum speedup to general lattices, we maintain a *compact, indexable* description of the automorphism group at each node of the section-quotient recursion. We first describe the group representation, and then show how to construct it at recursive nodes.

The data structure we use is a base and strong generating set (BSGS), and we represent automorphisms by integer matrices.

Definition 8.8 (BSGS)**.** *Let a finite group G act faithfully on a finite set Ω. A* base *is a sequence $(\beta_1, \dots, \beta_\ell) \in \Omega^\ell$ whose pointwise stabilizer in G is trivial. This determines the stabilizer chain $G = G^{(0)} \geq G^{(1)} \geq \cdots \geq G^{(\ell)} = \{1\}$, where $G^{(i)}$ fixes $\beta_1, \dots, \beta_i$. A* strong generating set *is a generating set S with $\langle S \cap G^{(i)} \rangle = G^{(i)}$ for all i. A* BSGS *is a base together with a strong generating set, plus a transversal of $G^{(i+1)}$ in $G^{(i)}$ at each level.*

Lemma 8.9 (BSGS properties [Sim70, Sim71, FHL80])**.** *Given a generating set S_0 of a finite group G acting faithfully on Ω, a BSGS is computable in time* $\mathrm{poly}(|\Omega|, \log|G|, |S_0|)$ *[FHL80]. From a BSGS one obtains, in* $\mathrm{poly}(|\Omega|, \log|G|)$ *per call:*

(i) $|G| = \prod_i [G^{(i-1)} : G^{(i)}]$ *and a membership test;*

(ii) a bijective unranking *map $f\colon \{0, \dots, |G|-1\} \to G$ and its inverse map;*

(iii) uniform sampling of G.

Every cost above is polynomial in the degree $|\Omega|$*, never in $|G|$. See [Ser03, Ch. 4–5] for a textbook treatment.*

Remark 8.10. *Item (ii) is what makes G efficiently and coherently searchable. An index in $\{0, \dots, |G|-1\}$ maps reversibly to a group element, so Fact 8.2 gives a coherent oracle $U_f\colon |r\rangle|0\rangle \mapsto |r\rangle|f(r)\rangle$ of size* $\mathrm{poly}(|\Omega|, \log|G|)$.

The following lemma from [ER65, Bab91] lets us obtain group generators from nearly uniform samples. The proof is deferred to Appendix C.

Lemma 8.11 (Random generation by growth)**.** *Let G be a finite group, and let ν be a distribution on G with*

$$\mathrm{TV}(\nu, \mathrm{Unif}(G)) = \delta \leq \frac{1}{4}.$$

Let $g_1, \dots, g_N$ be independent samples from ν. There is an absolute constant $C > 0$ such that, for every $u \geq 1$, if $N \geq C(\log_2|G| + u)$, then

$$\Pr[\langle g_1, \dots, g_N \rangle = G] \geq 1 - 2^{-u}.$$

Corollary 8.12. *In Lemma 8.11 take $G = \mathrm{Aut}(\mathcal{L})$ for a rank-m lattice $\mathcal{L}$. Then $O(m \log m + \kappa)$ independent samples from ν generate $\mathrm{Aut}(\mathcal{L})$ except with probability $\exp(-\Omega(\kappa))$.*

Proof. Combine Lemma 8.11 with Corollary 2.12 and take $u = \Theta(\kappa)$. □

Lemma 8.13 (Nearly uniform automorphisms from label collisions)**.** *Let $\mathcal{L}$ be well-rounded of rank $m \geq n_0$ for the constant n_0 fixed in Lemma 8.6, and set a parameter $\kappa = \Omega(m \log m)$. Run the collision sampler of the well-rounded case on $(\mathcal{L}, \mathcal{L})$ with table T and predicate*

$$P_T(W') = \big[\, W' \text{ resolving } \wedge\ \mathsf{Lab}(W') \in \mathsf{Lab}(T) \,\big].$$

Lift the returned pair (W_j, W'), where W_j is any table marker selected by a rule depending only on $\mathsf{Lab}(W')$, to a uniformly random $O \in \mathcal{S}(W_j, W')$ (recall we write $\mathcal{S}(W, W') = \{O \in \mathrm{Aut}(\mathcal{L}) : OW = W'\}$). Then, conditioned on a successful run, O is within total-variation distance $2^{-\Omega(m)}$ of uniform on $\mathrm{Aut}(\mathcal{L})$.

Proof. Fix $O \in \mathrm{Aut}(\mathcal{L})$, recall it is orthogonal with $O\mathcal{L}^* = \mathcal{L}^*$. We first prove uniformity under an ideal distribution, then compare it with the implemented sampler.

Let $\mu_\cap$ be $\mu = D_{\mathcal{L}^*,s}$ restricted to the ball $\{x : \|x\| \leq s\sqrt{\kappa}\}$ of Lemma 2.22 and renormalized. This law appears only in the analysis. By Lemmas 2.21 and 2.22,

$$\epsilon := \mathrm{TV}(\widetilde{\mu}, \mu_\cap) \leq \exp(-\Omega(\kappa)).$$

Both $\mu_\cap$ and the marked set

$$\mathrm{Mk}(T) := \{W' : W' \text{ resolving}, \mathsf{Lab}(W') \in \mathsf{Lab}(T)\}$$

are $\mathrm{Aut}(\mathcal{L})$-invariant. Indeed, automorphisms preserve the dual lattice and norms. They also preserve labels (Theorem 6.5(a)) and the set of resolving markers, since O permutes $\mathrm{Min}(\mathcal{L})$ and preserves inner products, so Ow has the same residual rank as w.

Fix a table T with $\mathrm{Mk}(T) \neq \varnothing$, and draw W' from $\mu_\cap|_{\mathrm{Mk}(T)}$. Select the table marker W_j with equal label by some deterministic rule, and lift ξ uniformly from $\mathcal{S}(W_j, W')$. The law $\mu_\cap|_{\mathrm{Mk}(T)}$ is automorphism-invariant, so W' and OW' are equidistributed for $O \in \mathrm{Aut}(\mathcal{L})$. Their labels are equal, so the selection rule returns the same W_j for both, and

$$\mathcal{S}(W_j, OW') = \{Y \in \mathrm{Aut}(\mathcal{L}) : YW_j = OW'\} = O\,\mathcal{S}(W_j, W').$$

Left translation by O carries the uniform law on $\mathcal{S}(W_j, W')$ to that on $\mathcal{S}(W_j, OW')$, so it follows that (W', ξ) and $(OW', O\xi)$ have the same distribution. The marginal distribution of ξ is then left-invariant on $\mathrm{Aut}(\mathcal{L})$, thus uniform.

To implement the uniform lift, use Theorem 5.14 to enumerate $\mathcal{S}(W_j, W')$ in $2^{O(m)}$ time, since resolving markers leave only $r \leq \lceil \frac{2m}{\log m} \rceil$ residual ranks. Put $M = |\mathcal{S}(W_j, W')|$, draw $\lceil \log_2 M \rceil$ bits, and accept a resulting index when it is below M. Repeating $\Theta(\kappa)$ times gives an exactly uniform index conditioned on success, with failure probability $\exp(-\Omega(\kappa))$.

Let ν denote the distribution of the returned automorphism conditioned on success. The joint distribution of the t implemented table samples is within $t\epsilon$ of t independent samples from $\mu_\cap$ in total variation. By Lemma 8.6 and the norm-tail bound, with probability at least $1 - 2^{-\Omega(m)} - t\epsilon$, the table T satisfies

$$p_\cap(T) := \mu_\cap(\mathrm{Mk}(T)) \geq p_0, \qquad p_0 = \Omega(t/S_2) \geq (Cm^2 \log m)^{-2m/3}.$$

Fix such a table. Our choice of κ ensures $\epsilon \le p_0/2$, so

$$\widetilde{\mu}(\mathrm{Mk}(T)) \ge p_\cap(T) - \epsilon \ge p_0/2.$$

Thus Lemma 8.4, with threshold $p_0/2$, gives constant success probability within $O(p_0^{-1/2})$ oracle calls. Conditioned on success, the probability that $p_\cap(T) < p_0$ remains $O(2^{-\Omega(m)} + t\epsilon)$, and the returned seed is uniform among those whose markers lie in $\mathrm{Mk}(T)$ (Lemma 8.4). Since each such marker w has $2^b\widetilde{\mu}(w)$ seed preimages, its distribution is $\widetilde{\mu}|_{\mathrm{Mk}(T)}$. The conditional total-variation bound gives

$$\mathrm{TV}\big(\widetilde{\mu}|_{\mathrm{Mk}(T)}, \mu_\cap|_{\mathrm{Mk}(T)}\big) \le \frac{2\mathrm{TV}(\widetilde{\mu}, \mu_\cap)}{\mu_\cap(\mathrm{Mk}(T))} \le \frac{2\epsilon}{p_\cap(T)} \le \frac{2\epsilon}{p_0}.$$

The preceding argument shows that a successful lift of $W' \sim \mu_\cap|_{\mathrm{Mk}(T)}$ gives a uniform element of $\mathrm{Aut}(\mathcal{L})$. Applying the same lifting rule to $W' \sim \widetilde{\mu}|_{\mathrm{Mk}(T)}$ changes the output distribution by at most $2\epsilon/p_0$. Adding this and $e^{-\Omega(\kappa)}$ for lift failure, we obtain

$$\mathrm{TV}\big(\nu, \mathrm{Unif}(\mathrm{Aut}(\mathcal{L}))\big) \le O\big(2^{-\Omega(m)} + t\epsilon + \epsilon/p_0 + e^{-\Omega(\kappa)}\big) = 2^{-\Omega(m)},$$

for a sufficiently large constant in $\kappa = \Omega(m \log m)$. □

8.6 Recursion for general lattices

Definition 8.14 (Solved description)**.** *For a rank-m integral lattice $\mathcal{L}$, a* solved description $\mathrm{Des}(\mathcal{L})$ *consists of*

(i) a finite set $B_\mathcal{L}$ with $|B_\mathcal{L}| \le 2^{O(m)}$, equipped with a faithful action of $\mathrm{Aut}(\mathcal{L})$; and

(ii) a BSGS of $\mathrm{Aut}(\mathcal{L})$ for its action on $B_\mathcal{L}$.

In the BSGS application, a group element (i.e., an automorphism) is represented as a permutation on $B_\mathcal{L}$. Using linear algebra, one can convert between matrix presentation and permutation presentation within $2^{O(m)}$ time.

Given $\mathrm{Des}(\mathcal{L}_1)$ and $O_0 \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$, we can index the isomorphisms through $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = O_0\mathrm{Aut}(\mathcal{L}_1)$, avoiding full enumeration.

Theorem 8.15. *In the QRAM model, there is a quantum algorithm that, given rank-n integral lattices $\mathcal{L}_1, \mathcal{L}_2$, runs in time $O(n^2 \log n)^{n/3}$ with QRAM of size $O(n^2 \log n)^{n/3}$, up to polynomial factors in the input encoding length, and has the following guarantee. With probability at least $1 - 2^{-\Omega(n \log n)}$, it outputs an isomorphism $O_0 \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$ together with generators of $\mathrm{Aut}(\mathcal{L}_1)$, or outputs $\varnothing$ if the lattices are not isomorphic.*

We prove the theorem by induction.

Lemma 8.16 (Base case)**.** *If $\mathcal{L}$ is well-rounded of rank m, then $\mathrm{Des}(\mathcal{L})$ and, for a well-rounded pair $\mathcal{L}_1, \mathcal{L}_2$, a matching isomorphism or $\varnothing$, are computable in quantum time $O(m^2 \log m)^{m/3}\ 2^{\mathrm{polylog}(\kappa)}$ and with success probability $1 - \exp(-\Omega(\kappa))$.*

Proof. For the finitely many ranks below the asymptotic threshold, use the deterministic small-rank canonicalizer of Theorem 5.12; hence assume m is large enough that Lemma 8.13 gives total-variation distance at most $1/4$. Take $B_{\mathcal{L}} = \mathrm{Min}(\mathcal{L})$. As $\mathcal{L}$ is well-rounded, its shortest vectors span the ambient space, so the $\mathrm{Aut}(\mathcal{L})$-action on $\mathrm{Min}(\mathcal{L})$ is faithful, and $|\mathrm{Min}(\mathcal{L})| \le 2^{O(m)}$ by the kissing bound [KL78].

To generate $\mathrm{Aut}(\mathcal{L})$, run the marker collision routine of Theorem 8.7 on $(\mathcal{L}, \mathcal{L})$, using the same global κ. A collision (W, W') with equal exact labels lifts to an automorphism O with $OW = W'$. One sample costs

$$S_2^{1/3} 2^{O(m)} 2^{\mathrm{polylog}(\kappa)} \le O(m^2 \log m)^{m/3}\ 2^{\mathrm{polylog}(\kappa)},$$

and its output distribution is within $2^{-\Omega(m)}$ of uniform by Lemma 8.13. Keep $O(m \log m + \kappa)$ verified samples. Corollary 8.12 shows that they generate $\mathrm{Aut}(\mathcal{L})$ except with probability $\exp(-\Omega(\kappa))$, and Lemma 8.9 returns the BSGS in $2^{O(m)}\mathrm{poly}(\kappa)$ time.

For a pair, a single label collision of Theorem 8.7 between markers from $\mathcal{L}_1^*$ and $\mathcal{L}_2^*$ yields one $O_0 \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$, which is then verified.

Using $\Theta(\kappa)$ capped repetitions for each required collision, and union-bounding over the polynomial number of repetitions and generated elements, makes the total failure probability $\exp(-\Omega(\kappa))$. □

For the inductive step, recall the section-quotient decomposition (Lemma 2.9): for non-well-rounded $\mathcal{L}$ with $V = \mathrm{span}(\mathrm{Min}(\mathcal{L}))$, set $A = \mathcal{L} \cap V$ (the *section*), $Q = \pi_{V^\perp}(\mathcal{L})$ (the *quotient*), and $\tau \colon Q \to D(A) = A^*/A$ the glue homomorphism. Then

$$\mathrm{Aut}(\mathcal{L}) = \big\{(\alpha, \beta) \in \mathrm{Aut}(A) \times \mathrm{Aut}(Q) : \overline{\alpha} \circ \tau = \tau \circ \beta\big\}, \tag{8.2}$$

where $\overline{\alpha}$ is the automorphism of $D(A)$ induced by α.

Lemma 8.17 (Inductive case)**.** *For $i = 1, 2$, let $\mathcal{L}_i$ have rank m and not be well-rounded, with section A_i of rank k and quotient Q_i of rank $m - k$. Given $\mathrm{Des}(A_i)$ and $\mathrm{Des}(Q_i)$, and one child isomorphism for each (if any), one computes $\mathrm{Des}(\mathcal{L}_i)$ and an isomorphism between $\mathcal{L}_1, \mathcal{L}_2$ (if any) in additional quantum time $O(m)^{m/2}\mathrm{poly}(\kappa)$, with probability $1 - \exp(-\Omega(\kappa))$ and using no QRAM.*

Proof. We omit subscripts when dealing with automorphisms. The section is well-rounded: $\mathrm{Min}(A) \supseteq \mathrm{Min}(\mathcal{L})$ spans V, so $\lambda_1(A) = \lambda_1(\mathcal{L})$ and A is a base-case leaf.

The two child BSGS give a coherent unranking of $K = \mathrm{Aut}(A) \times \mathrm{Aut}(Q)$ (Lemma 8.9(ii)), with $|K| \le k^k (m-k)^{m-k}\, 2^{O(m)} \le O(m)^m$ by Corollary 2.12. The predicate $P(\alpha, \beta) = [\overline{\alpha}\tau = \tau\beta]$ is checkable in $\mathrm{poly}(m, \log |D(A)|)$ time: $\overline{\alpha}\tau$ and $\tau\beta$ are homomorphisms $Q \to D(A)$, so it suffices to compare them on the $O(m)$ generators of Q, and arithmetic in $D(A)$ uses the fixed Smith presentation $D(A) \cong \bigoplus_i \mathbb{Z}/s_i\mathbb{Z}$ (Definition 5.6), so an element is $O(k)$ coordinates of total size $O(\log |D(A)|)$ bits. Fact 8.2 gives a coherent oracle of the same cost.

By (8.2), $\mathrm{Aut}(\mathcal{L}) \subseteq K$ is the set certified by P. By Lemma 8.4, amplitude amplification returns a *uniformly random* element of $\mathrm{Aut}(\mathcal{L})$ in $\sqrt{|K|/|\mathrm{Aut}(\mathcal{L})|} \le O(m)^{m/2}$ oracle calls. Accounting for the next-power-of-two search domain, use the conservative threshold $p_0 = 1/(2|K|)$ and $\Theta(\kappa)$ capped repetitions, and retain $O(m \log m + \kappa)$ successful samples. Corollary 8.12 then proves that these samples generate $\mathrm{Aut}(\mathcal{L})$ except with probability $\exp(-\Omega(\kappa))$.

For the BSGS, set $B_{\mathcal{L}} = B_A \sqcup B_Q$ (equivalently $(B_A \times \{0\}) \cup (B_Q \times \{1\})$, so each element carries a tag indicating whether it comes from B_A or from B_Q). Let $(\alpha, \beta) \in K$ act by α on B_A and by β on B_Q. This action is faithful because both child actions are faithful. Its restriction to $\mathrm{Aut}(\mathcal{L})$ is therefore faithful as well, and $|B_{\mathcal{L}}| \le 2^{O(k)} + 2^{O(m-k)} \le 2^{O(m)}$ (with the same hidden constant). Lemma 8.9 on the sampled generators then returns the BSGS in $2^{O(m)}$ time.

With child isomorphisms $O_A^0 \in \mathrm{Iso}(A_1, A_2)$ and $O_Q^0 \in \mathrm{Iso}(Q_1, Q_2)$, put $\psi = \overline{(O_A^0)}^{-1} \tau_2 O_Q^0$ mapping from $Q_1 \to D(A_1)$. By Lemma 2.9, $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \{(O_A^0 \alpha) \oplus (O_Q^0 \beta) : (\alpha, \beta) \in K,\ \overline{\alpha}\tau_1 = \psi\beta\}$. Using the same argument as above, the child descriptions give a coherent unranking of K, so we can run Lemma 8.4 with predicate $[\overline{\alpha}\tau_1 = \psi\beta]$ to obtain one isomorphism, or "none", in at most $\sqrt{|K|} \le O(m)^{m/2}$ calls. Again use threshold $1/(2|K|)$ and $\Theta(\kappa)$ capped repetitions. If no pair satisfies the glue condition, every run returns "none"; otherwise, the failure probability is $\exp(-\Omega(\kappa))$. The returned map is verified.

The success probability follows from the same analysis as Lemma 8.16. □

Proof of Theorem 8.15. Set $\kappa = \Theta(n \log n)$ once at the root and pass it unchanged through the recursion. Reject immediately if $\det(\mathcal{L}_1) \ne \det(\mathcal{L}_2)$ or $\lambda_1(\mathcal{L}_1) \ne \lambda_1(\mathcal{L}_2)$ or the shortest-vector spans differ in rank. If $V(\mathcal{L}_1)$ has full rank, then so does $V(\mathcal{L}_2)$ by the rank check, and both lattices are well-rounded. Apply Lemma 8.16 to the pair. Otherwise split by Lemma 2.9 into the well-rounded section A (Lemma 8.16) and the quotient Q, recurse on Q, and apply Lemma 8.17.

The recursion descends only on quotients, so it is a chain of length at most n. The section ranks k_i satisfy $\sum_i k_i \le n$. Every leaf rank $k \le n$ has cost $O(k^2 \log k)^{k/3} 2^{\mathrm{polylog}(\kappa)} \le O(n^2 \log n)^{n/3} 2^{\mathrm{polylog}(\kappa)}$, and the combining step at rank $m \le n$ costs at most $O(m)^{m/2}\mathrm{poly}(\kappa) \le O(n)^{n/2}\mathrm{poly}(\kappa)$. There are $O(n)$ nodes, so their sum is $O(n^2 \log n)^{n/3}$ with the $2^{\mathrm{polylog}(\kappa)}$ factor absorbed. QRAM is used only by the leaf collision finder, so its size is $\le O(n^2 \log n)^{n/3}$.

At the root the recursion returns $\mathrm{Des}(\mathcal{L}_1)$ and a matching O_0. The BSGS inside $\mathrm{Des}(\mathcal{L}_1)$ contains a strong generating set of $\mathrm{Aut}(\mathcal{L}_1)$, so O_0 together with these generators presents $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = O_0\,\mathrm{Aut}(\mathcal{L}_1)$, as claimed. This description has size $2^{O(n)}$.

Soundness is guaranteed as every returned matrix is checked, so no false isomorphism is returned. Every randomized node fails with probability $\exp(-\Omega(\kappa))$; a union bound over the $O(n)$ nodes preserves this bound, which is in particular $2^{-\Omega(n \log n)}$. This produces the generating set in time $O(n^2 \log n)^{n/3}$. The all-isomorphisms *listing* version remains barred by the output lower bound: the isomorphism set can have $2^n\, n!$ elements (Theorem 2.11). □

We end with a discussion of memory models for our quantum speedup.

Remark 8.18 (Memory models)**.** *Theorem 8.15 charges unit cost per access to a QRAM table of size $S_2^{1/3}$. Under cost models that charge for memory access or hardware, this advantage erodes [Ber09]. Generic low-memory collision tradeoffs do not immediately apply to our marker oracle, which is nonuniform and samples two different distributions, and the implemented discrete Gaussian sampler circuit itself has exponential workspace. We therefore make no QRAM-free running-time claim here.*

Acknowledgments

This work was supported by the NRF award titled "Computational Hardness of Lattice Problems and Implications" and A–Star grant titled "Research in Information–Theoretic Cryptography, Lattice Algorithms, and Fine–grained complexity".

AI Statement

The authors used OpenAI ChatGPT and Codex, and Anthropic Claude through Claude Code, for language editing, consistency checking, refining proof arguments, and drafting early versions of the proof overview (Section 1.2), which the authors subsequently rewrote extensively. These tools also suggested the use of matroids (Definition 2.23), Smith normal form (Definition 5.2), and bases and strong generating sets (Definition 8.8). The authors independently verified all AI-assisted content.

A The discriminant normal form

This appendix gives the construction behind Definition 5.6 and proves Proposition 5.8. Throughout, $\mathcal{L}$ is integral with Gram matrix $G \in \mathbb{Z}^{d\times d}$. That is, we identify $\mathcal{L} = \mathbb{Z}^d$ and equip it with the form $\langle x, y\rangle = x^\top G y$. Then

$$\mathcal{L}^* = G^{-1}\mathbb{Z}^d, \qquad D(\mathcal{L}) = \mathcal{L}^*/\mathcal{L} \;\cong\; \mathbb{Z}^d/G\mathbb{Z}^d, \tag{A.1}$$

with the last isomorphism sending $G^{-1}v + \mathcal{L} \mapsto v + G\mathbb{Z}^d$.

The fixed presentation. Let

$$P\,G\,Q = \mathrm{diag}(s_1, \ldots, s_d), \qquad s_i \mid s_{i+1}, \tag{A.2}$$

be the Smith form of Definition 5.2, s_i are unique and the same G gives the same P, Q by Proposition 5.3. Since $PG\mathbb{Z}^d = \mathrm{diag}(s_1, \ldots, s_d)\mathbb{Z}^d$,

$$\pi :\; \mathbb{Z}^d/G\mathbb{Z}^d \longrightarrow \bigoplus_{i:\, s_i>1} \mathbb{Z}/s_i\mathbb{Z}, \qquad v + G\mathbb{Z}^d \longmapsto Pv \bmod (s_i)_{s_i>1}, \tag{A.3}$$

is a well-defined isomorphism computed deterministically from G. Composing with (A.1) fixes $D(\mathcal{L}) \cong \bigoplus_{i:\, s_i>1} \mathbb{Z}/s_i\mathbb{Z}$.

The normal form. Let $H \le D(\mathcal{L})$ be generated by the cosets represented by $\xi_1, \ldots, \xi_m \in \mathcal{L}^*$. Put $k = |\{i : s_i > 1\}|$,

$$h_j := G\xi_j \in \mathbb{Z}^d, \qquad \widehat{h}_j := \pi(h_j + G\mathbb{Z}^d). \tag{A.4}$$

Then $\pi(H) = \langle \widehat{h}_1, \ldots, \widehat{h}_m\rangle$.

Let the columns of M_H be integer lifts of $\widehat{h}_j$, and also $s_i e_i$ (for $i : s_i > 1$). Compute

$$M_H U = [\, M_H^* \mid 0 \,], \qquad \mathrm{NF}_{D(\mathcal{L})}(H) := M_H^*, \tag{A.5}$$

where $M_H^* = \mathrm{HNF}(M_H)$ is a $k \times k$ matrix by the full row-rank convention of Definition 5.1. When the discriminant group is trivial ($k = 0$), M_H^* is the empty matrix.

Direct sums. For $D(S) = D(\mathcal{F}) \oplus D(R)$ we concatenate the two independently fixed Smith presentations. Then faithfulness for this concatenated presentation follows from faithfulness of both $\mathrm{NF}_{D(\mathcal{F})}$ and $\mathrm{NF}_{D(R)}$.

Proof of Proposition 5.8. The Smith form (A.2), the lifts (A.4), and the HNF (A.5) are each polynomial in the bit-size of G and the generators by Proposition 5.3.

The map π (A.3) is a fixed isomorphism and determined by G, so $\pi(H)$ depends only on H, not on its generators. Changing a representative $\xi_j \mapsto \xi_j + \ell$ with $\ell \in \mathcal{L}$ changes h_j by $G\ell \in G\mathbb{Z}^d$, which is absorbed by $v \mapsto v + G\mathbb{Z}^d$ and so leaves $\widehat{h}_j$ unchanged.

Uniqueness of s_i (Proposition 5.3) gives a unique finite group $\bigoplus_i \mathbb{Z}/s_i\mathbb{Z}$ from G. Let $\widetilde{H} \subseteq \mathbb{Z}^k$ be the column lattice of M_H. It is the full preimage of $\pi(H)$ under the quotient map $\mathbb{Z}^k \to \bigoplus_{i:s_i>1} \mathbb{Z}/s_i\mathbb{Z}$, so it is determined by H alone. Its unique Hermite basis is precisely M_H^*. Therefore

$$\mathrm{NF}_{D(\mathcal{L})}(H_1) = \mathrm{NF}_{D(\mathcal{L})}(H_2) \iff \widetilde{H}_1 = \widetilde{H}_2 \iff H_1 = H_2 \ .$$

□

Appendix B uses ordered tuples to record the images of basis vectors under a glue homomorphism, so we supply an additional normal form that deals with them.

Definition A.1 (Discriminant tuple normal form)**.** *In the setting of Definition 5.6, with the same fixed Smith presentation, for an ordered tuple $S = (u_1, \ldots, u_\ell)$ with $u_j \in \mathcal{L}^*$, let $\hat{u}_j \in \bigoplus_{i:s_i>1} \mathbb{Z}/s_i\mathbb{Z}$ be the coordinates of $u_j + \mathcal{L}$ as constructed above, each entry reduced into $\{0, \ldots, s_i - 1\}$. Set*

$$\mathrm{NF}_{D(\mathcal{L}),\mathrm{tup}}(S) = [\, \hat{u}_1 \ \cdots \ \hat{u}_\ell \,],$$

the matrix of reduced coordinate columns. No subgroup reduction is applied: the columns stay in their given order.

Proposition A.2. *In the setting of Definition A.1, $\mathrm{NF}_{D(\mathcal{L}),\mathrm{tup}}$ is computable in polynomial time and, for ordered tuples $S = (u_1, \ldots, u_\ell)$ and $S' = (u'_1, \ldots, u'_\ell)$ in $\mathcal{L}^*$,*

$$\mathrm{NF}_{D(\mathcal{L}),\mathrm{tup}}(S) = \mathrm{NF}_{D(\mathcal{L}),\mathrm{tup}}(S')$$

if and only if

$$u_j + \mathcal{L} = u'_j + \mathcal{L} \qquad \text{in } D(\mathcal{L}) \quad \text{for all } j.$$

Proof. In the fixed Smith presentation, each coset has unique coordinates in $\bigoplus_{i:s_i>1} \mathbb{Z}/s_i\mathbb{Z}$, which are computable in polynomial time. Hence the two matrices agree exactly when the cosets agree in every position. □

B The Haviv–Regev canonicalizer for lattices

For a rational lattice $\mathcal{L}$, recall the intrinsic scale

$$\mathfrak{a}(\mathcal{L}) = \alpha(\mathcal{L}) \cdot \mathbb{Z} = \big\langle \langle x, y\rangle : x, y \in \mathcal{L} \big\rangle_{\mathbb{Z}} \subseteq \mathbb{Q}.$$

The canonical string records $\alpha(\mathcal{L})$ and then uses the primitive integral form $\alpha(\mathcal{L})^{-1}\langle\cdot,\cdot\rangle$.

Definition B.1 (Haviv–Regev chain records in the well-rounded case)**.** *Let $\mathcal{L}$ be a rank-r primitive integral well-rounded lattice. Put*

$$\mathcal{W}(\mathcal{L}) := \mathcal{L}^* \cap 5r^{8.5} \cdot \lambda_1(\mathcal{L}^*) \cdot B_2^r.$$

For $w \in \mathcal{W}(\mathcal{L})$, define $x_1(w), \dots, x_r(w)$ recursively if possible by

$$x_j(w) := \arg\min\Big\{\langle w, x\rangle : x \in \mathrm{Min}(\mathcal{L}) \setminus \mathrm{span}(x_1(w), \dots, x_{j-1}(w))\Big\},\ j \in [r], \tag{B.1}$$

where the minimum must be achieved by a unique vector in $\mathrm{Min}(\mathcal{L})$.[6] *Let*

$$\mathcal{I}(\mathcal{L}) := \{w \in \mathcal{W}(\mathcal{L}) : \textit{the recursion (B.1) is uniquely defined for } j = 1, \dots, r\}.$$

For $w \in \mathcal{I}(\mathcal{L})$, let $X_w : \mathbb{R}^r \to \mathrm{span}(\mathcal{L})$ be the ordered real basis map ($\{e_i : 1 \le i \le r\}$ is the standard basis of $\mathbb{R}^r$)

$$X_w e_i = x_i(w).$$

Let $G_w := X_w^\top X_w = (\langle x_i(w), x_j(w)\rangle)_{i,j}$, $T_w := X_w^{-1}\mathcal{L} \subseteq \mathbb{Q}^r$. Using Definition 5.4, we define the *record* (for a well-rounded $\mathcal{L}$)

$$\mathrm{Rec}_{\mathrm{wr}}(\mathcal{L}, w) = \big(G_w, \mathrm{NF}_{\mathrm{mod}}(T_w)\big).$$

The field $\mathrm{NF}_{\mathrm{mod}}(T_w)$ is needed because the shortest chain $x_1, \dots, x_r$ need not be a $\mathbb{Z}$-basis of $\mathcal{L}$. Equivalently,

$$E_w := \sum_{i=1}^{r} \mathbb{Z} x_i(w) \subseteq \mathcal{L}$$

may have finite index, and $\mathcal{L}/E_w \simeq T_w/\mathbb{Z}^r$. Thus, the Gram matrix G_w alone canonicalizes the ordered shortest frame, not the whole lattice.

Lemma B.2 (Well-rounded Haviv–Regev canonical record)**.** *For every rank-r primitive integral well-rounded lattice $\mathcal{L}$, $\mathcal{I}(\mathcal{L}) \neq \varnothing$, and the string*[7]

$$\mathrm{Can}_{\mathrm{wr}}(\mathcal{L}) := \operatorname*{lex\,min}_{w \in \mathcal{I}(\mathcal{L})} \mathrm{Rec}_{\mathrm{wr}}(\mathcal{L}, w) \tag{B.2}$$

[6] We follow the arg min convention of [HR14] here. Since $\mathrm{Min}(\mathcal{L}) = -\mathrm{Min}(\mathcal{L})$, replacing w by $-w$ interchanges arg min and arg max, so this is equivalent to the arg max convention used for the marker machinery of Section 3.

[7] Here $\min^{\mathrm{lex}}$ denotes the minimum in the lexicographic order on records after serialization into an integer sequence. The same convention applies to every $\min^{\mathrm{lex}}$ below.

is a canonical form for $\mathcal{L}$. More precisely, if $\mathcal{L}_1, \mathcal{L}_2$ are well-rounded, then

$$\mathrm{Can}_{\mathrm{wr}}(\mathcal{L}_1) = \mathrm{Can}_{\mathrm{wr}}(\mathcal{L}_2) \iff \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) \neq \varnothing, \tag{B.3}$$

and all isometries can be enumerated in $r^{O(r)}$ time.

Proof. The non-emptiness of $\mathcal{I}(\mathcal{L})$ is the Haviv–Regev isolation theorem [HR14] in the case $\lambda_1(\mathcal{L}) = \lambda_r(\mathcal{L})$: there exists

$$w \in \mathcal{L}^*, \qquad \|w\| \le 5r^{8.5}\lambda_1(\mathcal{L}^*),$$

which uniquely defines a full independent chain of shortest vectors. This is the reason for the radius $5r^{8.5}\lambda_1(\mathcal{L}^*)$ in Definition B.1.

Suppose two lattices are isomorphic, let $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$. Then it preserves the shortest shell and by Lemma 2.8

$$O\mathrm{Min}(\mathcal{L}_1) = \mathrm{Min}(\mathcal{L}_2), \qquad O\mathcal{L}_1^* = \mathcal{L}_2^*, \qquad \lambda_1(\mathcal{L}_1^*) = \lambda_1(\mathcal{L}_2^*).$$

If $w \in \mathcal{I}(\mathcal{L}_1)$ and $x_1(w), \dots, x_r(w)$ is defined by (B.1), then $Ow \in \mathcal{I}(\mathcal{L}_2)$ and its chain is

$$x_i(Ow) = Ox_i(w) \qquad (1 \le i \le r).$$

Consequently

$$G_{Ow} = G_w, \qquad T_{Ow} = T_w, \qquad \mathrm{NF}_{\mathrm{mod}}(T_{Ow}) = \mathrm{NF}_{\mathrm{mod}}(T_w), \qquad \mathrm{Rec}_{\mathrm{wr}}(\mathcal{L}_2, Ow) = \mathrm{Rec}_{\mathrm{wr}}(\mathcal{L}_1, w),$$

where we use the uniqueness of $\mathrm{NF}_{\mathrm{mod}}$ (see Definition 5.4). Thus the set of candidate records is isometry-invariant, and the lexicographic minimum in (B.2) is the same for $\mathcal{L}_1, \mathcal{L}_2$. This also shows that for any $w \in \mathcal{I}(\mathcal{L}_1)$ achieving this minimum and $O \in \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2)$, Ow is also minimizing among $\mathcal{I}(\mathcal{L}_2)$. Thus every isometry is enumerated by going over all pairs of minimizing vectors in $\mathcal{I}(\mathcal{L}_1) \times \mathcal{I}(\mathcal{L}_2)$.

Conversely, suppose that two records agree:

$$\mathrm{Rec}_{\mathrm{wr}}(\mathcal{L}_1, w_1) = \mathrm{Rec}_{\mathrm{wr}}(\mathcal{L}_2, w_2).$$

Let $X_i = X_{w_i}$. The equality $G_{w_1} = G_{w_2}$ is equivalent to $X_1^\top X_1 = X_2^\top X_2$, which implies that the linear map

$$O = X_2 X_1^{-1} \tag{B.4}$$

is an orthogonal map. The equality $\mathrm{NF}_{\mathrm{mod}}(T_{w_1}) = \mathrm{NF}_{\mathrm{mod}}(T_{w_2})$ means that, from Definition 5.4, the two rational modules are the same:

$$X_1^{-1}\mathcal{L}_1 = T_{w_1} = T_{w_2} = X_2^{-1}\mathcal{L}_2.$$

Hence

$$O\mathcal{L}_1 = X_2 X_1^{-1}\mathcal{L}_1 = \mathcal{L}_2.$$

This proves (B.3), and it also gives the isometry associated with any matching pair of records.

It remains to bound the enumeration. By Lemma 2.19,

$$|\mathcal{W}(\mathcal{L})| \le (10r^{8.5}+1)^r = r^{O(r)}.$$

Thus the set $\mathcal{W}(\mathcal{L})$ can be enumerated in $r^{O(r)}$ time (e.g., using [HR14, Corollary 2.16]). The shortest shell $\mathrm{Min}(\mathcal{L})$ is enumerated in $2^{O(r)}$ time and space by Theorem 2.18; by Lemma 2.19 it has size at most 3^r. For each $w \in \mathcal{W}(\mathcal{L})$, checking the uniqueness of the recursion (B.1), computing G_w, and computing $\mathrm{NF}_{\mathrm{mod}}(T_w)$ are rational linear algebra plus polynomial-time HNF (Proposition 5.3). Therefore all minimizing records and all maps (B.4) are obtained in $r^{O(r)}$ time. Duplicates are removed by sorting canonical rational-matrix encodings. □

We now pass from the well-rounded case to the general case. The construction is the canonical version of the Haviv–Regev section/quotient recursion, and the notations follow from Lemma 2.9: $V = \mathrm{span}(\mathrm{Min}(\mathcal{L}))$, $A = \mathcal{L} \cap V$, $Q = \pi_{V^\perp}(\mathcal{L})$, and $\tau_{\mathcal{L}} : Q \to D(A)$. We use subscripts to distinguish objects from different lattices when necessary. For example, $V_1 = \mathrm{span}(\mathrm{Min}(\mathcal{L}_1))$.

Theorem 5.12 (Haviv–Regev-derived lattice canonicalizer)**.** *There is a deterministic algorithm that, given a rank-r positive definite rational lattice $\mathcal{L}$, computes a canonical string $\mathrm{Can}_0(\mathcal{L})$ and a complete* transporter list

$$\Phi_0(\mathcal{L}) = \{\phi : \mathcal{L} \to \mathcal{L}^{\mathrm{can}}\}$$

of isometries from $\mathcal{L}$ to the canonical representative $\mathcal{L}^{\mathrm{can}}$, in time $r^{O(r)}$ plus output. Hence, for two lattices $\mathcal{L}_1, \mathcal{L}_2$,

$$\mathrm{Can}_0(\mathcal{L}_1) = \mathrm{Can}_0(\mathcal{L}_2) \iff \mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) \neq \varnothing,$$

and in the equal case

$$\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \{\phi_2^{-1}\phi_1 : \phi_i \in \Phi_0(\mathcal{L}_i)\},$$

after duplicate removal.

Proof. We induct on r. Rank 0 is trivial. Put $\alpha_0 = \alpha(\mathcal{L})$ and let $\mathcal{L}^{\mathrm{prim}}$ be the same module with form $\alpha_0^{-1}\langle\cdot,\cdot\rangle$. All recursive computations below are performed on $\mathcal{L}^{\mathrm{prim}}$ (which we temporarily denote again by $\mathcal{L}$). The value α_0 is prepended to the final string. At the end we multiply the entire canonical form by α_0; by Lemma 2.1, the same linear maps are then transporters for the original lattice.

If $\mathcal{L}$ is well-rounded, use Lemma B.2. The canonical string stored is α_0 and $\mathrm{Can}_{\mathrm{wr}}(\mathcal{L})$. For any w attaining the minimum of (B.2), first reconstruct the primitive canonical representative from the rational module $T_w \subseteq \mathbb{Q}^r$ with Gram matrix G_w, and then restore the recorded scale. Thus $\mathcal{L}^{\mathrm{can}}$ is the rational lattice in standard module coordinates $\mathcal{L}^{\mathrm{can}} = (\mathbb{Z}^r, G)$ with

$$G_{\mathrm{prim}} = B^\top G_w B, \qquad G = \alpha_0 G_{\mathrm{prim}}, \qquad B = \tfrac{1}{d_w}\mathrm{HNF}(d_w T_w),$$

where $d_w = \min\{d \in \mathbb{Z}_{>0} : dT_w \subseteq \mathbb{Z}^r\}$, so B is the HNF $\mathbb{Z}$-basis of T_w. (Only the string $\mathrm{Can}_0(\mathcal{L}) = (\alpha_0, \mathrm{Can}_{\mathrm{wr}}(\mathcal{L}))$ is stored, and $\mathcal{L}^{\mathrm{can}}$ is rebuilt from it when transporters are assembled.) The primitive

transporter[8] attached to w is

$$\phi_w = B^{-1}X_w^{-1} : \mathcal{L}^{\mathrm{prim}} \longrightarrow (\mathbb{Z}^r, G_{\mathrm{prim}}).$$

Here the displayed map is first an isometry from $\mathcal{L}^{\mathrm{prim}}$ to $(\mathbb{Z}^r, G_{\mathrm{prim}})$ and, after common scaling, from the original $\mathcal{L}$ to $(\mathbb{Z}^r, G)$. The list $\Phi_0(\mathcal{L})$ consists of such ϕ_w for all minimizing w. Recall from the proof of Lemma B.2 that all minimizing w give the identical (T_w, G_w)-pair, hence they produce the same G_{prim}, and that for any $g \in \mathrm{Aut}(\mathcal{L})$ the witness gw is again minimizing with $\phi_{gw} = \phi_w g^{-1}$. As g ranges over $\mathrm{Aut}(\mathcal{L})$ these exhaust the isometries from $\mathcal{L}^{\mathrm{prim}}$ to $(\mathbb{Z}^r, G_{\mathrm{prim}})$; after scaling both forms by α_0, Lemma 2.1 gives all isometries from the original lattice to $(\mathbb{Z}^r, G)$.

Assume now that $0 < k = \dim(V(\mathcal{L})) < r$. Recursively compute

$$\big(A^{\mathrm{can}}, \Phi_0(A), \mathrm{Can}_0(A)\big), \qquad \big(Q^{\mathrm{can}}, \Phi_0(Q), \mathrm{Can}_0(Q)\big).$$

For every pair

$$\phi_A \in \Phi_0(A), \qquad \phi_Q \in \Phi_0(Q),$$

define the map in canonical coordinates:

$$\tau_{\phi_A,\phi_Q} = \overline{\phi_A} \circ \tau_{\mathcal{L}} \circ \phi_Q^{-1} : Q^{\mathrm{can}} \longrightarrow D(A^{\mathrm{can}}), \tag{B.5}$$

where $\overline{\phi_A}$ is the induced map on discriminant groups. Let $\mathrm{NF}(\tau_{\phi_A,\phi_Q})$ be the normal-form (Definition A.1) string of the homomorphism (B.5); explicitly, if $q_1, \ldots, q_{r-k}$ is the displayed basis (in this case, the standard basis) of Q^{can} and $\tilde{u}_i \in (A^{\mathrm{can}})^*$ are representatives of $\tau_{\phi_A,\phi_Q}(q_i)$, then

$$\mathrm{NF}(\tau_{\phi_A,\phi_Q}) = \mathrm{NF}_{D(A^{\mathrm{can}}),\mathrm{tup}}\big(\tilde{u}_1, \cdots, \tilde{u}_{r-k}\big).$$

Define

$$\tau^{\mathrm{can}} = \min_{\phi_A,\phi_Q}^{\mathrm{lex}} \mathrm{NF}(\tau_{\phi_A,\phi_Q}) \tag{B.6}$$

and

$$\mathrm{Can}_0(\mathcal{L}) = \Big(\alpha_0, \lambda_1(\mathcal{L})^2, k, \mathrm{Can}_0(A), \mathrm{Can}_0(Q), \tau^{\mathrm{can}}\Big). \tag{B.7}$$

The primitive canonical representative $\mathcal{L}^{\mathrm{prim,can}}$ is reconstructed from

$$A^{\mathrm{can}}, \quad Q^{\mathrm{can}}, \quad \tau^{\mathrm{can}}$$

by Lemma 2.9 and Remark 2.10. Here we retrieve the homomorphism from the canonical string τ^{can} using the Smith representation of $D(A^{\mathrm{can}})$. The final representative $\mathcal{L}^{\mathrm{can}}$ is the same module with the form of $\mathcal{L}^{\mathrm{prim,can}}$ multiplied by α_0.

The list $\Phi_0(\mathcal{L})$ consists of all direct-sum maps

$$\phi_A \oplus \phi_Q : \mathcal{L}^{\mathrm{prim}} \longrightarrow \mathcal{L}^{\mathrm{prim,can}}$$

[8] Note B^{-1} is an isometry from $(T_w, G_w) \to (\mathbb{Z}^r, G_{\mathrm{prim}})$.

for which (B.6) is attained, interpreted after common scaling as maps from the original $\mathcal{L}$ to $\mathcal{L}^{\mathrm{can}}$.

Correctness of $\mathrm{Can}_0(\mathcal{L})$. Every isometry $O : \mathcal{L}_1 \to \mathcal{L}_2$ sends shortest vectors to shortest vectors, hence sends V_1 to V_2, A_1 to A_2, and Q_1 to Q_2. Together with the inductive hypothesis, this shows that the first five elements in (B.7) remain the same for $\mathcal{L}_1, \mathcal{L}_2$.

By Lemma 2.9(ii), $O = O_A \oplus O_Q$ with $O_A \in \mathrm{Iso}(A_1, A_2)$, $O_Q \in \mathrm{Iso}(Q_1, Q_2)$ and $\overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q$. By induction A_1, A_2 share a canonical representative A^{can} and Q_1, Q_2 share Q^{can}, and—by completeness of the transporter lists—post-composition with O_A^{-1} and O_Q^{-1} are bijections $\Phi_0(A_1) \to \Phi_0(A_2)$ and $\Phi_0(Q_1) \to \Phi_0(Q_2)$. For every $(\phi_A, \phi_Q) \in \Phi_0(A_1) \times \Phi_0(Q_1)$,

$$\tau^{(2)}_{\phi_A O_A^{-1}, \phi_Q O_Q^{-1}} := \overline{\phi_A} \circ \overline{O}_A^{-1} \circ \tau_{\mathcal{L}_2} \circ O_Q \circ \phi_Q^{-1} = \overline{\phi_A} \circ \tau_{\mathcal{L}_1} \circ \phi_Q^{-1} =: \tau^{(1)}_{\phi_A, \phi_Q},$$

the middle step uses $\overline{O}_A \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q$. Hence the families $\{\mathrm{NF}(\tau_{\phi_A, \phi_Q})\}$ coincide for lattices $\mathcal{L}_1, \mathcal{L}_2$, so their lexicographic minima τ^{can} agree, which gives $\mathrm{Can}_0(\mathcal{L}_1) = \mathrm{Can}_0(\mathcal{L}_2)$.

Conversely, equality of the recursive canonical strings gives isometries between A_1, A_2 and Q_1, Q_2. Let $(\phi_A^{(i)}, \phi_Q^{(i)})$ attain the lexicographic minimum τ^{can} on side i. Since $\mathrm{Can}_0(\mathcal{L}_1) = \mathrm{Can}_0(\mathcal{L}_2)$, the two minimizing strings are equal, and as $\mathrm{NF}_{D,\mathrm{tup}}$ is faithful (Proposition A.2), the transported homomorphisms agree on the displayed basis of Q^{can}, hence as maps:

$$\overline{\phi_A^{(1)}} \circ \tau_{\mathcal{L}_1} \circ (\phi_Q^{(1)})^{-1} = \overline{\phi_A^{(2)}} \circ \tau_{\mathcal{L}_2} \circ (\phi_Q^{(2)})^{-1}. \tag{B.8}$$

Set $O_A = (\phi_A^{(2)})^{-1} \circ \phi_A^{(1)} \in \mathrm{Iso}(A_1, A_2)$ and $O_Q = (\phi_Q^{(2)})^{-1} \circ \phi_Q^{(1)} \in \mathrm{Iso}(Q_1, Q_2)$. From (B.8),

$$\overline{\phi_A^{(1)}} \circ \tau_{\mathcal{L}_1} = \overline{\phi_A^{(2)}} \circ \tau_{\mathcal{L}_2} \circ (\phi_Q^{(2)})^{-1} \circ \phi_Q^{(1)},$$

and composing with $(\overline{\phi_A^{(2)}})^{-1}$ on the left gives

$$\overline{O_A} \circ \tau_{\mathcal{L}_1} = \tau_{\mathcal{L}_2} \circ O_Q.$$

Using reconstruction (Lemma 2.9(ii)), this is the condition that $O_A \oplus O_Q$ sends $\mathcal{L}_1$ to $\mathcal{L}_2$, which proves $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) \neq \varnothing$.

Completeness of transporters. Recall $\Phi_0(\mathcal{L}) = \{\phi_A \oplus \phi_Q : (\phi_A, \phi_Q) \text{ attains } \tau^{\mathrm{can}}\}$. We first prove the statement for the primitive pair. By construction $\mathcal{L}^{\mathrm{prim,can}}$ is the reconstruction (Lemma 2.9(i)) of $A^{\mathrm{can}}, Q^{\mathrm{can}}$ along τ^{can}, so its glue homomorphism is $\tau_{\mathcal{L}^{\mathrm{prim,can}}} = \tau^{\mathrm{can}}$.

($\subseteq$) Each such $\phi_A \oplus \phi_Q$ is an isometry $\mathcal{L}^{\mathrm{prim}} \to \mathcal{L}^{\mathrm{prim,can}}$ by Lemma 2.9. ($\supseteq$) Let $\phi \in \mathrm{Iso}(\mathcal{L}^{\mathrm{prim}}, \mathcal{L}^{\mathrm{prim,can}})$. It preserves $\mathrm{Min}(\mathcal{L})$, hence V, so $\phi = \phi_A \oplus \phi_Q$ by Lemma 2.9(ii), and the transported glue map is $\overline{\phi}_A \circ \tau_{\mathcal{L}} \circ \phi_Q^{-1} = \tau_{\mathcal{L}^{\mathrm{prim,can}}} = \tau^{\mathrm{can}}$. Thus (ϕ_A, ϕ_Q) attains the minimum, i.e. ϕ has the defining form of $\Phi_0(\mathcal{L})$. Hence $\Phi_0(\mathcal{L}) = \mathrm{Iso}(\mathcal{L}^{\mathrm{prim}}, \mathcal{L}^{\mathrm{prim,can}})$. Scaling both forms by α_0 and applying Lemma 2.1 gives $\Phi_0(\mathcal{L}) = \mathrm{Iso}(\mathcal{L}, \mathcal{L}^{\mathrm{can}})$ for the original lattice.

If $\mathrm{Can}_0(\mathcal{L}_1) = \mathrm{Can}_0(\mathcal{L}_2)$, then $\mathcal{L}_1^{\mathrm{can}} = \mathcal{L}_2^{\mathrm{can}} =: \mathcal{L}^{\mathrm{can}}$ by reconstruction (Lemma 2.9(i)), and $\Phi_0(\mathcal{L}_i) = \mathrm{Iso}(\mathcal{L}_i, \mathcal{L}^{\mathrm{can}})$. Hence it is immediate that $\mathrm{Iso}(\mathcal{L}_1, \mathcal{L}_2) = \{\phi_2^{-1}\phi_1 : \phi_i \in \Phi_0(\mathcal{L}_i)\}$.

Cost. The normal form strings can be computed in polynomial time by Proposition A.2. Hence

for $0 < k < r$, the running time satisfies,

$$C(r) \le C(k) + C(r-k) + |\Phi_0(A)|\,|\Phi_0(Q)| + r^{O(r)}.$$

By induction and the automorphism bound in rank m (Corollary 2.12), $|\Phi_0(A)| \le k^{O(k)}, |\Phi_0(Q)| \le (r-k)^{O(r-k)}$. Therefore

$$|\Phi_0(A)|\,|\Phi_0(Q)| \le r^{O(r)}.$$

The recursion tree has $O(r)$ nodes, since child ranks sum to the parent rank, so the leaf ranks sum to r. Each node has rank at most r, hence non-recursive cost at most $r^{O(r)}$. A sum of $O(r)$ such terms is again $r^{O(r)}$, so $C(r) \le r^{O(r)}$. □

C Auxiliary probabilistic results

This section proves some probabilistic results in the main text. We first construct a discrete Gaussian sampler using finite randomness.

Lemma 2.21 (Finite-tape ADRS realization)**.** *Let $\mathcal{L}$ be a rank-n rational lattice, let $s > 0$ be efficiently computable, and let $\kappa \ge Cn$ for a sufficiently large absolute constant C. There exist integers b, G and a uniformly constructible deterministic Boolean circuit*

$$F_{\mathcal{L},s,\kappa} : \{0,1\}^b \longrightarrow \mathcal{L} \cup \{\perp\}$$

of size at most G, where the output lattice vector is represented by its integer coordinates in the displayed basis, such that

$$b, G \le 2^{O(n)+\mathrm{poly}\log(\kappa)}\mathrm{poly}(\mathrm{bits}(\mathcal{L}, s)).$$

For a uniform seed $R \in \{0,1\}^b$, the law of $F_{\mathcal{L},s,\kappa}(R)$ is within total variation distance $\exp(-\Omega(\kappa))$ of $D_{\mathcal{L},s}$, where the latter assigns mass zero to $\perp$. Every nonfailure output also satisfies $\|F_{\mathcal{L},s,\kappa}(R)\| \le s\sqrt{\kappa}$.

Proof. Run the batched algorithm underlying Theorem 2.20 and retain its first coordinate. Marginalization does not increase total variation distance.

Let

$$Q = 2^{O(n)+\mathrm{poly}\log(\kappa)}\mathrm{poly}(\mathrm{bits}(\mathcal{L}, s))$$

bound its running time and number of randomized operations. Truncate every Poisson draw and rejection loop so that each omitted tail has probability at most $\exp(-\Omega(\kappa))/Q$. The required truncation lengths are $O(\kappa + \log Q)$ by [ADRS15, Lemma 2.19] and [BLP+13, Section 5]. For the $p_{\max}$ estimator of [ADRS15, Proposition 3.1], there are 2^n cosets modulo $2\mathcal{L}$, so $p_{\max} \ge 2^{-n}$; we stop after $n+1$ halvings and output $\perp$ if the estimator has not halted. For $\kappa \ge Cn$, the total probability of a timeout or overflow is $\exp(-\Omega(\kappa))$.

Approximate every remaining coin probability and truncated distribution function to $q = \Theta(\kappa + \log Q)$ binary digits and implement it using fair random bits. A hybrid argument over all randomized operations bounds the resulting change in distribution by $Q \cdot 2^{-q} = \exp(-\Omega(\kappa))$.

After truncation, the computation has at most $Q\mathrm{poly}(\kappa + \log Q)$ elementary steps. Since each randomized step uses $O(q)$ fair bits, its random-tape length satisfies $b \le Q\mathrm{poly}(\kappa + \log Q)\, q$. Compiling the same computation into Boolean gates gives $G \le Q\mathrm{poly}\big(\kappa + \log Q, q, \mathrm{bits}(\mathcal{L}, s)\big)$. By the definition of Q and q, both quantities are at most $2^{O(n)+\mathrm{poly}\log(\kappa)}\mathrm{poly}(\mathrm{bits}(\mathcal{L}, s))$.

Finally, output $\perp$ whenever the sampled vector has norm larger than $s\sqrt{\kappa}$. The discrete-Gaussian tail estimate used in Lemma 2.22 bounds the removed mass by $\exp(-\Omega(\kappa))$. This norm cutoff also gives a polynomial-bit coordinate representation in the displayed rational basis. Padding all branches with unused random bits and dummy gates produces a fixed tape and a fixed circuit. Combining the sampling, truncation, finite-precision, and norm-tail errors proves the claim. □

We next study the output of amplitude amplification. The original results of [BHT98, BBHT98, BHMT02] did not explicitly include the output distribution of amplitude amplification conditioned on success. However, the conditional law is almost immediate from their procedure and Grover search [Gro96]. For completeness, we prove it in the next lemma.

Lemma C.1 (Conditional output law of Search)**.** *In the setting of Lemma 8.4, suppose g is a deterministic predicate computed exactly and reversibly, and $\mathcal{A}$ makes no measurements and acts on a fixed-length seed register. Then, conditioned on success, the returned x is distributed exactly as measuring $\mathcal{A}|0\rangle$ conditioned on $\{g = 1\}$:*

$$\Pr[\, x = z \mid \text{success}\,] \;=\; \frac{\|((|z\rangle\langle z|) \otimes I_{\text{work}})|\Psi_1\rangle\|^2}{a}, \qquad a = \||\Psi_1\rangle\|^2,$$

where $|\Psi_1\rangle = \Pi_1\mathcal{A}|0\rangle$ and Π_1 is the projector onto good output values tensored with the identity on any work registers. In particular, from a uniform-amplitude seed register, x is uniform over $\{g = 1\}$.

Proof. Let $\Pi_1 = (\sum_{z:\, g(z)=1}|z\rangle\langle z|) \otimes I_{\text{work}}$, $\Pi_0 = I - \Pi_1$, and write $|\Psi_1\rangle = \Pi_1\mathcal{A}|0\rangle$, $|\Psi_0\rangle = \Pi_0\mathcal{A}|0\rangle$, so $\mathcal{A}|0\rangle = |\Psi_1\rangle + |\Psi_0\rangle$ with $\langle\Psi_1|\Psi_0\rangle = 0$. By [BHMT02, §2] the Grover iterate $Q = -\mathcal{A}S_0\mathcal{A}^{-1}O_g$, where O_g is the phase oracle (Lemma 8.4) and $S_0 = I - 2|0\rangle\langle 0|$ (thus $\mathcal{A}S_0\mathcal{A}^{-1}$ is the reflection about $\mathcal{A}|0\rangle$), preserves the real plane $\mathrm{span}\{|\Psi_1\rangle, |\Psi_0\rangle\}$, so for every $j \ge 0$ the good component is a scalar multiple of the fixed vector $|\Psi_1\rangle$:

$$\Pi_1 Q^j \mathcal{A}|0\rangle \;=\; c_j\,|\Psi_1\rangle, \qquad c_j \in \mathbb{C}.$$

Measuring $Q^j\mathcal{A}|0\rangle$ in the computational basis and conditioning on $g = 1$ therefore returns z with probability

$$\frac{|c_j|^2\|((|z\rangle\langle z|) \otimes I_{\text{work}})|\Psi_1\rangle\|^2}{|c_j|^2\||\Psi_1\rangle\|^2} = \frac{\|((|z\rangle\langle z|) \otimes I_{\text{work}})|\Psi_1\rangle\|^2}{a}.$$

This is independent of j: the factor $|c_j|^2$ cancels, so every fixed-j run yields the same conditional law. Two observations extend this to Search itself. First, Search never disturbs this law before the final measurement: it applies only $\mathcal{A}, \mathcal{A}^{-1}, O_g, S_0$ and then measures in the computational basis, with no intermediate measurement or amplitude estimation. Second, its schedule (random j, geometric growth, restarts with a fresh $\mathcal{A}$) selects the number of iterations without reference to which good outcome z occurs. Hence the law given eventual success is a mixture over j of one and

the same law, and therefore equals it. The uniform case follows since a uniform-amplitude seed makes $|\Psi_1\rangle \propto \sum_{z:\, g(z)=1} |z\rangle$. □

Finally, we give a proof of the finite-group generation lemma.

Lemma 8.11 (Random generation by growth)**.** *Let G be a finite group, and let ν be a distribution on G with*

$$\mathrm{TV}(\nu, \mathrm{Unif}(G)) = \delta \le \frac{1}{4}.$$

Let $g_1, \dots, g_N$ be independent samples from ν. There is an absolute constant $C > 0$ such that, for every $u \ge 1$, if $N \ge C(\log_2 |G| + u)$, then

$$\Pr[\langle g_1, \dots, g_N \rangle = G] \ge 1 - 2^{-u}.$$

Proof. Let

$$H_i = \langle g_1, \dots, g_i \rangle, \qquad H_0 = \{1\}.$$

Call step $i+1$ successful if $g_{i+1} \notin H_i$. If $H_i < G$ is a proper subgroup, then $|H_i|/|G| \le 1/2$. Hence, by the total variation bound,

$$\Pr[g_{i+1} \in H_i \mid g_1, \dots, g_i] = \nu(H_i) \le \frac{|H_i|}{|G|} + \delta \le \frac{1}{2} + \delta \le \frac{3}{4}.$$

Therefore, whenever $H_i < G$ is proper,

$$\Pr[g_{i+1} \notin H_i \mid g_1, \dots, g_i] \ge \frac{1}{4}.$$

Moreover, if $g_{i+1} \notin H_i$, then H_i is a proper subgroup of $H_{i+1} = \langle H_i, g_{i+1} \rangle$. Thus

$$|H_{i+1}| \ge 2|H_i|.$$

Consequently, after $r := \lceil \log_2 |G| \rceil$ successful growth steps, the generated subgroup must be all of G.

It remains to bound the number of growth steps. Couple the process with independent Bernoulli variables $Y_1, \dots, Y_N$ of mean $1/4$ so that, as long as $H_{i-1} \neq G$, the event $Y_i = 1$ implies $g_i \notin H_{i-1}$. Hence, if $\sum_i Y_i \ge r$, then at least r proper growth steps occur and therefore $H_N = G$.

Now $Y := \sum_i Y_i \sim \mathrm{Bin}(N, 1/4)$. For a sufficiently large absolute constant C, if $N \ge C(r + u)$, then

$$\Pr[Y < r] \le 2^{-u}$$

by the Chernoff bound. Since $r = \lceil \log_2 |G| \rceil$, this proves the claim (possibly adjusting C). □

References

[ABGS21] Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. Fine-grained hardness of CVP(P)—everything that we can prove and

nothing else. In *SODA*, pages 1816–1835, 2021.

[ADRS15] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time using discrete Gaussian sampling. In *STOC*, pages 733–742, 2015.

[ARLW24] Léo Ackermann, Adeline Roux-Langlois, and Alexandre Wallet. Public-key encryption from the lattice isomorphism problem. In *WCC*, 2024.

[AS18] Divesh Aggarwal and Noah Stephens-Davidowitz. Just take the average! An embarrassingly simple 2^n-time algorithm for SVP (and CVP). In *SOSA*, pages 12:1–12:19, 2018.

[Bab91] László Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *STOC*, pages 164–174, 1991.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *STOC*, pages 684–697, 2016.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–505, 1998.

[BDS16] Huck Bennett, Daniel Dadush, and Noah Stephens-Davidowitz. On the lattice distortion problem. In *ESA*, pages 9:1–9:17, 2016.

[Ben73] Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17(6):525–532, 1973.

[Ber09] Daniel J. Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? In *SHARCS*, 2009.

[BF26] Huck Bennett and Kyle Fridberg. On (non-)isomorphism of self-dual lattices and codes. arXiv:2606.18662, 2026.

[BGPS23] Huck Bennett, Atul Ganju, Pura Peetathawatchai, and Noah Stephens-Davidowitz. Just how hard are rotations of $\mathbb{Z}^n$? Algorithms and cryptography with the simplest lattice. In *EUROCRYPT*, pages 252–281, 2023.

[BGS17] Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the quantitative hardness of CVP. In *FOCS*, pages 13–24, 2017.

[BHLS26] Huck Bennett, Minki Hhan, Zihan Li, and Noah Stephens-Davidowitz. Lattices with large kissing number from local density. arXiv:2610.14380, 2026.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Information*, volume 305 of *Contemporary Mathematics*, pages 53–74. American Mathematical Society, 2002.

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *LATIN*, pages 163–169, 1998.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, pages 575–584, 2013.

[BLS26] Huck Bennett, Zhengnan Lai, and Noah Stephens-Davidowitz. Advanced cryptography from lattice isomorphism—new constructions of IBE and FHE. In *CRYPTO*, 2026. Full version: Cryptology ePrint Archive, Paper 2026/465.

[BMM25] Pedro Branco, Giulio Malavolta, and Zayd Maradni. Fully-homomorphic encryption from lattice isomorphism. In *TCC*, pages 220–252, 2025.

[Bru69] Richard A. Brualdi. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society*, 1:161–167, 1969.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics*. Springer, 1993.

[Dad12] Daniel Nicolas Dadush. *Integer Programming, Lattice Algorithms, and Deterministic Volume Estimation*. PhD thesis, Georgia Institute of Technology, 2012.

[DPPvW22] Léo Ducas, Eamonn W. Postlethwaite, Ludo N. Pulles, and Wessel P. J. van Woerden. Hawk: Module LIP makes lattice signatures fast, compact and simple. In *ASIACRYPT*, pages 65–94, 2022.

[DSHVvW20] Mathieu Dutour Sikirić, Anna Haensch, John Voight, and Wessel P. J. van Woerden. A canonical form for positive definite matrices. *Open Book Series*, 4(1):179–195, 2020. Proceedings of ANTS XIV.

[DSSV09] Mathieu Dutour Sikirić, Achill Schürmann, and Frank Vallentin. Complexity and algorithms for computing Voronoi cells of lattices. *Mathematics of Computation*, 78(267):1713–1731, 2009.

[DvW22] Léo Ducas and Wessel P. J. van Woerden. On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In *EUROCRYPT*, pages 643–673, 2022.

[Edm71] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[ER65] Paul Erdős and Alfréd Rényi. Probabilistic methods in group theory. *Journal d'Analyse Mathématique*, 14:127–138, 1965.

[Fei96] Walter Feit. Orders of finite linear groups. In *Proceedings of the First Jamaican Conference on Group Theory and its Applications*, pages 9–11, 1996.

[FHL80] Merrick Furst, John Hopcroft, and Eugene Luks. Polynomial-time algorithms for permutation groups. In *FOCS*, pages 36–41, 1980.

[GLM08] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical Review Letters*, 100(16):160501, 2008.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219, 1996.

[HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *STOC*, pages 469–477, 2007.

[HR14] Ishay Haviv and Oded Regev. On the lattice isomorphism problem. In *SODA*, pages 391–404, 2014.

[Hun24] Christoph Hunkenschröder. Deciding whether a lattice has an orthonormal basis is in co-NP. *Mathematical Programming*, 208(1):763–775, 2024.

[JL26] Kaijie Jiang and Yinchen Liu. Exploiting the complexity of lattice isomorphism problem via irreducible decomposition. In *CRYPTO*, 2026. Full version: Cryptology ePrint Archive, Paper 2026/1139.

[JWL+25a] Kaijie Jiang, Anyu Wang, Hengyi Luo, Guoxiao Liu, Gang Tang, Yanbin Pan, and Xiaoyun Wang. Re-randomize and extract: A novel commitment construction framework based on group actions. In *EUROCRYPT*, pages 124–153, 2025.

[JWL+25b] Kaijie Jiang, Anyu Wang, Hengyi Luo, Guoxiao Liu, Yang Yu, and Xiaoyun Wang. Exploiting the symmetry of $\mathbb{Z}^n$: Randomization and the automorphism problem. *Journal of Cryptology*, 38(3):27, 2025.

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

[KB79] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979.

[KL78] Grigorii A. Kabatiansky and Vladimir I. Levenshtein. Bounds for packings on a sphere and in space. *Problemy Peredachi Informatsii*, 14(1):3–25, 1978.

[KS26] Veronika Kuchta and Francesco Sica. (Mis)using the lattice isomorphism problem. Cryptanalysis of the double-LIP and construction of LIP-based blind signatures. Cryptology ePrint Archive, Paper 2026/722, 2026.

[LJJ+26] Hengyi Luo, Kaijie Jiang, Renjie Jin, Yanbin Pan, and Anyu Wang. Commitment schemes based on module-LIP. In *CRYPTO*, 2026. Full version: Cryptology ePrint Archive, Paper 2025/431.

[LK26] Thijs Laarhoven and Scott Duke Kominers. A lattice family with kissing numbers $\tau(\mathcal{L}_n) \geq e^{2\sqrt{n}}$. arXiv:2609.13608, 2026.

[LLL82] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[LRvW25] Alberto Leporati, Lorenzo Rovida, and Wessel van Woerden. Beyond LWE: a lattice framework for homomorphic encryption. Cryptology ePrint Archive, Paper 2025/1171, 2025.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[MV13] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. *SIAM Journal on Computing*, 42(3):1364–1391, 2013.

[Pha25] Harry Pham. Cryptographic primitives from the lattice isomorphism problems. Master's thesis, Aalto University, 2025.

[PP85] Wilhelm Plesken and Michael Pohst. Constructing integral lattices with prescribed minimum. I. *Mathematics of Computation*, 45(171):209–221, 1985.

[PS97] Wilhelm Plesken and Bernd Souvignier. Computing isometries of lattices. *Journal of Symbolic Computation*, 24(3-4):327–334, 1997.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):34:1–34:40, 2009.

[Sch86] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

[Ser03] Ákos Seress. *Permutation Group Algorithms*, volume 152 of *Cambridge Tracts in Mathematics*. Cambridge University Press, 2003.

[Sim70] Charles C. Sims. Computational methods in the study of permutation groups. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 169–183. Pergamon, Oxford, 1970.

[Sim71] Charles C. Sims. Computation with permutation groups. In *SYMSAC*, pages 23–28, 1971.

[Ste16] Noah Stephens-Davidowitz. Discrete Gaussian sampling reduces to CVP and SVP. In *SODA*, pages 1748–1764, 2016.

[vNdP26] Chris van Noorden and Paola de Perthuis. Post-quantum anonymous signatures from the lattice isomorphism group action. In *SCN*, 2026. Full version: Cryptology ePrint Archive, Paper 2026/436.

[VV85] Leslie G. Valiant and Vijay V. Vazirani. NP is as easy as detecting unique solutions. In *STOC*, pages 458–463, 1985.

[vW24] Wessel P. J. van Woerden. Dense and smooth lattices in any genus. In *ASIACRYPT*, pages 386–417, 2024.