Exact ℓ_p-CVP via Euclidean SVP in Polynomial Space

Jiseung Kim
Jeonbuk National University
jiseungkim@jbnu.ac.kr

Minkyu Kim
The Affiliated Institute of ETRI
mkkim@nsr.re.kr

Hyung Tae Lee
Chung-Ang University
hyungtaelee@cau.ac.kr

Abstract

We give a randomized, rank-preserving reduction from exact ℓ_p-CVP to exact Euclidean SVP for every integer $p \geq 2$. For rational inputs of bit length ℓ on full-rank lattices $L \subset \mathbb{R}^N$, the reduction uses $(Cp)^N \operatorname{poly}(p, \ell)$ oracle calls and additional bit operations, where C is an absolute constant. It uses $\operatorname{poly}(p, \ell)$ working space outside the oracle, and all oracle queries have rank N and $\operatorname{poly}(p, \ell)$ bit complexity.

For $p = 2$, the reduction transfers polynomial-space exact SVP algorithms to exact CVP with $2^{O(N)} \operatorname{poly}(\ell)$ time overhead. Instantiating it with the SVP algorithms of Kannan and of Feng, Gao and Liu yields randomized polynomial-space CVP algorithms running in time $N^{N/(2e)+o(N)}$ and $N^{N/(4e)+o(N)}$, respectively. For every fixed integer $p \geq 3$, the same reduction yields exact ℓ_p-CVP algorithms for arbitrary rational targets either in randomized time and space $2^{O(N)}$, or in randomized time $N^{N/(4e)+o(N)}$ and polynomial space. These algorithmic bounds are up to polynomial factors in ℓ.

The reduction performs local search sequentially over the cosets of $2L$, where L is the input lattice, while tracking the excess of the current objective value over the optimum. Within a coset containing a closest vector, midpoint inequalities bound the unknown correction in terms of this excess, which in turn allows us to construct many short corrections that reduce the excess bound by a factor of four. A counting argument shows that these improving corrections form a sufficiently large fraction of the lattice points in a known search region, and a sparsification-based sampler assigns sufficiently large probability to every nonzero vector in that region, using at most one SVP call per sample. Repeating this improvement over polynomially many stages yields exact recovery. For $p \geq 3$, the same framework is extended using a diagonal weighted norm computed from the current point.

1 Introduction

The Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) are two fundamental computational problems on lattices. A rank-N lattice $L \subset \mathbb{R}^N$ is the set of all integer combinations of N linearly independent vectors. Given such a lattice, SVP asks for a shortest nonzero vector of L, while CVP asks for a lattice vector closest to a given target $t \in \mathbb{R}^N$. We study the exact versions of these problems, in which an optimal vector must be found.

In the Euclidean norm, SVP reduces to CVP without increasing the rank [GMSS99]. In contrast, reducing CVP to SVP is particularly interesting under space constraints. With exponential space, exact Euclidean CVP can be solved in time $2^{N+o(N)}$ [ADS15], while exact Euclidean SVP admits faster algorithms, most recently Hhan's algorithm with time and space $2^{N/2+o(N)}$ [Hha26, Theorem 1.1]. In polynomial space, however, a substantial gap has persisted. Kannan's enumeration algorithm solves SVP in time $N^{N/(2e)+o(N)}$, whereas the corresponding CVP bound is $N^{N/2+o(N)}$ [Kan83, HS07]; Fu [Fu16] improves only a lower-order factor in the latter bound. More recently, Feng, Gao and Liu announced a randomized polynomial-space SVP algorithm running in time $N^{N/(4e)+o(N)}$ [FGL26]. Thus, the best known worst-case polynomial-space running times for SVP and CVP have differed by a factor of $N^{\Omega(N)}$.

This gap reflects a limitation of known algorithms rather than a known separation between the problems. The fastest exact CVP algorithms rely on large collections of lattice objects, such as Voronoi-relevant vectors [MV10] or exponentially many discrete Gaussian samples [ADS15]. This raises a natural question: can exact CVP be solved using an SVP oracle while maintaining only polynomial working space? We answer this question affirmatively with a randomized, rank-preserving reduction from exact CVP to exact Euclidean SVP that uses polynomial working space outside the oracle.

Our approach also addresses a longstanding question about exact CVP in non-Euclidean norms. For an integer $p \geq 1$, let $\|z\|_p = (\sum_i |z_i|^p)^{1/p}$, and define ℓ_p-CVP as the problem of finding a lattice vector minimizing $\|x - t\|_p$. For every fixed integer $p \geq 3$, single-exponential algorithms for exact ℓ_p-CVP were known when the distance from the target to the lattice is at most a constant multiple of the shortest-vector length [DPV11], [DV12, Theorem 1.9], but not for arbitrary targets. Micciancio and Voulgaris [MV10, Section 1.2] explicitly identified extending single-exponential exact CVP algorithms to other norms, even with randomization, as an open problem. Blömer and Kohn [BK18, Theorem 7] showed that the number of Voronoi-relevant vectors can be unbounded even for the ℓ_3 norm in dimension three, illustrating a fundamental obstacle to extending the Euclidean Voronoi-cell approach. For arbitrary targets, previous algorithms achieve $N^{O(N)}$ time in polynomial space [BN11] or $(\log N)^{O(N)}$ time via integer programming [RR26, Theorem 5]. Known reductions between norms [ACK+21, EV20] incur approximation loss and therefore do not resolve the exact problem.

Our reduction addresses both gaps within a unified framework. For $p = 2$, it transfers polynomial-space exact SVP algorithms to exact CVP with only single-exponential overhead, while preserving the rank. For every fixed integer $p \geq 3$, it converts Euclidean SVP algorithms into arbitrary-target exact ℓ_p-CVP algorithms, yielding either a single-exponential time-and-space instantiation or a polynomial-space instantiation. In both cases, the reduction itself uses only polynomial working space outside the oracle.

1.1 Our Results

We now state the reduction precisely. For $p = 2$, the input consists of a positive definite rational Gram matrix $G = B^T B$ and the coordinates $\bar{t} \in \mathbb{Q}^N$ of the target with respect to B, so that $t = B\bar{t}$; the basis B itself need not be given. For $p \geq 3$, we consider full-rank lattices in $\mathbb{R}^N$ given by nonsingular rational $N \times N$ bases, together with rational targets. Let ℓ denote the total bit length of the input, including the encodings of p and the requested failure probability ε.

Theorem 1.1. *There exist an absolute constant C and a randomized algorithm with access to an exact Euclidean SVP oracle such that, for every integer $p \geq 2$, every rational $\varepsilon \in (0, 1/2)$, and every input as described above, the algorithm outputs an ℓ_p-closest vector to the target with probability at*

least $1-\varepsilon$. *For* $p=2$, *the output is represented by its coefficient vector with respect to* B. *On every execution, it*

(i) makes at most $(Cp)^N \operatorname{poly}(p,\ell)$ *oracle calls and performs as many additional bit operations,*
(ii) queries the oracle only on lattices of rank N *given by* $\operatorname{poly}(p,\ell)$*-bit Gram matrices, and*
(iii) uses $\operatorname{poly}(p,\ell)$ *space outside the oracle.*

With the same asymptotic bounds and success probability, the reduction can output all closest vectors exactly once. There are at most 2^N such vectors, and this is tight for $L=\mathbb{Z}^N$ and $t=(1/2,\dots,1/2)$.

For $p=2$, the main consequence of Theorem 1.1 is a transfer of polynomial-space SVP bounds to exact Euclidean CVP. Plugging in the best available polynomial-space SVP algorithms gives the following concrete bounds.

Corollary 1.2. *Exact Euclidean CVP on the rank-*N *inputs described in Theorem 1.1 can be solved with probability at least* $1-\varepsilon$ *by randomized polynomial-space algorithms with running times*

$$N^{N/(2e)+o(N)}\operatorname{poly}(\ell) \qquad \text{and} \qquad N^{N/(4e)+o(N)}\operatorname{poly}(\ell),$$

obtained respectively from Kannan's SVP algorithm [HS07, Theorem 2] and the algorithm of Feng, Gao and Liu [FGL26, Theorem 4.1].

The underlying parameterized transfer is stated in Corollary 7.3. Together with the reduction of [GMSS99] in the opposite direction, it also shows that the randomized polynomial-space time complexities of exact Euclidean SVP and CVP agree up to a factor $2^{O(N)}\operatorname{poly}(\ell)$; see Corollary 7.4.

For every fixed integer $p\geq 3$, Theorem 1.1 also yields two complementary time-space tradeoffs for full-rank lattices with arbitrary rational targets.

Corollary 1.3. *For every fixed integer* $p\geq 3$, *exact* ℓ_p*-CVP on full-rank lattices given by rational bases, with arbitrary rational targets, can be solved with probability at least* $1-\varepsilon$ *either in randomized time and space* $2^{O(N)}\operatorname{poly}(\ell)$, *or in randomized time* $N^{N/(4e)+o(N)}\operatorname{poly}(\ell)$ *and polynomial space.*

The first bound uses the deterministic single-exponential SVP algorithm of Micciancio and Voulgaris [MV10] after the rank-preserving integer-basis conversion of Lemma A.1. The second uses the polynomial-space algorithm of Feng, Gao and Liu [FGL26, Theorem 4.1], together with Lemmas A.1 and A.2.

1.2 Technical Overview

The reduction is a randomized local search for a closest vector. We first explain the Euclidean case. Fix a target t, a closest vector y, and a current lattice point x. If y were known, the vector $y-x$ would move x to the optimum in one step. Since y is unknown, the algorithm instead searches for short lattice vectors that make substantial progress toward the optimum.

To make such a local search possible, we need to relate progress in the objective to geometric distance from y. For a lattice point x, define its *excess* by $\|x-t\|^2-\|y-t\|^2$. This quantity measures how far the objective value at x is above the optimum. The algorithm does not know the excess, because it does not know y; it is used only in the analysis. Ideally, a small excess would imply that x is close to y, because then the unknown correction $y-x$ would necessarily be short. This implication, however, is false in general.

From small excess to geometric closeness. For example, take $L = \mathbb{Z}$ and $t = 1/2 - \delta$ with $0 < \delta < 1/4$. The target lies slightly to the left of the midpoint between 0 and 1, so its closest lattice vector is $y = 0$. Consider instead the neighboring lattice point $x = 1$. Its excess is $(1/2+\delta)^2 - (1/2-\delta)^2 = 2\delta$. As δ tends to zero, this excess becomes arbitrarily small: the two lattice points 0 and 1 are almost equally good with respect to the objective. Nevertheless, $|x - y| = 1$ remains fixed. Thus, having a small excess does not by itself imply being geometrically close to a closest vector, and hence does not guarantee that the desired correction $y - x$ has small length.

This difficulty disappears once we restrict attention to a single coset of the sublattice $2L$. Suppose that x and y lie in the same coset, so that $x - y \in 2L$. Then, their midpoint $(x+y)/2$ is itself a lattice point. Since y is closest to t, this midpoint cannot be closer to t than y: $\|(x+y)/2 - t\|^2 \geq \|y - t\|^2$. Expanding this inequality yields a special case of the parity-clustering inequality of Aggarwal, Dadush and Stephens-Davidowitz [ADS15, Lemma 5.3]:

$$\|x - y\|^2 \leq 2\big(\|x - t\|^2 - \|y - t\|^2\big).$$

Therefore, within the coset of y, an excess of at most Δ implies $\|x - y\|^2 \leq 2\Delta$. In other words, within a coset of $2L$, a small excess guarantees that the current point is geometrically close to a closest vector, and hence that the required correction is short. Notice that the two points 0 and 1 in the example above belong to different cosets of $2\mathbb{Z}$, so the counterexample no longer applies.

This provides the geometric control needed for local search, but only in a coset containing a closest vector. Rather than identifying such a coset in advance, the algorithm runs the same search independently in all 2^N cosets of $2L$ in L. In a coset containing a closest vector y, the bound above ensures that a small excess makes the correction $y - x$ short, and the analysis below shows that the search recovers y with high probability. In other cosets, no such guarantee is needed. After processing all cosets, the algorithm compares the resulting candidates by their distance to t and returns the closest one.

From one unknown correction to many good corrections. It remains to explain how the search makes progress in a coset containing a closest vector. Suppose that x lies in the same coset of $2L$ as a closest vector y and has excess at most Δ, and let $\eta = y - x$. The bound above gives $\|\eta\|^2 \leq 2\Delta$, so the length of the unknown correction η is controlled by the current excess bound. The key observation is that the algorithm does not need to recover η itself.

Let $S = 2L \cap \mathcal{B}(\sqrt{\Delta/6})$, where $\mathcal{B}(r)$ denotes the Euclidean ball of radius r centered at the origin. For every $w \in S$, the point $y + w$ has excess at most $\Delta/4$: by Lemma 4.1,

$$\|y + w - t\|^2 - \|y - t\|^2 \leq \tfrac{3}{2}\|w\|^2 \leq \tfrac{\Delta}{4}.$$

Starting from x, the same point is reached by the correction $\eta + w$, since $x + (\eta + w) = y + w$. Hence, every vector in the translated set $\eta + S$ is a successful correction that reduces the excess bound from Δ to $\Delta/4$. Thus, instead of relying on the single unknown correction η, we have an entire set $\eta + S$ of successful corrections, with $|\eta + S| = |S|$. Figure 1(a) illustrates the good points around y, while Figure 1(b) shows the corresponding corrections in $\eta + S$.

It therefore suffices to draw a random vector $V \sim \mathcal{D}$ from a distribution $\mathcal{D}$ on $2L$ such that $V \in \eta + S$ with sufficient probability; a realized sample is denoted by v in Figure 1. All vectors of $\eta + S$ have length less than $2\sqrt{\Delta}$, a radius that the algorithm knows. To compare the $|S|$ successful corrections with all lattice vectors in this search ball, cover $\mathcal{B}(2\sqrt{\Delta})$ by balls of radius $\frac{1}{2}\sqrt{\Delta/6}$. By Lemma 2.2, at most 21^N such balls are needed. If one of these smaller balls contains a lattice point $x_k \in 2L$, then any other lattice point $x \in 2L$ in the same ball satisfies $\|x - x_k\| \leq \sqrt{\Delta/6}$. Since

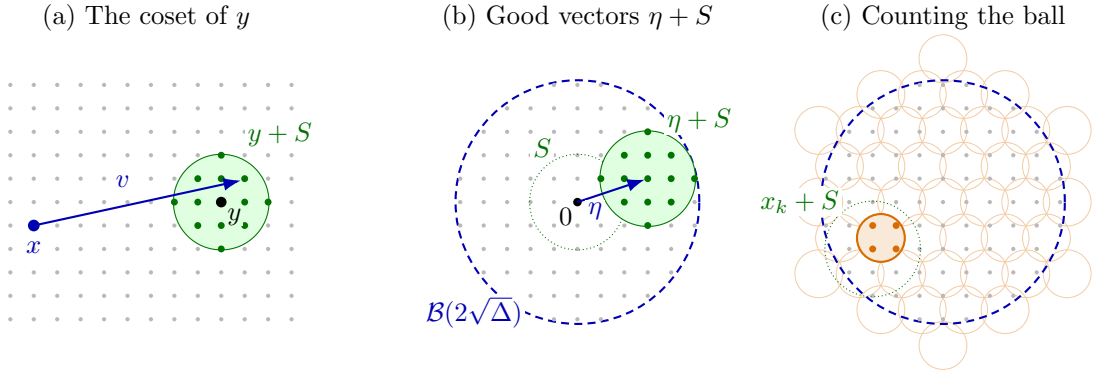


Figure 1: Constructing and counting improving corrections in the coset of a closest vector y, where $\eta = y - x$.

$x - x_k \in 2L$, we have $x - x_k \in S$, or equivalently $x \in x_k + S$. Thus, every smaller ball contains at most $|S|$ points of $2L$ (Lemma 2.1), and hence

$$|2L \cap \mathcal{B}(2\sqrt{\Delta})| \;\le\; 21^N \, |S|$$

for every lattice, independently of its shape. This counting argument is illustrated in Figure 1(c). Consequently, if $\mathcal{D}$ assigns probability at least $\gamma/|2L \cap \mathcal{B}(2\sqrt{\Delta})|$ to every nonzero point of $2L \cap \mathcal{B}(2\sqrt{\Delta})$, then $\Pr[V \in \eta + S]$ is at least $\gamma \, 21^{-N}/2$ (Lemma 4.2). Neither $\mathcal{D}$ nor the radius depends on y.

Sampling only what the reduction needs. We now need a way to sample vectors from the search ball using an SVP oracle such that every nonzero lattice vector in the search ball has a sufficiently large chance of being returned. Some vectors may be sampled more often than others; the reduction only needs to ensure that none of them is too unlikely. The sampler of Theorem 3.1 provides exactly this guarantee, with $\gamma = 2^{-O(N)}/\operatorname{poly}(\ell)$, using at most one exact SVP call per sample.

The basic idea is to use lattice sparsification to isolate one short lattice direction. Fix any nonzero vector z in the search ball. There is a unique way to write $z = mu$, where $m \ge 1$ is an integer and u is primitive, meaning that u is not itself an integer multiple of another lattice vector. Thus, to produce z, it is enough to recover the direction u and guess the multiplier m.

To recover u, the sampler chooses a prime q and imposes a random linear equation modulo q on the coefficient vectors of the lattice. Only the vectors satisfying this equation are kept, and they form a full-rank sublattice. A fixed primitive vector survives this filtering with probability $1/q$. Thus, q controls how strongly the lattice is sparsified: increasing q removes more competing vectors, but also makes it less likely that u itself survives.

For an appropriate scale of q, there is sufficient probability that $\pm u$ survive while all other vectors that could be as short as u are removed. On this event, $\pm u$ are the unique shortest nonzero vectors of the sparsified lattice, so a single SVP call returns u up to sign. Multiplying the result by the guessed value of m then gives $\pm z$.

The sampler does not know the correct multiplier m or the appropriate scale of q in advance, so it guesses both. The analysis shows that, despite these guesses, every nonzero vector in the search ball is returned with sufficiently large probability. This sparsification step follows the primitive-vector

sampling approach of Stephens-Davidowitz [Ste16, Section 4.1], building on Khot [Kho05] and Dadush and Kun [DK16]; see Remark 3.2 for a brief comparison and Section 6.3 for further details.

From local progress to exact recovery. The search proceeds in stages with excess bounds decreasing by a factor of four. In the coset containing y, $2^{O(N)}\,\mathrm{poly}(\ell)$ samples suffice at each stage to obtain the next-scale improvement with high probability. After polynomially many stages, the excess bound is below $\lambda_1(2L)^2/2$, and the midpoint inequality forces the current point to equal y. Since the 2^N cosets and all samples are processed sequentially, the reduction stores only a constant number of lattice points, one sampler workspace, and a few counters, yielding polynomial working space outside the oracle.

The ℓ_p case. For an integer $p \geq 3$, the same strategy still applies: work within cosets of $2L$, show that many short corrections make a constant-factor improvement, and use the same sparsification sampler to hit one of them. The new difficulty is geometric. The Euclidean length of a correction alone no longer provides the control needed to bound its effect on $F_t(z) = \sum_i |z_i - t_i|^p$. A displacement in a coordinate with large residual $|z_i - t_i|$ is more expensive than the same displacement in a coordinate with small residual.

We therefore replace the Euclidean norm by a diagonal weighted norm computed from the current point x:

$$\|w\|_{H_x}^2 = \textstyle\sum_i h_i w_i^2, \qquad h_i = |x_i - t_i|^{p-2} + \rho, \qquad \rho = \Theta\big((\Delta/N)^{1-2/p}\big).$$

The curvature relevant to perturbations around the optimum depends on the unknown minimizer y, rather than the current point x. Midpoint inequalities are what make the construction usable: if x and y lie in the same coset and $F_t(x) - F_t(y) \leq \Delta$, then they imply that $y - x$ is short in the computable norm H_x and that the weights at x control those around y (Lemmas 5.1 and 5.2).

The good perturbations around y are now chosen from a coordinate box rather than a ball. This gives coordinatewise control of the Taylor remainder of F_t without losing an $N^{\Omega(N)}$ factor in volume. By symmetry, at least half of the lattice points in this box also make the first-order term nonpositive. A covering argument then shows that a $(Cp)^{-N}$ fraction of the lattice points in a known H_x-ball yield the same factor-four improvement (Lemma 5.4). Since Theorem 3.1 works for any rational positive-definite metric, the same sparsification sampler applies to H_x. Finally, the rationality of the input makes the possible values of F_t discrete, so once the maintained excess bound is smaller than the minimum positive difference between distinct objective values, the current point is exactly optimal.

1.3 Related Work

Ajtai, Kumar and Sivakumar [AKS02] solve $(1+\varepsilon)$-approximate CVP in time $2^{O(N(1+1/\varepsilon))}$ and exponential space. Naszódi and Venzin [NV22] solve $(1+\varepsilon)$-approximate CVP in ℓ_p norms for fixed p, whereas our result concerns exact ℓ_p-CVP. Arya, da Fonseca and Mount [AdFM24] give a randomized $(1+\varepsilon)$-CVP algorithm for arbitrary norms with running time $2^{O(N)}\varepsilon^{-(N-1)/2}$, up to polynomial factors in the input size. Dubey and Holenstein [DH11] give a polynomial-time Turing reduction from $\gamma^2\sqrt{N}$-approximate Euclidean CVP to γ-approximate Euclidean SVP. The CVP-to-SVP reductions of [ACK+21] incur approximation loss and increase the lattice rank from N to $N+1$. Dadush and Micciancio [DM13, Section 5.4] discuss a potential application of densest-sublattice preprocessing to faster polynomial-space enumeration for CVP with preprocessing, under a heuristic assumption. Hunkenschröder, Reuland and Schymura [HRS20] solve CVP in polynomial

space given a basis in which every facet normal of the Voronoi cell has coefficients bounded by some c, in time $(2c+1)^{O(N)}\,\mathrm{poly}(N)$. For Haar-random lattices, Laarhoven [Laa26, Theorem 1.8] obtains a polynomial-space exact CVP with preprocessing (CVPP) algorithm with query time $N^{0.0892N+o(N)}$, excluding preprocessing costs. Stephens-Davidowitz [Ste16] gives polynomial-time, dimension-preserving reductions from centered discrete Gaussian sampling to SVP and from shifted discrete Gaussian sampling to CVP. The exact CVP algorithm of [ADS15] uses shifted samples and requires $2^{N+o(N)}$ time and space, so these results together do not give a polynomial-space reduction from CVP to SVP. Dadush, Eisenbrand and Rothvoss [DER25] make approximate integer programming exact when the residues of a solution modulo $\Theta(N)$ are known, and guessing these residues costs $N^{O(N)}$. In contrast, our reduction enumerates only the 2^N residue classes modulo 2, using midpoint inequalities specific to the ℓ_p objective.

Organization. Section 2 introduces notation and two counting lemmas; Section 3 presents the short-vector sampler and states its guarantees; and Sections 4 and 5 prove Theorem 1.1. Section 6 proves the sampling theorem, Section 7 gives the deferred CVP proofs and polynomial-space consequences, and Appendix A treats oracle implementation details.

2 Preliminaries

We introduce the notation and geometric tools used throughout the paper.

Lattices. A lattice of rank N is $L = B\mathbb{Z}^N$ for a nonsingular $B \in \mathbb{R}^{N\times N}$ with columns $b_1, \dots, b_N$. Its Gram matrix is $G = B^T B$, and we write $\lambda_1(L)$ for the minimum length of a nonzero vector of L.

For the Euclidean reduction, a lattice vector Bz is represented by its coefficient vector $z \in \mathbb{Z}^N$, and the target $t = B\bar{t}$ by $\bar{t} \in \mathbb{Q}^N$. Thus, $\|Bz - t\|^2 = (z-\bar{t})^T G(z-\bar{t})$. In our reduction for $p = 2$, only the positive definite rational matrix G and the rational vector $\bar{t}$ are given; a basis B with $B^T B = G$ is never computed. We will use a simple lower bound on the least eigenvalue of G to convert length bounds into bounds on coefficient vectors. Put $\iota(G) = \det G/(\operatorname{Tr} G)^{N-1}$. Since each of the remaining $N-1$ eigenvalues is at most $\operatorname{Tr} G$, $\det G \le \lambda_{\min}(G)(\operatorname{Tr} G)^{N-1}$, and hence $\iota(G) \le \lambda_{\min}(G)$. Therefore,

$$z^T G z \;\ge\; \iota(G)\,\|z\|^2 \qquad \text{for all } z \in \mathbb{R}^N. \tag{1}$$

When G is rational, $\iota(G)$ is computed exactly from G. In particular, $\lambda_1(L)^2 \ge \iota(G)$, because every nonzero lattice vector has a nonzero integer coefficient vector.

We write $\operatorname{dist}(t, L) = \min_{x\in L}\|x - t\|$, and log for the natural logarithm. The cosets of $2L$ in L are $Ba + 2L$, $a \in \{0,1\}^N$.

Norms given by positive definite matrices. For symmetric matrices A and B, we write $A \succeq B$ if $A - B$ is positive semidefinite, and $A \succ B$ if $A - B$ is positive definite. In particular, $H \succ 0$ means that H is positive definite. For a symmetric positive definite matrix H, which we call a *metric*, let $\|x\|_H = (x^T H x)^{1/2}$, $\mathcal{B}_H(r) = \{x \in \mathbb{R}^N : \|x\|_H \le r\}$ and $\lambda_1(L, H) = \min_{0\neq x\in L}\|x\|_H$. We drop H when $H = I$. For $L = B\mathbb{Z}^N$, we have $\|Ba\|_H^2 = a^T(B^T H B)a$, so the pair (L, H) is described by the Gram matrix $B^T H B$. In our algorithms, this matrix is rational and squared lengths are compared exactly.

SVP oracles. An SVP oracle receives a positive definite rational Gram matrix and returns the coefficient vector of a shortest nonzero vector. When there are several shortest vectors, the oracle may choose among them by any rule that depends on the history of the computation and on independent random bits. Our guarantees hold for every such rule. In Appendix A, Lemma A.1 converts a rational Gram matrix into an integer basis of the same rank whose shortest coefficient vectors are shortest for the original matrix, and Lemma A.2 shows how to simulate the exact oracle using an SVP algorithm that succeeds with probability 2/3.

Primitive vectors. A nonzero $u \in L$ is *primitive* if $u/k \notin L$ for every integer $k \geq 2$, equivalently if the greatest common divisor of its coefficients is 1. Every nonzero $x \in L$ equals mu for exactly one integer $m \geq 1$ and one primitive $u \in L$. Following [Ste16], for $s > 0$ we write $\xi_{L,H}(s)$ for the number of primitive vectors of L in $\mathcal{B}_H(s)$, counting each pair $\{u, -u\}$ once.

Counting lattice points. Let $\omega_N = \pi^{N/2}/\Gamma(N/2+1)$ be the volume of the Euclidean unit ball, so that $\mathrm{vol}(\mathcal{B}(R)) = \omega_N R^N$. Since $\Gamma(z+1) \geq (z/e)^z$ for $z > 0$,

$$\omega_N \;\leq\; (2\pi e/N)^{N/2}. \tag{2}$$

The next two lemmas are standard packing and covering bounds; their proofs are deferred to Section 7. They compare the numbers of lattice points in two sets of different sizes or shapes and require no assumption on the lattice. If a large set is covered by T translates of half of a small symmetric convex set Q, then it contains at most T times as many lattice points as Q.

Lemma 2.1. *Let $L \subset \mathbb{R}^N$ be a lattice and $Q \subset \mathbb{R}^N$ a convex, compact, symmetric set. If $X \subseteq \bigcup_{k=1}^{T}(c_k + \frac{1}{2}Q)$ for some $c_1, \ldots, c_T \in \mathbb{R}^N$, then $|L \cap X| \leq T \cdot |L \cap Q|$.*

Lemma 2.2. *The following covering bounds hold.*

(i) For every positive definite metric H and $R, \delta > 0$, the ball $\mathcal{B}_H(R)$ is contained in a union of at most $(1 + 2R/\delta)^N$ translates of $\mathcal{B}_H(\delta)$.

(ii) For the Euclidean norm, $R > 0$ and $0 < \alpha \leq R/\sqrt{N}$, the ball $\mathcal{B}(R)$ is contained in a union of at most $\omega_N \cdot (2R/\alpha)^N$ translates of the cube $[0, \alpha]^N$.

3 Sampling Short Lattice Vectors with an SVP Oracle

Throughout this section, K denotes the lattice given to the sampler, to distinguish it from the CVP lattice L used in Sections 4 and 5. The progress arguments in those sections require only a pointwise lower bound on the probability of every nonzero lattice vector in a known ball. More precisely, for a distribution $\mathcal{D}$ on K, a radius $r > 0$, and a number $\gamma \in (0, 1]$, we require

$$\mathcal{D}(x) \;\geq\; \frac{\gamma}{|K \cap \mathcal{B}_H(r)|} \qquad \text{for every } x \in (K \cap \mathcal{B}_H(r)) \setminus \{0\}. \tag{3}$$

If $V \sim \mathcal{D}$, then summing over x in (3) gives $\Pr[V \in E] = \mathcal{D}(E) \geq \gamma\,|E \setminus \{0\}|/|K \cap \mathcal{B}_H(r)|$ for every $E \subseteq K \cap \mathcal{B}_H(r)$. No upper bound on $\mathcal{D}(x)$ is needed, and $\mathcal{D}$ may assign large probability to 0. Algorithm 1 gives the sampler used in the reduction; the following theorem states its guarantees.

Theorem 3.1. *Let $K = B_K\mathbb{Z}^N$ and let $H \succ 0$ be a metric such that $G_K := B_K^T H B_K \in \mathbb{Q}^{N\times N}$. Let $r^2 > 0$ and $\lambda_0^2 > 0$ be rational numbers satisfying $\lambda_0 \leq \lambda_1(K, H)$. There is a randomized algorithm with access to an exact SVP oracle that, given G_K, r^2, and λ_0^2, outputs a coefficient vector representing a lattice vector in K. Let $\mathcal{D}$ be the induced distribution of this lattice vector, and let $V \sim \mathcal{D}$. Then the following properties hold.*

Algorithm 1: Sampling short lattice vectors

Input : Rational $G_K \succ 0$, $r^2 > 0$, and $0 < \lambda_0^2 \le \lambda_1(K,H)^2$
Output : Coefficient vector of a point of $K \cap \mathcal{B}_H(r)$

if $r^2 < \lambda_0^2$ **then return** 0
$J_{\mathrm{s}} \leftarrow \max\{\kappa \ge 0 : 4^\kappa \lambda_0^2 \le r^2\}$
$i_{\max} \leftarrow N(J_{\mathrm{s}} + 4) + 1$
choose $\kappa \in \{0, \dots, J_s\}$, $i \in \{1, \dots, i_{\max}\}$, and then $m \in [2^\kappa, 2^{\kappa+1})$ by procedure (1), using independent random bits
choose a prime $q \in [2^i, 2^{i+1})$ by procedure (2) and $z \in \mathbb{F}_q^N$ by procedure (3)
if any of procedures (1)–(3) fails **then return** 0
compute D_z as defined at the beginning of Section 6
$a \leftarrow D_z a'$, where a' is the oracle answer on $D_z^T G_K D_z$
if $m^2 a^T G_K a \le r^2$ **then return** ma or $-ma$, each with probability $1/2$
return 0

(a) $\mathrm{supp}(\mathcal{D}) \subseteq K \cap \mathcal{B}_H(r)$*; equivalently,* $V \in K \cap \mathcal{B}_H(r)$ *with probability* 1*.*

(b) $\mathcal{D}$ *satisfies* (3) *with*

$$\gamma = \frac{9^{-N}}{512\, N \left(\log_2 \max\{1, r/\lambda_0\} + 4\right)^2},$$

conditionally on any history of earlier computations and for every rule by which the oracle chooses among shortest vectors.

(c) The algorithm makes at most one oracle call, on a full-rank sublattice of K *in the metric* H *given by a Gram matrix whose bit length is polynomial in that of the input. Its other bit operations and its space are polynomial in the bit lengths of the input and of the oracle answer.*

The reductions supply a computable lower bound λ_0 on $\lambda_1(K,H)$; the proof of Theorem 3.1 is deferred to Section 6.

Construction. Write every nonzero $x \in K$ uniquely as $x = mu$ with u primitive. The sampler guesses m and a sparsification scale, and uses a random congruence to isolate $\pm u$ in a full-rank sublattice. Writing $K = B_K \mathbb{Z}^N$, for a prime q and $z \in \mathbb{F}_q^N$ define

$$\Lambda_z = \{a \in \mathbb{Z}^N : \langle z, a \rangle \equiv 0 \pmod q\}, \qquad K_z = B_K \Lambda_z.$$

If D_z is a basis matrix of Λ_z, then the query Gram matrix is $D_z^T G_K D_z$ and an oracle answer a' corresponds to the coefficient vector $D_z a'$ in K.

For a fixed $x = mu \in K \cap \mathcal{B}_H(r)$, one choice of the multiplier range and prime scale makes the random congruence isolate $\pm u$ with sufficient probability. The basis D_z, the bounded-randomness procedures (1)–(3), the counting and isolation bounds, and the exact calculation of (3) are specified in Section 6.

Remark 3.2. The sparsification step follows Stephens-Davidowitz [Ste16, Section 4.1]. Our reduction does not use the shifted/discrete-Gaussian part of that work: it needs only centered short-vector samples and the pointwise bound (3). The detailed comparison is in Section 6.

4 Exact CVP in the Euclidean Norm

We first prove Theorem 1.1 for $p = 2$. Throughout the section, $L = B\mathbb{Z}^N$ has positive definite rational Gram matrix G, and $t = B\bar{t}$ with $\bar{t} \in \mathbb{Q}^N$.

4.1 Geometry and One-Step Progress

The following two lemmas formalize the Euclidean ingredients of Section 1.2; full proofs are deferred to Section 7.2. The consequence $\|x - y\|^2 \le 2\Delta$ in Lemma 4.1 is the exact-CVP specialization of the parity-clustering inequality of Aggarwal, Dadush and Stephens-Davidowitz [ADS15, Lemma 5.3]; here we use it as part of the construction of a sufficiently large set of improving corrections.

Lemma 4.1. *Let y be a closest vector of L to t. For every $w \in 2L$,*

$$\tfrac{1}{2}\|w\|^2 \;\le\; \|y + w - t\|^2 - \|y - t\|^2 \;\le\; \tfrac{3}{2}\|w\|^2.$$

Consequently, for every $\Delta \ge 0$ and every $x \in y + 2L$ with $\|x - t\|^2 \le \|y - t\|^2 + \Delta$,

(i) $\|x - y\|^2 \le 2\Delta$;

(ii) if $\Delta < \lambda_1(2L)^2/2$, then $x = y$.

In particular, each coset of $2L$ in L contains at most one closest vector of L to t.

Lemma 4.2. *Let y be a closest vector of L to t, let $\Delta > 0$, and let $x \in y + 2L$ satisfy $x \ne y$ and $\|x - t\|^2 \le \|y - t\|^2 + \Delta$. Let $\mathcal{D}$ be a distribution on $2L$ satisfying (3) with $K = 2L$, $H = I$, $r = 2\sqrt{\Delta}$ and some $\gamma \in (0, 1]$, and let $V \sim \mathcal{D}$. Then*

$$\Pr\big[\|x + V - t\|^2 \le \|y - t\|^2 + \Delta/4\big] \;\ge\; \tfrac{1}{2} \cdot 21^{-N}\gamma.$$

For the proof, put $\eta = y - x$, $S = 2L \cap \mathcal{B}(\sqrt{\Delta/6})$, and $E = \eta + S$. Lemma 4.1 makes every vector of E improving and places E inside the sampled ball, while Lemmas 2.2 and 2.1 give $|2L \cap \mathcal{B}(2\sqrt{\Delta})| \le 21^N |S|$. Thus at least a $\frac{1}{2}21^{-N}$ fraction of the relevant nonzero vectors are good; see Section 7.2.

4.2 Algorithm for Exact Euclidean CVP

Algorithm 2 presents our algorithm for exact Euclidean CVP. It processes the 2^N cosets of $2L$ separately and uses stages with excess bounds decreasing by a factor of four. Put

$$U = (N + 1)(\bar{t}^T G \bar{t} + \operatorname{Tr} G), \qquad \Delta_j = U/4^j, \qquad J = \min\{j \ge 0 : \Delta_j < 2\iota(G)\}.$$

Here, U bounds the squared distance from t to every starting point Ba, $a \in \{0, 1\}^N$, and $\lambda_1(2L)^2 \ge 4\iota(G)$, so the final scale forces exactness. Here, $J \ge 1$ since $U \ge 2 \operatorname{Tr} G \ge 2\iota(G)$. In stage j the sampler is called on $(4G, 4\Delta_j, 4\iota(G))$. With

$$\theta = \frac{189^{-N}}{1024N(\lceil \log_2(U/\iota(G)) \rceil + 4)^2}, \qquad \mathcal{M} = \theta^{-1}\left\lceil \log_2 \frac{4 \cdot 2^N J}{\varepsilon} \right\rceil,$$

we have $J = \operatorname{poly}(\ell)$ and $\mathcal{M} = 2^{O(N)} \operatorname{poly}(\ell)$.

Correctness and resources. For the coset containing a closest vector y, maintain $\|x_j - t\|^2 \le \|y - t\|^2 + \Delta_j$. By Lemma 4.2 and Theorem 3.1, each trial reaches the next scale with conditional probability at least θ; repetition and a union bound over stages and cosets give the required success probability, and the last scale forces $x_J = y$. The algorithm makes $2^N J\mathcal{M} = 2^{O(N)} \operatorname{poly}(\ell)$ rank-N oracle calls, and all stored quantities and queries have polynomial bit length, so the non-oracle space is polynomial. The two-run all-closest-vector variant and the bounded-error-oracle substitution use Lemmas 4.1 and A.2. Hence, Theorem 1.1 holds for $p = 2$; full details are in Section 7.2.

Algorithm 2: Exact CVP in the Euclidean norm

Input : Rational Gram matrix G, target coefficients $\bar{t}$, rational $\varepsilon \in (0, 1/2)$
Output : Coefficient vector of a closest lattice point in the Euclidean norm

$c_{\text{best}} \leftarrow 0$
for $a \in \{0,1\}^N$ **do**
  $c \leftarrow a$
  for $j = 0, \dots, J-1$ **do**
    $c_{\text{old}} \leftarrow c$
    for $\mathcal{M}$ times **do**
      $\bar{c} \leftarrow$ output of Algorithm 1 on $\big(4G,\, 4\Delta_j,\, 4\iota(G)\big)$
      if $\|c_{\text{old}} + 2\bar{c} - \bar{t}\|_G^2 < \|c - \bar{t}\|_G^2$ **then** $c \leftarrow c_{\text{old}} + 2\bar{c}$
  if $\|c - \bar{t}\|_G^2 < \|c_{\text{best}} - \bar{t}\|_G^2$ **then** $c_{\text{best}} \leftarrow c$
return c_{best}

5 Exact CVP in ℓ_p Norms

We now prove Theorem 1.1 for an integer $p \geq 3$. The reduction keeps the coset-by-coset structure of Algorithm 2, but measures corrections in a diagonal metric computed from the current point. Throughout the section, $L = B\mathbb{Z}^N$ for a nonsingular $B \in \mathbb{Q}^{N\times N}$ and $t \in \mathbb{Q}^N$. Write

$$F_t(z) = \sum_{i=1}^{N} |z_i - t_i|^p, \qquad \phi(\alpha) = |\alpha|^p.$$

A closest vector in the ℓ_p norm is a minimizer of F_t on L.

5.1 Midpoint Inequalities

Let y be a minimizer and let $x \in y + 2L$. Since $(x+y)/2 \in L$, minimality of y gives a midpoint inequality. The following bounds are the quantitative form needed below; the proof is deferred to Section 7.4.

Lemma 5.1. *Let y be a minimizer of F_t on L, let $\Delta > 0$, and let $x \in y+2L$ satisfy $F_t(x) \leq F_t(y)+\Delta$. Write $\eta = y - x$, $\sigma = x - t$ and $\sigma' = y - t$. Then,*

$$\sum_i |\eta_i|^p \leq 2^p\Delta, \qquad \sum_i \max\{|\sigma_i|, |\sigma_i'|\}^{p-2}\eta_i^2 \leq 4\Delta, \qquad \sum_i \big(|\sigma_i'|^{p/2} - |\sigma_i|^{p/2}\big)^2 \leq 2\Delta.$$

5.2 A Weighted Norm Computed at the Current Point

The local curvature of F_t at the unknown minimizer y is unavailable to the algorithm. We instead use weights computed from the current point x. For $\Delta > 0$, put $\beta = (\Delta/N)^{1/p}$ and let ρ be the least number of the form 2^k, $k \in \mathbb{Z}$, such that $\rho^p \geq (\Delta/N)^{p-2}$. Thus $\beta^{p-2} \leq \rho < 2\beta^{p-2}$. Define

$$h_i = |x_i - t_i|^{p-2} + \rho, \qquad H_x = \operatorname{diag}(h_1, \dots, h_N), \qquad \beta_i = \max\{|y_i - t_i|, \beta\}. \tag{4}$$

The metric H_x is rational and computable from x and Δ; the numbers β_i are used only in the analysis.

Lemma 5.2. *Let y be a minimizer of F_t on L, let $\Delta > 0$, and let $x \in y+2L$ satisfy $F_t(x) \le F_t(y)+\Delta$. With (4),*

$$\|y-x\|_{H_x}^2 \le 12\Delta, \qquad \sum_i \frac{\beta_i^{p-2}}{h_i} \le 7N, \qquad h_i\beta_i^2 \ge \frac{\Delta}{N} \quad \text{for every } i.$$

The first bound places the unknown correction $y-x$ inside a known H_x-ball. The other two bounds compare the computable weights at x with the curvature around y. The proof is given in Section 7.4.

5.3 Progress in the Weighted Norm

We use the following coordinatewise Taylor bound; its proof is also deferred to Section 7.4.

Lemma 5.3. *For $a, w \in \mathbb{R}$ and $\beta_0 > 0$ with $\beta_0 \ge |a|$,*

$$0 \;\le\; \phi(a+w)-\phi(a)-\phi'(a)w \;\le\; \tfrac{p^2}{2}\beta_0^{p-2}e^{p|w|/\beta_0}w^2.$$

The next lemma is the analogue of Lemma 4.2. It is the only new geometric ingredient needed to reuse the Euclidean iteration.

Lemma 5.4. *Let y be a minimizer of F_t on L, let $\Delta > 0$, and let $x \in y+2L$ satisfy $x \ne y$ and $F_t(x) \le F_t(y)+\Delta$. Let H_x be defined by (4), let $\mathcal{D}$ be a distribution on $2L$ satisfying (3) with $K = 2L$, $H = H_x$, $r = 4\sqrt{\Delta}$ and some $\gamma \in (0,1]$, and let $V \sim \mathcal{D}$. Then*

$$\Pr\big[F_t(x+V) \le F_t(y)+\Delta/4\big] \;\ge\; \tfrac{1}{4}\cdot(188p)^{-N}\gamma.$$

Proof sketch. Let $\eta = y-x$, $K = 2L$, and $Q = \{w : h_i w_i^2 \le \Delta/(32Np^2) \text{ for all } i\}$. Keep the half of $K \cap Q$ on which the linear term $\sum_i \phi'(y_i - t_i)w_i$ is nonpositive. Symmetry preserves at least half the points, and Lemmas 5.2 and 5.3 show that every retained w gives $F_t(y+w) - F_t(y) < \Delta/4$ and that $\eta + w$ lies in $\mathcal{B}_{H_x}(4\sqrt{\Delta})$. After the map $w \mapsto H_x^{1/2}w$, Lemmas 2.1 and 2.2 show that this box contains, up to the factor $(188p)^N$, enough points of the sampled ball. Hence at least a $\frac{1}{4}(188p)^{-N}$ fraction of its relevant nonzero vectors are improving, and (3) gives the lemma. See Section 7.4 for the coordinate estimates and covering calculation. □

Reduction and completion of the proof. Use Algorithm 2, but at the beginning of stage j freeze $H = H_{x_{\text{old}}}$ from (4) and use this metric for every sample in that stage. If d is the least common multiple of the denominators of B and t, put

$$U = \textstyle\sum_i \Big(|t_i| + \sum_k |B_{ik}|\Big)^p, \qquad \Delta_j = U/4^j, \qquad J = \min\{j \ge 0 : \Delta_j < d^{-p}\}.$$

Then $F_t(Ba) \le U$ for every starting coset representative, while $F_t(L) \subset d^{-p}\mathbb{Z}$, so the last scale forces exact optimality. Also $J \ge 1$, since nonsingularity of B gives $U \ge Nd^{-p}$. In stage j the sampler input and the uniform success bound are

$$\big((2B)^T H(2B), 16\Delta_j, 4\rho/d^2\big), \qquad \theta = \frac{(1692p)^{-N}}{2048N(\lceil \log_2(4Ud^{p+2}N)\rceil + 4)^2},$$

and we take $\mathcal{M} = \theta^{-1}\lceil \log_2(4\cdot 2^N J/\varepsilon)\rceil = (1692p)^N \operatorname{poly}(p,\ell)$. By Lemma 5.4 and Theorem 3.1, repetition preserves the invariant $F_t(x_j) \le F_t(y) + \Delta_j$ through all stages of each coset containing a minimizer with the required probability. All metrics, coefficient vectors, and queries have $\operatorname{poly}(p,\ell)$ bits, giving $(Cp)^N \operatorname{poly}(p,\ell)$ time outside the oracle and polynomial working space. Hence Theorem 1.1 holds for every integer $p \ge 3$; full pseudocode, resource bounds, the all-closest-vector variant, and the bounded-error-oracle argument are in Section 7.4.

6 Proof of the Short-Vector Sampler Theorem

In this section, we prove Theorem 3.1 for Algorithm 1. Let $\mathcal{D}$ denote the distribution of the lattice vector represented by the algorithm's output coefficient vector, and let $V \sim \mathcal{D}$. Fix a nonzero vector $x = mu$, where $m \geq 1$ and u is primitive. The proof has three steps. We first establish counting bounds for lattice points in balls. We then show that a suitable random sparsification isolates $\pm u$ with sufficient probability. Finally, we combine these bounds with the random choices of Algorithm 1 to prove the pointwise probability bound (3) and the claimed resource bounds.

Throughout this section, $K = B_K\mathbb{Z}^N$, H and $G_K = B_K^T H B_K$ are as in Section 3. All norms and balls are taken with respect to H, and we drop H from the notation. We write $\lambda = \lambda_1(K, H)$ and $\xi = \xi_{K,H}$. For a lattice vector u, we write a_u for its coefficient vector.

Basis for the Sparsified Sublattices. For a prime q and $z \in \mathbb{F}_q^N$, recall

$$\Lambda_z = \{a \in \mathbb{Z}^N : \langle z, a\rangle \equiv 0 \pmod q\}, \qquad K_z = B_K\Lambda_z.$$

If $z = 0$, take $D_z = I_N$. If $z \neq 0$, fix a coordinate l_0 with $z_{l_0} \neq 0$ and let $\bar{z}_l \in \{0, \dots, q-1\}$ represent $z_l z_{l_0}^{-1}$ for $l \neq l_0$. Taking qe_{l_0} and $e_l - \bar{z}_l e_{l_0}$, $l \neq l_0$, as columns gives an integer matrix D_z with $|\det D_z| = q$ whose columns form a basis of Λ_z. Hence, K_z has Gram matrix $D_z^T G_K D_z$, whose bit length is polynomial in that of G_K and in $\log q$, and an oracle answer a' for K_z corresponds to the coefficient vector $D_z a'$ in K.

Relation to Stephens–Davidowitz. The sparsification step follows the primitive-vector sampler of Stephens-Davidowitz [Ste16, Section 4.1], which builds on Khot [Kho05] and Dadush and Kun [DK16]. The results of Stephens-Davidowitz [Ste16, Lemma 2.18, Theorem 4.1, Proposition 4.2] isolate a primitive direction under conditions involving the number of short primitive vectors and a lower bound on $\lambda_1(K, H)$. The lemmas below replace these conditions by the single inequality $q > |K \cap \mathcal{B}_H(2s)|$ for $s \geq \|u\|_H$, counting all lattice points and absorbing the loss through Lemma 6.2. We also guess the order of magnitude of q instead of estimating the number of short primitive vectors. Finally, unlike the discrete-Gaussian sampler of [Ste16, Theorem 4.6], our construction needs only the centered pointwise lower bound (3); no shifted sampler is used.

6.1 Counting Lattice Points in Balls

The first lemma bounds the number of lattice points in a ball in terms of the minimum distance, and compares balls whose radii differ by a factor of two.

Lemma 6.1. *For every $R > 0$, $|K \cap \mathcal{B}(R)| \leq (1 + 2R/\lambda)^N$. For every $s > 0$, $|K \cap \mathcal{B}(2s)| \leq 9^N |K \cap \mathcal{B}(s)|$.*

Proof. Distinct points of K are at distance at least λ. Hence the open balls of radius $\lambda/2$ centered at the points of $K \cap \mathcal{B}(R)$ are pairwise disjoint, and they are contained in $\mathcal{B}(R + \lambda/2)$. Comparing volumes gives $|K \cap \mathcal{B}(R)|\,(\lambda/2)^N \leq (R + \lambda/2)^N$, which is the first bound. For the second bound, Lemma 2.2(i) with $R = 2s$ and $\delta = s/2$ covers $\mathcal{B}(2s)$ by 9^N translates of $\mathcal{B}(s/2)$, and Lemma 2.1 with $Q = \mathcal{B}(s)$ gives the claim. □

The second lemma compares the number of lattice points in a ball of radius r with the number in a ball of radius $r/2^\kappa$. Every primitive vector u of length at most $r/2^\kappa$ has the signed multiples $\pm u, \dots, \pm 2^\kappa u$ in the larger ball, and Lemma 6.2 makes this count precise when u ranges over all primitive vectors.

Lemma 6.2. *Let $r > 0$, let $\kappa \geq 0$ be an integer, and let $s = r/2^\kappa$. Then,*

$$|K \cap \mathcal{B}(r)| - 1 \;\geq\; 2^\kappa \big(|K \cap \mathcal{B}(s)| - 1\big).$$

Proof. Since every nonzero vector is mu for a unique $m \geq 1$ and a unique primitive u,

$$|K \cap \mathcal{B}(R)| - 1 \;=\; 2 \sum_{m \geq 1} \xi(R/m) \qquad \text{for every } R > 0.$$

For $m \geq 1$, let $\mathcal{I}_m = \{k \in \mathbb{Z} : 2^\kappa(m-1) < k \leq 2^\kappa m\}$. The sets $\mathcal{I}_m$ partition the positive integers into blocks of 2^κ consecutive integers. Since ξ is nondecreasing, every $k \in \mathcal{I}_m$ satisfies $\xi(r/k) \geq \xi(r/(2^\kappa m)) = \xi(s/m)$. Therefore,

$$\sum_{k \geq 1} \xi(r/k) \;=\; \sum_{m \geq 1} \sum_{k \in \mathcal{I}_m} \xi(r/k) \;\geq\; 2^\kappa \sum_{m \geq 1} \xi(s/m),$$

and the lemma follows from the identity above. □

6.2 Sparsification

We use the sublattices K_z defined in Section 3. To isolate a primitive vector u, the other short primitive vectors must be independent of u modulo q, and no multiple of a removed primitive vector may be shorter than u. Lemmas 6.3 and 6.4 establish both properties under the single condition $q > |K \cap \mathcal{B}(2s)|$ for a radius $s \geq \|u\|$. They are the variant of [Ste16, Proposition 4.2] described in Remark 3.2.

Lemma 6.3. *Let $s > 0$ and let q be a prime with $q > |K \cap \mathcal{B}(2s)|$. If $u_1, u_2 \in K \cap \mathcal{B}(s)$ are primitive and $u_2 \neq \pm u_1$, then the coefficient vectors of u_1 and u_2 are linearly independent over $\mathbb{F}_q$.*

Proof. We show that a dependence modulo q forces at least q lattice points into the parallelogram spanned by u_1 and u_2, which lies in $\mathcal{B}(2s)$. The coefficient vectors a_{u_1} and a_{u_2} have greatest common divisor 1, so neither vanishes modulo q. Suppose that they are linearly dependent over $\mathbb{F}_q$. Then $a_{u_2} \equiv k_0 a_{u_1} \pmod q$ for some integer k_0, that is, $u_2 - k_0 u_1 \in qK$. Two primitive vectors on a common line through the origin are equal up to sign, so u_1 and u_2 are linearly independent over $\mathbb{R}$. Thus, $K_{u_1,u_2} = K \cap \operatorname{span}(u_1, u_2)$ is a lattice of rank two, in which u_1 is primitive. Since $qK \cap \operatorname{span}(u_1, u_2) = qK_{u_1,u_2}$, the vector u_2 lies in the sublattice $K' = \mathbb{Z}u_1 + qK_{u_1,u_2}$.

We compute the index of K'. The quotient $K_{u_1,u_2}/qK_{u_1,u_2} \cong \mathbb{F}_q^2$ has q^2 elements, and the image of K' in it is the subgroup generated by the image of u_1. This image is nonzero because u_1 is primitive in K_{u_1,u_2}, so the subgroup has q elements and $[K_{u_1,u_2} : K'] = q$. Since $\mathbb{Z}u_1 + \mathbb{Z}u_2 \subseteq K'$, it follows that $[K_{u_1,u_2} : \mathbb{Z}u_1 + \mathbb{Z}u_2] \geq q$.

The half-open parallelogram $\{\alpha_1 u_1 + \alpha_2 u_2 : \alpha_1, \alpha_2 \in [0,1)\}$ is a fundamental domain of $\mathbb{Z}u_1 + \mathbb{Z}u_2$, so it contains exactly $[K_{u_1,u_2} : \mathbb{Z}u_1 + \mathbb{Z}u_2] \geq q$ points of K_{u_1,u_2}. Each of them has norm less than $\|u_1\| + \|u_2\| \leq 2s$. Hence $|K \cap \mathcal{B}(2s)| \geq q$, contradicting the choice of q. □

Lemma 6.4. *Let $s > 0$, let q be a prime with $q > |K \cap \mathcal{B}(2s)|$, and let $u \in K$ be primitive with $\|u\| \leq s$. If z is uniform in $\mathbb{F}_q^N$, then with probability at least $1/(2q)$ the shortest nonzero vectors of K_z are exactly u and $-u$.*

Proof. We first bound the probability that u is, up to sign, the only primitive vector of $\mathcal{B}(s)$ in K_z. We then show that in this case no other nonzero vector of K_z is as short as u.

The vector 0 and the two vectors in each of the $\xi(s)$ primitive pairs in $\mathcal{B}(s)$ are distinct points of $K \cap \mathcal{B}(2s)$. Hence, $2\xi(s) < q$. Let $\mathcal{E}$ be the event that $u \in K_z$ and that $u' \notin K_z$ for every primitive

$u' \in \mathcal{B}(s)$ with $u' \neq \pm u$. Choose one representative u' of each of the $\xi(s) - 1$ other primitive pairs in $\mathcal{B}(s)$. By Lemma 6.3, the coefficient vectors of u and u' are linearly independent over $\mathbb{F}_q$, so a union bound gives

$$\Pr[\mathcal{E}] \;\geq\; \Pr[u \in K_z] - \sum_{u'} \Pr[u \in K_z \text{ and } u' \in K_z] \;=\; \frac{1}{q} - \frac{\xi(s)-1}{q^2} \;\geq\; \frac{1}{q} - \frac{1}{2q} \;=\; \frac{1}{2q}.$$

Now assume $\mathcal{E}$, and let $w \in K_z$ be nonzero with $\|w\| \leq \|u\|$. Write $w = kw'$ with $k \geq 1$ and w' primitive. Since $\|w'\| \leq \|w\| \leq \|u\| \leq s$, we have $w' \in \mathcal{B}(s)$. If $w' \in K_z$, then $w' = \pm u$ by the definition of $\mathcal{E}$, and $\|w\| \leq \|u\|$ forces $w = \pm u$. If $w' \notin K_z$, then $\langle z, a_{w'} \rangle \not\equiv 0$ while $k\langle z, a_{w'} \rangle \equiv 0 \pmod q$, so q divides k and $\|w\| \geq q\lambda$. The multiples of a shortest nonzero vector show that $|K \cap \mathcal{B}(2s)| \geq 1 + 2\lfloor 2s/\lambda \rfloor > 2s/\lambda$, so $q\lambda > 2s > \|u\|$, which is impossible. Hence on $\mathcal{E}$ the shortest nonzero vectors of K_z are exactly u and $-u$. □

6.3 Analysis of Theorem 3.1

We begin by restating Theorem 3.1 for convenience.

Theorem 3.1 (restated). *Let $K = B_K\mathbb{Z}^N$ and let $H \succ 0$ be a metric such that $G_K := B_K^T H B_K \in \mathbb{Q}^{N\times N}$. Let $r^2 > 0$ and $\lambda_0^2 > 0$ be rational numbers satisfying $\lambda_0 \leq \lambda_1(K, H)$. There is a randomized algorithm with access to an exact SVP oracle that, given G_K, r^2, and λ_0^2, outputs a coefficient vector representing a lattice vector in K. Let $\mathcal{D}$ be the induced distribution of this lattice vector, and let $V \sim \mathcal{D}$. Then the following properties hold.*

(a) $\operatorname{supp}(\mathcal{D}) \subseteq K \cap \mathcal{B}_H(r)$*; equivalently, $V \in K \cap \mathcal{B}_H(r)$ with probability* 1*.*

(b) $\mathcal{D}$ satisfies (3) with

$$\gamma = \frac{9^{-N}}{512\, N \left(\log_2 \max\{1, r/\lambda_0\} + 4\right)^2},$$

conditionally on any history of earlier computations and for every rule by which the oracle chooses among shortest vectors.

(c) The algorithm makes at most one oracle call, on a full-rank sublattice of K in the metric H given by a Gram matrix whose bit length is polynomial in that of the input. Its other bit operations and its space are polynomial in the bit lengths of the input and of the oracle answer.

We first describe the random sampling procedures used by Algorithm 1, and then prove the three guarantees of Theorem 3.1.

Random choices. Lines 4 and 5 of Algorithm 1 use the following procedures.

(1) To sample from a set $\{a_0, \ldots, a_0 + n - 1\}$ of n consecutive integers using a bounded number of random bits, the procedure draws $\lceil \log_2 n \rceil$ random bits and reads them as an integer ν. It returns $a_0 + \nu$ if $\nu < n$, and returns FAIL otherwise. Conditioned on success, the returned element is uniform over the set. Unconditionally, each element is returned with probability $2^{-\lceil \log_2 n \rceil} \geq 1/(2n)$, which equals $1/n$ when n is a power of 2.

(2) A prime in $[2^i, 2^{i+1})$, $i \geq 1$, is obtained by drawing at most $3(i+1)$ uniform integers from this interval and returning the first prime among them, where primality is tested deterministically [AKS04]. By [Sho03, Theorem 5.11], $\pi(2m) - \pi(m) > m/(3\log(2m))$ for $m \geq 2$. With $m = 2^i$, a uniform integer from $[2^i, 2^{i+1})$ is prime with probability greater than $1/(3(i+1))$, so a prime is returned with probability greater than $1 - e^{-1} > 1/2$.

(3) A uniform vector $z \in \mathbb{F}_q^N$, for a prime $q \in [2^i, 2^{i+1})$, is obtained coordinate by coordinate. For each coordinate the procedure draws up to $\lceil \log_2 N \rceil + 2$ integers of $i+1$ bits and keeps the first that is less than q. Each draw is kept with probability at least $1/2$, so a given coordinate fails with probability at most $1/(4N)$, and all coordinates are obtained with probability at least $3/4$. Conditioned on this event, z is uniform on $\mathbb{F}_q^N$.

Support (proof of Theorem 3.1(a)). In any execution, the output is 0 or a coefficient vector $\pm ma$ representing the realized lattice vector $v = B_K(\pm ma)$. The output test ensures

$$\|v\|_H^2 = m^2 a^T G_K a \le r^2,$$

so $v \in K \cap \mathcal{B}_H(r)$. Thus, $\mathrm{supp}(\mathcal{D}) \subseteq K \cap \mathcal{B}_H(r)$, proving part (a) of Theorem 3.1.

Pointwise lower bound (proof of Theorem 3.1(b)). Fix a nonzero $x \in K \cap \mathcal{B}(r)$, so that $r \ge \lambda \ge \lambda_0$. We identify random choices under which Algorithm 1 returns x, bound the product of their probabilities, and compare it with $1/|K \cap \mathcal{B}(r)|$.

Write $x = mu$ with $m \ge 1$ and u primitive, and let $\kappa = \lfloor \log_2 m \rfloor$, $s = r/2^\kappa$ and $\Xi = |K \cap \mathcal{B}(2s)|$. Since $2^\kappa \le m \le r/\|u\| \le r/\lambda_0$, we have $4^\kappa \lambda_0^2 \le r^2$, that is, $\kappa \le J_{\mathrm{s}}$, as defined in Line 2 of Algorithm 1. Since $m \ge 2^\kappa$, we have $\|u\| \le r/m \le s$, and in particular $s \ge \lambda$. The next claim shows that some exponent i in the range of the algorithm yields primes for which Lemma 6.4 applies.

Claim 6.5. *Let i^* be the least integer $i \ge 1$ with $2^i > \Xi$. Then $i^* \le i_{\max}$, and every prime $q \in [2^{i^*}, 2^{i^*+1})$ satisfies $\Xi < q \le 4\Xi$.*

Proof. By Lemma 6.1, $\Xi \le (1 + 4s/\lambda)^N \le (1 + 2^{J_{\mathrm{s}}+3})^N \le 2^{N(J_{\mathrm{s}}+4)} < 2^{i_{\max}}$, where we used $s \le r < 2^{J_{\mathrm{s}}+1}\lambda_0 \le 2^{J_{\mathrm{s}}+1}\lambda$. Hence, $i^* \le i_{\max}$. Since $K \cap \mathcal{B}(2s)$ contains 0 and $\pm u$, we have $\Xi \ge 3$, so $i^* \ge 2$ and $2^{i^*-1} \le \Xi$. Therefore, $\Xi < 2^{i^*} \le q < 2^{i^*+1} \le 4\Xi$. □

The algorithm returns x if the following events occur. Each probability bound holds conditionally on the preceding events and on any earlier history.

(E1) Procedure (1) returns this κ, $i = i^*$ and this m: probability at least $\frac{1}{2(J_{\mathrm{s}}+1)} \cdot \frac{1}{2i_{\max}} \cdot 2^{-\kappa}$, since the range of m has 2^κ elements.

(E2) Procedures (2) and (3) succeed: probability at least $\frac{1}{2} \cdot \frac{3}{4}$. Then z is uniform on $\mathbb{F}_q^N$.

(E3) The shortest nonzero vectors of K_z are exactly $\pm u$: probability at least $\frac{1}{2q} \ge \frac{1}{8\Xi}$, by Lemma 6.4 and Claim 6.5. On this event the oracle returns a coefficient vector a' with $D_z a' = \pm a_u$ for every rule of choosing among shortest vectors, and the output test passes because $\|mu\| \le r$.

(E4) The random sign of the output is the one that gives a_x: probability $\frac{1}{2}$.

Multiplying these bounds gives

$$\mathcal{D}(x) \;\ge\; \frac{3 \cdot 2^{-\kappa}}{512\,(J_{\mathrm{s}}+1)\,i_{\max}\,\Xi}.$$

It remains to compare $2^\kappa \Xi$ with $|K \cap \mathcal{B}(r)|$. By Lemma 6.1, the inequality $|K \cap \mathcal{B}(s)| \ge 3$, and Lemma 6.2,

$$2^\kappa \Xi \;\le\; 9^N 2^\kappa |K \cap \mathcal{B}(s)| \;\le\; \tfrac{3}{2}\, 9^N 2^\kappa \big(|K \cap \mathcal{B}(s)| - 1\big) \;\le\; \tfrac{3}{2}\, 9^N |K \cap \mathcal{B}(r)|.$$

Finally, $(J_{\mathrm{s}}+1) i_{\max} \le 2N(J_{\mathrm{s}}+4)^2$ and $J_{\mathrm{s}} \le \log_2(r/\lambda_0) = \log_2 \max\{1, r/\lambda_0\}$. Combining the three displays,

$$\mathcal{D}(x) \;\ge\; \frac{9^{-N}}{256\,(J_{\mathrm{s}}+1)\,i_{\max}\,|K \cap \mathcal{B}(r)|} \;\ge\; \frac{9^{-N}}{512\,N\,\big(\log_2 \max\{1, r/\lambda_0\} + 4\big)^2\,|K \cap \mathcal{B}(r)|},$$

which is (3) with the value of γ in Theorem 3.1(b). This proves part (b).

Resources (proof of Theorem 3.1(c)). If $r^2 < \lambda_0^2$, the algorithm returns 0 before making an oracle call, and the resource bounds are immediate. Hence assume $r^2 \geq \lambda_0^2$. The algorithm makes at most one oracle call, on the full-rank sublattice K_z of K, which is K itself for $z = 0$. Since $4^{J_s} \leq r^2/\lambda_0^2$, the numbers J_s and $i_{\max}$ are polynomial in the input length, and so are the bit lengths of m, q, z and D_z, the number of random bits, and the number of primality tests. In particular, $q < 2^{i_{\max}+1}$ has at most $i_{\max} + 1 = N(J_s + 4) + 2$ bits, by the definition of $i_{\max}$. Since the entries of D_z have $O(i_{\max})$ bits and G_K is a rational input matrix, standard rational matrix multiplication shows that every entry of the query $D_z^T G_K D_z$ has bit length polynomial in the input length. All remaining computations involve rationals whose bit length is polynomial in the bit lengths of the input and of the oracle answer. This proves part (c) and completes the proof of Theorem 3.1.

7 Deferred Proofs and Details for the CVP Reductions

This section contains the proofs and detailed bookkeeping deferred from Sections 2, 4 and 5. After the Euclidean proofs, we record polynomial-space consequences of the CVP reductions before turning to the deferred ℓ_p proofs.

7.1 Counting and Covering Lemmas

We prove Lemmas 2.1 and 2.2, restating them for convenience.

Lemma 2.1 (restated). *Let $L \subset \mathbb{R}^N$ be a lattice and $Q \subset \mathbb{R}^N$ a convex, compact, symmetric set. If $X \subseteq \bigcup_{k=1}^{T} (c_k + \frac{1}{2}Q)$ for some $c_1, \ldots, c_T \in \mathbb{R}^N$, then $|L \cap X| \leq T \cdot |L \cap Q|$.*

Proof of Lemma 2.1. Fix k such that $L \cap (c_k + \frac{1}{2}Q)$ contains a point x_k. For every x in this set, $x - x_k \in \frac{1}{2}Q - \frac{1}{2}Q$, and $\frac{1}{2}Q - \frac{1}{2}Q = Q$ because Q is convex and symmetric. As $x - x_k \in L$, we obtain $L \cap (c_k + \frac{1}{2}Q) \subseteq x_k + (L \cap Q)$. Summing over k proves the lemma. □

Lemma 2.2 (restated). *The following covering bounds hold.*

(i) For every positive definite metric H and $R, \delta > 0$, the ball $\mathcal{B}_H(R)$ is contained in a union of at most $(1 + 2R/\delta)^N$ translates of $\mathcal{B}_H(\delta)$.

(ii) For the Euclidean norm, $R > 0$ and $0 < \alpha \leq R/\sqrt{N}$, the ball $\mathcal{B}(R)$ is contained in a union of at most $\omega_N \cdot (2R/\alpha)^N$ translates of the cube $[0, \alpha]^N$.

Proof of Lemma 2.2. (i) Let $Y \subseteq \mathcal{B}_H(R)$ be maximal subject to $\|x - x'\|_H > \delta$ for distinct $x, x' \in Y$. The balls $x + \mathcal{B}_H(\delta/2)$, $x \in Y$, are disjoint and contained in $\mathcal{B}_H(R + \delta/2)$, so $|Y| \leq (1 + 2R/\delta)^N$. By maximality, $\mathcal{B}_H(R) \subseteq \bigcup_{x \in Y} (x + \mathcal{B}_H(\delta))$.

(ii) The cubes $\alpha(k + [0,1)^N)$, $k \in \mathbb{Z}^N$, are disjoint and have diameter $\alpha\sqrt{N} \leq R$. Each cube that meets $\mathcal{B}(R)$ is contained in $\mathcal{B}(2R)$, so at most $\omega_N \cdot (2R/\alpha)^N$ cubes meet $\mathcal{B}(R)$, and their closures cover $\mathcal{B}(R)$. □

7.2 Deferred Euclidean Proofs

We first prove Lemmas 4.1 and 4.2. We then give the detailed correctness and resource analysis of Algorithm 2 and complete the proof of Theorem 1.1 for $p = 2$.

Lemma 4.1 (restated). *Let y be a closest vector of L to t. For every $w \in 2L$,*

$$\tfrac{1}{2}\|w\|^2 \;\le\; \|y + w - t\|^2 - \|y - t\|^2 \;\le\; \tfrac{3}{2}\|w\|^2.$$

Consequently, for every $\Delta \ge 0$ and every $x \in y + 2L$ with $\|x - t\|^2 \le \|y - t\|^2 + \Delta$,
(i) $\|x - y\|^2 \le 2\Delta$;
(ii) if $\Delta < \lambda_1(2L)^2/2$, then $x = y$.
In particular, each coset of $2L$ in L contains at most one closest vector of L to t.

Proof of Lemma 4.1. Since $w \in 2L$, both $y + w/2$ and $y - w/2$ lie in L. As y is a closest vector, neither of them is closer to t than y, and expanding the squares gives

$$0 \;\le\; \|y \pm w/2 - t\|^2 - \|y - t\|^2 \;=\; \pm\langle y - t, w\rangle + \tfrac{1}{4}\|w\|^2.$$

Hence, $|\langle y - t, w\rangle| \le \|w\|^2/4$. Inserting this bound into the identity $\|y + w - t\|^2 - \|y - t\|^2 = \|w\|^2 + 2\langle y - t, w\rangle$ gives both inequalities.

For (i), apply the lower bound to $w = x - y \in 2L$. For (ii), suppose that $x \ne y$. Then $x - y$ is a nonzero vector of $2L$, so $\|x - y\|^2 \ge \lambda_1(2L)^2 > 2\Delta$, which contradicts (i). The last statement is (ii) with $\Delta = 0$. □

Lemma 4.2 (restated). *Let y be a closest vector of L to t, let $\Delta > 0$, and let $x \in y + 2L$ satisfy $x \ne y$ and $\|x - t\|^2 \le \|y - t\|^2 + \Delta$. Let $\mathcal{D}$ be a distribution on $2L$ satisfying (3) with $K = 2L$, $H = I$, $r = 2\sqrt{\Delta}$ and some $\gamma \in (0, 1]$, and let $V \sim \mathcal{D}$. Then*

$$\Pr\big[\|x + V - t\|^2 \le \|y - t\|^2 + \Delta/4\big] \;\ge\; \tfrac{1}{2} \cdot 21^{-N}\gamma.$$

Proof of Lemma 4.2. Let $\eta = y - x$, let $S = 2L \cap \mathcal{B}(\sqrt{\Delta/6})$, and let $E = \eta + S$. We show that
(a) $\|x + z - t\|^2 \le \|y - t\|^2 + \Delta/4$ for every $z \in E$,
(b) $E \subseteq 2L \cap \mathcal{B}(2\sqrt{\Delta})$, and
(c) $|E \setminus \{0\}| \ge \tfrac{1}{2} \cdot 21^{-N}\,|2L \cap \mathcal{B}(2\sqrt{\Delta})|$.
Then (3) gives $\Pr[V \in E] = \mathcal{D}(E) \ge \gamma\,|E \setminus \{0\}|/|2L \cap \mathcal{B}(2\sqrt{\Delta})| \ge \tfrac{1}{2} \cdot 21^{-N}\gamma$, and (a) proves the lemma.

- For (a), let $z = \eta + w$ with $w \in S$. Then $x + z = y + w$, and the upper bound of Lemma 4.1 gives $\|y + w - t\|^2 - \|y - t\|^2 \le \tfrac{3}{2}\|w\|^2 \le \Delta/4$.
- For (b), $\eta \in 2L$ because $x \in y + 2L$, and $\|\eta\| \le \sqrt{2\Delta}$ by Lemma 4.1(i). Hence, every $w \in S$ satisfies $\|\eta + w\| \le (\sqrt{2} + 1/\sqrt{6})\sqrt{\Delta} < 2\sqrt{\Delta}$.
- For (c), Lemma 2.2(i) with $R = 2\sqrt{\Delta}$ and $\delta = \tfrac{1}{2}\sqrt{\Delta/6}$ covers $\mathcal{B}(2\sqrt{\Delta})$ by at most $(1+8\sqrt{6})^N \le 21^N$ translates of $\tfrac{1}{2}\mathcal{B}(\sqrt{\Delta/6})$. By Lemma 2.1 with $Q = \mathcal{B}(\sqrt{\Delta/6})$,

$$|2L \cap \mathcal{B}(2\sqrt{\Delta})| \;\le\; 21^N|S|.$$

Finally $|E| = |S|$, and E contains $\eta \ne 0$ because $0 \in S$. Hence $|E\setminus\{0\}| \ge \max\{1, |S|-1\} \ge |S|/2$. □

Detailed probability and resource analysis. We provide the detailed correctness and resource analysis underlying the proof sketch in Section 4.2.

Lemma 7.1. *Assume that the oracle is exact. Let y be a closest vector of L to t, and let $a \in \{0,1\}^N$ satisfy $y \in Ba + 2L$. Then, with probability at least $1 - \varepsilon/(4 \cdot 2^N)$, the iteration of Algorithm 2 for a ends with $Bc = y$.*

Proof. For $0 \le j \le J$, let x_j be the current point Bc at the start of stage j, where x_J denotes the point at the end of the last stage. All these points lie in $y + 2L$, since the algorithm adds only vectors of $2L$ to Ba. We have $\|x_0 - t\|^2 \le U = \Delta_0 \le \|y - t\|^2 + \Delta_0$, so the invariant holds initially.

Fix $j < J$ and condition on the computation up to the start of stage j, assuming that $\|x_j - t\|^2 \le \|y - t\|^2 + \Delta_j$. We show that $\|x_{j+1} - t\|^2 \le \|y - t\|^2 + \Delta_{j+1}$ with probability at least $1 - \varepsilon/(4 \cdot 2^N J)$. If $x_j = y$, then $x_{j+1} = y$, because the current point changes only when its distance to t decreases strictly. Otherwise, Lemma 4.2 with $x = x_j$ and $\Delta = \Delta_j$, together with Theorem 3.1(b), shows that each sample $\bar{c}$ of stage j satisfies $\|x_j + 2B\bar{c} - t\|^2 \le \|y - t\|^2 + \Delta_{j+1}$ with probability at least θ, conditionally on the earlier samples. Hence, all $\mathcal{M}$ samples fail with probability at most

$$(1-\theta)^{\mathcal{M}} \;\le\; e^{-\theta\mathcal{M}} \;\le\; \frac{\varepsilon}{4 \cdot 2^N J},$$

where we used $\log_2 z \ge \log z$ for $z \ge 1$. If some sample succeeds, then x_{j+1} is at least as close to t as $x_j + 2B\bar{c}$ for that sample, which proves the claim.

By a union bound over the J stages, $\|x_J - t\|^2 \le \|y - t\|^2 + \Delta_J$ with probability at least $1 - \varepsilon/(4 \cdot 2^N)$. Since $\Delta_J < 2\iota(G) \le \lambda_1(2L)^2/2$, Lemma 4.1(ii) then gives $x_J = y$. □

Lemma 7.2 bounds the resources of Algorithm 2. It also applies to an oracle that may err, provided that each answer is an integer vector of bounded bit length, a condition enforced by Lemma A.2.

Lemma 7.2. *Algorithm 2 makes at most $2^N J\mathcal{M}$ oracle calls, each on a full-rank sublattice of $2L$ given by a $\mathrm{poly}(\ell)$-bit Gram matrix. If every oracle answer is an integer vector of $\mathrm{poly}(\ell)$ bits, in particular if the oracle is exact, then the algorithm uses $2^{O(N)}\,\mathrm{poly}(\ell)$ other bit operations and $\mathrm{poly}(\ell)$ space outside the oracle.*

Proof. The algorithm makes $2^N J\mathcal{M}$ calls of the sampler, and each of them makes at most one oracle call by Theorem 3.1(c). The input $(4G, 4\Delta_j, 4\iota(G))$ of every call of the sampler has bit length polynomial in ℓ, so Theorem 3.1(c) bounds the bit lengths of the queries and the cost of each call by $\mathrm{poly}(\ell)$. Since $\mathcal{M} = 2^{O(N)}\,\mathrm{poly}(\ell)$ and $J = \mathrm{poly}(\ell)$, the number of bit operations is $2^{O(N)}\,\mathrm{poly}(\ell)$.

For the space bound, the current point Bc satisfies $\|Bc - t\|^2 \le U$ at all times, because it starts at a point Ba with $\|Ba - t\|^2 \le U$ and its distance to t never increases. By (1), $\|c - \bar{t}\|^2 \le U/\iota(G)$, so c has $\mathrm{poly}(\ell)$ bits. The same holds for c_{old} and c_{best}. Every sample lies in $\mathcal{B}(2\sqrt{U})$ by Theorem 3.1(a), and (1) bounds its coefficients in the same way. Besides the workspace of one call of the sampler, which is reused, the algorithm stores three coefficient vectors and the loop counters. □

Proof of Theorem 1.1 for $p = 2$. Suppose first that the oracle is exact. Let y be a closest vector of L to t, and let $a \in \{0,1\}^N$ satisfy $y \in Ba + 2L$. By Lemma 7.1, with probability at least $1 - \varepsilon$ the iteration for a ends at y. In this event, the returned coefficient vector c_{best} represents Bc_{best}, which is at least as close to t as y and hence is a closest vector. Lemma 7.2 gives the bounds (i)–(iii) of Theorem 1.1.

To output every closest vector, we run Algorithm 2 twice with independent random bits and modify the final comparison of each coset. Let $x_a^{(1)}$ denote the point at the end of the iteration for

a in the first run, and put $\mu = \min_a \|x_a^{(1)} - t\|^2$. In the second run, define $c_a^{(2)}$ and $x_a^{(2)} = Bc_a^{(2)}$ analogously, and output $c_a^{(2)}$ whenever $\|x_a^{(2)} - t\|^2 = \mu$. By Lemma 4.1, every coset contains at most one closest vector. By Lemma 7.1 and a union bound over the at most $2 \cdot 2^N$ iterations for cosets that contain a closest vector, with probability at least $1 - \varepsilon/2$ every such iteration ends at the closest vector in both runs. In this event $\mu = \operatorname{dist}(t, L)^2$, and the second run outputs the coefficient vectors representing exactly the closest vectors, each of them once.

Finally, suppose that the oracle is replaced by an SVP algorithm that succeeds with probability at least $2/3$. We run the algorithm, in either variant, with $\varepsilon/2$ in place of ε, so that $\mathcal{M}$ is computed for $\varepsilon/2$. Before its first oracle call, the algorithm computes $T_{\mathrm{o}} = 2^{N+1} J\mathcal{M}$, which by Lemma 7.2 bounds the number of oracle calls of two runs, and a bound $\ell_0 = \operatorname{poly}(\ell)$ on the bit lengths of all queries and of their shortest vectors (see the remark after Lemma A.2). Lemma A.2 with $\varepsilon_1 = \varepsilon_2 = \varepsilon/2$ shows that the success probability is at least $1 - \varepsilon$. Each oracle call is answered by $\lceil \log_3(2T_{\mathrm{o}}/\varepsilon) \rceil = \operatorname{poly}(\ell)$ runs of the SVP algorithm, and every answer that passes the check of Lemma A.2 has at most ℓ_0 bits, so Lemma 7.2 applies. □

7.3 Polynomial-Space Consequences

We first present the parameterized transfer underlying the Euclidean bounds in Corollary 1.2.

Corollary 7.3. *Suppose that an algorithm solves exact SVP with probability at least $2/3$ on every rank-N lattice given by an ℓ_0-bit integer basis, in time $T_1(N, \ell_0)$ and space $T_2(N, \ell_0)$ on every execution, where T_1 and T_2 are nondecreasing in ℓ_0. Then, there is a polynomial P such that, for every rational $\varepsilon \in (0, 1/2)$, exact CVP on rank-N inputs of bit length ℓ, where ℓ includes the bit length of ε, can be solved with probability at least $1 - \varepsilon$ in time $2^{O(N)} P(\ell)\, T_1(N, P(\ell))$ and space $P(\ell) + T_2(N, P(\ell))$.*

Proof. Apply Theorem 1.1 with $p = 2$ and the SVP algorithm in place of the oracle, as in the detailed proof in Section 7.2. All queries have rank N and $\operatorname{poly}(\ell)$ bits, and there are $2^{O(N)} \operatorname{poly}(\ell)$ of them. Lemma A.1 converts each query into an integer basis of $\operatorname{poly}(\ell)$ bits whose shortest coefficient vectors are shortest for the query. Using Lemma A.2 as in Section 7.2, we run the reduction with $\varepsilon/2$ in place of ε and amplify the bounded-error SVP algorithm using $\operatorname{poly}(\ell)$ repetitions per oracle call, so that the oracle simulation fails with probability at most $\varepsilon/2$. The queries are answered one at a time, so the space used by the SVP algorithm is reused. □

The reduction of Goldreich, Micciancio, Safra and Seifert [GMSS99] goes in the opposite direction. Together with Corollary 7.3 it shows that the two problems have the same time complexity for randomized polynomial-space algorithms, up to a factor $2^{O(N)} \operatorname{poly}(\ell)$.

Corollary 7.4. *Let f be a function. If exact Euclidean SVP on lattices of rank N can be solved with probability at least $2/3$ in time $f(N) \operatorname{poly}(\ell)$ and space $\operatorname{poly}(\ell)$, then so can exact Euclidean CVP in time $2^{O(N)} f(N) \operatorname{poly}(\ell)$. Conversely, if exact Euclidean CVP on lattices of rank N can be solved with probability at least $2/3$ in time $f(N) \operatorname{poly}(\ell)$ and space $\operatorname{poly}(\ell)$, then so can exact Euclidean SVP.*

Proof. The first statement is Corollary 7.3 with $T_1(N, \ell_0) = f(N) \operatorname{poly}(\ell_0)$, $T_2(N, \ell_0) = \operatorname{poly}(\ell_0)$ and $\varepsilon = 1/3$. For the second, let $L = B\mathbb{Z}^N$ with columns $b_1, \ldots, b_N$. For $i = 1, \ldots, N$, let $L^{(i)} \subseteq L$ be the sublattice of vectors whose ith coefficient is even, and let y_i be a closest vector of $L^{(i)}$ to b_i. The vector $y_i - b_i \in L$ has an odd ith coefficient, so it is nonzero. Let Bz be a shortest nonzero vector of L. Some coefficient z_i is odd, since otherwise $B(z/2)$ would be a shorter nonzero vector of L. Then, $Bz + b_i \in L^{(i)}$, so $\|y_i - b_i\| \le \|Bz\|$, and the shortest of $y_1 - b_1, \ldots, y_N - b_N$ is a shortest

nonzero vector of L. The N instances of CVP have rank N and $\mathrm{poly}(\ell)$ bits, and they are solved one after another. Running the CVP algorithm independently $\lceil \log_3(3N) \rceil$ times on each instance and keeping the closest valid lattice vector makes all N answers correct with probability at least $2/3$. □

We next justify the polynomial-space instantiation in Corollary 1.3. For every fixed integer $p \geq 3$, the reduction of Theorem 1.1 can be instantiated with the polynomial-space SVP algorithm of Feng, Gao and Liu [FGL26, Theorem 4.1], using Lemmas A.1 and A.2. Lemma A.1 converts each rank-N rational Gram-matrix query into an equivalent rank-N integer-basis instance of polynomial bit length, and Lemma A.2 simulates the exact SVP oracle using the bounded-error SVP algorithm with only polynomial overhead. Since $(Cp)^N = 2^{O(N)}$, which is $N^{o(N)}$ for fixed p, the $N^{N/(4e)+o(N)}\,\mathrm{poly}(\ell)$ running time of [FGL26, Theorem 4.1] absorbs the overhead of the reduction. Thus, exact ℓ_p-CVP on full-rank lattices with arbitrary rational targets can be solved in randomized time $N^{N/(4e)+o(N)}\,\mathrm{poly}(\ell)$ and polynomial space. The oracle calls are made sequentially, so the workspace used to answer them can be reused.

7.4 Deferred Proofs for the ℓ_p Reduction

We first prove Lemmas 5.1–5.4. We then give the full pseudocode and parameter choices for Algorithm 3, analyze its correctness and resource bounds, and complete the proof of Theorem 1.1 for $p \geq 3$.

7.4.1 Geometric and Progress Lemmas

Lemma 5.1 (restated). *Let y be a minimizer of F_t on L, let $\Delta > 0$, and let $x \in y + 2L$ satisfy $F_t(x) \leq F_t(y) + \Delta$. Write $\eta = y - x$, $\sigma = x - t$ and $\sigma' = y - t$. Then,*

$$\sum_i |\eta_i|^p \leq 2^p\Delta, \qquad \sum_i \max\{|\sigma_i|, |\sigma_i'|\}^{p-2}\eta_i^2 \leq 4\Delta, \qquad \sum_i \big(|\sigma_i'|^{p/2} - |\sigma_i|^{p/2}\big)^2 \leq 2\Delta.$$

Proof of Lemma 5.1. For $a, b \in \mathbb{R}$ let $g(a,b) = \frac{1}{2}(|a|^p + |b|^p) - |\frac{a+b}{2}|^p$. Since $(x+y)/2 \in L$ and y is a minimizer,

$$\sum_i g(\sigma_i, \sigma_i') \;=\; \tfrac{1}{2}\big(F_t(x) + F_t(y)\big) - F_t\big(\tfrac{x+y}{2}\big) \;\leq\; \tfrac{1}{2}\big(F_t(x) - F_t(y)\big) \;\leq\; \tfrac{\Delta}{2}. \tag{5}$$

We bound g from below in two ways. Put $\bar{a} = (a+b)/2$, $\omega = |a-b|/2$ and $M = |\bar{a}| + \omega = \max\{|a|, |b|\}$, and assume $\omega > 0$, since both bounds are trivial for $\omega = 0$. Taylor's formula with integral remainder, applied to ϕ at $\bar{a}$ in the directions $\pm\omega$, gives

$$g(a,b) = \frac{p(p-1)}{2}\int_0^\omega (\omega - \alpha)\big(|\bar{a} + \alpha|^{p-2} + |\bar{a} - \alpha|^{p-2}\big)\, d\alpha.$$

One of $|\bar{a} + \alpha|$ and $|\bar{a} - \alpha|$ equals $|\bar{a}| + \alpha \geq (\alpha/\omega)M$. Hence,

$$g(a,b) \;\geq\; \frac{p(p-1)}{2}\,\frac{M^{p-2}}{\omega^{p-2}}\int_0^\omega (\omega - \alpha)\alpha^{p-2}\, d\alpha \;=\; \frac{M^{p-2}\omega^2}{2} \;=\; \frac{1}{8}M^{p-2}(a-b)^2.$$

For the second bound, the triangle inequality and convexity of $u \mapsto u^{p/2}$ on $[0, \infty)$ give

$$\Big|\frac{a+b}{2}\Big|^p \leq \Big(\frac{|a| + |b|}{2}\Big)^p \leq \Big(\frac{|a|^{p/2} + |b|^{p/2}}{2}\Big)^2,$$

so $g(a,b) \ge \frac{1}{4}(|a|^{p/2}-|b|^{p/2})^2$. Summing each bound over i with $(a,b)=(\sigma_i,\sigma_i')$ and using (5) proves the second and third inequalities. The first follows from the second, because $|\eta_i| \le 2\max\{|\sigma_i|,|\sigma_i'|\}$ gives $|\eta_i|^p \le 2^{p-2}\max\{|\sigma_i|,|\sigma_i'|\}^{p-2}\eta_i^2$. □

Lemma 5.2 (restated). *Let y be a minimizer of F_t on L, let $\Delta > 0$, and let $x \in y + 2L$ satisfy $F_t(x) \le F_t(y) + \Delta$. With (4),*

$$\|y-x\|_{H_x}^2 \le 12\Delta, \qquad \sum_i \frac{\beta_i^{p-2}}{h_i} \le 7N, \qquad h_i\beta_i^2 \ge \frac{\Delta}{N} \quad \text{for every } i.$$

Proof of Lemma 5.2. We use the notation η, σ, σ' of Lemma 5.1, so that $h_i = |\sigma_i|^{p-2}+\rho$ and $\beta_i = \max\{|\sigma_i'|,\beta\}$. By Hölder's inequality and Lemma 5.1, $\|\eta\|^2 \le N^{1-2/p}(\sum_i |\eta_i|^p)^{2/p} \le 4N^{1-2/p}\Delta^{2/p}$. With $\rho < 2(\Delta/N)^{1-2/p}$ this gives $\rho\|\eta\|^2 < 8\Delta$. Since $|\sigma_i| \le \max\{|\sigma_i|,|\sigma_i'|\}$, Lemma 5.1 also gives $\sum_i |\sigma_i|^{p-2}\eta_i^2 \le 4\Delta$. Adding the two bounds proves $\|\eta\|_{H_x}^2 \le 12\Delta$.

Since $|\sigma_i'|^{p/2} \le |\sigma_i|^{p/2} + \big||\sigma_i'|^{p/2} - |\sigma_i|^{p/2}\big|$, raising to the power $2-4/p \in (0,2)$ and using $(a+b)^{2-4/p} \le 2\big(a^{2-4/p}+b^{2-4/p}\big)$ for $a,b \ge 0$ gives

$$|\sigma_i'|^{p-2} \;\le\; 2|\sigma_i|^{p-2} + 2\,\big||\sigma_i'|^{p/2} - |\sigma_i|^{p/2}\big|^{2-4/p}.$$

Divide by h_i, which is at least $|\sigma_i|^{p-2}$ and at least ρ, and bound the sum of the last terms by Hölder's inequality and the third bound of Lemma 5.1:

$$\sum_i \frac{|\sigma_i'|^{p-2}}{h_i} \;\le\; 2N + \frac{2}{\rho}N^{2/p}(2\Delta)^{1-2/p} \;\le\; 2N + 2^{2-2/p}N \;\le\; 6N,$$

where the middle step uses $\rho \ge (\Delta/N)^{1-2/p}$. Since $\beta_i^{p-2} \le |\sigma_i'|^{p-2} + \beta^{p-2}$ and $\beta^{p-2} \le \rho \le h_i$, the second claim follows. Finally $h_i\beta_i^2 \ge \rho\beta^2 \ge \beta^p = \Delta/N$. □

Lemma 5.3 (restated). *For $a, w \in \mathbb{R}$ and $\beta_0 > 0$ with $\beta_0 \ge |a|$,*

$$0 \;\le\; \phi(a+w) - \phi(a) - \phi'(a)w \;\le\; \tfrac{p^2}{2}\beta_0^{p-2}e^{p|w|/\beta_0}w^2.$$

Proof of Lemma 5.3. The lower bound holds because ϕ is convex. By Taylor's theorem the middle term equals $\frac{1}{2}\phi''(\zeta)w^2$ for some ζ between a and $a+w$. As $|\zeta| \le \beta_0 + |w|$,

$$\phi''(\zeta) = p(p-1)|\zeta|^{p-2} \le p^2\beta_0^{p-2}(1+|w|/\beta_0)^{p-2} \le p^2\beta_0^{p-2}e^{p|w|/\beta_0}.$$ □

Lemma 5.4 (restated). *Let y be a minimizer of F_t on L, let $\Delta > 0$, and let $x \in y + 2L$ satisfy $x \ne y$ and $F_t(x) \le F_t(y) + \Delta$. Let H_x be defined by (4), let $\mathcal{D}$ be a distribution on $2L$ satisfying (3) with $K = 2L$, $H = H_x$, $r = 4\sqrt{\Delta}$ and some $\gamma \in (0,1]$, and let $V \sim \mathcal{D}$. Then*

$$\Pr\big[F_t(x+V) \le F_t(y) + \Delta/4\big] \;\ge\; \tfrac{1}{4}\cdot(188p)^{-N}\gamma.$$

Full proof of Lemma 5.4. We use the notation η, σ, σ' of Lemma 5.1 and h_i, β_i of (4), and we let $K = 2L$. We construct a set $E \subseteq K \cap \mathcal{B}_{H_x}(4\sqrt{\Delta})$ such that every $z \in E$ satisfies $F_t(x+z) - F_t(y) < \Delta/4$ and $|E \setminus \{0\}| \geq \frac{1}{4}(188p)^{-N}|K \cap \mathcal{B}_{H_x}(4\sqrt{\Delta})|$. Then (3) gives the lemma. The set is $E = \eta + S_-$, where S_- will be chosen so that every $w \in S_-$ satisfies $F_t(y+w) - F_t(y) < \Delta/4$. We define S_- through a box and a sign condition. The box gives the bound on each $|w_i|$ required by Lemma 5.3 at a smaller cost in the count than a ball, and the sign condition ensures that the linear term of the Taylor expansion of F_t at y is nonpositive.

Let $\tau = (\Delta/(32N))^{1/2}/p$, and define

$$Q = \{w \in \mathbb{R}^N : \ h_i w_i^2 \leq \tau^2 \text{ for all } i\}, \qquad S_- = \Big\{w \in K \cap Q : \ \sum_i \phi'(\sigma_i')\, w_i \leq 0\Big\}.$$

The set $K \cap Q$ is symmetric and the sum in the definition of S_- changes sign with w, so w or $-w$ lies in S_- for every $w \in K \cap Q$. Hence $|S_-| \geq |K \cap Q|/2$.

Claim 7.5 (Improvement)**.** *For every $w \in S_-$, $F_t(y+w) - F_t(y) < \Delta/4$.*

Proof. Let $w \in S_-$. By Lemma 5.2, $h_i\beta_i^2 \geq \Delta/N$, so

$$\frac{p|w_i|}{\beta_i} \ \leq \ \frac{p\tau}{\sqrt{h_i}\,\beta_i} \ \leq \ \frac{1}{\sqrt{32}}, \qquad p^2 w_i^2 \ \leq \ \frac{p^2\tau^2}{h_i} \ = \ \frac{\Delta}{32Nh_i}.$$

We have

$$F_t(y+w) - F_t(y) \ = \ \sum_i \phi'(\sigma_i')\, w_i + \sum_i \big(\phi(\sigma_i' + w_i) - \phi(\sigma_i') - \phi'(\sigma_i')w_i\big),$$

and the first sum is at most 0 because $w \in S_-$. Bounding each term of the second sum by Lemma 5.3 with $a = \sigma_i'$ and $\beta_0 = \beta_i \geq |\sigma_i'|$, and then using Lemma 5.2, we obtain

$$F_t(y+w) - F_t(y) \ \leq \ \frac{e^{1/\sqrt{32}}}{2} \sum_i p^2 \beta_i^{p-2} w_i^2 \ \leq \ \frac{e^{1/\sqrt{32}}}{2} \cdot \frac{\Delta}{32N} \sum_i \frac{\beta_i^{p-2}}{h_i} \ \leq \ \frac{7}{64} e^{1/\sqrt{32}}\, \Delta.$$

Since $\frac{7}{64}e^{1/\sqrt{32}} < \frac{1}{4}$, the right-hand side is less than $\Delta/4$. □

Claim 7.6 (Containment in the sampling ball)**.** $\eta + S_- \subseteq K \cap \mathcal{B}_{H_x}(4\sqrt{\Delta})$.

Proof. For $w \in Q$, $\|w\|_{H_x} \leq \sqrt{N}\tau = \sqrt{\Delta/32}/p < 0.06\sqrt{\Delta}$ since $p \geq 3$. With Lemma 5.2, $\|\eta + w\|_{H_x} \leq (\sqrt{12} + 0.06)\sqrt{\Delta} < 4\sqrt{\Delta}$. □

Claim 7.7 (Counting bound)**.** $|K \cap \mathcal{B}_{H_x}(4\sqrt{\Delta})| \leq (188p)^N\, |K \cap Q|$.

Proof. The linear map $w \mapsto H_x^{1/2} w$ sends K to the lattice $H_x^{1/2}K$, the set Q to the cube $[-\tau, \tau]^N$, and $\mathcal{B}_{H_x}(4\sqrt{\Delta})$ to $\mathcal{B}(4\sqrt{\Delta})$. Since $\tau \leq 4\sqrt{\Delta}/\sqrt{N}$, Lemma 2.2(ii) with $\alpha = \tau$ and (2) cover $\mathcal{B}(4\sqrt{\Delta})$ by at most

$$\omega_N \cdot \Big(\frac{8\sqrt{\Delta}}{\tau}\Big)^N \ \leq \ \Big(\frac{2\pi e}{N}\Big)^{N/2} \big(8\sqrt{32N}\, p\big)^N \ = \ \big(64\sqrt{\pi e}\, p\big)^N \ < \ (188p)^N$$

translates of $[0, \tau]^N$, which is a translate of $\frac{1}{2}[-\tau, \tau]^N$. Lemma 2.1 applied to $H_x^{1/2}K$ proves the claim. □

Let $E = \eta + S_-$. By Claim 7.5, every $z \in E$ satisfies $F_t(x+z) - F_t(y) < \Delta/4$. Since $|E| = |S_-|$ and E contains $\eta \neq 0$, the argument in the proof of Lemma 4.2 gives $|E \setminus \{0\}| \geq |S_-|/2$. By Claims 7.6 and 7.7 and (3),

$$\Pr[V \in E] = \mathcal{D}(E) \;\geq\; \gamma \frac{|S_-|/2}{(188p)^N |K \cap Q|} \;\geq\; \gamma \frac{|S_-|/2}{2(188p)^N |S_-|} \;=\; \tfrac{1}{4}\gamma\,(188p)^{-N}. \qquad \square$$

7.4.2 Full pseudocode and parameters

Algorithm 3 presents our algorithm for exact ℓ_p-CVP for integer $p \geq 3$, following the structure of Algorithm 2. It represents a lattice point Bc by its coefficient vector $c \in \mathbb{Z}^N$ and evaluates $F_t(Bc)$ exactly in rational arithmetic.

Description. The algorithm processes the cosets $Ba + 2L$, $a \in \{0,1\}^N$, one at a time and runs the same fixed sequence of stages in each coset. At the start of stage j, it stores the current point $x_{\text{old}} = Bc_{\text{old}}$ and computes the diagonal metric $H = H_{x_{\text{old}}}$ from (4) with $\Delta = \Delta_j$. This metric is kept fixed throughout the stage. Every sampled correction is added to the same point x_{old}, and the algorithm keeps the best point found so far according to F_t. Thus, the structure is the same as in Algorithm 2, except that the metric is recomputed at the start of each stage and the sampler is run in that weighted metric.

Parameters. Let d be the least common multiple of the denominators of the entries of B and t. The algorithm uses the numbers

$$U = \sum_i \Big(|t_i| + \sum_k |B_{ik}|\Big)^p, \qquad \Delta_j = U/4^j \;\; (j \geq 0), \qquad J = \min\{j \geq 0 :\; \Delta_j < d^{-p}\}.$$

For $a \in \{0,1\}^N$, we have $|(Ba)_i - t_i| \leq |t_i| + \sum_k |B_{ik}|$, so $F_t(Ba) \leq U$. The number J of stages is chosen as follows. Every coordinate of a lattice point or of t lies in $d^{-1}\mathbb{Z}$, so every value of F_t on L lies in $d^{-p}\mathbb{Z}$. Hence, a lattice point x with $F_t(x) \leq F_t(y) + \Delta_J$ for a minimizer y satisfies $F_t(x) = F_t(y)$. Since U and d^{-p} are positive rationals of $\text{poly}(p,\ell)$ bit length, we have $J = O(1 + \log_2 \max\{1, Ud^p\}) = \text{poly}(p,\ell)$. Each row of the nonsingular matrix B has a nonzero entry, whose absolute value is at least $1/d$. Hence $U \geq Nd^{-p}$, and in particular $J \geq 1$.

At stage j, the lattice $2L$ in the metric $H = H_{x_{\text{old}}}$ has Gram matrix $(2B)^T H (2B)$. A nonzero vector of $2L$ has a coordinate of absolute value at least $2/d$, so its squared H-length is at least $4\rho/d^2$. The algorithm calls the sampler of Theorem 3.1 on $((2B)^T H (2B), 16\Delta_j, 4\rho/d^2)$, that is, with radius $r = 4\sqrt{\Delta_j}$ and $\lambda_0^2 = 4\rho/d^2$.

To bound the success probability uniformly over the stages, we bound r/λ_0. For $j < J$, we have $\Delta_j \geq d^{-p}$, and since $0 < 1 - 2/p < 1$,

$$\rho \;\geq\; (\Delta_j/N)^{1-2/p} \;\geq\; \min\{1, \Delta_j/N\} \;\geq\; d^{-p}/N.$$

Hence $(r/\lambda_0)^2 = 16\Delta_j d^2/(4\rho) \leq 4Ud^{p+2}N$, and $4Ud^{p+2}N \geq 1$ because $U \geq Nd^{-p}$. Thus $\log_2 \max\{1, r/\lambda_0\} \leq \lceil \log_2(4Ud^{p+2}N) \rceil$ in every stage. By Lemma 5.4 and Theorem 3.1(b), a sample of stage j therefore succeeds with probability at least

$$\theta = \frac{(1692p)^{-N}}{2048\,N\,(\lceil \log_2(4Ud^{p+2}N) \rceil + 4)^2}, \qquad \text{and we put} \qquad \mathcal{M} = \theta^{-1} \Big\lceil \log_2 \frac{4 \cdot 2^N J}{\varepsilon} \Big\rceil.$$

Both θ^{-1} and $\mathcal{M}$ are integers, and $\mathcal{M} = (1692p)^N \,\text{poly}(p,\ell)$.

Algorithm 3: Exact CVP in the ℓ_p norm for an integer $p \geq 3$

Input : Nonsingular $B \in \mathbb{Q}^{N\times N}$, target $t \in \mathbb{Q}^N$, integer $p \geq 3$, rational $\varepsilon \in (0, 1/2)$
Output : Coefficient vector of a closest lattice point in the ℓ_p norm

$c_{\text{best}} \leftarrow 0$
for $a \in \{0,1\}^N$ **do**
  $c \leftarrow a$
  for $j = 0, \ldots, J-1$ **do**
    $c_{\text{old}} \leftarrow c$
    $H \leftarrow H_{Bc_{\text{old}}}$ computed by (4) with $\Delta = \Delta_j$
    for $\mathcal{M}$ times **do**
      $\bar{c} \leftarrow$ output of Algorithm 1 on $\big((2B)^T H(2B),\ 16\Delta_j,\ 4\rho/d^2\big)$
      if $F_t(B(c_{\text{old}} + 2\bar{c})) < F_t(Bc)$ **then** $c \leftarrow c_{\text{old}} + 2\bar{c}$
  if $F_t(Bc) < F_t(Bc_{\text{best}})$ **then** $c_{\text{best}} \leftarrow c$
return c_{best}

7.4.3 Correctness and Resource Analysis

We prove the correctness and resource bounds of Algorithm 3, completing the proof of Theorem 1.1 for $p \geq 3$.

Lemma 7.8. *Assume that the oracle is exact. Let y be a minimizer of F_t on L, and let $a \in \{0,1\}^N$ satisfy $y \in Ba + 2L$. Then, with probability at least $1 - \varepsilon/(4 \cdot 2^N)$, the iteration of Algorithm 3 for a ends with $Bc = y$.*

Proof. The proof follows that of Lemma 7.1. Let x_j be the current point at the start of stage j, and let x_J be the point at the end of the last stage. All these points lie in $y + 2L$, and

$$F_t(x_0) \leq U = \Delta_0 \leq F_t(y) + \Delta_0,$$

so the invariant holds initially. Fix $j < J$ and condition on the computation up to the start of stage j, assuming that $F_t(x_j) \leq F_t(y) + \Delta_j$. If $x_j = y$, then $x_{j+1} = y$, because the current point changes only when F_t decreases strictly. Otherwise, Lemma 5.4 with $x = x_j$ and $\Delta = \Delta_j$, together with Theorem 3.1(b), shows that each sample of stage j yields a point x' with $F_t(x') \leq F_t(y) + \Delta_{j+1}$ with probability at least θ, conditionally on the earlier samples. As in the proof of Lemma 7.1, all $\mathcal{M}$ samples fail with probability at most $\varepsilon/(4 \cdot 2^N J)$. A union bound over the J stages shows that $F_t(x_J) \leq F_t(y) + \Delta_J$ with probability at least $1 - \varepsilon/(4 \cdot 2^N)$. In this event, $F_t(x_J) = F_t(y)$ by the choice of J, so x_J is also a minimizer of F_t on L. Since F_t is strictly convex, two distinct minimizers y, y' in the same coset of $2L$ would give a lattice point $(y + y')/2$ with $F_t((y + y')/2) < F_t(y)$. Hence, $x_J = y$. □

Lemma 7.9. *Algorithm 3 makes at most $2^N J\mathcal{M}$ oracle calls, each on a full-rank sublattice of $2L$ in a diagonal metric, given by a $\text{poly}(p, \ell)$-bit Gram matrix. If every oracle answer is an integer vector of $\text{poly}(p, \ell)$ bits, in particular if the oracle is exact, then the algorithm uses $(Cp)^N \text{poly}(p, \ell)$ other bit operations, for an absolute constant C, and $\text{poly}(p, \ell)$ space outside the oracle.*

Proof. Each of the $2^N J$ stages makes $\mathcal{M}$ calls of the sampler, and each of them makes at most one oracle call by Theorem 3.1(c). The current point satisfies $F_t(Bc) \leq U$ at all times, because

it starts at a point Ba with $F_t(Ba) \le U$ and F_t never increases. Hence, $|(Bc)_i - t_i| \le \max\{1, U\}$, and since $c = B^{-1}(Bc)$, the vector c has $\mathrm{poly}(p, \ell)$ bits. The same holds for c_{old} and c_{best}. The number ρ lies between d^{-p}/N and $2\max\{1, U\}$, so H, $4\rho/d^2$, and $16\Delta_j$ have $\mathrm{poly}(p, \ell)$ bits. Hence, Theorem 3.1(c) implies that every SVP query has $\mathrm{poly}(p, \ell)$-bit Gram matrix. For an exact oracle, every shortest coefficient vector for such a query has $\mathrm{poly}(p, \ell)$ bits by the coefficient bound following Lemma A.2. Under the stated assumption on oracle answers, Theorem 3.1(c) therefore implies that each call of the sampler uses $\mathrm{poly}(p, \ell)$ non-oracle bit operations and space. Every sample lies in $\mathcal{B}_H(4\sqrt{U})$ by Theorem 3.1(a). Since $H \succeq \rho I$, its ambient coordinates have $\mathrm{poly}(p, \ell)$ bits; applying the rational matrix $(2B)^{-1}$ then shows that its coefficient vector with respect to the basis $2B$ also has $\mathrm{poly}(p, \ell)$ bits. Since $\mathcal{M} = (1692p)^N \mathrm{poly}(p, \ell)$ and $J = \mathrm{poly}(p, \ell)$, the number of bit operations is $(Cp)^N \mathrm{poly}(p, \ell)$. Besides the workspace of one call of the sampler, the algorithm stores three coefficient vectors, one diagonal matrix, and the loop counters. □

Proof of Theorem 1.1 for $p \ge 3$. The proof follows that for $p = 2$, with Lemmas 7.8 and 7.9 in place of Lemmas 7.1 and 7.2. By Lemma 7.8, the coefficient vector returned by Algorithm 3 represents a minimizer of F_t on L, and hence an ℓ_p-closest vector, with probability at least $1 - \varepsilon$. Each coset of $2L$ contains at most one minimizer by the strict convexity argument at the end of the proof of Lemma 7.8. Applying the two-run procedure of Section 7.2 to Algorithm 3, with comparisons made using F_t, outputs every minimizer exactly once with probability at least $1 - \varepsilon/2$. For an SVP algorithm that succeeds with probability at least $2/3$, Lemma A.2 applies as in the proof for $p = 2$, with $\varepsilon/2$ in place of ε and $T_{\mathrm{o}} = 2^{N+1} J\mathcal{M}$. □

8 Conclusion

We have presented a randomized, rank-preserving reduction from exact ℓ_p-CVP to exact Euclidean SVP for every integer $p \ge 2$, using only polynomial working space outside the oracle. For $p = 2$, our reduction shows that the randomized polynomial-space time complexities of exact Euclidean SVP and CVP agree up to a $2^{O(N)} \mathrm{poly}(\ell)$ factor. For every fixed integer $p \ge 3$, it yields exact ℓ_p-CVP algorithms for arbitrary rational targets with either single-exponential time and space or $N^{N/(4e)+o(N)} \mathrm{poly}(\ell)$ time and polynomial space. The reduction combines local search within cosets of $2L$, counting arguments for improving corrections, and SVP-based sampling. An interesting direction for future work is to reduce the exponential overhead while preserving its rank and space guarantees.

A Oracle Details

This appendix presents two implementation facts used in the reductions. First, we show that an SVP query given by a positive definite rational Gram matrix can be converted, without increasing the rank, into an equivalent query on an integer basis. Second, we show how an exact SVP oracle can be replaced by a bounded-error SVP algorithm with only polynomial overhead.

Lemma A.1 (Rational Gram matrices)**.** *Exact SVP for a positive definite rational $N \times N$ Gram matrix G reduces in polynomial time to exact SVP on an integer square basis of rank N, in the sense that every shortest coefficient vector for the new basis is a shortest coefficient vector for G.*

Proof. Let D_0 clear the denominators of G, and put $a_{\max} = \max_i G_{ii}$, $\iota = \iota(G)$ and $b_0^2 = 4a_{\max}/\iota$. Distinct values of $a^T G a$, $a \in \mathbb{Z}^N$, differ by at least $1/D_0$. Choose a positive rational ε_0 satisfying

$$\varepsilon_0 < \min\{\iota/4,\ 1/(8D_0 b_0^2)\}.$$

A rational LDL^T decomposition of G, followed by sufficiently accurate dyadic approximations of the square roots of the diagonal entries in the decomposition, gives a rational square matrix $\widetilde{B}$ with $\|\widetilde{B}^T\widetilde{B} - G\|_{\mathrm{op}} < \varepsilon_0$. Here, an entrywise error bound multiplied by N bounds the operator norm. Write $\widetilde{G} = \widetilde{B}^T\widetilde{B}$.

Since $\iota \le \lambda_{\min}(G) \le a_{\max}$, we have $\varepsilon_0 < a_{\max}/4$, and the least eigenvalue of $\widetilde{G}$ is at least $\frac{3}{4}\iota$. Let a^* and $\tilde{a}$ be shortest coefficient vectors for G and for $\widetilde{G}$, respectively. Comparing with unit vectors gives $a^{*T}Ga^* \le a_{\max}$ and $\tilde{a}^T\widetilde{G}\tilde{a} \le a_{\max} + \varepsilon_0 < \frac{5}{4}a_{\max}$. Hence, $\|a^*\|^2 \le a_{\max}/\iota < b_0^2$ by (1), and $\|\tilde{a}\|^2 < \frac{5}{4}a_{\max}/(\frac{3}{4}\iota) < b_0^2$. For both vectors the two quadratic forms differ by less than $\varepsilon_0 b_0^2 < 1/(8D_0)$. If $\tilde{a}^TG\tilde{a} > a^{*T}Ga^*$, then $\tilde{a}^TG\tilde{a} \ge a^{*T}Ga^* + 1/D_0$, and therefore $\tilde{a}^T\widetilde{G}\tilde{a} > a^{*T}\widetilde{G}a^* + 3/(4D_0)$, which contradicts the choice of $\tilde{a}$. Hence $\tilde{a}$ is shortest for G.

Let D be a positive integer clearing the denominators of $\widetilde{B}$. Then $\widehat{B} = D\widetilde{B}$ is an integer basis with Gram matrix $\widehat{B}^T\widehat{B} = D^2\widetilde{G}$, so its shortest coefficient vectors are exactly those of $\widetilde{G}$. The entries of the rational LDL^T decomposition are ratios of minors of G, and $\log(1/\varepsilon_0)$ is polynomial in the bit length of G because D_0, $a_{\max}$ and ι are. Thus, ε_0 can be chosen with polynomial bit length, and all numbers involved have polynomial bit length. □

Lemma A.2 (Bounded-error SVP algorithms)**.** *Let $0 \le \varepsilon_1 < 1$ and $0 < \varepsilon_2 < 1$. Let $\mathcal{R}$ be a randomized algorithm with an exact SVP oracle that succeeds with probability at least $1-\varepsilon_1$ for every rule by which the oracle chooses among shortest vectors. Suppose that before its first call $\mathcal{R}$ computes $T_{\mathrm{o}} \ge 1$ and ℓ_0 such that, for all answers that pass the check below, it makes at most T_{o} calls, each on a lattice whose Gram matrix and shortest nonzero coefficient vectors have at most ℓ_0 bits. Let $\mathcal{S}$ be an SVP algorithm that always halts within its time bound and returns a shortest nonzero vector with probability at least $2/3$. Answer each call by running $\mathcal{S}$ independently $k = \lceil \log_3(T_{\mathrm{o}}/\varepsilon_2) \rceil$ times and returning the shortest output that is a nonzero integer coefficient vector of the correct dimension and of at most ℓ_0 bits, or aborting if there is none. Then, $\mathcal{R}$ succeeds with probability at least $1-\varepsilon_1-\varepsilon_2$, and each call costs k runs of $\mathcal{S}$.*

Proof. The idea is to amplify each oracle call enough that, with high probability, at least one of the repeated runs returns an actual shortest vector. Conditioned on the history, all k runs fail to return a shortest vector with probability at most $3^{-k} \le \varepsilon_2/T_{\mathrm{o}}$. If at least one run succeeds, then one retained output is a shortest vector; since every retained nonzero integer vector represents a lattice vector, the shortest retained output is also shortest.

Couple the execution with one that uses an exact oracle returning the same answer whenever the simulation returns a shortest vector. The two executions agree until the first call at which the simulation fails to return a shortest vector, including the case that it aborts, and a union bound over at most T_{o} calls bounds the probability of such a failure by ε_2. Therefore, the overall success probability is at least $1-\varepsilon_1-\varepsilon_2$. □

The coefficient bound required in Lemma A.2 can be computed directly from the query Gram matrix. For a query with Gram matrix G', a shortest vector with coefficient vector a satisfies

$$\iota(G')\|a\|^2 \le a^TG'a \le \min_j G'_{jj},$$

where the upper bound follows by comparing the shortest vector with the unit coefficient vectors. Hence,

$$\|a\|^2 \le \frac{\min_j G'_{jj}}{\iota(G')}.$$

Thus, when G' has polynomial bit length, the right-hand side has polynomial bit length, and the coordinates of every shortest coefficient vector have polynomial bit length as well.

AI Methodology

We used OpenAI Codex, ChatGPT and Claude to assist with developing and checking proofs, comparing the literature, and editing. The tools materially affected Sections Sections 3, 4, 5 and 6, in particular the counting arguments of Lemmas 4.2 and 5.4, and the short-vector sampler in Section 3 and its analysis in Section 6.

References

[ACK+21] Divesh Aggarwal, Yanlin Chen, Rajendra Kumar, Zeyong Li, and Noah Stephens-Davidowitz. Dimension-preserving reductions between SVP and CVP in different p-norms. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2444–2462, 2021. doi:10.1137/1.9781611976465.145.

[AdFM24] Sunil Arya, Guilherme D. da Fonseca, and David M. Mount. Economical convex coverings and applications. *SIAM Journal on Computing*, 53(4):1002–1038, 2024. doi:10.1137/23M1568351.

[ADS15] Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the closest vector problem in 2^n time—the discrete Gaussian strikes again!, 2015. Version 2, 29 September 2015. URL: https://arxiv.org/abs/1504.01995v2, arXiv:1504.01995.

[AKS02] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. Sampling short lattice vectors and the closest lattice vector problem. In *Proceedings of the 17th IEEE Annual Conference on Computational Complexity*, pages 53–57, 2002. URL: https://research.ibm.com/publications/sampling-short-lattice-vectors-and-the-closest-lattice-vector-problem.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004. doi:10.4007/annals.2004.160.781.

[BK18] Johannes Blömer and Kathlén Kohn. Voronoi cells of lattices with respect to arbitrary norms. *SIAM Journal on Applied Algebra and Geometry*, 2(2):314–338, 2018. doi:10.1137/17M1132045.

[BN11] Johannes Blömer and Stefanie Naewe. Solving the closest vector problem with respect to ℓ_p norms, 2011. Version 4, 26 September 2011. URL: https://arxiv.org/abs/1104.3720v4, arXiv:1104.3720.

[DER25] Daniel Dadush, Friedrich Eisenbrand, and Thomas Rothvoss. From approximate to exact integer programming. *Mathematical Programming*, 210:223–241, 2025. Full version arXiv:2211.03859v4, 8 April 2024. URL: https://arxiv.org/abs/2211.03859v4, doi:10.1007/s10107-024-02084-1.

[DH11] Chandan Dubey and Thomas Holenstein. Approximating the closest vector problem using an approximate shortest vector oracle. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 6845 of *Lecture Notes in Computer Science*, pages 184–193. Springer, 2011. doi:10.1007/978-3-642-22935-0_16.

[DK16] Daniel Dadush and Gábor Kun. Lattice sparsification and the approximate closest vector problem. *Theory of Computing*, 12(2):1–34, 2016. doi:10.4086/toc.2016.v012a002.

[DM13] Daniel Dadush and Daniele Micciancio. Algorithms for the densest sub-lattice problem, 2013. Author full version dated 24 December 2012; SODA 2013. URL: https://homepages.cwi.nl/~dadush/papers/min-det.pdf.

[DPV11] Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative lattice algorithms in any norm via M-ellipsoid coverings. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 580–589, 2011. doi:10.1109/FOCS.2011.31.

[DV12] Daniel Dadush and Santosh Vempala. Near-optimal deterministic algorithms for volume computation via M-ellipsoids, 2012. Author version, May 2012. URL: https://homepages.cwi.nl/~dadush/papers/m-ellipsoid.pdf.

[EV20] Friedrich Eisenbrand and Moritz Venzin. Approximate CVP_p in time $2^{0.802n}$. In *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of *Leibniz International Proceedings in Informatics*, pages 43:1–43:15, 2020. doi:10.4230/LIPIcs.ESA.2020.43.

[FGL26] Yansong Feng, Yiming Gao, and Jiaqi Liu. Faster SVP in polynomial space, 2026. Version 1, 18 September 2026. URL: https://arxiv.org/abs/2609.21612v1, arXiv:2609.21612.

[Fu16] Lianzhi Fu. Improvements in the analysis of Kannan's CVP algorithm. *Theoretical Computer Science*, 621:103–112, 2016. doi:10.1016/j.tcs.2016.01.038.

[GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999. URL: https://cseweb.ucsd.edu/~daniele/papers/GMSS.pdf, doi:10.1016/S0020-0190(99)00083-6.

[Hha26] Minki Hhan. Finding a shortest vector and more in $2^{n/2+o(n)}$ time using q-ary coset difference tree, 2026. Version 1, 2 September 2026. URL: https://arxiv.org/abs/2609.02764v1, arXiv:2609.02764.

[HRS20] Christoph Hunkenschröder, Gina Reuland, and Matthias Schymura. On compact representations of Voronoi cells of lattices. *Mathematical Programming*, 183:337–358, 2020. doi:10.1007/s10107-019-01463-3.

[HS07] Guillaume Hanrot and Damien Stehlé. Improved analysis of Kannan's shortest lattice vector algorithm, 2007. Full version of the CRYPTO 2007 paper, version 2, 9 May 2007. URL: https://arxiv.org/abs/0705.0965v2, arXiv:0705.0965.

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing (STOC)*, pages 193–206, 1983. doi:10.1145/800061.808749.

[Kho05] Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, 2005. doi:10.1145/1089023.1089027.

[Laa26] Thijs Laarhoven. Asymptotics of random-lattice enumeration: Beyond HKZ via dense sublattices, 2026. Version 1, 11 October 2026. URL: https://arxiv.org/abs/2610.15265v1, arXiv:2610.15265.

[MV10] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations, 2010. Full version, ECCC TR10-014, revision 1. URL: https://eccc.weizmann.ac.il/report/2010/014/revision/1/download.

[NV22] Márton Naszódi and Moritz Venzin. Covering convex bodies and the closest vector problem. *Discrete & Computational Geometry*, 67:1191–1210, 2022. doi:10.1007/s00454-022-00392-x.

[RR26] Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming, 2026. Full version of the FOCS 2023 paper; version 5, 27 March 2026. URL: https://arxiv.org/abs/2303.14605v5, arXiv:2303.14605.

[Sho03] Victor Shoup. A computational introduction to number theory and algebra, 2003. Beta version 1, Theorem 5.11, p. 51. URL: https://shoup.net/ntb/ntb-b1.pdf.

[Ste16] Noah Stephens-Davidowitz. Discrete Gaussian sampling reduces to CVP and SVP, 2016. Full version of the SODA 2016 paper, version 4, 19 April 2016. URL: https://arxiv.org/abs/1506.07490v4, arXiv:1506.07490.