# Fast Provable Algorithms for Approximate CVP and Exact SVP on Random $q$-ary Lattices

Jiseung Kim[1] Junhyuk Kwon[2] Changmin Lee[2,*] JeongHwan Lee[2]

[1]Jeonbuk National University, Jeonju, Republic of Korea
[2]Korea University, Seoul, Republic of Korea

jiseungkim@jbnu.ac.kr; {kwon76,changminlee,hwani0814}@korea.ac.kr

[*]Corresponding author

**Abstract**

We give provable randomized algorithms for the $\gamma$-approximate Closest Vector Problem ($\gamma$-CVP) with $\gamma = 5/3$ and the exact Shortest Vector Problem (SVP) in the Euclidean norm on random $q$-ary lattices. These lattices consist of the integer vectors mapped to zero modulo $q$ by a uniformly random matrix $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, for any integer $q \geq 2$. For input density $\delta = m \log_2 q/n$ with $0 \leq \delta \leq 1$, the running times are $2^{0.2668n+o(n)}$ for 5/3-CVP and $2^{0.2741n+o(n)}$ for exact SVP, with polynomial input-length factors suppressed. For $1 < \delta \leq 3/2$, the respective bounds are $2^{0.56n+o(n)}$ and $2^{0.468n+o(n)}$. For all but a negligible fraction of matrices, both algorithms succeed with overwhelming probability over their internal randomness. Notably, for $0 \leq \delta \leq 1$, our provable bounds fall below even the $2^{0.2925n+o(n)}$ heuristic estimate for classical lattice sieving.

As a cryptographic application, we prove that signatures in the original construction of Gentry, Peikert, and Vaikuntanathan (GPV) can be forged in time $2^{0.0534n+o(n)}$ for $n \geq 5m \log_2 q$. For a 128-bit security target, a leading-term estimate gives $n \approx 2400$ and a public key of roughly 144 kB, indicating a substantial storage burden.

**Keywords:** Random $q$-ary Lattices, Closest Vector Problem, Shortest Vector Problem, Representation Method.

# 1 Introduction

For an integer $n \geq 1$, let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice. For a target $\boldsymbol{y} \in \mathbb{R}^n$ and an approximation factor $\gamma > 1$, the $\gamma$-approximate closest vector problem ($\gamma$-CVP) asks for a vector $\boldsymbol{x} \in \Lambda$ satisfying $\|\boldsymbol{x} - \boldsymbol{y}\| \leq \gamma \min_{\boldsymbol{v} \in \Lambda} \|\boldsymbol{v} - \boldsymbol{y}\|$. The shortest vector problem (SVP) asks for a nonzero vector in $\Lambda$ of minimum length $\lambda_1(\Lambda) = \min_{\boldsymbol{v} \in \Lambda \setminus \{\mathbf{0}_n\}} \|\boldsymbol{v}\|$. Throughout, we use the Euclidean norm.[1]

In this work, we study these problems on random $q$-ary lattices defined as

$$\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^n : \mathbf{H}\boldsymbol{x} \equiv \mathbf{0}_m \pmod{q}\} \ ,$$

where $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ is chosen uniformly at random for integers $n \geq 1$, $q \geq 2$, and $m \geq 0$.

Studying $\gamma$-CVP and SVP on random $q$-ary lattices is important for understanding the cost of finding short solutions to modular equations. The short integer solution (SIS) problem asks, given $\mathbf{H}$ and a bound $B > 0$, for a nonzero $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying $\mathbf{H}\boldsymbol{x} \equiv \mathbf{0}_m \pmod{q}$ and $\|\boldsymbol{x}\| \leq B$. Such a vector describes a nontrivial integer relation among the columns of $\mathbf{H}$, with small coefficients as measured by the Euclidean norm. These relations are precisely the nonzero vectors of $\Lambda$. SVP finds the shortest such relation and therefore solves SIS whenever one exists within the bound $B$.

The inhomogeneous short integer solution (ISIS) problem instead asks for a short vector whose image is a prescribed $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$: $\mathbf{H}\boldsymbol{x} \equiv \boldsymbol{t} \pmod{q}$ and $\|\boldsymbol{x}\| \leq B$. To see its connection to $\gamma$-CVP, first compute any integer preimage $\boldsymbol{z}$, without requiring it to be short. This can be done in polynomial time whenever the congruence is solvable. Subtracting a lattice vector $\boldsymbol{v} \in \Lambda$ preserves the prescribed image, since $\mathbf{H}(\boldsymbol{z} - \boldsymbol{v}) \equiv \mathbf{H}\boldsymbol{z} - \mathbf{H}\boldsymbol{v} \equiv \boldsymbol{t} \pmod{q}$. The length of the resulting preimage $\boldsymbol{z} - \boldsymbol{v}$ is exactly the distance from $\boldsymbol{z}$ to $\boldsymbol{v}$, and every integer preimage arises this way. Thus, finding a nearby lattice vector amounts to finding a short preimage. A $\gamma$-CVP algorithm returns a preimage at most $\gamma$ times as long as the shortest one, solving ISIS whenever a preimage of norm at most $B/\gamma$ exists.

SIS and ISIS are central to the security analysis of lattice-based cryptographic primitives [Ajt96, MR07]. For the map $\boldsymbol{x} \mapsto \mathbf{H}\boldsymbol{x} \bmod q$, two distinct inputs of norm at most $B$ with the same image yield, by subtraction, a nonzero kernel vector of norm at most $2B$. Thus, finding a collision gives an SIS solution. The hash-and-sign construction of Gentry, Peikert, and Vaikuntanathan (GPV) illustrates the connection with ISIS [GPV08]. It uses a public matrix $\mathbf{H}$ drawn from a distribution statistically close to uniform, together with a secret trapdoor: auxiliary information that enables the signer to sample short preimages efficiently. A signature is a short preimage of the message hash, so finding such a preimage for a fresh message within the verification bound produces a forgery. This is precisely an ISIS instance, which an attacker must solve without the trapdoor. Algorithms for SVP and $\gamma$-CVP therefore provide upper bounds on the cost of solving these underlying SIS and ISIS search problems, subject to the required length bounds.

For SVP on random $q$-ary lattices, the best known provable classical time bound available from general-lattice algorithms is $2^{n/2+o(n)}$, due to Hhan [Hha26], whose algorithm applies to general lattices. For $\gamma$-CVP, the algorithm of Aggarwal, Dadush, and Stephens-Davidowitz [ADSD15] finds an exact closest vector in time $2^{n+o(n)}$, the best known provable classical bound for the exact problem. It therefore also provides this bound for every fixed $\gamma > 1$. [2] However, these general-lattice bounds do not exploit the distribution of the uniformly random matrix $\mathbf{H}$ defining our input lattice.

We define $\delta = m \log_2 q/n$ as the *input density*. The regime $\delta \approx 1$ is particularly interesting because it lies near the counting balance $2^n = q^m$ between binary vectors of length $n$ and possible

[1] Both $\gamma$-CVP and SVP can also be defined with respect to other norms.

[2] Heuristic sieving algorithms offer smaller estimated time exponents [BDGL16, Laa17], but our focus is on provable guarantees.

images under $\mathbf{H}$. When two distinct binary vectors have the same image, their difference is a nonzero vector in $\Lambda$ with entries in $\{-1, 0, 1\}$. More generally, short solutions to $\mathbf{H}\boldsymbol{x} \equiv \boldsymbol{t} \pmod{q}$ can be sought combinatorially: writing $\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2$ turns this congruence into the matching condition $\mathbf{H}\boldsymbol{x}_1 \equiv \boldsymbol{t} - \mathbf{H}\boldsymbol{x}_2 \pmod{q}$. The independence and uniformity of the columns of $\mathbf{H}$ provide additional structure for analyzing these matches.

This combinatorial approach has been explored for ISIS. Bai, Galbraith, Li, and Sheffield [BGLS19] develop algorithms based on representation methods [HGJ10, BCJ11], with heuristic estimates for their performance on random instances. Provable combinatorial algorithms for SIS and ISIS are also known in other length regimes [DEL25]. However, these results do not directly provide the $\gamma$-CVP and SVP guarantees sought here. Extending the representation-based approach to these lattice problems requires overcoming four obstacles. First, searching within a prescribed coefficient set or length bound does not by itself solve $\gamma$-CVP: the returned lattice point must be within $\gamma$ times the distance from the target to its closest lattice point. Second, success for a uniformly random prescribed modular value independent of $\mathbf{H}$ does not ensure success for every $\gamma$-CVP target, including targets chosen after $\mathbf{H}$ is known. Third, SVP requires a shortest nonzero lattice vector, so finding a vector below a prescribed length bound does not suffice. Moreover, the analysis must account for the dependence of shortest vectors on $\mathbf{H}$. The gap is therefore to turn combinatorial searches for short modular relations into provable algorithms for $\gamma$-CVP on all targets and SVP on random $q$-ary lattices in the density regimes considered here.

These observations lead to our main question:

> *Can combinatorial searches for short modular relations yield provable algorithms for $\gamma$-CVP on arbitrary targets and SVP on random $q$-ary lattices, with better running times than general-lattice algorithms?*

### 1.1 Results

We answer this question in the affirmative for $0 \le \delta \le 3/2$. The following theorems give bounds uniform over all integer moduli $q \ge 2$ on each of two density intervals.

**Theorem 1.1** (Informal: 5/3-CVP)**.** *For $0 \le \delta \le 3/2$, there is a randomized algorithm that, given $\mathbf{H}$ and $\mathbf{y} \in \mathbb{Q}^n$, has time and space bounds*

| *Density* | *Time* | *Space* |
|---|---|---|
| $0 \le \delta \le 1$ | $2^{0.2668n+o(n)}$ | $2^{0.2628n+o(n)}$ |
| $1 < \delta \le 3/2$ | $2^{0.56n+o(n)}$ | $2^{0.325n+o(n)}$ |

*on every input and execution, up to factors polynomial in the binary input length. For sufficiently large $n$, all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices $\mathbf{H}$ have the following property: with $\mathbf{H}$ held fixed, with probability at least $1 - 2^{-\Omega(n)}$ over the internal randomness, the algorithm outputs $\mathbf{x} \in \Lambda$ satisfying*

$$\|\mathbf{x} - \mathbf{y}\| \le \frac{5}{3} \min_{\mathbf{v} \in \Lambda} \|\mathbf{v} - \mathbf{y}\|$$

*simultaneously for every $\mathbf{y} \in \mathbb{Q}^n$.*

Formal proofs of these bounds and success guarantees are given in Section 4.

**Theorem 1.2** (Informal: SVP)**.** *For $0 \le \delta \le 3/2$, there is a randomized algorithm that, given $\mathbf{H}$, always outputs a nonzero vector in $\Lambda$. Its time and space bounds are*

| *Density* | *Time* | *Space* |
|---|---|---|
| $0 \le \delta \le 1$ | $2^{0.2741n+o(n)}$ | $2^{0.2614n+o(n)}$ |
| $1 < \delta \le 3/2$ | $2^{0.468n+o(n)}$ | $2^{0.329n+o(n)}$ |

| Problem and reference | Lattice model | Analysis | Time exponent | Space exponent |
|---|---|---|---|---|
| CVP [ADSD15] | General lattices | Provable | 1 | 1 |
| CVP [DL26] | Haar-random lattices | Provable | 0.2925 | 0.2076 |
| CVP [Laa17] | General lattices | Heuristic | 0.2925 | 0.2076 |
| 3.67312-CVP [Hha26] | General lattices | Provable | 0.5 | 0.5 |
| **5/3-CVP, this work** | Random $q$-ary, $0 \le \delta \le 1$ | Provable | **0.2668** | **0.2628** |
| **5/3-CVP, this work** | Random $q$-ary, $1 < \delta \le 3/2$ | Provable | **0.56** | **0.325** |
| SVP [Hha26] | General lattices | Provable | 0.5 | 0.5 |
| SVP [DL26, ZWW26] | Haar-random lattices | Provable | 0.2925 | 0.2076 |
| SVP [BDGL16] | General lattices | Heuristic | 0.2925 | 0.2076 |
| **SVP, this work** | Random $q$-ary, $0 \le \delta \le 1$ | Provable | **0.2741** | **0.2614** |
| **SVP, this work** | Random $q$-ary, $1 < \delta \le 3/2$ | Provable | **0.468** | **0.329** |

Table 1: Classical Euclidean CVP and SVP algorithms. An exponent $\alpha$ denotes a bound of $2^{\alpha n+o(n)}$, with polynomial factors omitted.
The Haar-random CVP bound assumes a uniform target class modulo the lattice [DL26]. Our CVP guarantee holds simultaneously for every rational target.

*on every input and execution, up to factors polynomial in the binary input length. For sufficiently large $n$, all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices $\mathbf{H}$ have the following property: with $\mathbf{H}$ held fixed, the algorithm finds a shortest vector with probability at least $1 - 2^{-\Omega(n)}$ over its internal randomness.*

Formal proofs of these bounds and success guarantees are given in Section 5.

**Comparison with prior bounds.** Throughout $0 \le \delta \le 3/2$, our time bounds for 5/3-CVP and SVP on random $q$-ary lattices improve on those obtained by applying the general-lattice exact algorithms in Table 1. Even for $1 < \delta \le 3/2$, our CVP algorithm achieves a 5/3-approximation simultaneously for all rational targets with time exponent 0.56. Solving CVP on a general lattice gives the same approximation guarantee with time exponent 1 [ADSD15]. In the same range, our SVP exponent is 0.468, below the general-lattice bound of 0.5 [Hha26].

For $0 \le \delta \le 1$, our time exponent for a 5/3-approximation to CVP is 0.2668, compared with the heuristic sieving exponent of 0.2925 for CVP on general lattices [Laa17]. Our provable time exponent of 0.2741 for SVP is below the heuristic sieving exponent of 0.2925 for SVP on general lattices [BDGL16].

**Application to ISIS and SIS.** For $0 \le \delta \le 3/2$, Theorem 1.1 solves ISIS with bound $B$ whenever a preimage of norm at most $3B/5$ exists, and Theorem 1.2 solves SIS whenever $\lambda_1(\Lambda) \le B$.

For short preimages, the correction procedure in our CVP algorithm needs only the first $d = \lceil m \log_2 q \rceil$ columns of $\mathbf{H}$. Suppose $q \ge 3$ and $\delta_0 \le \delta \le 1$ for a fixed $\delta_0 > 0$. Given $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, the procedure finds $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying $\mathbf{H}\boldsymbol{x} \equiv \boldsymbol{t} \pmod q$, and $\|\boldsymbol{x}\|^2 \le 0.2271d$, in time $2^{0.2668d+o(n)}$ and space $2^{0.2628d+o(n)}$, up to polynomial input-length factors. For all but a negligible fraction of matrices $\mathbf{H}$, the runs on all $\boldsymbol{t}$ succeed simultaneously with overwhelming probability (Corollary 4.10).

Our ISIS result complements the subexponential-time result of Ducas, Engelberts, and Loyer [DEL25, Theorem 4.2] by giving a substantially stronger bound on the preimage norm. For prime $q = m^{\Theta(1)}$, their theorem gives subexponential time for permitted norms of size $q\sqrt{m}/m^{o(1)}$, whereas our algorithm guarantees norm at most $\sqrt{0.2271d} = O(\sqrt{m \log q})$ in time $2^{0.2668d+o(n)}$.

**Application to GPV-type signatures.** Our short-preimage algorithm gives a forgery for GPV-type signatures whose verifier accepts any integer preimage of the message hash with norm at most the verification bound $B$. Suppose $m \geq 1$, $q \geq 3$, and $n \geq d = \lceil m \log_2 q \rceil$, with $B^2 \geq 0.2271d$, and assume that the public matrix $\mathbf{H}$ is within negligible statistical distance of uniform. Apply Corollary 4.10 to the first $d$ columns of $\mathbf{H}$ and the hash value used to verify a fresh message. This submatrix has density $(m \log_2 q)/d \in (1/2, 1]$ and remains negligibly close to uniform. Appending zeros to the output gives a preimage of the same hash value under $\mathbf{H}$ without changing its norm. The procedure therefore yields a valid signature with overwhelming probability as $d \to \infty$, in time $2^{0.2668d+o(d)}$ and space $2^{0.2628d+o(d)}$, up to polynomial factors in the binary input length.

The original GPV construction satisfies these distribution and norm conditions for prime $q \geq 3$ polynomially bounded in $m$ and $n \geq 5m \log_2 q$ [GPV08, full version, Proposition 5.8 and Section 5.3.2]. Since $d \leq n/5 + 1$, the resulting bounds throughout this range are time $2^{0.0534n+o(n)}$ and space $2^{0.0526n+o(n)}$.

For fixed $m, q$, increasing $n$ does not increase the leading time exponent of this attack, but increases the size of the public matrix. The smallest such matrix has $n = \lceil 5m \log_2 q \rceil$. To relate the leading time exponent to a 128-bit security target, set $0.2668d = 128$, which gives $d \approx 480$ and, at this smallest dimension, $n \approx 2400$. Counting approximately $\log_2 q$ bits per matrix entry, storing this public matrix takes about $mn \log_2 q \approx n^2/5$ bits, or 144 kB, indicating a substantial public-key storage burden.[3]

**Code availability.** The code and results for verifying the parameter choices and complexity bounds in Appendices F and G are available at

`https://anonymous.4open.science/r/qary-lattice-certificates-0120/`.

## 1.2 Technical overview

Both our $\gamma$– CVP algorithm and our SVP algorithm build on a common procedure for finding short integer solutions to linear congruences modulo $q$. In this subsection, we first describe this procedure and its probabilistic guarantees, and then outline how $\gamma$-CVP and SVP reduce to it.

### 1.2.1 Short-congruence search and its analysis

Our CVP and SVP algorithms use a common short-solution procedure for $q \geq 3$; the case $q = 2$ is treated separately.[4] Given $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n_0}$ and $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, where $m, n_0 \geq 1$, the common procedure searches for vectors $\boldsymbol{v} \in \{-1, 0, 1\}^{n_0}$ satisfying $\mathbf{A}\boldsymbol{v} \equiv \boldsymbol{t} \pmod q$, with prescribed and equal numbers of 1 and $-1$ entries. These sign counts fix the norm of every output (Section 3.1).

[3] This estimate omits lower-order terms and polynomial factors. It is not a concrete security estimate or a proved minimum dimension for 128-bit security.

[4] For $q = 2$, the CVP algorithm uses dynamic programming or enumeration of the binary kernel in the corresponding parameter ranges, and a separate sampler for short binary solutions in the remaining correction branch (Algorithm 3 and Lemma 4.2). The SVP algorithm enumerates nonzero integer vectors of norm at most 2 (Algorithm 4).

We use the representation method of Howgrave-Graham and Joux [HGJ10] in Ozerov's recursive framework [Oze16]. Each candidate admits many decompositions into two ternary summands with prescribed sign counts. At each recursive step, random intermediate congruences restrict the two lists of summands. The procedure generates these lists recursively and merges pairs whose sums satisfy the parent congruence and sign counts. The multiple decompositions give each candidate many ways to be generated despite the intermediate restrictions.

The main difficulty is to prove that these recursive restrictions preserve a candidate fixed independently of the matrix. Fix such a candidate $\boldsymbol{v}$, take $\mathbf{A}$ uniformly at random, and set $\boldsymbol{t} = \mathbf{A}\boldsymbol{v} \bmod q$. In the analysis, we choose a decomposition uniformly among those satisfying the random intermediate congruence. Although $\boldsymbol{v}$ is fixed independently of $\mathbf{A}$, the selected summands depend on $\mathbf{A}$. Thus the same fixed-vector argument cannot be applied directly to the recursive calls.

We compare this experiment with one in which the decomposition is chosen uniformly and independently of the entire matrix, and the value in the intermediate congruence is then set to the image of the first summand under the map defining that congruence. A second-moment bound on the number of compatible decompositions bounds the statistical distance between the two experiments. In the latter experiment, the failure bound for each recursive call applies to its summand; a union bound combines these bounds, and the statistical distance controls the additional error in transferring them to the original experiment (Lemma B.3 and Appendix B.4). The comparison retains the entire matrix because rows outside a child's congruence can still determine the order in which its outputs are merged.

At the bottom level, Schroeppel–Shamir ordered generation [SS81] produces vectors in merge order without storing the full output list. The inclusion guarantee holds for every permitted ordering, so this implementation remains compatible with the recursive analysis (Lemma B.2). Deterministic list limits bound the time and space of every execution; the probability that these limits discard a required vector is bounded separately (Sections 3.1.1 and 3.1.2).

This analysis supplies two guarantees. One guarantees that, for most fixed matrices $\mathbf{A}$, the procedure produces at least one solution with constant probability over $\boldsymbol{t}$ sampled uniformly and independently of $\mathbf{A}$ and over the procedure's internal randomness, provided that the number of ternary candidates exceeds $q^m$ by an exponential factor (Corollary 3.3). The other guarantees that, for any fixed candidate $\boldsymbol{v}$ independent of a uniform $\mathbf{A}$, the procedure includes $\boldsymbol{v}$ among its outputs on input $(\mathbf{A}, \mathbf{A}\boldsymbol{v} \bmod q)$ with high probability over $\mathbf{A}$ and the procedure's internal randomness (Lemma 3.2).

However, these guarantees leave a gap for each lattice problem. For $\gamma$-CVP, each target determines a specific value in the congruence, but the first guarantee provides only average success over an independent uniform value. The $\gamma$-CVP reduction must therefore convert this average guarantee into success at every required value. For SVP, the desired shortest vector depends on the input matrix, but the second guarantee applies to a candidate fixed independently of that matrix. The SVP reduction must therefore account for this dependence when using the second guarantee to find a shortest vector.

### 1.2.2 Reduction for $\gamma$-CVP

We show how to combine the short-congruence search with exact search to solve $\gamma$-CVP for arbitrary rational targets, where $\gamma > 1$ is a fixed rational approximation factor. We describe the main case $q \geq 3$ and $1/8 < \delta \leq 1$.

For a target $\boldsymbol{y} \in \mathbb{Q}^n$, we seek an integer vector $\boldsymbol{x}$ close to $\boldsymbol{y}$ satisfying $\mathbf{H}\boldsymbol{x} \equiv \mathbf{0}_m \pmod q$. If we temporarily omit the congruence, a closest integer vector is obtained by coordinatewise rounding: $\boldsymbol{z} = \lfloor \boldsymbol{y} \rceil$. This gives $|z_i - y_i| \leq 1/2$ for every coordinate; put $\boldsymbol{f} = \boldsymbol{z} - \boldsymbol{y} \in [-1/2, 1/2]^n$. However, $\boldsymbol{z}$

need not satisfy the congruence.

We therefore seek an integer vector $\boldsymbol{w}$ satisfying

$$\mathbf{H}\boldsymbol{w} \equiv -\mathbf{H}\boldsymbol{z} \pmod{q}.$$

Then $\boldsymbol{z} + \boldsymbol{w}$ belongs to $\Lambda$, and its distance from the target is $\|\boldsymbol{f} + \boldsymbol{w}\|$ (Figure 1). The reduction must both find such a vector $\boldsymbol{w}$ and control this distance. We search among vectors $\boldsymbol{w}$ supported on the first $d = \lceil m \log_2 q \rceil \le n$ coordinates.

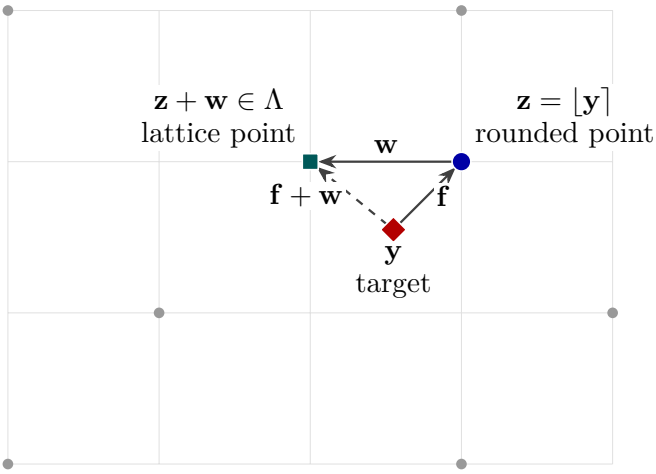


Figure 1: Illustration of the rounding and correction steps. The faint grid represents integer points, and the gray dots are points of an illustrative two-dimensional lattice. Rounding the target $\mathbf{y}$ gives $\mathbf{z}$; adding a correction $\mathbf{w}$ yields a lattice point $\mathbf{z} + \mathbf{w}$. The dashed arrow shows the final displacement $\mathbf{f} + \mathbf{w}$.

Two requirements guide this construction. The value $-\mathbf{H}\boldsymbol{z} \bmod q$ is determined by $\mathbf{H}$ and the target, so the average success guarantee for a uniform congruence value does not suffice. We also need a distance bound that makes the subsequent exact search sufficiently cheap. Fix a rational $0 < \beta < 1$ and consider integer vectors $\boldsymbol{w}$ supported on the first $d$ coordinates and satisfying $\mathbf{H}\boldsymbol{w} \equiv -\mathbf{H}\boldsymbol{z} \pmod{q}$ and $\|\boldsymbol{w}\|^2 \le \beta d$. Integer coordinates $w_i$ satisfy $|w_i| \le w_i^2$. This inequality and $|f_i| \le 1/2$ give $2\langle \boldsymbol{f}, \boldsymbol{w} \rangle \le \sum_{i=1}^{d} |w_i| \le \|\boldsymbol{w}\|^2$, so this norm bound alone yields squared distance at most $\|\boldsymbol{f}\|^2 + 2\beta d$. Averaging several such vectors can improve this bound. For any nonempty collection of these vectors, write $\overline{\boldsymbol{w}}$ for their average. Averaging their squared distances and using Cauchy–Schwarz gives

$$\min_{\boldsymbol{w}} \|\boldsymbol{f} + \boldsymbol{w}\|^2 \le \|\boldsymbol{f}\|^2 + \beta d + \sqrt{d}\,\|\overline{\boldsymbol{w}}\|,$$

where the minimum is over this collection. Here the factor $\sqrt{d}$ follows from the bound $\sqrt{d}/2$ on the norm of the first $d$ coordinates of $\boldsymbol{f}$. Thus, for a fixed rational $\beta < c < 2\beta$, it suffices to obtain $\|\overline{\boldsymbol{w}}\| \le (c - \beta)\sqrt{d}$: at least one vector in the collection then satisfies

$$\|\boldsymbol{z} + \boldsymbol{w} - \boldsymbol{y}\|^2 \le \|\boldsymbol{f}\|^2 + cd. \tag{1}$$

The algorithm retains the closest lattice point found; the average is used only in the analysis.

We next construct a randomized procedure that, given $\mathbf{H}$ and $\boldsymbol{s} \in (\mathbb{Z}/q\mathbb{Z})^m$, returns an integer vector $\boldsymbol{w}$ supported on the first $d$ coordinates and satisfying $\mathbf{H}\boldsymbol{w} \equiv \boldsymbol{s} \pmod{q}$ and $\|\boldsymbol{w}\|^2 \le \beta d$, or reports failure. An attempt is successful when it returns such a vector. For most fixed matrices $\mathbf{H}$, we require a constant success probability and a small mean conditioned on success for every $\boldsymbol{s}$. For

CVP, we use $\boldsymbol{s} = -\mathbf{H}\boldsymbol{z} \bmod q$. We reserve $\Theta(d^{2/3})$ of the first $d$ columns of $\mathbf{H}$ in two blocks and let $\mathbf{A}$ contain the remaining columns used by the common procedure. Fix a matrix $\mathbf{A}$ for which the average success guarantee holds. The reserved columns remain independent and uniform under this conditioning. Each call to the common procedure supplies its first output, or reports failure if it produces none.

First, for an input value $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, choose a uniform sign $\sigma \in \{-1, 1\}$, call the common procedure on $(\mathbf{A}, \sigma\boldsymbol{t})$, and return $\sigma\boldsymbol{v}$ if the call returns $\boldsymbol{v}$. For each fixed $\boldsymbol{t}$, consider the success probability and expected output of this call, averaging over the sign and the procedure's internal randomness and counting a failed output as zero. These two quantities may vary with $\boldsymbol{t}$. Their averages over uniform $\boldsymbol{t}$ are the original average success probability and zero, respectively.

We now apply this call to $\boldsymbol{s}$ minus the sum of a uniformly chosen subset of columns from the first reserved block. For a fixed block and a fixed $\boldsymbol{s}$, the resulting success probability and expected output are the averages of the two quantities above over the subset choice. The Erdős–Rényi second-moment identity [ER65] controls how far these subset averages are from their averages over uniform input values. Specifically, consider the squared deviation of the success probability from its average over uniform inputs, and the squared norm of the expected output, whose average is zero. Averaged over the random block and uniform $\boldsymbol{s}$, both quantities are reduced by a factor exponential in the number of columns in the block, compared with their corresponding averages before averaging over subsets. Markov's inequality first gives that most blocks have small average errors over $\boldsymbol{s}$. For each such fixed block, a second application gives constant success probability and a small expected output for all but a small fraction of values $\boldsymbol{s}$.

The second reserved block extends these bounds to every $\boldsymbol{s}$. For a fixed $\boldsymbol{s}$, subtracting each of its independent uniform columns gives independent uniform values to which the preceding construction applies. Concentration bounds the fraction of these values lying in the exceptional set, strongly enough to sum the failure probabilities over all $q^m$ choices of $\boldsymbol{s}$. Thus, for most choices of this block, subtracting a uniformly selected column gives constant success probability and a small output mean for every $\boldsymbol{s}$. To obtain a correction for the original congruence, we append the subset coefficients and the indicator vector of the selected column to the signed output. Their contributions under $\mathbf{H}$ restore the two subtracted shifts. The block sizes leave enough room in the norm bound for these appended coordinates, so the complete correction satisfies $\|\boldsymbol{w}\|^2 \le \beta d$. Their contribution to the norm of the mean is only $O(d^{1/3})$. For fixed $\mathbf{H}$ and $\boldsymbol{s}$, counting failed attempts as zero makes the expected vector equal to the success probability times the expected vector conditioned on success. The constant lower bound on the success probability therefore gives a small conditional mean. Consequently, for every fixed $\varepsilon > 0$ and sufficiently large $n$, all but a $2^{-\Omega(n^{2/3})}$ fraction of input matrices have, simultaneously for every $\boldsymbol{s} \in (\mathbb{Z}/q\mathbb{Z})^m$,

$$\Pr[\text{success}] = \Omega(1), \qquad \big\|\mathbb{E}[\boldsymbol{w} \mid \text{success}]\big\| \le \varepsilon\sqrt{d}. \tag{2}$$

Probability and expectation here concern one attempt with $\mathbf{H}$ and $\boldsymbol{s}$ fixed (Lemmas 4.4 and 4.5). Figure 2 shows how the reserved columns and their coefficients form the correction.

To turn this conditional-mean bound into the sample-mean bound required by (1), fix such a matrix and choose $\varepsilon$ sufficiently small relative to $c - \beta$. A sufficiently large constant multiple of $n$ independent attempts gives enough returned vectors whose average has norm at most $(c - \beta)\sqrt{d}$, except with exponentially small probability. This follows from concentration for the number of successes and for the bounded vectors returned by successful attempts. Use the same random bits in corresponding attempts for each $\boldsymbol{s}$ and sum the failure probabilities over its $q^m$ possible values. Since $q^m \le 2^n$, the repetition constant can be chosen so that this event holds for all values with probability $1 - 2^{-\Omega(n)}$. Targets with the same value of $-\mathbf{H}\lfloor\boldsymbol{y}\rceil \bmod q$ use the same correction samples, and the

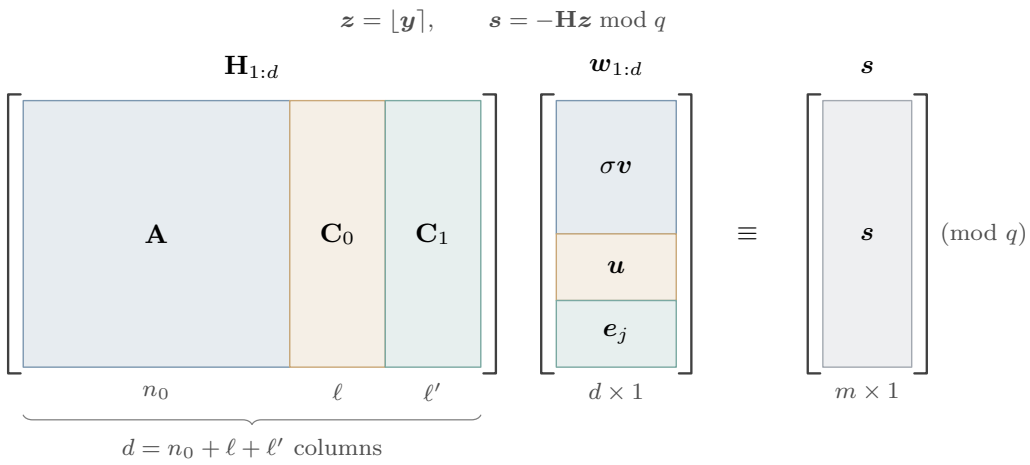


Figure 2: $\gamma$-CVP correction construction. The first $d = n_0 + \ell + \ell'$ columns of $\mathbf{H}$ form $[\mathbf{A}\ \mathbf{C}_0\ \mathbf{C}_1]$, with respective widths $n_0, \ell, \ell'$. The vector $\boldsymbol{u}$ contains the binary coefficients selecting a subset of columns of $\mathbf{C}_0$, and $\boldsymbol{e}_j$ is the $j$th unit vector, selecting one column of $\mathbf{C}_1$. The correction $\boldsymbol{w}$ is zero outside its first $d$ coordinates.

distance bound holds for every compatible rounding error $\boldsymbol{f}$. Hence one choice of random bits gives the required candidate for every rational target (Lemma 4.6).

It remains to obtain a relative approximation. The algorithm first performs exact search in the full $n$-dimensional lattice, within the ball around $\boldsymbol{y}$ of squared radius

$$b = (\|\boldsymbol{f}\|^2 + cd)/\gamma^2.$$

If this ball contains a lattice point, the search returns a closest one. Otherwise, the optimal squared distance exceeds $b$, while (1) gives a correction candidate at squared distance at most $\gamma^2 b$. This proves the $\gamma$-approximation (Theorem 4.7).

Since $c < 2\beta$, the mean bound reduces the exact-search radius below the one given by the correction-norm bound alone, thereby reducing the cost of that stage. The shifted theta-function bound controls the rounding error's contribution to this search cost. There are only $O(n)$ correction attempts and their storage is reused, so the leading time and space exponents are the larger of those for correction generation and exact search. At $\gamma = 5/3$ and density one, the certified choices make exact search cheaper than correction generation (Corollary 4.8 and Section 4.4). Binary and low-density inputs use separate procedures (Algorithm 3 and Lemmas 4.1 and 4.2); Corollary 4.11 treats $1 < \delta \leq 3/2$.

### 1.2.3 Reduction for SVP

We show how the guarantee for a ternary vector fixed independently of its input matrix yields an algorithm for finding a shortest nonzero vector of $\Lambda$, even though a shortest vector depends on $\mathbf{H}$. The reduction must also accommodate coordinates outside $\{-1, 0, 1\}$ and unequal numbers of 1 and

$-1$ entries. We describe the main search branch, where $q \geq 3$, $q^4 > n$, and $\delta \leq 1$ is bounded below by a positive constant.

We first restrict the search to a ball that contains a shortest vector for most input matrices and whose integer points can be counted independently of $\mathbf{H}$. Write $\boldsymbol{e}_i$ for the $i$th coordinate unit vector. The algorithm retains $q\boldsymbol{e}_1 \in \Lambda$, so it suffices to find a shortest vector of length less than $q$ whenever one exists. Counting integer vectors determines a ball centered at zero with squared radius $O(n)$, depending only on $n, m, q$. A second-moment bound shows that this ball contains a nonzero lattice vector for all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices. It therefore contains a shortest vector for those matrices. Its total number of integer points is at most $q^m 2^{n^{2/3}+O((\log n)^2)}$; this bound will control the sum of failure probabilities over candidates (Lemma 5.3 and Eq. (58)).

Counting the coordinates of vectors in this ball further restricts what the algorithm must search. For most matrices, every nonzero lattice vector in the ball of length less than $q$ has a coordinate equal to 1 or $-1$. Such an entry makes the lattice congruence uniquely determine one matrix column from the others; we use this fact below to analyze the search conditioned on lattice membership. The same analysis gives bounds, determined by $n, m, q$, on the numbers of entries equal to 1 and to $-1$ and on the possible positions and values of coordinates with absolute value at least two (Lemma 5.5). These bounds hold simultaneously for all such lattice vectors, and hence for a shortest vector chosen after $\mathbf{H}$ is known. We use them to enumerate the coordinates of absolute value at least two and the sign counts; the short-solution procedure finds the remaining ternary coordinates.

For each allowed choice, let $\boldsymbol{z} \in \mathbb{Z}^n$ record the specified coordinates of absolute value at least two, with zeros elsewhere. Choose $i \notin \operatorname{supp}(\boldsymbol{z})$ and $\sigma \in \{-1, 1\}$ for one further coordinate, and let $N_+, N_-$ be the specified numbers of remaining 1 and $-1$ entries. Write $\mathcal{I} = [n] \setminus (\operatorname{supp}(\boldsymbol{z}) \cup \{i\})$ for the positions still to be filled. The required counts $N_+, N_-$ need not be equal, whereas the common procedure requires equal counts. To meet this requirement, we add coordinates with prescribed values and independent uniform columns to the matrix. Let $n_0 > n$, with $n_0 = O(n)$, be the enlarged dimension and $a_0$ the required number of entries of each nonzero sign. Choose $\boldsymbol{d} \in \{-1, 0, 1\}^{n_0 - |\mathcal{I}|}$ with $a_0 - N_+$ ones and $a_0 - N_-$ minus ones, and all other entries zero. We choose $n_0$ and $a_0$ so that these counts are nonnegative and fit in the added positions for every allowed choice. Sample an independent uniform matrix $\mathbf{D} \in (\mathbb{Z}/q\mathbb{Z})^{m \times (n_0 - |\mathcal{I}|)}$ and set

$$\mathbf{A} = [\mathbf{H}_{\mathcal{I}}\ \mathbf{D}], \qquad \boldsymbol{t} = \left(-\mathbf{H}(\boldsymbol{z} + \sigma\boldsymbol{e}_i) + \mathbf{D}\boldsymbol{d}\right) \bmod q.$$

Apply the common procedure to $(\mathbf{A}, \boldsymbol{t})$ and inspect its outputs. If an output has the prescribed added coordinates $\boldsymbol{d}$, place its first $|\mathcal{I}|$ entries in the positions $\mathcal{I}$ and restore the coordinates specified by $\boldsymbol{z}$ and $\sigma\boldsymbol{e}_i$. The resulting vector $\boldsymbol{x}'$ satisfies $\mathbf{H}\boldsymbol{x}' \equiv \mathbf{0}_m \pmod q$, because the term $\mathbf{D}\boldsymbol{d}$ cancels. Its $i$th coordinate is $\sigma$, so it is nonzero. Removing the added coordinates leaves exactly $N_+$ ones and $N_-$ minus ones, giving $\|\boldsymbol{x}'\|^2 = 1 + \|\boldsymbol{z}\|^2 + N_+ + N_-$. Thus each accepted output has the length specified by this choice of coordinates and counts.

We now justify why this search finds a vector as short as a given candidate when that candidate belongs to the lattice. In the analysis, fix an integer vector $\boldsymbol{x}$ in the ball, of length less than $q$ and satisfying the coordinate restrictions above, independently of $\mathbf{H}$. Consider the enumerated choice matching its coordinates of absolute value at least two, a coordinate $x_i = \sigma$, and its remaining sign counts. Conditioning on $\mathbf{H}\boldsymbol{x} \equiv \mathbf{0}_m \pmod q$ gives

$$\mathbf{H}\boldsymbol{e}_i \equiv -\sigma \sum_{j \neq i} x_j \mathbf{H}\boldsymbol{e}_j \pmod q.$$

For every assignment of the other columns, exactly one of the $q^m$ possible values of column $i$ satisfies this congruence. Consequently, $\Pr[\mathbf{H}\boldsymbol{x} \equiv \mathbf{0}_m \pmod q] = q^{-m}$. Moreover, every assignment of the

other columns has the same probability of satisfying the condition, so their joint distribution remains independent and uniform after conditioning. This uses only the invertibility of $\sigma = \pm 1$ and holds for composite $q$ as well.

Under this conditioning, $\mathbf{A}$ is uniform, since its original columns exclude column $i$ and its added columns are independent and uniform. The vector $\boldsymbol{v}$ formed by appending $\boldsymbol{d}$ to $\boldsymbol{x}_{\mathcal{I}}$ is fixed independently of $\mathbf{A}$, has $a_0$ entries of each nonzero sign, and satisfies $\mathbf{A}\boldsymbol{v} \equiv \boldsymbol{t} \pmod{q}$. The fixed-vector guarantee therefore applies to this call. With high probability, $\boldsymbol{v}$ appears among the outputs, passes the added-coordinate check, and reconstructs $\boldsymbol{x}$ (Lemma 5.6). This probability is over $\mathbf{H}$ conditioned on $\mathbf{H}\boldsymbol{x} \equiv \mathbf{0}_m \pmod{q}$ and the independent randomness of the call. The guarantee concerns inclusion among the outputs, so the algorithm continues inspecting the same call's outputs after one fails the added-coordinate check. Figure 3 illustrates the construction and the conditional distribution used in its analysis.

To apply this argument to a shortest vector, we must account for all candidates that could become shortest as $\mathbf{H}$ varies. For each fixed candidate, bound the probability that it belongs to $\Lambda$ and the corresponding call fails to find a vector at least as short. This is $q^{-m}$ times the conditional failure probability. The list limits in the common procedure can be chosen so that the latter is at most $2^{-2n^{2/3}}$ for sufficiently large $n$. Summing over the integer candidates satisfying the coordinate restrictions, whose number is bounded by the total number of points in the ball, gives at most

$$(q^m 2^{n^{2/3}+O((\log n)^2)})q^{-m}2^{-2n^{2/3}} = 2^{-\Omega(n^{2/3})}.$$

Thus, except with this probability, every such candidate belonging to the lattice has a corresponding call that produces a vector no longer than it. On the radius and coordinate events established above, this applies to a shortest vector unless $q\boldsymbol{e}_1$ is already shortest.

One round consists of searching all allowed coordinate choices and retaining the shortest nonzero lattice vector found. Including the exceptions in the radius and coordinate bounds, its failure probability is $2^{-\Omega(n^{2/3})}$ over $\mathbf{H}$ and the round's randomness. Markov's inequality implies that all but a $2^{-\Omega(n^{2/3})}$ fraction of fixed matrices have one-round failure probability at most $1/4$. For each such matrix, $n$ independent rounds all fail with probability at most $4^{-n}$ (Theorem 5.7).

The cost comes from enumerating the allowed positions and values of coordinates of absolute value at least two and making one short-solution call for each choice of these coordinates, $(i, \sigma)$, and $(N_+, N_-)$. The choices of $i, \sigma, N_+, N_-$ and the $n$ rounds contribute only polynomial factors. Hence the number of allowed vectors $\boldsymbol{z}$ multiplies the time for one call in dimension $n_0$. Calls are performed successively, reusing the memory required by one execution of the short-solution procedure. The current coordinate choice and the shortest vector found so far require only polynomial additional space, so the number of choices does not multiply the space bound. The increase from $n$ to $n_0$ is included in both exponents (Theorem 5.8). At lower densities, exact search in the same ball gives better bounds. For $q = 2$ or $q^4 \leq n$, enumerating all nonzero integer vectors of norm at most $q$ suffices because $q\boldsymbol{e}_1 \in \Lambda$. For $1 < \delta \leq 3/2$, certified parameter choices give the stated bounds using the same reduction (Theorem 5.9).

### 1.3 Related work

**Algorithms for CVP and SVP.** Approximation factors, preprocessing costs, and input distributions determine how other lattice algorithms compare with the bounds in Table 1. Aggarwal and Stephens-Davidowitz [ASD18] give a direct Euclidean CVP algorithm using $2^{n+o(n)}$ time and space for every fixed approximation factor greater than one, and for factors approaching one. Eisenbrand and Venzin [EV22] and Rothvoss and Venzin [RV22] give constant-factor algorithms for CVP and

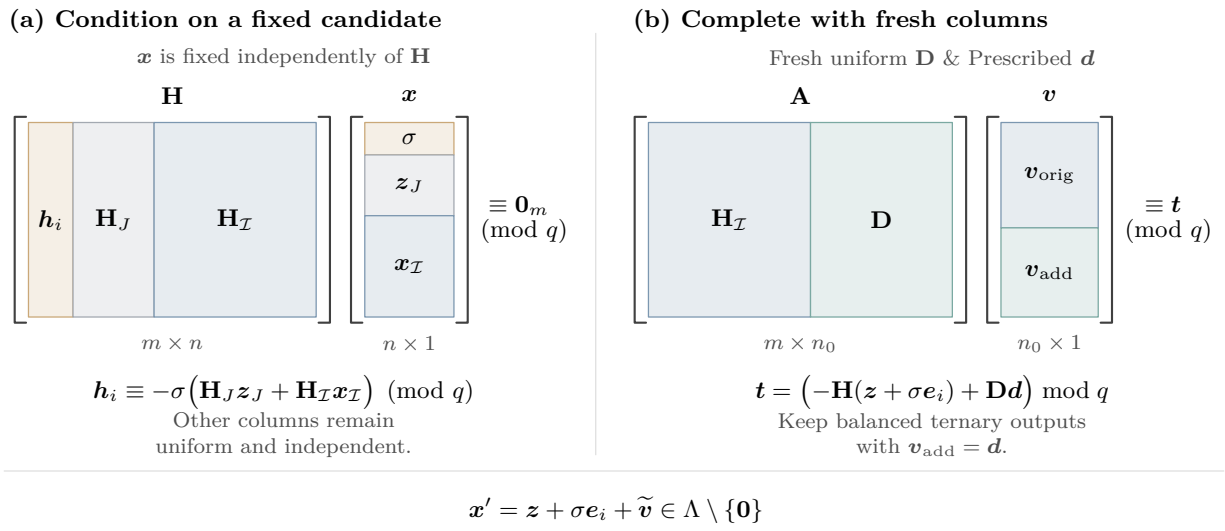


Figure 3: SVP conditioning and completion. For a fixed candidate $\boldsymbol{x}$ with $x_i = \sigma \in \{-1, 1\}$, let $J = \{j : |x_j| \geq 2\}$ and $\mathcal{I} = [n] \setminus (J \cup \{i\})$. Here $\boldsymbol{h}_i = \mathbf{H}\boldsymbol{e}_i$, and $\boldsymbol{z}$ agrees with $\boldsymbol{x}$ on $J$ and is zero elsewhere. The added columns $\mathbf{D}$ have prescribed coordinates $\boldsymbol{d}$; $\widetilde{\boldsymbol{v}}$ embeds the original-coordinate part $\boldsymbol{v}_{\text{orig}}$ in $\mathcal{I}$, with zeros elsewhere.

SVP in $\ell_p$ norms ($1 \leq p \leq \infty$) and arbitrary norms, respectively. Their time bound $2^{(0.802+\varepsilon)n}$, for each fixed $\varepsilon > 0$, allows a sufficiently large constant approximation factor depending on $\varepsilon$.

Ducas, Laarhoven, and van Woerden [DLvW20] give heuristic tradeoffs between query time and storage for CVP after lattice-dependent preprocessing. These query costs exclude preprocessing, which must be included in a comparison of the total cost for one target. Doulgerakis and Laarhoven [DL26] study a different input distribution: Haar-random determinant-one lattices. They solve CVP for random targets and SVP in this model. The distribution of lattices and the restriction to random CVP targets are both part of these guarantees; sampling a uniform congruence matrix specifies a different input distribution.

**Representation methods and their analysis.** The recursive search in our common procedure builds on existing representation methods and their rigorous analysis for random subset sum. Howgrave-Graham and Joux [HGJ10] introduced the method for subset sum, and Becker, Coron, and Joux [BCJ11] extended it using ternary decompositions. Ozerov [Oze16, Algorithm 9 and Theorem 43] analyzes a recursive construction with bounds on intermediate lists, proving running-time and success guarantees, including inclusion of a fixed solution among the outputs. We follow this recursive construction; the analysis for systems of congruences and the space bounds for ordered generation used here are given in Section 3.1 and Appendix B.4. Representation methods have also been applied to lattice problems: Becker, Gama, and Joux [BGJ14] search for CVP and SVP solutions through overlattices, with a complexity analysis based on the Gaussian heuristic.

**Short solutions to random linear congruences.** For short-congruence search, the distribution of the congruence value and the permitted output weight are part of the problem specification. Chailloux, Debris-Alazard, and Etinski [CDAE21] analyze information-set decoding over prime fields with additive weight functions, where a vector's weight is the sum of its coordinate weights. They sample a uniform full-row-rank matrix and, independently, a uniform vector of prescribed

weight, then take its image under the matrix as the value in the congruence. Ducas, Engelberts, and Loyer [DEL25] analyze a randomized variant of Wagner's generalized-birthday algorithm through discrete Gaussian sampling; its permitted norm regime is compared with our short-preimage bound in Section 1.1. Bollauf, Pouly, and Shen [BPS26] obtain SIS solutions by discrete Gaussian sampling for norm bounds below $q/2$, and prove bounds on the probability of obtaining a nonzero vector within the required norm bound. Their concrete estimates of total cost additionally use models for basis reduction and sampling, so the proved sampling guarantee and the estimated cost of generating the samples are separate parts of the analysis.

**Preimage attacks on GPV-type signatures.** Concrete preimage attacks optimize lattice reduction and search as a function of the verification bound as well as the public-key parameters. Ducas, Espitau, and Postlethwaite [DEP23] exploit the geometry of BKZ-reduced $q$-ary bases at small moduli, and Abapour [Aba26] refines their model of sieve outputs and attack success. These cost estimates use models for lattice reduction and sieving. The representation-based ISIS algorithms of Bai, Galbraith, Li, and Sheffield [BGLS19] provide another approach, with heuristic analyses of the combinatorial search. A comparison with the bounds in Section 1.1 must use the same public-key distribution and parameters $(m, n, q, B)$, and optimize each attack's parameters: the verification bound $B$ affects the reduction required and the probability that the search produces an accepted preimage.

# 2 Preliminaries

We fix the notation and conventions for computation costs, and recall the probability tools.

## 2.1 Notation

Matrices use bold uppercase letters and column vectors bold lowercase letters; $\top$ denotes transpose. For integers $k \geq 0$, $[k] = \{1, \ldots, k\}$, with $[0] = \varnothing$. The zero vector is $\mathbf{0}_k \in \mathbb{Z}^k$. A *list* over a set $\mathcal{X}$ is a function $\mathcal{L} : [k] \to \mathcal{X}$ for an integer $k \geq 0$; its length is $|\mathcal{L}| = k$. For $x \in \mathcal{X}$, $x \in \mathcal{L}$ means $x = \mathcal{L}(i)$ for some $i \in [k]$. The case $k = 0$ is the empty list, denoted $[\,]$. For an integer $n \geq 1$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^n x_i y_i$, $\|\boldsymbol{x}\| = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$. We write $\boldsymbol{x}[j] = x_j$ for $j \in [n]$. For $\mathcal{I} \subseteq [n]$, $\boldsymbol{x}_\mathcal{I} = (x_i)_{i \in \mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ and $\mathbb{Z}^\mathcal{I} = \{(z_i)_{i \in \mathcal{I}} : z_i \in \mathbb{Z}\}$, with indices in increasing order. For integers $q \geq 2$, $m \geq 0$, and $n \geq 1$, let $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$. Write $H_{ij} \in \mathbb{Z}/q\mathbb{Z}$ for its entry in row $i \in [m]$ and column $j \in [n]$, and $\boldsymbol{e}_j$ for the $j$th standard unit vector, of the dimension required by the context; thus $\mathbf{H}\boldsymbol{e}_j$ is column $j$ of $\mathbf{H}$. Its submatrices are

$$\mathbf{H}_\mathcal{I} = [\mathbf{H}\boldsymbol{e}_i]_{i \in \mathcal{I}} \in (\mathbb{Z}/q\mathbb{Z})^{m \times |\mathcal{I}|} \quad (\mathcal{I} \subseteq [n]), \qquad \mathbf{H}_{a:b} = \mathbf{H}_{\{a,\ldots,b\}} \quad (a, b \in [n],\ a \leq b).$$

For $y \in \mathbb{Q}$, set $\lfloor y \rceil = \lfloor y + 1/2 \rfloor$. For $\boldsymbol{y} \in \mathbb{Q}^n$, rounding is coordinatewise: $\lfloor \boldsymbol{y} \rceil = (\lfloor y_i \rceil)_{i=1}^n \in \mathbb{Z}^n$, with $\lfloor \boldsymbol{y} \rceil - \boldsymbol{y} \in [-1/2, 1/2]^n$. $\perp$ denotes failure. We use log and ln to denote the base-2 and natural logarithms, respectively. For $a \in \mathbb{R}$, write $[a]_+ = \max\{a, 0\}$.

We write $\Pr[\mathcal{E}]$ for the probability of an event $\mathcal{E}$ and $\mathbb{E}[X]$ for the expectation of an integrable real random variable $X$. For random vectors, expectation is taken coordinatewise. Subscripts on Pr and $\mathbb{E}$ indicate which random choices the probability or expectation is taken over. The arrow $\xleftarrow{\$}$ denotes uniform sampling from a nonempty finite set.

Unless stated otherwise, asymptotic notation refers to $n \to \infty$. For functions $f : \mathbb{Z}_{\geq 1} \to \mathbb{R}$ and $g : \mathbb{Z}_{\geq 1} \to (0, \infty)$, we write $f(n) = O(g(n))$ if $|f(n)| \leq Cg(n)$ for some constant $C > 0$ and all

sufficiently large $n$, and $f(n) = \Omega(g(n))$ if $f(n) \geq cg(n)$ for some constant $c > 0$ and all sufficiently large $n$. We write $f(n) = \Theta(g(n))$ if both bounds hold, and $f(n) = o(g(n))$ if $f(n)/g(n) \to 0$. Implicit constants may depend on parameters explicitly held fixed.

Integers and rationals are encoded in binary, and arithmetic on them is exact. Time is measured in bit operations and space in bits. The notation $\mathrm{poly}(\cdot)$ denotes a polynomial factor in its arguments. Unless specified otherwise, poly denotes a polynomial factor in the binary input length; such factors are omitted from exponential time and space bounds.

## 2.2 Probability tools

We recall Shannon entropy and total variation distance for discrete probability distributions. For a distribution $P$, $P(x)$ is the probability that a sample from $P$ equals $x$.

**Definition 2.1** (Shannon entropy)**.** For a distribution $P$ on a countable set $\mathcal{X}$, define $\mathcal{H}(P) = -\sum_{x \in \mathcal{X}} P(x) \log P(x)$, with $0 \log 0 = 0$. For a finite distribution with probabilities $p_1, \ldots, p_k$, we also write $\mathcal{H}(p_1, \ldots, p_k)$. The binary entropy function is $h(t) = \mathcal{H}(t, 1-t)$ for $t \in [0, 1]$.

**Definition 2.2** (Total variation distance)**.** For distributions $P, Q$ on a finite set $\mathcal{X}$, $\mathrm{TV}(P, Q) = \frac{1}{2} \sum_{x \in \mathcal{X}} |P(x) - Q(x)|$.

We recall several inequalities for bounding probabilities.

**Lemma 2.3** (Markov's inequality)**.** *If $X$ is a nonnegative random variable and $a > 0$, then*

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$

*Proof.* The inequality follows from $\mathbb{E}[X] \geq a \Pr[X \geq a]$. □

**Lemma 2.4** (Chernoff bound)**.** *For an integer $\ell \geq 1$, let $X_1, \ldots, X_\ell$ be independent scalar random variables taking values in $\{0, 1\}$. Put $X = \sum_{j=1}^{\ell} X_j$ and $\mu = \mathbb{E}[X]$. Then, for $0 < \tau < 1$,*

$$\Pr[X \leq (1-\tau)\mu] \leq e^{-\tau^2 \mu / 2}.$$

*Proof.* Independence and $1 + u \leq e^u$ give $\mathbb{E}\left[e^{-tX}\right] \leq \exp(\mu(e^{-t} - 1))$ for $t > 0$. Apply Lemma 2.3 to $e^{-tX}$ with $t = -\ln(1-\tau)$, and use $-(1-\tau)\ln(1-\tau) - \tau \leq -\tau^2/2$. □

**Lemma 2.5.** *Let $\ell, d \geq 1$ be integers and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell \in \mathbb{R}^d$ be independent random column vectors with $\mathbb{E}[\boldsymbol{y}_i] = \mathbf{0}_d$ and $\|\boldsymbol{y}_i\| \leq b$ almost surely, where $b > 0$ is a scalar. For every $t \geq 0$,*

$$\Pr\left[\Big\|\sum_{i=1}^{\ell} \boldsymbol{y}_i\Big\| \geq b\sqrt{\ell} + t\right] \leq \exp\left(-\frac{t^2}{2\ell b^2}\right).$$

*Proof.* Independence and the zero means imply

$$\mathbb{E}\left[\Big\|\sum_i \boldsymbol{y}_i\Big\|\right] \leq \left(\mathbb{E}\left[\Big\|\sum_i \boldsymbol{y}_i\Big\|^2\right]\right)^{1/2} = \left(\sum_i \mathbb{E}\left[\|\boldsymbol{y}_i\|^2\right]\right)^{1/2} \leq b\sqrt{\ell}.$$

Changing one $\boldsymbol{y}_i$ can change $\|\sum_i \boldsymbol{y}_i\|$ by at most $2b$. The bounded-differences inequality of McDiarmid [McD89] therefore bounds a deviation $t$ above the mean by $\exp(-2t^2/(\ell(2b)^2))$. □

### 2.3 Lattices and computational problems

Let $n \geq 1$, $q \geq 2$, and $m \geq 0$ be integers. For a matrix $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, we write

$$\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^n : \mathbf{H}\boldsymbol{x} \equiv \mathbf{0}_m \pmod{q}\}$$

for the associated $q$-ary lattice.[5] We use the Euclidean norm throughout. In the problems below, the lattice is given by the matrix $\mathbf{H}$.

**Definition 2.6** ($\gamma$-CVP)**.** Fix a rational approximation factor $\gamma > 1$. Given a matrix $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ and a target $\boldsymbol{y} \in \mathbb{Q}^n$, return a vector $\boldsymbol{x} \in \Lambda$ satisfying

$$\|\boldsymbol{x} - \boldsymbol{y}\| \leq \gamma \min_{\boldsymbol{a} \in \Lambda} \|\boldsymbol{a} - \boldsymbol{y}\|.$$

**Definition 2.7** (SVP)**.** Given a matrix $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, return a vector $\boldsymbol{x} \in \Lambda \setminus \{\mathbf{0}_n\}$ satisfying

$$\|\boldsymbol{x}\| = \min_{\boldsymbol{a} \in \Lambda \setminus \{\mathbf{0}_n\}} \|\boldsymbol{a}\|.$$

**Random input model.** We consider the above problems when the entries of $\mathbf{H}$ are sampled independently and uniformly from $\mathbb{Z}/q\mathbb{Z}$. This induces a distribution on the associated $q$-ary lattice $\Lambda$. For $\gamma$-CVP, the target $\boldsymbol{y} \in \mathbb{Q}^n$ is arbitrary and may be chosen after $\mathbf{H}$ is known.

## 3 Common Algorithms for Linear Congruences

We develop two procedures for finding integer solutions to linear congruences. The randomized procedure in Section 3.1 produces ternary vectors of prescribed length, used to construct candidate lattice vectors in SVP and correction vectors in $\gamma$-CVP. The deterministic procedure in Section 3.2 finds a closest solution inside a prescribed ball or certifies that it contains no such solution. $\gamma$-CVP uses this exact search to handle nearby targets and obtain a lower bound on the distance when the search fails; SVP uses its variant restricted to nonzero vectors.

### 3.1 Finding short solutions

We first consider the problem of finding short integer solutions to linear congruences. We focus on ternary vectors containing a prescribed number of entries equal to 1 and the same number of entries equal to $-1$. This restriction fixes their Euclidean norm.

**Definition 3.1** (Short ternary solutions)**.** Let $n, m \geq 1$ and $q \geq 3$ be integers. Given $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, and an integer $a_0 \geq 0$ with $2a_0 \leq n$, output a list of distinct vectors $\boldsymbol{x} \in \{-1, 0, 1\}^n$ satisfying

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{t}, \qquad |\{j \in [n] : x_j = 1\}| = |\{j \in [n] : x_j = -1\}| = a_0 \ .$$

The procedure is the representation technique of Howgrave-Graham and Joux [HGJ10], in Ozerov's recursive formulation and analysis [Oze16, Algorithm 9 and Theorem 43], with opposite signs on zero coordinates as in Becker, Coron, and Joux [BCJ11, Section 3.1]. It generates, level by

[5] The set $\Lambda$ is a subgroup of $\mathbb{Z}^n$ containing $q\mathbb{Z}^n$. Every subgroup of $\mathbb{Z}^n$ is free abelian of rank at most $n$, and the inclusion $q\mathbb{Z}^n \subseteq \Lambda$ forces its rank to be $n$. Hence $\Lambda = \mathbf{B}\mathbb{Z}^n$ for some nonsingular $\mathbf{B} \in \mathbb{Z}^{n \times n}$, whose columns form a lattice basis.

level, vectors with prescribed sign counts that satisfy prescribed congruences, and combines pairs of them into vectors satisfying stronger congruences, up to the input congruence.

The analysis for a fixed solution and the restrictions on list sizes follow this framework. Here we also account for ternary vectors over even moduli, moduli growing with the dimension, and sequential generation of the last-level vectors within the stated space bound.

Definition 3.1 specifies the valid entries of the output list. An empty list is allowed and does not certify that no solution exists. Under the parameter conditions stated below, we prove two probability bounds. For SVP, Lemma 3.2 shows that a prescribed vector $\boldsymbol{x}$ with the required sign counts, fixed independently of a uniform $\mathbf{A}$, is among the outputs on input $(\mathbf{A}, \mathbf{A}\boldsymbol{x})$ with high probability over the matrix and the internal randomness. For $\gamma$-CVP, Corollary 3.3 gives, under additional parameter conditions, constant success probability for most fixed matrices, averaged over an independent uniform right-hand side and the internal randomness. Section 3.1.2 bounds the time and space of every execution, including unsuccessful ones.

**Candidate sets and decompositions.** Fix an integer $k \geq 1$, independent of $n$, and number the levels $0, \ldots, k$; level zero contains the outputs. Choose rational constants $0 < t_i < 1/2$ for $0 \leq i \leq k$ subject to

$$\frac{t_{i-1}}{2} \leq t_i \leq \frac{1 - t_{i-1}}{2} \qquad (1 \leq i \leq k), \tag{3}$$

and make the sign counts integral by setting

$$a_0 = 2^k \left\lfloor \frac{t_0 n}{2^k} \right\rfloor, \qquad a_i = \frac{a_{i-1}}{2} + 2^{k-i} \left\lfloor \frac{(t_i - t_{i-1}/2)n}{2^{k-i}} \right\rfloor \quad (1 \leq i \leq k),$$

so that $a_i/n$ approaches $t_i$. Write $a'_i = a_i - a_{i-1}/2$ for $1 \leq i \leq k$. At level $i$, let $\mathcal{T}_i \subseteq \{-1, 0, 1\}^n$ consist of the vectors with exactly $a_i$ entries of each nonzero sign. We write $\mathcal{T} = \mathcal{T}_0$ and $\beta = 2t_0$; every vector in $\mathcal{T}$ has squared length $2a_0 \leq \beta n$, and the procedure solves Definition 3.1 with this $a_0$.

For a parent $\boldsymbol{x} \in \mathcal{T}_{i-1}$, a decomposition is a pair with

$$\boldsymbol{x} = \boldsymbol{p} + \boldsymbol{u}, \qquad \boldsymbol{p}, \boldsymbol{u} \in \mathcal{T}_i \qquad (1 \leq i \leq k).$$

The coordinatewise possibilities are

$$\begin{aligned} 1 &= 1 + 0 = 0 + 1, \\ -1 &= (-1) + 0 = 0 + (-1), \\ 0 &= 0 + 0 = 1 + (-1) = (-1) + 1. \end{aligned}$$

The decomposition rule selects those decompositions in which each child receives half of the parent's positive entries and half of its negative entries, and in which, on the parent's zero coordinates, $a'_i$ positions carry $(p_j, u_j) = (1, -1)$, $a'_i$ other positions carry $(p_j, u_j) = (-1, 1)$, and the remaining positions carry $(0, 0)$. The lower bound in (3) allows each child to receive half of each nonzero sign, and the upper bound leaves enough zero coordinates for the cancelling signs: these choices are possible because $0 \leq 2a'_i \leq n - 2a_{i-1}$. For example,

$$\begin{aligned} (1, 1, -1, -1, 0, 0) &= (1, 0, -1, 0, 1, -1) + (0, 1, 0, -1, -1, 1) \\ &= (1, 0, -1, 0, -1, 1) + (0, 1, 0, -1, 1, -1). \end{aligned}$$

Each child has two entries of each nonzero sign. Changing the cancelling signs on the last two coordinates preserves the parent; distributing its nonzero entries differently gives further decompositions.

For $\boldsymbol{x} \in \mathcal{T}_{i-1}$, let $\mathcal{D}_i(\boldsymbol{x})$ be the set of first children $\boldsymbol{p}$ obtained by this rule; the second child is then uniquely $\boldsymbol{x} - \boldsymbol{p}$.

Decompositions are a device of the analysis. The algorithm generates vectors of $\mathcal{T}_i$ satisfying prescribed congruences and combines pairs whose sum lies in $\mathcal{T}_{i-1}$, including pairs outside the rule above. The multiplicity of decompositions is what permits restricting the children by congruences; the analysis in Section 3.1.1 uses only the counted subset $\mathcal{D}_i(\boldsymbol{x})$.

The algorithm uses the sizes

$$|\mathcal{T}_i| = \frac{n!}{a_i!^2(n-2a_i)!}, \qquad |\mathcal{D}_i(\boldsymbol{x})| = \binom{a_{i-1}}{a_{i-1}/2}^2 \frac{(n-2a_{i-1})!}{a_i'!^2(n-2a_{i-1}-2a_i')!}.$$

The second count is independent of $\boldsymbol{x}$: the two binomial factors split its positive and negative coordinates, and the remaining factor assigns cancelling signs on its zero coordinates. Their exponential rates are

$$\begin{aligned} h_i &= \mathcal{H}(t_i, t_i, 1-2t_i) && (0 \le i \le k), \\ r_i &= 2t_{i-1} + (1-2t_{i-1})\,\mathcal{H}\Big(\frac{t_i - t_{i-1}/2}{1-2t_{i-1}}, \frac{t_i - t_{i-1}/2}{1-2t_{i-1}}, \frac{1-t_{i-1}-2t_i}{1-2t_{i-1}}\Big) && (1 \le i \le k), \end{aligned} \tag{4}$$

so that $|\mathcal{T}_i| = 2^{h_i n + O(\log n)}$ and $|\mathcal{D}_i(\boldsymbol{x})| = 2^{r_i n + O(\log n)}$.

**Congruences and right-hand sides.** Let $\mathcal{G}_i$ be finite products of cyclic groups and $\mathbf{A}_i : \mathbb{Z}^n \to \mathcal{G}_i$ be homomorphisms for $0 \le i \le k$. The congruences become weaker down the levels: for surjective homomorphisms $\pi_{i-1} : \mathcal{G}_{i-1} \to \mathcal{G}_i$, require

$$\mathbf{A}_i = \pi_{i-1} \circ \mathbf{A}_{i-1} \qquad (1 \le i \le k).$$

Let $\rho > 0$ be fixed, with input dimensions satisfying $m \log q = \rho n + o(n)$, and choose a rational constant $\eta > 0$. Put

$$d_0 = \rho, \qquad d_i = r_i - \eta \quad (1 \le i \le k), \qquad \log|\mathcal{G}_i| = d_i n + o(n),$$

and require

$$\rho > d_1 > \cdots > d_k > 0. \tag{5}$$

These inequalities, with fixed positive margins, allow each child's congruence to retain less information than its parent's. The margin $\eta$ below the decomposition rate $r_i$ is what leaves decompositions of a parent after its children have been restricted; Section 3.1.1 quantifies this.

For $q^{\lceil \log(n+2) \rceil} \le 2^n$, so that $\log q = o(n)$, obtain the maps by retaining rows of $\mathbf{A}$. Put $m_0 = m$ and

$$m_i = \max\Big\{ r \in \mathbb{Z}_{\ge 0} : q^r 2^{\lceil \eta n \rceil} \le |\mathcal{D}_i(\boldsymbol{x})| \Big\} \qquad (1 \le i \le k). \tag{6}$$

Then $m_i \log q = d_i n + o(n)$. Take $\mathcal{G}_i = (\mathbb{Z}/q\mathbb{Z})^{m_i}$ and let $\mathbf{A}_i$ consist of the last $m_i$ rows of $\mathbf{A}$; in particular, $\mathbf{A}_0 = \mathbf{A}$. The map $\pi_{i-1}$ drops the first $m_{i-1} - m_i$ coordinates. For larger moduli, Appendix B.5 constructs the groups and maps from base-three digits of $\mathbf{A}$ reduced modulo a power of three.

A call at level $i$ receives a right-hand side $\boldsymbol{t} \in \mathcal{G}_i$ and must produce vectors $\boldsymbol{x} \in \mathcal{T}_i$ with $\mathbf{A}_i \boldsymbol{x} = \boldsymbol{t}$. For $i < k$, it chooses $\boldsymbol{b}$ uniformly in $\mathcal{G}_{i+1}$ and asks its two children for vectors of $\mathcal{T}_{i+1}$ with right-hand sides $\boldsymbol{b}$ and $\pi_i(\boldsymbol{t}) - \boldsymbol{b}$. Since $\mathbf{A}_{i+1} = \pi_i \circ \mathbf{A}_i$ is a homomorphism, every vector $\boldsymbol{p}$ from the first child and every vector $\boldsymbol{u}$ from the second satisfy

$$\mathbf{A}_{i+1}(\boldsymbol{p} + \boldsymbol{u}) = \pi_i(\boldsymbol{t}),$$

which is the image of the required congruence $\mathbf{A}_i(\boldsymbol{p} + \boldsymbol{u}) = \boldsymbol{t}$ under $\pi_i$, but not that congruence itself.

**List limits.** Fix an integer $\nu \geq 1$, taking $\nu = 1$ unless stated otherwise, and define

$$w = 2^{\nu\lceil n^{2/3}\rceil}, \qquad u_i = \left\lceil w\left(1 + \frac{|\mathcal{T}_i|}{|\mathcal{G}_i|}\right)\right\rceil, \qquad v_i = \left\lceil w\left(1 + \frac{|\mathcal{T}_{i+1}|}{|\mathcal{G}_i|}\right)\right\rceil, \tag{7}$$

where $u_i$ is defined for $0 \leq i \leq k$ and $v_i$ for $0 \leq i < k$. The algorithm enforces two limits: a level-$i$ call returns at most $u_i$ distinct vectors, and a combination at level $i$ pairs each right vector with at most the first $v_i$ left vectors having its matching value in $\mathcal{G}_i$. The latter restriction already appears in Ozerov's framework [Oze16, Algorithm 9]. Note that $w = 2^{o(n)}$. Section 3.1.1 bounds the probability that these limits discard a desired vector, and Section 3.1.2 uses them to bound the work and storage of every execution.

Algorithm 1 computes the quantities above from the parameters and the input and calls the recursive procedure Combine with the input right-hand side. Its guards return the empty list when $|\mathcal{D}_i(\boldsymbol{x})| < 2^{\lceil \eta n\rceil}$ for some $i$ or when the row counts (6) do not satisfy $m_0 > m_1 > \cdots > m_k \geq 1$; the parameter conditions exclude both cases for all sufficiently large $n$.

**Algorithm 1** Enumerating short solutions by retaining rows

**Require:** The procedure Combine (Algorithm 5), using the level data computed below.

**Parameters:** Integers $k, \nu \geq 1$, rational $\eta > 0$, and rational $0 < t_i < 1/2$ for $0 \leq i \leq k$, with $t_{i-1}/2 \leq t_i \leq (1 - t_{i-1})/2$ for $1 \leq i \leq k$.

**Input:** $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ and $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, where $n, m \geq 1$, $q \geq 3$, and $q^{\lceil \log(n+2) \rceil} \leq 2^n$.

**Output:** Distinct $\boldsymbol{x} \in \{-1, 0, 1\}^n$ with $\mathbf{A}\boldsymbol{x} = \boldsymbol{t}$ and $a_0$ entries of each nonzero sign; possibly an empty list.

$a_0 \leftarrow 2^k \left\lfloor t_0 n / 2^k \right\rfloor$
**for** $i = 1, \ldots, k$ **do**
    $a_i \leftarrow a_{i-1}/2 + 2^{k-i} \left\lfloor (t_i - t_{i-1}/2) n / 2^{k-i} \right\rfloor$
    $a'_i \leftarrow a_i - a_{i-1}/2$
**end for**
**for** $i = 0, \ldots, k$ **do**
    Define $\mathcal{T}_i = \{\boldsymbol{x} \in \{-1, 0, 1\}^n : |\{j : x_j = 1\}| = |\{j : x_j = -1\}| = a_i\}$.
**end for**
**for** $i = 1, \ldots, k$ **do**
    Define $\mathcal{D}_i(\boldsymbol{x})$ by the decomposition rule above, for $\boldsymbol{x} \in \mathcal{T}_{i-1}$.
    **if** $|\mathcal{D}_i(\boldsymbol{x})| < 2^{\lceil \eta n \rceil}$ **then**
        **return** [ ]
    **end if**
    $m_i \leftarrow \max\{r \in \mathbb{Z}_{\geq 0} : q^r 2^{\lceil \eta n \rceil} \leq |\mathcal{D}_i(\boldsymbol{x})|\}$
**end for**
$m_0 \leftarrow m$
**if** $m_0 > m_1 > \cdots > m_k \geq 1$ does not hold **then**
    **return** [ ]
**end if**
**for** $i = 0, \ldots, k$ **do**
    $\mathbf{A}_i \leftarrow$ the last $m_i$ rows of $\mathbf{A}$; $\mathcal{G}_i \leftarrow (\mathbb{Z}/q\mathbb{Z})^{m_i}$
**end for**
**for** $i = 0, \ldots, k-1$ **do**
    $\pi_i \leftarrow$ projection from $\mathcal{G}_i$ onto its last $m_{i+1}$ coordinates
**end for**
$w \leftarrow 2^{\nu \lceil n^{2/3} \rceil}$
**for** $i = 0, \ldots, k$ **do**
    $u_i \leftarrow \lceil w\,(1 + |\mathcal{T}_i|/|\mathcal{G}_i|) \rceil$
**end for**
**for** $i = 0, \ldots, k-1$ **do**
    $v_i \leftarrow \lceil w\,(1 + |\mathcal{T}_{i+1}|/|\mathcal{G}_i|) \rceil$
**end for**
**return** Combine$(0, \boldsymbol{t})$

**The recursive combination.** For $0 \leq i < k$ and $\boldsymbol{t} \in \mathcal{G}_i$, the call Combine$(i, \boldsymbol{t})$ returns a list $\mathcal{L}_i$ of at most $u_i$ distinct vectors $\boldsymbol{x} \in \mathcal{T}_i$ with $\mathbf{A}_i\boldsymbol{x} = \boldsymbol{t}$; Algorithm 5 in Appendix B.2 specifies it. The call chooses $\boldsymbol{b}$ uniformly in $\mathcal{G}_{i+1}$, independently of all other random choices, and puts $\boldsymbol{c} = \pi_i(\boldsymbol{t}) - \boldsymbol{b}$. If $i < k-1$, it obtains the left list $\mathcal{X} =$ Combine$(i+1, \boldsymbol{b})$ and the right list $\mathcal{Y} =$ Combine$(i+1, \boldsymbol{c})$ and sorts $\mathcal{X}$ by the ordering value $\mathbf{A}_i\boldsymbol{p}$ of its vectors $\boldsymbol{p}$ and $\mathcal{Y}$ by the ordering value $\boldsymbol{t} - \mathbf{A}_i\boldsymbol{u}$ of

its vectors $\boldsymbol{u}$. If $i = k - 1$, it reads the left vectors from the last-level generator with right-hand side $\boldsymbol{b}$ and the right vectors from the one with right-hand side $\boldsymbol{c}$, described below; these deliver their vectors one at a time in nondecreasing order of the same ordering values, and neither list is stored. Group elements are compared lexicographically by the canonical representatives of their cyclic coordinates, and ties among stored vectors are broken lexicographically by the vectors.

The two ordered sequences are then merged. Equal ordering values, $\mathbf{A}_i\boldsymbol{p} = \boldsymbol{t} - \mathbf{A}_i\boldsymbol{u}$, are exactly the pairs with $\mathbf{A}_i(\boldsymbol{p} + \boldsymbol{u}) = \boldsymbol{t}$. For each value attained on both sides, the call keeps the first $v_i$ left vectors with that value in a buffer and skips any further left vectors with that value; then, for each right vector $\boldsymbol{u}$ with that value and each buffered $\boldsymbol{p}$, it forms $\boldsymbol{x} = \boldsymbol{p} + \boldsymbol{u}$ over $\mathbb{Z}$, tests whether $\boldsymbol{x} \in \mathcal{T}_i$ by counting its entries of each sign, and appends $\boldsymbol{x}$ to $\mathcal{L}_i$ unless a dictionary shows that it is already present. As soon as $|\mathcal{L}_i| = u_i$, the call returns $\mathcal{L}_i$ immediately; otherwise it returns $\mathcal{L}_i$ when one side is exhausted. Returned lists retain insertion order. The root call $\textsc{Combine}(0, \boldsymbol{t})$ thus returns solutions of the input congruence, and its list is the output of Algorithm 1.

**Generating the last-level vectors.** A last-level generator receives $\boldsymbol{t} \in \mathcal{G}_k$ together with the shift $\boldsymbol{g} \in \mathcal{G}_{k-1}$ and the sign $\varepsilon \in \{\pm 1\}$ that define the ordering value $\boldsymbol{g} + \varepsilon\mathbf{A}_{k-1}\boldsymbol{x}$ required by its parent; Algorithm 6 in Appendix B.3 specifies it. It samples $\ell = n(n+1)^3$ coordinate partitions, independent of each other and of the elements $\boldsymbol{b}$, each consisting of a uniform first block of $n'$ positions and its complement, where $n' = 2\lfloor a_k/2 \rfloor + \lfloor (n - 2a_k)/2 \rfloor$. For each partition it lists every vector on the first block with $\lfloor a_k/2 \rfloor$, $\lfloor a_k/2 \rfloor$, and $\lfloor (n - 2a_k)/2 \rfloor$ entries equal to 1, $-1$, and 0, and every vector on the second block with the complementary counts, so that each sum of one vector from each list lies in $\mathcal{T}_k$. Pairs whose sum $\boldsymbol{x}$ satisfies $\mathbf{A}_k\boldsymbol{x} = \boldsymbol{t}$ are generated in nondecreasing order of the ordering value by the ordered pair generation of Schroeppel and Shamir [SS81], without storing all pairs, and a heap merges the partitions. The generator inspects at most $u_k$ pairs from each partition, emits a vector only from the first partition that is compatible with it, meaning that the first block carries exactly the prescribed counts of its entries, which removes duplicates, and stops after emitting $u_k$ distinct vectors. The parent reads the emitted vectors one at a time; the generator stores its block lists and merge data but no output list.

**Large moduli, randomness, and output modes.** When $q^{\lceil \log(n+2) \rceil} > 2^n$, Algorithm 7 in Appendix B.5 replaces Algorithm 1. It reduces the canonical integer entries of $\mathbf{A}$ and $\boldsymbol{t}$ modulo the largest power of three $q_0$ with $q_0 w \le q$, takes $\mathcal{G}_i$ and $\mathbf{A}_i$ from prefixes of the base-three digits of the reduced matrix, runs the same $\textsc{Combine}$ once for each integer lift $\boldsymbol{t} + q\boldsymbol{z}$ with $\boldsymbol{z} \in \{-n, \ldots, n\}^m$, and emits a candidate only after checking the input congruence over the integers.

We write $\omega$ for the random choices of an execution: the elements $\boldsymbol{b}$ of the recursive calls and the coordinate partitions of the last-level generators, sampled independently of the input. Appendix B.6 implements them with a tape of polynomially many independent fair bits, assigned in fixed blocks to the possible calls and partitions and reused for all lifts. The input and $\omega$ determine the execution.

Outputs are emitted in the insertion order of the root list. A polynomial-time acceptance test may be applied to each emitted vector; a rejection continues the enumeration and changes no list limit. We write $\mathcal{S}(\mathbf{A}, \boldsymbol{t}; \omega)$ for the variant that returns the first emitted vector of the appropriate branch, without an acceptance test, and $\perp$ if no vector is emitted. The complete enumeration is used for SVP and the first-output variant $\mathcal{S}$ for $\gamma$-CVP.

The following lemma bounds the probability that a solution fixed independently of the matrix is among the outputs. For even moduli, vectors with the same parity have additional collision probabilities; the following conditions control their contribution.

**Lemma 3.2.** *Fix integers $k, \nu \geq 1$, a real constant $\rho > 0$, and rational constants $\eta > 0$ and $0 < t_i < 1/2$ for $0 \leq i \leq k$ satisfying (3) and (5), with $h_i, r_i$ as in (4), $d_0 = \rho$, and $d_i = r_i - \eta$. If even moduli are allowed, also assume*

$$h(2t_0) > \frac{\rho}{2}, \qquad h(2t_i) > \frac{d_{i-1}}{2}, \qquad r_i > \frac{d_i}{2} + 2t_i - t_{i-1} \quad (1 \leq i \leq k). \tag{8}$$

*Let $n$ tend to infinity, and let integers $q \geq 3$ and $m \geq 1$ depend on $n$ with $m \log q = \rho n + o(n)$. For every fixed $\boldsymbol{x} \in \mathcal{T}$, chosen independently of a uniform $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, the procedure (Algorithm 1, or Algorithm 7 when $q^{\lceil \log(n+2) \rceil} > 2^n$) on input $(\mathbf{A}, \mathbf{A}\boldsymbol{x})$ satisfies, for sufficiently large $n$,*

$$\Pr_{\mathbf{A},\omega}[\boldsymbol{x} \text{ is among its outputs}] \geq 1 - 2^{-(\nu - o(1))n^{2/3}},$$

*where $\omega$ denotes its random bits, sampled independently of the input. In particular, for every fixed $\nu'$ with $0 < \nu' < \nu$, the failure probability is at most $2^{-\nu' n^{2/3}}$ for sufficiently large $n$.*

### 3.1.1 Correctness

**Validity of the outputs.** Every output of the procedure, in either modulus branch and in either output mode, lies in $\{\boldsymbol{x} \in \mathcal{T} : \mathbf{A}\boldsymbol{x} = \boldsymbol{t}\}$, the outputs are distinct, and each has squared length $2a_0 \leq \beta n$. This holds for every input and every execution, including executions that reach a list limit. The argument is an induction over the levels. A last-level generator emits distinct vectors of $\mathcal{T}_k$ whose $\mathbf{A}_k$-image is its right-hand side, because the two block lists carry complementary sign counts, only pairs with the required image are generated, and each vector is emitted only from its first compatible partition. At level $i < k$, equal ordering values give $\mathbf{A}_i(\boldsymbol{p} + \boldsymbol{u}) = \boldsymbol{t}$, the sign-count test retains only sums in $\mathcal{T}_i$, and the dictionary removes duplicates, so $\textsc{Combine}(i, \boldsymbol{t})$ returns distinct vectors of $\mathcal{T}_i$ with image $\boldsymbol{t}$; at the root, $\mathbf{A}_0 = \mathbf{A}$ gives the input congruence. In the large-modulus branch, the integer check enforces the input congruence directly, and the uniqueness of the integer lift $(\mathbf{A}\boldsymbol{x} - \boldsymbol{t})/q$ of a solution $\boldsymbol{x}$ prevents repeated outputs across lifts (Appendix B.5). Validity says nothing about which solutions are found; the two statements below concern that question.

**Finding a fixed vector.** The following argument proves Lemma 3.2.

*Proof of Lemma 3.2.* Throughout, $\mathbf{A}$ is uniform, $\boldsymbol{x} \in \mathcal{T}$ is fixed independently of $\mathbf{A}$, and the input is $(\mathbf{A}, \mathbf{A}\boldsymbol{x})$: the right-hand side depends on the matrix, the desired vector does not. We first treat the branch $q^{\lceil \log(n+2) \rceil} \leq 2^n$ with exact uniform random choices and show, by induction over the levels, that the probability of omitting $\boldsymbol{x}$ is $O(1/w) + 2^{-\Omega(n)}$. The collision and distribution estimates used below are proved in Appendix B.4.

*Decompositions that pass the congruence.* Fix $0 \leq i < k$ and $\boldsymbol{x} \in \mathcal{T}_i$; the vector of the lemma is the case $i = 0$. Hold $\mathbf{A}$ fixed and consider the call $\textsc{Combine}(i, \mathbf{A}_i\boldsymbol{x})$. Each $\boldsymbol{p} \in \mathcal{D}_{i+1}(\boldsymbol{x})$ satisfies $\mathbf{A}_{i+1}\boldsymbol{p} + \mathbf{A}_{i+1}(\boldsymbol{x} - \boldsymbol{p}) = \mathbf{A}_{i+1}\boldsymbol{x} = \pi_i(\mathbf{A}_i\boldsymbol{x})$, so the second child's congruence holds as soon as the first child's does. The element $\boldsymbol{b}$ of the call is uniform in $\mathcal{G}_{i+1}$, so a fixed $\boldsymbol{p}$ satisfies $\mathbf{A}_{i+1}\boldsymbol{p} = \boldsymbol{b}$ with probability $1/|\mathcal{G}_{i+1}|$, and linearity of expectation gives

$$\mathbb{E}_{\boldsymbol{b}}\big[\big|\{\boldsymbol{p} \in \mathcal{D}_{i+1}(\boldsymbol{x}) : \mathbf{A}_{i+1}\boldsymbol{p} = \boldsymbol{b}\}\big|\big] = \frac{|\mathcal{D}_{i+1}(\boldsymbol{x})|}{|\mathcal{G}_{i+1}|} = 2^{\eta n + o(n)},$$

by (4), $\log|\mathcal{G}_{i+1}| = d_{i+1}n + o(n)$, and $d_{i+1} = r_{i+1} - \eta$. The congruences thus restrict the children strongly while leaving, on average, exponentially many decompositions of $\boldsymbol{x}$. This expectation alone

does not show that the procedure outputs $\boldsymbol{x}$ with high probability: a decomposition compatible with $\boldsymbol{b}$ is selected after $\boldsymbol{b}$, and conditioned on $\boldsymbol{b}$ it depends on the matrix.

*Comparison with an independent decomposition.* Lemma B.3 compares the algorithm's experiment, in which $\mathbf{A}$ and $\boldsymbol{b}$ are sampled and then $\boldsymbol{z}$ is chosen uniformly among the elements of the fixed set $\mathcal{D}_{i+1}(\boldsymbol{x})$ with $\mathbf{A}_{i+1}\boldsymbol{z} = \boldsymbol{b}$, or $\boldsymbol{z} = \perp$ if there is none, with the experiment in which $\boldsymbol{z}$ is uniform in $\mathcal{D}_{i+1}(\boldsymbol{x})$ and independent of the entire matrix, and $\boldsymbol{b} = \mathbf{A}_{i+1}\boldsymbol{z}$. The choice of $\boldsymbol{z}$ is a device of the analysis; the algorithm does not perform it. For odd $q$, two distinct ternary vectors differ by a unit modulo $q$ in some coordinate, so their images under $\mathbf{A}_{i+1}$ coincide with probability exactly $1/|\mathcal{G}_{i+1}|$, and the two joint laws of $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z})$ are within total variation

$$\frac{1}{2}\sqrt{\frac{|\mathcal{G}_{i+1}| - 1}{|\mathcal{D}_{i+1}(\boldsymbol{x})|}} \leq \frac{1}{2}\, 2^{-\lceil \eta n \rceil / 2}$$

by (6). For even $q \geq 4$, vectors of equal parity collide more often; Lemma B.5 adds a term that the third inequality in (8) makes $2^{-\Omega(n)}$. The comparison retains the full matrix, because its rows beyond $\mathbf{A}_{i+1}$ determine the ordering values by which the children's vectors are merged, including the order in which a last-level generator emits them. After the comparison, both $\boldsymbol{z}$ and $\boldsymbol{x} - \boldsymbol{z}$ are independent of the entire matrix and of the random choices of the two child calls, which receive the right-hand sides $\mathbf{A}_{i+1}\boldsymbol{z}$ and $\mathbf{A}_{i+1}(\boldsymbol{x} - \boldsymbol{z})$.

*From the child outputs to the parent.* For $0 \leq i < k$, let $\delta_i$ be the maximum over $\boldsymbol{x} \in \mathcal{T}_i$ of $\Pr_{\mathbf{A},\omega}[\textsc{Combine}(i, \mathbf{A}_i\boldsymbol{x})$ does not return $\boldsymbol{x}]$, and let $\delta_k$ be the maximum probability that a last-level generator with right-hand side $\mathbf{A}_k\boldsymbol{x}$ omits a fixed $\boldsymbol{x} \in \mathcal{T}_k$, over all shifts $\boldsymbol{g} \in \mathcal{G}_{k-1}$ and signs $\varepsilon$, including choices that depend on the full matrix. By Lemma B.2, which holds for every ordering value, a fixed vector is omitted at level $k$ only if none of the $\ell$ partitions is compatible with it, which has probability at most $2^{-n}$, or if more than $u_k$ vectors of $\mathcal{T}_k$ share its image. At level $i < k$, in the compared experiment each child is a vector independent of the matrix, so by induction each is omitted with probability at most $\delta_{i+1}$. If both are returned, their ordering values agree, since $\mathbf{A}_i\boldsymbol{z} = \mathbf{A}_i\boldsymbol{x} - \mathbf{A}_i(\boldsymbol{x} - \boldsymbol{z})$, and the merge forms their sum $\boldsymbol{x} \in \mathcal{T}_i$ unless a limit interferes: $\boldsymbol{z}$ is in the buffer when at most $v_i$ left vectors share its ordering value, and the output limit cannot exclude $\boldsymbol{x}$ when at most $u_i$ vectors of $\mathcal{T}_i$ share the image $\mathbf{A}_i\boldsymbol{x}$, because a list of $u_i$ distinct solutions already contains all of them.

*Omission by the limits.* For odd $q$, the expected number, over $\mathbf{A}$, of vectors of $\mathcal{T}_j$ that share the image of a fixed vector under $r$ retained rows is $1 + (|\mathcal{T}_j| - 1)/q^r$, and $u_i$ and $v_i$ in (7) exceed the corresponding expectations by the factor $w$; Markov's inequality (Lemma 2.3) therefore bounds each probability of exceeding a limit by $1/w$. For even $q \geq 4$, Lemma B.4 and the first two inequalities in (8) bound the additional collisions among vectors of equal parity by the main term, which gives $2/w$. Hence $\delta_k \leq 2^{-n} + 2/w$, and a union bound over the two children, the two limits, and the comparison error, without conditioning on the success of any other call, gives

$$\delta_i \leq 2\delta_{i+1} + 4/w + 2^{-\Omega(n)} \qquad (0 \leq i < k).$$

Since $k$ is fixed, $\delta_0 \leq O(1/w) + 2^{-\Omega(n)}$.

*Large moduli and finite randomness.* In the branch $q^{\lceil \log(n+2) \rceil} > 2^n$, the reduced matrix is within total variation $mn/w$ of a uniform matrix over $\mathbb{Z}/q_0\mathbb{Z}$, under which the same collision and comparison estimates hold for the digit-prefix groups, and the lift of the input right-hand side that equals $\mathbf{A}\boldsymbol{x}$ over the integers is among the processed lifts. Applying the bound above to the call at that lift, without conditioning on earlier lifts, gives omission probability at most $O(1/w) + mn/w + 2^{-\Omega(n)}$ (Appendix B.5). Replacing the exact uniform choices by the tape of fair bits changes the probability of every event determined by the execution by at most $2^{-\Omega(n^2)}$ (Appendix B.6). Since $w = 2^{\nu\lceil n^{2/3} \rceil}$,

the probability of omitting $\boldsymbol{x}$ is at most $\mathrm{poly}(n)/w + 2^{-\Omega(n)} + 2^{-\Omega(n^2)} = 2^{-(\nu-o(1))n^{2/3}}$ in both branches, which is the claim; the bound for $\nu' < \nu$ follows for sufficiently large $n$. □

**Average success for a uniform right-hand side.** The $\gamma$-CVP construction needs success averaged over a uniform right-hand side that is independent of the matrix, for the first-output variant $\mathcal{S}$. To obtain this bound, we additionally assume $h_0 > \rho$, so that $\mathcal{T}$ has exponentially more vectors than $(\mathbb{Z}/q\mathbb{Z})^m$ has elements. Lemma 3.2 does not require this inequality.

**Corollary 3.3.** *Fix integers $k, \nu \geq 1$, a real constant $\rho > 0$, and rational constants $\eta > 0$ and $0 < t_i < 1/2$ for $0 \leq i \leq k$ satisfying (3) and (5), with $h_i, r_i$ as in (4), $d_0 = \rho$, and $d_i = r_i - \eta$, and assume additionally that $h_0 > \rho$. If even moduli are allowed, also assume (8). Let $n$ tend to infinity, and let integers $q \geq 3$ and $m \geq 1$ depend on $n$ with $m \log q = \rho n + o(n)$. Let $\mathcal{S}(\mathbf{A}, \boldsymbol{t}; \omega)$ be the first-output variant of the same procedure. For every fixed $p \in (0,1)$ and sufficiently large $n$, all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ satisfy*

$$\Pr_{\boldsymbol{t},\omega}[\mathcal{S}(\mathbf{A}, \boldsymbol{t}; \omega) \neq \perp] \geq p,$$

*where $\mathbf{A}$ is held fixed, $\boldsymbol{t}$ is uniform in $(\mathbb{Z}/q\mathbb{Z})^m$, and $\omega$ is independent of $\boldsymbol{t}$.*

*Proof.* Let $\boldsymbol{z}$ be uniform in $\mathcal{T}$ and let $\mathbf{A}$ be uniform, with $\boldsymbol{z}$, $\mathbf{A}$, and $\omega$ independent. For each fixed value of $\boldsymbol{z}$, Lemma 3.2 bounds the probability over $(\mathbf{A}, \omega)$ that $\boldsymbol{z}$ is not output on input $(\mathbf{A}, \mathbf{A}\boldsymbol{z})$ by $2^{-(\nu-o(1))n^{2/3}}$; averaging over $\boldsymbol{z}$ gives the same bound for the joint probability over $(\mathbf{A}, \boldsymbol{z}, \omega)$. If $\boldsymbol{z}$ is output, the output list is nonempty, so $\mathcal{S}(\mathbf{A}, \mathbf{A}\boldsymbol{z}; \omega) \neq \perp$.

Let $\boldsymbol{t}$ be uniform in $(\mathbb{Z}/q\mathbb{Z})^m$ and independent of $\mathbf{A}$. For odd $q$, Lemma B.3 with all $m$ rows gives

$$\mathrm{TV}\big((\mathbf{A}, \mathbf{A}\boldsymbol{z}), (\mathbf{A}, \boldsymbol{t})\big) \leq \frac{1}{2}\sqrt{\frac{q^m - 1}{|\mathcal{T}|}} = 2^{(\rho - h_0)n/2 + o(n)} = 2^{-\Omega(n)},$$

by $h_0 > \rho$. For even $q \geq 4$, each parity class of $\mathcal{T}$ contains at most $\binom{2a_0}{a_0} \leq 2^{2t_0 n}$ vectors, and Lemma B.5 adds under the square root a term of order $2^{(\rho/2 + 2t_0 - h_0)n + o(n)}$, which is $2^{-\Omega(n)}$ because $h_0 - 2t_0 = h(2t_0) > \rho/2$ by the first inequality in (8). The randomness $\omega$ is independent of both pairs and can be appended without changing these bounds. Hence $\Pr_{\mathbf{A},\boldsymbol{t},\omega}[\mathcal{S}(\mathbf{A}, \boldsymbol{t}; \omega) = \perp] \leq 2^{-\Omega(n^{2/3})}$.

Finally, hold $\mathbf{A}$ fixed and consider the conditional failure probability over $(\boldsymbol{t}, \omega)$. By Markov's inequality (Lemma 2.3), the fraction of matrices for which it exceeds $1 - p$ is at most the joint failure probability divided by $1 - p$, which is $2^{-\Omega(n^{2/3})}$ for every fixed $p \in (0,1)$. □

**Uniformity over parameter choices.** The SVP algorithm and the density-dependent $\gamma$-CVP algorithm compute the proportions $t_i$ from the input dimensions. Keep $k, \nu, \eta$ fixed, put $\rho = m \log q / n$, and compute the rational $t_i$ deterministically in polynomial time from $n, m, q$ alone. Suppose the tuples $(\rho, t_0, \ldots, t_k)$ lie in a fixed compact set on which $\rho > 0$, $0 < t_i < 1/2$,

$$\frac{t_{i-1}}{2} < t_i < \frac{1 - t_{i-1}}{2} \qquad (1 \leq i \leq k),$$

and the strict equation and applicable parity conditions (5) and (8) hold. Then Lemma 3.2 holds uniformly over these choices: the implicit constants and the threshold for $n$ can be chosen independently of the tuple, and the $o(1)$ term admits a bound of that order over the entire compact set. Compactness and continuity give one positive lower bound for every strict margin, the strict coordinate conditions keep all fractions in the decomposition counts bounded away from zero, and

choosing the proportions from the dimensions alone leaves the matrix distribution unchanged and keeps the fixed vector independent of the matrix, so the argument above applies to every such choice with a uniform error term. If additionally $h_0 > \rho$ throughout the same compact set, Corollary 3.3 also holds uniformly for each fixed $p \in (0,1)$, since compactness supplies a positive lower bound for $h_0 - \rho$. Appendix B.7 proves these extensions, together with the uniform time and space bounds stated in Section 3.1.2.

The two probability statements have different uses. Corollary 3.3 supplies the average success bound required by the $\gamma$-CVP transformation in Lemma 4.4, which obtains success at every right-hand side. SVP applies Lemma 3.2 after accounting for the dependence of a shortest vector on the matrix. The statement for a uniform right-hand side alone does not imply a positive success probability at $\boldsymbol{t} = \mathbf{0}_m$, and neither statement alone proves that the procedure finds a shortest vector of the lattice defined by $\mathbf{A}$.

### 3.1.2 Computation costs

The bounds below hold for every input matrix and right-hand side and for every execution, including executions that reach a list limit or return no vector, and they use no probabilistic assumption. Polynomial factors in the binary input length are written poly and are omitted from exponential bounds. The two list limits are what make the bounds independent of the input: without them, the number of candidates sharing a matching value, and with it the work and the stored lists, would depend on the matrix.

**Last-level generation.** For each of its $\ell = n(n+1)^3$ partitions, a last-level generator lists the vectors on the two blocks with the prescribed sign counts. The block counts are halves of $(a_k, a_k, n-2a_k)$ up to rounding, so the factorial formula for $|\mathcal{T}_k|$ gives each block list $\sqrt{|\mathcal{T}_k|}\,2^{O(\log n)} = 2^{h_k n/2 + o(n)}$ entries. Grouping one list by $\mathbf{A}_k$-images, building the search structures for the ordered pair generation, and inserting the first pairs into the heap cost $\sqrt{|\mathcal{T}_k|}$ poly per partition, and each further pair costs polynomial time (Lemma B.1); at most $\ell u_k$ pairs are inspected in total. One generation therefore takes time $(\sqrt{|\mathcal{T}_k|} + u_k)$ poly and space $\sqrt{|\mathcal{T}_k|}$ poly, the space holding the block lists and merge data of the suspended generator; emitted vectors are not stored (Lemma B.2).

**List sizes and combinations.** By (7), $\log|\mathcal{T}_i| = h_i n + O(\log n)$, $\log|\mathcal{G}_i| = d_i n + o(n)$, and $\log w = o(n)$,

$$\log u_i = [h_i - d_i]_+ n + o(n), \qquad \log v_i = [h_{i+1} - d_i]_+ n + o(n).$$

The limits are deterministic caps; the ratios $|\mathcal{T}_i|/|\mathcal{G}_i|$ and $|\mathcal{T}_{i+1}|/|\mathcal{G}_i|$ enter only through their definition. A combination at level $i-1$, for $1 \le i \le k$, reads at most $u_i$ vectors from each child and, when $i < k$, sorts the two stored child lists in time $u_i$ poly; the last-level generators deliver their vectors in order and need no sorting. It inspects at most $u_i v_{i-1}$ pairs, since each right vector is paired with at most $v_{i-1}$ buffered left vectors, and each inspected pair costs polynomial time for the integer addition, the sign-count test, the dictionary lookup, and the insertion. As $v_{i-1} \ge 1$, the reading and sorting cost is absorbed, and the combination takes time $u_i v_{i-1}$ poly $= 2^{([h_i - d_i]_+ + [h_i - d_{i-1}]_+)n + o(n)}$.

**Time.** The recursion has fixed depth $k$, so an execution makes $2^k - 1$ calls to Combine and $2^k$ last-level generations. Summing the bounds above gives time

$$\Big(\sqrt{|\mathcal{T}_k|} + \sum_{i=1}^{k} u_i v_{i-1}\Big)\,\text{poly},$$

which also pays for reading the root list and for a polynomial-time acceptance test on every emitted vector. In the large-modulus branch, the $(2n+1)^m = 2^{O((\log n)^2)}$ lifts are processed in sequence and the integer check is polynomial per candidate, which multiplies the time by $2^{o(n)}$. Sampling and decoding the polynomial-length random tape take polynomial time and space.

**Space.** The calls run depth first. While a call at level $i < k-1$ computes its right child, it stores the finished left child list, of size at most $u_{i+1}$; while it merges, it stores both child lists, its buffer, its dictionary, and the output list under construction, of size at most $u_i$. At level $k-1$, the two generators are suspended instead of stored, and the buffer holds at most $v_{k-1}$ vectors. At any time, the stored data therefore consist of one active call per level, each holding a constant number of lists: lists of size at most $u_i$ for $0 \le i < k$, buffers, and the data of at most two suspended generators. For $i < k-1$, $v_i \le u_{i+1}$ because $|\mathcal{G}_i| \ge |\mathcal{G}_{i+1}|$, so only the buffer at level $k-1$ needs a separate term, and no level-$k$ list is ever stored. Storage is reused between successive calls and between the lifts of the large-modulus branch. The simultaneous space is at most

$$\Big(\sqrt{|\mathcal{T}_k|} + \sum_{i=0}^{k-1} u_i + v_{k-1}\Big)\,\mathrm{poly}\,.$$

**Time and space exponents.** With the estimates for $u_i$ and $v_i$, define

$$\begin{aligned} T_{\mathrm{in}} &= \max\left\{\frac{h_k}{2}, \max_{1\le i\le k}\big([h_i-d_i]_+ + [h_i-d_{i-1}]_+\big)\right\},\\ S_{\mathrm{in}} &= \max\left\{\frac{h_k}{2}, \max_{0\le i<k}[h_i-d_i]_+,\ [h_k-d_{k-1}]_+\right\}. \end{aligned} \tag{9}$$

Then every execution terminates in time $2^{T_{\mathrm{in}}n+o(n)}$ and space $2^{S_{\mathrm{in}}n+o(n)}$, up to factors polynomial in the binary input length. In $T_{\mathrm{in}}$, the term $h_k/2$ is the last-level listing and the term $[h_i-d_i]_+ + [h_i-d_{i-1}]_+$ is the pair inspection of the combination at level $i-1$; in $S_{\mathrm{in}}$, the three terms are the block lists of a suspended generator, the stored lists at the levels below $k$, and the last buffer. Appendix B.5 verifies the bounds for the large-modulus branch, and Appendix B.6 shows that the finite implementation uses polynomially many random bits. For parameters chosen from the input dimensions as in Section 3.1.1, the bounds hold uniformly: $T_{\mathrm{in}}, S_{\mathrm{in}}$ are evaluated at the chosen tuple, the $o(n)$ terms admit bounds of that order over the entire compact set, the guards pass above one threshold for $n$ independent of the tuple, and computing the rational proportions and the integer counts adds a polynomial factor (Appendix B.7).

## 3.2 Exact search inside a prescribed ball

We next consider the problem of finding integer solutions to linear congruences within a prescribed Euclidean ball. Unlike the previous procedure, this algorithm performs an exact search and finds a solution closest to the center of the ball, if one exists.

**Definition 3.4** (Closest lattice vector within a ball)**.** Let $n \ge 1$, $m \ge 0$, and $q \ge 2$ be integers. Given $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, $\boldsymbol{y} \in \mathbb{Q}^n$, and $b \in \mathbb{Q}_{\ge 0}$, return a vector $\boldsymbol{x} \in \mathbb{Z}^n$ minimizing $\|\boldsymbol{x}-\boldsymbol{y}\|^2$ subject to

$$\mathbf{H}\boldsymbol{x} = \mathbf{0}_m, \qquad \|\boldsymbol{x}-\boldsymbol{y}\|^2 \le b\ ,$$

or $\perp$ when no such vector exists.

We give a deterministic algorithm for Definition 3.4, denoted by $\mathcal{E}_\tau$ for a rational parameter $\tau \in [0,1]$ (Algorithm 2). It returns $\perp$ exactly when the ball contains no integer vector satisfying the congruence, and otherwise a closest such vector. The parameter $\tau$ enters only the squared radii of the partial lists defined below. In the $\gamma$-CVP application, it is chosen so that the rounding-error contribution to the search radius cancels from the exponent in the cost bound (Section 4.3).

Algorithm 2 separates coordinates according to their contributions to the squared distance from the target. Coordinates with large contributions are handled directly, while the remaining coordinates are processed through four lists of partial integer assignments. Combining these lists allows the search to avoid storing all pairs of assignments. We describe the construction below.

**Fixing the coordinates with large errors.** Put $\boldsymbol{f} = \lfloor \boldsymbol{y} \rceil - \boldsymbol{y}$, so that $|f_i| \le 1/2$ for every $i \in [n]$. If $b < \|\boldsymbol{f}\|^2$, the procedure returns $\perp$ at once. Otherwise put

$$\Delta = \lceil \sqrt{n} \rceil, \qquad b_\tau = b - \tau \|\boldsymbol{f}\|^2.$$

The procedure enumerates every set $\mathcal{J} \subseteq [n]$ of at most $\lfloor b/\Delta \rfloor$ coordinates and every assignment $\boldsymbol{u} \in \mathbb{Z}^{\mathcal{J}}$ satisfying

$$\Delta < (u_i - y_i)^2 \le b \quad (i \in \mathcal{J}), \qquad \sum_{i \in \mathcal{J}} (u_i - y_i)^2 \le b.$$

Such a pair $(\mathcal{J}, \boldsymbol{u})$ prescribes the coordinates with squared error exceeding $\Delta$ and their values; every other coordinate will be restricted to squared error at most $\Delta$. Fixing $\boldsymbol{x}_{\mathcal{J}} = \boldsymbol{u}$ turns the congruence $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$ into $\mathbf{H}_{[n]\setminus\mathcal{J}} \boldsymbol{x}_{[n]\setminus\mathcal{J}} = \boldsymbol{s}'$ with $\boldsymbol{s}' = -\mathbf{H}_{\mathcal{J}} \boldsymbol{u}$, which is passed to the combination step.

**Dividing the remaining coordinates.** The procedure $\mathcal{C}$ of Algorithm 8 takes a list of distinct indices and outputs, one pair at a time, the two halves of each cyclic rotation of the list: an odd-length list is first padded with one padding entry, each rotation is split into two halves of equal length, and the padding entry is deleted from the halves. The empty list has the unique split into two empty lists. Algorithm 2 applies $\mathcal{C}$ to the list $[n] \setminus \mathcal{J}$ in increasing order and then, independently, to each of the two output halves. This gives polynomially many partitions of the remaining coordinates into four lists $\mathcal{I}_1, \ldots, \mathcal{I}_4$, each of length at most $\lceil n/4 \rceil$. Each $\mathcal{I}_j$ is a list of distinct indices in $[n]$, which we identify with the set of its entries; vector restrictions and matrix columns use these indices in increasing order, as in Section 2.1. The procedure does not select a partition: it processes every output of the three calls, and the analysis shows that one of them balances the four parts.

**Forming four lists of partial solutions.** For each part, the procedure lists all assignments $\boldsymbol{v} \in \mathbb{Z}^{\mathcal{I}_j}$ with coordinate squared errors at most $\Delta$ and total squared error at most

$$b_j = \frac{b_\tau}{4} + 2\Delta + \tau \sum_{i \in \mathcal{I}_j} f_i^2.$$

The procedure $\mathcal{L}$ of Algorithm 9 takes a rational target and a squared radius and outputs every integer vector within that squared distance of the target exactly once. It assigns integer coordinates successively, discarding a partial assignment if its squared error plus the minimum rounding error of the unassigned coordinates exceeds the radius, and stores only the current partial assignment. Algorithm 2 calls $\mathcal{L}(\boldsymbol{y}_{\mathcal{I}_j}, b_j)$ and keeps the output vectors whose coordinate squared errors are all at most $\Delta$. The resulting list $\mathcal{X}_j$ stores records $X = (\boldsymbol{g}_X, c_X, \boldsymbol{v}_X)$, where

$$\boldsymbol{g}_X = \mathbf{H}_{\mathcal{I}_j} \boldsymbol{v}, \qquad c_X = \|\boldsymbol{v} - \boldsymbol{y}_{\mathcal{I}_j}\|^2, \qquad \boldsymbol{v}_X = \boldsymbol{v}.$$

**Algorithm 2** Closest lattice vector within a ball: $\mathcal{E}_\tau$

**Require:** The procedure $\mathcal{C}$, enumerate prescribed splits of ordered index lists (Algorithm 8).

**Require:** The procedure $\mathcal{L}$, enumerate integer vectors within a squared-distance bound (Algorithm 9).

**Require:** The procedure $\mathcal{M}$, find a minimum-cost four-list congruence match or return $\perp$ (Algorithm 11).

**Parameters:** A rational constant $\tau \in [0, 1]$.

**Input:** $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, $\boldsymbol{y} \in \mathbb{Q}^n$, and $b \in \mathbb{Q}_{\geq 0}$, where $n \geq 1$, $m \geq 0$, and $q \geq 2$.

**Output:** A closest $\boldsymbol{x} \in \mathbb{Z}^n$ to $\boldsymbol{y}$ satisfying $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$ and $\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq b$, or $\perp$ if none exists.

$\boldsymbol{f} \leftarrow \lfloor \boldsymbol{y} \rceil - \boldsymbol{y}$
**if** $b < \|\boldsymbol{f}\|^2$ **then**
    **return** $\perp$
**end if**
$\Delta \leftarrow \lceil \sqrt{n} \rceil$; $b_\tau \leftarrow b - \tau\|\boldsymbol{f}\|^2$; $\boldsymbol{x} \leftarrow \perp$; $b_{\text{best}} \leftarrow b$
**for all** $\mathcal{J} \subseteq [n]$ with $|\mathcal{J}| \leq \lfloor b/\Delta \rfloor$ **do**
    **for all** $\boldsymbol{u} \in \mathbb{Z}^{\mathcal{J}}$ with $\Delta < (u_i - y_i)^2 \leq b$ for every $i \in \mathcal{J}$ **do**
        **if** $\sum_{i \in \mathcal{J}} (u_i - y_i)^2 > b$ **then**
            **continue**
        **end if**
        $\boldsymbol{s}' \leftarrow -\mathbf{H}_{\mathcal{J}}\boldsymbol{u}$; $\mathcal{K} \leftarrow [n] \setminus \mathcal{J}$ in increasing order
        **for all** $(\mathcal{K}_1, \mathcal{K}_2)$ output by $\mathcal{C}(\mathcal{K})$ **do**
            **for all** $(\mathcal{I}_1, \mathcal{I}_2)$ output by $\mathcal{C}(\mathcal{K}_1)$ **do**
                **for all** $(\mathcal{I}_3, \mathcal{I}_4)$ output by $\mathcal{C}(\mathcal{K}_2)$ **do**
                    **for** $j = 1, \ldots, 4$ **do**
                        $b_j \leftarrow b_\tau/4 + 2\Delta + \tau \sum_{i \in \mathcal{I}_j} f_i^2$; $\mathcal{X}_j \leftarrow [\,]$
                        **for all** $\boldsymbol{v}$ output by $\mathcal{L}(\boldsymbol{y}_{\mathcal{I}_j}, b_j)$ **do**
                            **if** $(v_i - y_i)^2 \leq \Delta$ for all $i \in \mathcal{I}_j$ **then**
                                Append $(\mathbf{H}_{\mathcal{I}_j}\boldsymbol{v}, \|\boldsymbol{v} - \boldsymbol{y}_{\mathcal{I}_j}\|^2, \boldsymbol{v})$ to $\mathcal{X}_j$
                            **end if**
                        **end for**
                    **end for**
                    $R \leftarrow \mathcal{M}(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4, \boldsymbol{s}')$
                    **if** $R = \perp$ **then**
                        **continue**
                    **end if**
                    $(X_1, \ldots, X_4) \leftarrow R$
                    $\boldsymbol{z}_{\mathcal{J}} \leftarrow \boldsymbol{u}$; $\boldsymbol{z}_{\mathcal{I}_j} \leftarrow \boldsymbol{v}_{X_j}$ for $j = 1, \ldots, 4$
                    **if** $\|\boldsymbol{z} - \boldsymbol{y}\|^2 \leq b_{\text{best}}$ **then**
                        $(\boldsymbol{x}, b_{\text{best}}) \leftarrow (\boldsymbol{z}, \|\boldsymbol{z} - \boldsymbol{y}\|^2)$
                    **end if**
                **end for**
            **end for**
        **end for**
    **end for**
**end for**
**return** $\boldsymbol{x}$

Thus the fields are the element $\boldsymbol{g}_X \in (\mathbb{Z}/q\mathbb{Z})^m$, the squared distance $c_X$, which we call the record's cost, and the partial integer vector $\boldsymbol{v}_X$. An empty part has one record with $\boldsymbol{g}_X = \mathbf{0}_m$, zero cost, and the empty vector. The four lists, together with $\boldsymbol{s}'$, are the input of the combination step.

**Combining the lists without storing all pairs.** The procedure $\mathcal{M}$ of Algorithm 11 takes the four lists and the required sum $\boldsymbol{s}'$ and returns records $X_1, \dots, X_4$, one from each list, with $\sum_j \boldsymbol{g}_{X_j} = \boldsymbol{s}'$ and minimum total cost $\sum_j c_{X_j}$, or $\perp$ when no such quadruple exists. We adapt the ordered generation of pair sums of Schroeppel and Shamir [SS81], as described by Shallue [Sha12, Section 4]. The combination also works in a product $\mathcal{G} = \prod_{i=1}^{m} \mathbb{Z}/q_i\mathbb{Z}$ with integers $q_i \geq 2$; the lists above have $q_i = q$ for every $i$. Elements of $\mathcal{G}$ are compared lexicographically using their canonical residue coordinates. For $m = 0$, $\mathcal{G}$ has one element. The combination uses only each record's element of $\mathcal{G}$, rational cost, and retained partial vector. Within each list, it first retains only a minimum-cost record for each element of $\mathcal{G}$; when costs tie, it retains the first such record in input order. For lists $\mathcal{X}, \mathcal{Y}$, a shift $\boldsymbol{t} \in \mathcal{G}$, and $\varepsilon \in \{-1, 1\}$, a pair record $W$ retains $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ and has element $\boldsymbol{g}_W = \boldsymbol{t} + \varepsilon(\boldsymbol{g}_X + \boldsymbol{g}_Y)$ and cost $c_X + c_Y$. The procedure $\mathcal{P}(\mathcal{X}, \mathcal{Y}; \boldsymbol{t}, \varepsilon)$ outputs a minimum-cost pair for each attained element of $\mathcal{G}$, in increasing order of those elements. It consumes all pairs with the same element before emitting their minimum-cost pair, retaining the first generated pair when costs tie. The complete set of pairs is never stored. Section C.1.4 gives the implementation, extending the binary-coordinate construction of Dietzfelbinger, Schlag, and Walzer [DSW18, Algorithm 3] to general moduli.

The procedure $\mathcal{M}$ runs $\mathcal{P}(\mathcal{X}_1, \mathcal{X}_2; \mathbf{0}_m, 1)$ and $\mathcal{P}(\mathcal{X}_3, \mathcal{X}_4; \boldsymbol{s}', -1)$ and merges their outputs as two sorted sequences: it advances the procedure whose current element is smaller, and when the two current elements are equal, it records the four underlying records if their total cost is smaller than the best total found so far, and then advances both. It returns the recorded quadruple, or $\perp$ if the two outputs never agree. When $\mathcal{M}$ returns $X_1, \dots, X_4$, Algorithm 2 assembles the candidate $\boldsymbol{z} \in \mathbb{Z}^n$ with $\boldsymbol{z}_{\mathcal{J}} = \boldsymbol{u}$ and $\boldsymbol{z}_{\mathcal{I}_j} = \boldsymbol{v}_{X_j}$ for $j = 1, \dots, 4$, and accepts it if $\|\boldsymbol{z} - \boldsymbol{y}\|^2 \leq b_{\text{best}}$, where $b_{\text{best}}$ is $b$ initially and the squared distance of the closest accepted candidate afterwards. Thus a candidate whose squared distance exceeds $b$ is discarded, and after all choices $(\mathcal{J}, \boldsymbol{u})$ and all partitions the procedure returns the closest accepted candidate, or $\perp$ if none was accepted.

**Excluding the zero vector.** For SVP we use a variant $\mathcal{E}_\tau^{\neq 0}(\mathbf{H}, \boldsymbol{y}, b)$ that returns a closest vector among the nonzero solutions, or $\perp$ when the ball contains no nonzero solution (Algorithm 12 in Appendix C.3). After the rounding test of $\mathcal{E}_\tau$, it enumerates a coordinate $i \in [n]$ and a nonzero integer $z$ with $(z - y_i)^2 \leq b$, and fixes $x_i = z$. This leaves, on the remaining coordinates, the problem of finding a closest vector $\boldsymbol{x}' \in \mathbb{Z}^{n-1}$ to $\boldsymbol{y}_{[n]\setminus\{i\}}$ subject to

$$\mathbf{H}_{[n]\setminus\{i\}}\boldsymbol{x}' = -z\mathbf{H}\boldsymbol{e}_i, \qquad \|\boldsymbol{x}' - \boldsymbol{y}_{[n]\setminus\{i\}}\|^2 \leq b - (z - y_i)^2,$$

whose right-hand side is no longer zero and whose squared radius is reduced by $(z - y_i)^2$. If $n = 1$, no coordinate remains, and the variant tests $z\mathbf{H}\boldsymbol{e}_1 = \mathbf{0}_m$ directly. Otherwise it calls the auxiliary procedure $\mathcal{E}_\tau^{\text{aux}}$ of Appendix C.3, which is Algorithm 2 with the required sum $\boldsymbol{s}' = -\mathbf{H}_{\mathcal{J}}\boldsymbol{u}$ replaced by $\boldsymbol{s}' = \boldsymbol{s} - \mathbf{H}_{\mathcal{J}}\boldsymbol{u}$ for a prescribed right-hand side $\boldsymbol{s}$, here $\boldsymbol{s} = -z\mathbf{H}\boldsymbol{e}_i$; the lists and their combination are unchanged. Every returned vector is completed by $x_i = z$, and the closest completed vector over all choices of $i$ and $z$ is returned. The nonzero coordinate is fixed before the search, rather than removing the zero vector after the minimum-cost combination, because the retention of minimum-cost records could otherwise discard every nonzero partial vector in favor of a zero one with the same element of $\mathcal{G}$.

The following lemma states the output conditions and costs used in the $\gamma$-CVP and SVP algorithms. For $a > 0$ and $u \in \mathbb{R}$, write

$$\Theta_u(a) = \sum_{j \in \mathbb{Z}} e^{-a(j-u)^2}, \qquad \Theta(a) = \Theta_0(a)$$

as in (48). The cost bound uses the inequality

$$\Theta_u(a) \le \Theta(a) e^{-a\tau u^2} \qquad (|u| \le 1/2). \tag{10}$$

The parameter $a$ is used only in the analysis.

**Lemma 3.5.** *Fix a rational constant $\tau \in [0,1]$. For integers $n \ge 1$, $m \ge 0$, and $q \ge 2$, the deterministic procedure $\mathcal{E}_\tau$ of Algorithm 2 takes*

$$\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}, \qquad \boldsymbol{y} \in \mathbb{Q}^n, \qquad b \in \mathbb{Q}_{\ge 0}, \quad b = O(n).$$

*It returns a vector of minimum distance from $\boldsymbol{y}$ in*

$$\{\boldsymbol{x} \in \mathbb{Z}^n : \mathbf{H}\boldsymbol{x} = \mathbf{0}_m,\ \|\boldsymbol{x} - \boldsymbol{y}\|^2 \le b\},$$

*or $\perp$ exactly when this set is empty. Put $\boldsymbol{f} = \lfloor \boldsymbol{y} \rceil - \boldsymbol{y}$ and $b_\tau = b - \tau\|\boldsymbol{f}\|^2$. If $b < \|\boldsymbol{f}\|^2$, the procedure returns $\perp$ in polynomial time. Otherwise, for any $a = a(n) > 0$ with $a = \Omega(1)$ and $a = O(\log(n+2))$ that satisfies (10), every execution uses time $2^{\frac{ab_\tau + n \ln \Theta(a)}{2 \ln 2} + O(\sqrt{n}\log(n+2))}$ and space $2^{\frac{ab_\tau + n \ln \Theta(a)}{4 \ln 2} + O(\sqrt{n}\log(n+2))}$, omitting factors polynomial in the binary input length. The same guarantees hold for a deterministic variant $\mathcal{E}_\tau^{\neq 0}(\mathbf{H}, \boldsymbol{y}, b)$ restricted to nonzero solutions.*

The next two subsections prove the correctness and cost claims, respectively. The subroutine implementations and their proofs are given in Appendix C.

### 3.2.1 Correctness

We show that Algorithm 2 returns a vector of minimum distance from $\boldsymbol{y}$ in $\{\boldsymbol{x} \in \mathbb{Z}^n : \mathbf{H}\boldsymbol{x} = \mathbf{0}_m,\ \|\boldsymbol{x} - \boldsymbol{y}\|^2 \le b\}$, or $\perp$ exactly when this set is empty. The argument uses only the output specifications of the subroutines; the growth conditions and (10) play no role in it.

**Immediate rejection.** For every $i \in [n]$, $\lfloor y_i \rceil$ is an integer nearest to $y_i$, so $|z - y_i| \ge |f_i|$ for every integer $z$. Summing the squares over the coordinates gives $\|\boldsymbol{x} - \boldsymbol{y}\|^2 \ge \|\boldsymbol{f}\|^2$ for every $\boldsymbol{x} \in \mathbb{Z}^n$. Hence, if $b < \|\boldsymbol{f}\|^2$, the ball contains no integer vector, and returning $\perp$ is correct. Otherwise $b_\tau = b - \tau\|\boldsymbol{f}\|^2 \ge b - \|\boldsymbol{f}\|^2 \ge 0$, since $0 \le \tau \le 1$.

**Accepted candidates are solutions.** Every candidate $\boldsymbol{z}$ accepted in Algorithm 2 satisfies

$$\mathbf{H}\boldsymbol{z} = \mathbf{H}_{\mathcal{J}}\boldsymbol{u} + \sum_{j=1}^{4} \mathbf{H}_{\mathcal{I}_j}\boldsymbol{v}_{X_j} = \mathbf{H}_{\mathcal{J}}\boldsymbol{u} + \boldsymbol{s}' = \mathbf{0}_m,$$

because the records returned by $\mathcal{M}$ satisfy $\sum_j \boldsymbol{g}_{X_j} = \boldsymbol{s}'$, and $\|\boldsymbol{z} - \boldsymbol{y}\|^2 \le b_{\text{best}} \le b$ by the acceptance test. Thus the returned vector, if any, lies in the feasible set, and the final value of $b_{\text{best}}$ is its squared distance. It remains to show that no feasible vector is closer.

**Enumeration of the coordinates with large errors.** Suppose the feasible set is nonempty and fix a closest vector $\boldsymbol{x}$ in it. Consider the iteration in which $\mathcal{J} = \{i \in [n] : (x_i - y_i)^2 > \Delta\}$ and $\boldsymbol{u} = \boldsymbol{x}_{\mathcal{J}}$. This iteration exists: each $i \in \mathcal{J}$ contributes more than $\Delta$ to $\|\boldsymbol{x} - \boldsymbol{y}\|^2 \le b$, so $|\mathcal{J}|\Delta \le b$ and $|\mathcal{J}| \le \lfloor b/\Delta \rfloor$; moreover $\Delta < (u_i - y_i)^2 \le b$ for every $i \in \mathcal{J}$ and $\sum_{i\in\mathcal{J}}(u_i - y_i)^2 \le b$. On every remaining coordinate $i \in [n] \setminus \mathcal{J}$,

$$0 \le (x_i - y_i)^2 - \tau f_i^2 \le \Delta,$$

where the lower bound uses $(x_i - y_i)^2 \ge f_i^2 \ge \tau f_i^2$ and the upper bound uses $(x_i - y_i)^2 \le \Delta$. Since the terms omitted below are nonnegative for the same reason,

$$\sum_{i\in[n]\setminus\mathcal{J}} \left((x_i - y_i)^2 - \tau f_i^2\right) \le \sum_{i\in[n]} \left((x_i - y_i)^2 - \tau f_i^2\right) = \|\boldsymbol{x} - \boldsymbol{y}\|^2 - \tau\|\boldsymbol{f}\|^2 \le b_\tau.$$

**A balanced partition for a closest vector.** The adjusted squared errors $(x_i - y_i)^2 - \tau f_i^2$ on the remaining coordinates thus lie in $[0, \Delta]$ and sum to at most $b_\tau$. We claim that among the partitions enumerated for this iteration there is one in which every part satisfies

$$\sum_{i\in\mathcal{I}_j} \left((x_i - y_i)^2 - \tau f_i^2\right) \le \frac{b_\tau}{4} + \frac{3}{2}\Delta \qquad (j = 1, \ldots, 4).$$

Consider the first call of $\mathcal{C}$, on the list $[n] \setminus \mathcal{J}$ padded to even length if necessary, where the padding entry is assigned adjusted error zero. The two halves of a rotation have adjusted errors summing to the total. Advancing the rotation by one position removes one entry from the first half and appends one, so the adjusted error of the first half changes by at most $\Delta$. The first half of the rotation starting at the middle position is the second half of the initial rotation, so as the rotation advances from the initial position to the middle position, the adjusted error of the first half moves from one side of half the total to the other in steps of at most $\Delta$. Hence some rotation has a first half with adjusted error within $\Delta$ of half the total, and both halves of that rotation have adjusted error at most half the total plus $\Delta$. Deleting the padding entry changes no adjusted error. Applying the same argument to the second calls, on each of these two halves, gives four parts with adjusted error at most a quarter of the total plus $\Delta/2 + \Delta$, which is at most $b_\tau/4 + 3\Delta/2$. This is Lemma C.4, which also bounds the sizes of the parts. Adding $\tau \sum_{i\in\mathcal{I}_j} f_i^2$ to both sides of the claimed bound gives, for that partition,

$$\sum_{i\in\mathcal{I}_j} (x_i - y_i)^2 \le \frac{b_\tau}{4} + \frac{3}{2}\Delta + \tau \sum_{i\in\mathcal{I}_j} f_i^2 \le b_j \qquad (j = 1, \ldots, 4).$$

Thus $\boldsymbol{x}_{\mathcal{I}_j}$ lies within squared distance $b_j$ of $\boldsymbol{y}_{\mathcal{I}_j}$, so it is output by $\mathcal{L}(\boldsymbol{y}_{\mathcal{I}_j}, b_j)$ (Lemma C.2), and it passes the filter because $(x_i - y_i)^2 \le \Delta$ on every remaining coordinate. Hence the four lists of this iteration contain the records of the four restrictions of $\boldsymbol{x}$.

**Finding a minimum-cost combination.** Two properties of $\mathcal{M}$ remain. First, retaining only a minimum-cost record for each element of $\mathcal{G}$ in each list loses no minimum-cost quadruple: the parts are disjoint, so the total cost of a quadruple is the sum of its four costs, and the constraint $\sum_j \boldsymbol{g}_{X_j} = \boldsymbol{s}'$ depends on the records only through their elements; replacing a record by the retained record with the same element therefore preserves the constraint and does not increase the total cost. Second, a pair from $\mathcal{P}(\mathcal{X}_1, \mathcal{X}_2; \mathbf{0}_m, 1)$ and a pair from $\mathcal{P}(\mathcal{X}_3, \mathcal{X}_4; \boldsymbol{s}', -1)$ have equal elements exactly when

$$\boldsymbol{g}_{X_1} + \boldsymbol{g}_{X_2} = \boldsymbol{s}' - (\boldsymbol{g}_{X_3} + \boldsymbol{g}_{X_4}),$$

that is, when $\sum_j \boldsymbol{g}_{X_j} = \boldsymbol{s}'$. For a fixed common element, the two sides are chosen independently, so the minimum-cost pair of each side combines to a minimum-cost quadruple attaining that element. Since each of the two procedures emits its attained elements in increasing order, the merge in $\mathcal{M}$ visits every common element exactly once, with a minimum-cost pair from each side, and keeps the smallest total. Hence $\mathcal{M}$ returns a quadruple of minimum total cost with $\sum_j \boldsymbol{g}_{X_j} = \boldsymbol{s}'$, or $\perp$ when none exists; Lemma C.3 proves this, including the ordering property of the pair generation.

In the iteration and partition found above, the restrictions of $\boldsymbol{x}$ form such a quadruple, since $\sum_j \mathbf{H}_{\mathcal{I}_j}\boldsymbol{x}_{\mathcal{I}_j} = \mathbf{H}\boldsymbol{x} - \mathbf{H}_{\mathcal{J}}\boldsymbol{u} = \boldsymbol{s}'$. The quadruple returned by $\mathcal{M}$ therefore has total cost at most $\sum_j \|\boldsymbol{x}_{\mathcal{I}_j} - \boldsymbol{y}_{\mathcal{I}_j}\|^2$, and the assembled candidate $\boldsymbol{z}$ satisfies

$$\|\boldsymbol{z} - \boldsymbol{y}\|^2 = \sum_{i \in \mathcal{J}} (u_i - y_i)^2 + \sum_{j=1}^{4} c_{X_j} \le \|\boldsymbol{x} - \boldsymbol{y}\|^2 \le b.$$

Since $\boldsymbol{z}$ is feasible and $\boldsymbol{x}$ is a closest feasible vector, in fact $\|\boldsymbol{z} - \boldsymbol{y}\|^2 = \|\boldsymbol{x} - \boldsymbol{y}\|^2$. Every accepted candidate is feasible, so $b_{\text{best}}$ never drops below $\|\boldsymbol{x} - \boldsymbol{y}\|^2$; hence $\boldsymbol{z}$ is accepted, after which $b_{\text{best}} = \|\boldsymbol{x} - \boldsymbol{y}\|^2$ and only candidates at this distance are accepted. The returned vector is therefore a closest feasible vector. If the feasible set is empty, no candidate is accepted, and the procedure returns $\perp$.

**The nonzero variant.** Every vector $\boldsymbol{x}$ returned by $\mathcal{E}_\tau^{\neq 0}$ has its fixed coordinate $x_i = z \neq 0$, and its remaining coordinates satisfy the constraints of the reduced problem, so

$$\mathbf{H}\boldsymbol{x} = z\mathbf{H}\boldsymbol{e}_i + \mathbf{H}_{[n]\setminus\{i\}}\boldsymbol{x}_{[n]\setminus\{i\}} = \mathbf{0}_m,$$
$$\|\boldsymbol{x} - \boldsymbol{y}\|^2 = (z - y_i)^2 + \|\boldsymbol{x}_{[n]\setminus\{i\}} - \boldsymbol{y}_{[n]\setminus\{i\}}\|^2 \le b.$$

Conversely, every nonzero solution $\boldsymbol{x}$ in the ball has a coordinate $x_i = z \neq 0$ with $(z - y_i)^2 \le \|\boldsymbol{x} - \boldsymbol{y}\|^2 \le b$, so the pair $(i, z)$ is enumerated, and the two identities show that $\boldsymbol{x}_{[n]\setminus\{i\}}$ satisfies the constraints of the reduced problem. For $n = 1$, the direct test decides that problem. Otherwise the correctness argument above applies to $\mathcal{E}_\tau^{\text{aux}}$ without change, since a prescribed right-hand side $\boldsymbol{s}$ alters only the required sum $\boldsymbol{s}'$ of the combination, so the auxiliary search returns a completion of $z$ at squared distance at most $\|\boldsymbol{x} - \boldsymbol{y}\|^2$. Taking the closest completion over all pairs $(i, z)$ therefore gives a closest nonzero solution, or $\perp$ exactly when none exists.

### 3.2.2 Computation costs

We bound the time and space of Algorithm 2 for $b = O(n)$, omitting factors polynomial in the binary input length. Coordinatewise rounding and its rejection test use polynomial time and space, so assume that the test does not reject; then $b_\tau \ge 0$.

**Enumerating coordinates, values, and partitions.** Since $\Delta = \lceil \sqrt{n} \rceil$ and $b = O(n)$, at most $\lfloor b/\Delta \rfloor = O(\sqrt{n})$ coordinates are fixed, so there are at most $(n+1)^{\lfloor b/\Delta \rfloor}$ sets $\mathcal{J}$. Each fixed coordinate takes an integer value $u_i$ with $(u_i - y_i)^2 \le b$, which allows at most $2\sqrt{b} + 1 = O(\sqrt{n})$ values. Hence the pairs $(\mathcal{J}, \boldsymbol{u})$ number $2^{O(\sqrt{n}\log(n+2))}$; this is where the growth condition $b = O(n)$ enters, through both the number of fixed coordinates and the number of their values. For each pair, the three calls of $\mathcal{C}$ output at most $n + 1$ pairs each, so there are at most $(n + 1)^3$ partitions, generated in polynomial time and space (Lemma C.1), and every part has at most $\lceil n/4 \rceil$ coordinates (Lemma C.4).

**List sizes and listing time.** Fix an analysis parameter $a > 0$ satisfying (10). Each vector in the $j$th list contributes at least $e^{-ab_j}$ to the product of the one-dimensional sums,

$$\prod_{i\in\mathcal{I}_j} \Theta_{y_i}(a) = \sum_{\boldsymbol{v}\in\mathbb{Z}^{\mathcal{I}_j}} e^{-a\|\boldsymbol{v}-\boldsymbol{y}_{\mathcal{I}_j}\|^2},$$

so $|\mathcal{X}_j| \le e^{ab_j} \prod_{i\in\mathcal{I}_j} \Theta_{y_i}(a)$. By Lemma C.2, the listing time is at most

$$\left(1 + e^{ab_j} \prod_{i\in\mathcal{I}_j} \Theta_{y_i}(a)\right) \text{poly},$$

since $b_j = O(n)$. The additive constant accounts for the initial test when the listing ball contains no integer point. By (50), $\Theta_{y_i}(a) = \Theta_{f_i}(a)$, and $|f_i| \le 1/2$, so (10) applies to every coordinate and gives

$$\begin{aligned}|\mathcal{X}_j| &\le e^{ab_j} \prod_{i\in\mathcal{I}_j} \Theta_{y_i}(a) \le e^{ab_j}\, \Theta(a)^{|\mathcal{I}_j|} e^{-a\tau \sum_{i\in\mathcal{I}_j} f_i^2} \\ &= e^{ab_\tau/4 + 2a\Delta}\, \Theta(a)^{|\mathcal{I}_j|}.\end{aligned}$$

Since $b_\tau \ge 0$ and $\Theta(a) \ge 1$, the last expression is at least one and absorbs the additive constant in the listing-time bound. The contribution $\tau \sum_{i\in\mathcal{I}_j} f_i^2$ of the rounding error to $b_j$ is cancelled by the decrease $e^{-a\tau \sum_{i\in\mathcal{I}_j} f_i^2}$ in the shifted sums. Filtering out the vectors with a coordinate error exceeding $\Delta$ cannot increase either bound. Now let $a = a(n) > 0$ satisfy $a = \Omega(1)$ and $a = O(\log(n+2))$. Since $\Theta$ is decreasing (Lemma A.2), $a = \Omega(1)$ gives $\ln \Theta(a) = O(1)$, and with $|\mathcal{I}_j| \le \lceil n/4 \rceil$ this yields $\Theta(a)^{|\mathcal{I}_j|} \le \Theta(a)^{n/4} 2^{O(1)}$; and $a = O(\log(n+2))$ gives $2a\Delta = O(\sqrt{n}\log(n+2))$. Therefore every list has size, and is generated in time, at most

$$2^{\frac{ab_\tau + n\ln\Theta(a)}{4\ln 2} + O(\sqrt{n}\log(n+2))}.$$

**Combining the four lists.** Let $\ell \ge 1$ bound the lengths of the four lists. The ordered procedure $\mathcal{P}$ reads the generator $\mathcal{Q}$ of Algorithm 10, which emits the at most $\ell^2$ pair records of two lists in nondecreasing order of $\boldsymbol{g}_W$; it stores a prefix tree of one list, one iterator and one heap entry per record of the other list, and the current output, but never the emitted pairs (Section C.1.3). Hence a complete traversal takes time $\ell^2$ poly and space $\ell$ poly, where the polynomial factors account for the record lengths, the tree and heap operations, and exact modular and rational arithmetic. Sorting and deduplicating the four lists take $\ell$ poly time and space, and the merge in $\mathcal{M}$ retains only the two current pairs and the best quadruple. Thus $\mathcal{M}$ runs in time $\ell^2$ poly and space $\ell$ poly (Lemma C.3): the time is quadratic in the list length because every pair may be generated, while the space stays linear because the pairs are generated one at a time and never stored together. With the list bound above, one combination takes time $2^{\frac{ab_\tau + n\ln\Theta(a)}{2\ln 2} + O(\sqrt{n}\log(n+2))}$ and space $2^{\frac{ab_\tau + n\ln\Theta(a)}{4\ln 2} + O(\sqrt{n}\log(n+2))}$.

**The complete search.** The pairs $(\mathcal{J}, \boldsymbol{u})$ and the partitions are processed sequentially. Each of the $2^{O(\sqrt{n}\log(n+2))}$ iterations generates four lists and combines them, so the total time is

$$2^{\frac{ab_\tau + n\ln\Theta(a)}{2\ln 2} + O(\sqrt{n}\log(n+2))}.$$

Apart from polynomial auxiliary storage, only the four current lists, their combination data, and the closest candidate are retained, and this storage is reused from one iteration to the next; hence the space is

$$2^{\frac{ab_\tau + n\ln\Theta(a)}{4\ln 2} + O(\sqrt{n}\log(n+2))}.$$

Every record has encoding length polynomial in the binary input length. All arithmetic and comparisons are exact on integers and rationals; no division in $\mathbb{Z}/q\mathbb{Z}$ is used.

**The nonzero variant.** There are $O(n(\sqrt{b}+1))$ choices of the fixed coordinate $i$ and its nonzero value $z$, processed sequentially; this count again uses $b = O(n)$. For $n = 1$ the direct tests use polynomial time and space, so let $n \geq 2$. The call for a choice $(i, z)$ has dimension $n-1$ and squared radius $b-(z-y_i)^2$, so its adjusted squared radius is

$$b-(z-y_i)^2-\tau\sum_{j\neq i} f_j^2 = b_\tau - \big((z-y_i)^2-\tau f_i^2\big) \leq b_\tau,$$

since $(z-y_i)^2 \geq f_i^2 \geq \tau f_i^2$. If the rounding test of the call rejects, the call costs polynomial time and space. Otherwise the reduced adjusted radius is nonnegative, and the bounds above apply to the reduced instance with the same analysis parameter $a$, with $ab_\tau + n\ln\Theta(a)$ replaced by

$$a\big(b_\tau - ((z-y_i)^2-\tau f_i^2)\big) + (n-1)\ln\Theta(a) \leq ab_\tau + n\ln\Theta(a),$$

where the inequality uses $\ln\Theta(a) \geq 0$. The time and space bounds of the variant are therefore the same as above, up to polynomial factors.

With $\boldsymbol{y} = \mathbf{0}_n$, the nonzero variant returns a shortest nonzero lattice vector whenever the supplied ball contains one. Taking $\tau = 0$, the inequality $\Theta_u(a) \leq \Theta(a)$ holds for every $a > 0$ by Lemma A.3, so every $a$ in the stated growth range can be used. For $\gamma$-CVP, a positive $\tau$ allows the cost bound to account for the target's rounding error. Each application chooses the radius and analyzes its total cost before selecting parameters.

# 4 $\gamma$-CVP

We solve the problem defined in Definition 2.6 by combining short solutions to linear congruences (Section 3.1) with exact search within a prescribed ball (Section 3.2). We present the algorithm and then analyze its correctness and complexity.

### 4.1 The $\gamma$-CVP algorithm

**Algorithm 3** $\gamma$-CVP

**Parameters:** A rational constant $\gamma > 1$.
**Input:** $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$ and $\boldsymbol{y} \in \mathbb{Q}^n$, with $q \geq 2$, $n \geq 1$, $m \geq 0$, and $\delta \leq 3/2$.
**Output:** A point of $\Lambda$.

$\boldsymbol{z} \leftarrow \lfloor \boldsymbol{y} \rceil$; **if** $m = 0$ **then return** $\boldsymbol{z}$
**if** $q^{8m} \leq 2^n$ **then return** $\mathcal{A}(\mathbf{H}, \boldsymbol{y})$
$\hat{\boldsymbol{x}} \leftarrow q\,\lfloor \boldsymbol{y}/q \rceil$; $b_{\text{best}} \leftarrow \|\hat{\boldsymbol{x}} - \boldsymbol{y}\|^2$
**if** $q = 2$ and $m \geq 3n/4$ **then**
    Compute a basis $\mathbf{B}$ of $\ker \mathbf{H}$ over $\mathbb{F}_2$
    **if** $\dim \ker \mathbf{H} > \max\{0, n-m\} + \lceil n^{2/3} \rceil$ **then return** $\hat{\boldsymbol{x}}$
    **for** $\boldsymbol{\alpha} \in \{0,1\}^{\dim \ker \mathbf{H}}$ **do**
        $\boldsymbol{x} \leftarrow \mathbf{B}\boldsymbol{\alpha} \bmod 2$; $\boldsymbol{x} \leftarrow \boldsymbol{x} + 2\,\lfloor (\boldsymbol{y} - \boldsymbol{x})/2 \rceil$
        **if** $\|\boldsymbol{x} - \boldsymbol{y}\|^2 < b_{\text{best}}$ **then** $(\hat{\boldsymbol{x}}, b_{\text{best}}) \leftarrow (\boldsymbol{x}, \|\boldsymbol{x} - \boldsymbol{y}\|^2)$
    **end for**
    **return** $\hat{\boldsymbol{x}}$
**end if**
Choose the correction parameters from $n, m, q$ as specified below
Set $d, \ell, \ell', n_0$ by (15)
**if** $n_0 \leq 0$ or ($q = 2$ and $n_0 \leq m$) **then return** $\mathcal{A}(\mathbf{H}, \boldsymbol{y})$
**if** $q \geq 3$ and the stated feasibility checks fail **then return** $\mathcal{A}(\mathbf{H}, \boldsymbol{y})$
$b \leftarrow (\|\boldsymbol{z} - \boldsymbol{y}\|^2 + cd)/\gamma^2$
$\boldsymbol{x} \leftarrow \mathcal{E}_{\gamma^{-2}}(\mathbf{H}, \boldsymbol{y}, b)$
**if** $\boldsymbol{x} \neq \perp$ **then return** $\boldsymbol{x}$
$[\mathbf{A}\ \mathbf{C}_0\ \mathbf{C}_1] \leftarrow \mathbf{H}_{1:d}$ with widths $n_0, \ell, \ell'$; $\boldsymbol{s} \leftarrow -\mathbf{H}\boldsymbol{z}$
**for** $i = 1, \ldots, r$ **do**
    $\boldsymbol{u} \xleftarrow{\$} \{0,1\}^{\ell}$, $j \xleftarrow{\$} [\ell']$, $\sigma \xleftarrow{\$} \{-1, 1\}$
    $\boldsymbol{t} \leftarrow \sigma(\boldsymbol{s} - \mathbf{C}_0\boldsymbol{u} - \mathbf{C}_1\boldsymbol{e}_j)$; $\boldsymbol{v} \leftarrow \mathcal{S}_q(\mathbf{A}, \boldsymbol{t})$
    **if** $\boldsymbol{v} \neq \perp$ **then**
        $\boldsymbol{w} \leftarrow (\sigma\boldsymbol{v}^\top, \boldsymbol{u}^\top, \boldsymbol{e}_j^\top, \mathbf{0}_{n-d}^\top)^\top$; $\boldsymbol{x} \leftarrow \boldsymbol{z} + \boldsymbol{w}$
        **if** $\|\boldsymbol{x} - \boldsymbol{y}\|^2 < b_{\text{best}}$ **then** $(\hat{\boldsymbol{x}}, b_{\text{best}}) \leftarrow (\boldsymbol{x}, \|\boldsymbol{x} - \boldsymbol{y}\|^2)$
    **end if**
**end for**
**return** $\hat{\boldsymbol{x}}$

The algorithm first handles special cases using deterministic procedures. For the remaining inputs, it performs exact search within a prescribed ball, followed by randomized correction attempts if the search fails.

**Choosing the correction parameters.** For $\delta \leq 1$, fix rational short-solution parameters satisfying Corollary 3.3 at $\rho = 1$, including $h_0 > 1$ and the conditions for even moduli. With $\beta = 2t_0$ as in Section 3.1, fix a rational $c > \beta$, and additionally require

$$0 < \beta < 1/4, \qquad h(\beta) > 3/4. \tag{11}$$

The same $\beta, c$ then serve both the common short solver and the binary sampler. A concrete choice is verified in Section 4.4.

For $1 < \delta \le 3/2$, the correction branch has $q \ge 3$. For $\delta \in [1, 3/2]$, let $u_\delta \in (0, 1/3)$ be the unique solution of

$$\mathcal{H}(u_\delta, u_\delta, 1 - 2u_\delta) = \delta. \tag{12}$$

The entropy is strictly increasing on this interval. Define

$$t_i(\delta) = c_i(u_\delta + 10^{-7}) \quad (0 \le i \le 3), \qquad (c_0, c_1, c_2, c_3) = \left(1, \frac{7}{10}, \frac{47}{100}, \frac{6}{25}\right), \tag{13}$$

and the limiting length parameters

$$\beta(\delta) = 2t_0(\delta), \qquad c(\delta) = \beta(\delta) + 2 \cdot 10^{-5}. \tag{14}$$

Here the $c_i$ are coordinate ratios and $c(\delta)$ is the squared-distance allowance. For the algorithm, compute a rational upper approximation $\widehat{u}_n$ satisfying

$$0 \le \widehat{u}_n - u_\delta \le \min\{n^{-2}, 10^{-12}\},$$

and use the rational proportions $t_i = c_i(\widehat{u}_n + 10^{-7})$. Set $\beta = 2t_0$, $c = \beta + 2 \cdot 10^{-5}$, $k = 3$, $\eta = 10^{-8}$, and $\nu = 16$. Certified bisection computes $\widehat{u}_n$ in polynomial time: the entropy derivative has a positive lower bound on the relevant compact interval. The closed interval $[1, 3/2]$ is used for the verification in Lemma F.5; at $\delta = 1$ the algorithm uses the preceding fixed choice.

**Dimensions and feasibility checks.** Corrections use the first $d$ coordinates, where

$$d = \begin{cases} \min\{n, \lceil \delta n \rceil\}, & q \ge 3, \\ n, & q = 2, \end{cases} \qquad \ell = \lceil d^{2/3} \rceil, \qquad \ell' = \lceil \ell/\beta \rceil, \qquad n_0 = d - \ell - \ell'. \tag{15}$$

Use $\mathcal{A}$ if $n_0 \le 0$ or if $q = 2$ and $n_0 \le m$. For $q \ge 3$ with $n_0 > 0$, let $\mathcal{S}$ be the procedure of Corollary 3.3 in dimension $n_0$ with the chosen parameters. Its actual equation rate is

$$\rho_n = \frac{m \log q}{n_0} = \max\{1, \delta\} + O(n^{-1/3})$$

on the correction branch. Check the coordinate inequalities (3) by exact rational arithmetic. Check the strict conditions (5) and (8), and $h_0 > \rho_n$, at this actual rate using outward intervals of width at most $n^{-2}$. Use $\mathcal{A}$ unless all these conditions are certified. The fixed margins for $\delta \le 1$ and the uniform margins in Lemma F.5 ensure that these fallbacks occur only in bounded dimension. For $q = 2$ in the correction branch, use the binary sampler $\mathcal{S}_2$ of Lemma 4.2.

**Exact search and corrections.** Let $\boldsymbol{z} = \lfloor \boldsymbol{y} \rceil$ and $\boldsymbol{f} = \boldsymbol{z} - \boldsymbol{y}$. A correction $\boldsymbol{w}$ yields a lattice point exactly when $\mathbf{H}\boldsymbol{w} = -\mathbf{H}\boldsymbol{z}$. We seek one satisfying

$$\|\boldsymbol{z} + \boldsymbol{w} - \boldsymbol{y}\|^2 \le \|\boldsymbol{f}\|^2 + cd. \tag{16}$$

Since $\boldsymbol{w}$ is integral and $|f_i| \le 1/2$, the length bound $\|\boldsymbol{w}\|^2 \le \beta d$ gives

$$2\langle \boldsymbol{f}, \boldsymbol{w} \rangle \le \sum_{i=1}^{d} |w_i| \le \|\boldsymbol{w}\|^2 \le \beta d.$$

Thus the length bound alone gives squared distance at most $\|\boldsymbol{f}\|^2 + 2\beta d$. A small average correction will instead prove (16) for any fixed $c > \beta$, arbitrarily close to $\beta$. Before generating corrections, use $\mathcal{E}_{\gamma^{-2}}$ with

$$b = (\|\boldsymbol{f}\|^2 + cd)/\gamma^2. \tag{17}$$

If this ball contains a lattice point, exact search returns a closest one. Otherwise the optimum squared distance exceeds $b$, and (16) suffices to solve $\gamma$-CVP.

Partition the first $d$ columns as $[\mathbf{A}\ \mathbf{C}_0\ \mathbf{C}_1]$ with widths $n_0, \ell, \ell'$. A uniform $\boldsymbol{u} \in \{0,1\}^\ell$, a uniform column index $j \in [\ell']$, and an independent uniform sign shift and symmetrize the input to the short solver. The appended coordinates contribute at most $\ell + 1$ to the squared length. Let $\mathcal{S}_q = \mathcal{S}$ for $q \geq 3$ and $\mathcal{S}_q = \mathcal{S}_2$ for $q = 2$. Each of the $r = \lceil Cn \rceil$ attempts uses fresh independent bits, including the bits inside $\mathcal{S}_q$. The constant $C$ is chosen uniformly over the two parameter ranges in Theorem 4.7.

The tests $q^{8m} \leq 2^n$ and, for $q = 2$, $m \geq 3n/4$ implement $\delta \leq 1/8$ and $\delta \geq 3/4$, respectively, using exact integer arithmetic. For $q \geq 3$, $d = \min\{n, \lceil \log(q^m) \rceil\}$ is obtained from the binary length of $q^m$, distinguishing powers of two; the test $q^m \leq 2^n$ selects the parameter range. The algorithm keeps only the closest candidate; corresponding attempts use fixed blocks of random bits when executions at different targets are compared.

## 4.2 Correctness

Algorithm 3 uses two auxiliary procedures, an exact procedure for small $q^m$ and a short-solution procedure for binary congruences. We state their guarantees here and give the implementations and proofs in Appendix D.

**Lemma 4.1** (Exact dynamic programming over $(\mathbb{Z}/q\mathbb{Z})^m$)**.** *Let $q \geq 2$, $m \geq 1$, and $n \geq 1$ be integers. There is a deterministic procedure $\mathcal{A}(\mathbf{H}, \boldsymbol{y})$ that, for every $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ and $\boldsymbol{y} \in \mathbb{Q}^n$, returns a point of $\Lambda$ closest to $\boldsymbol{y}$. Its time and space are, respectively,*

$$nq^{m+1}\,\mathrm{poly}, \qquad nq^m\,\mathrm{poly}\ .$$

*In terms of the density $\delta = m \log_2 q/n$, these are at most $2^{2\delta n}\,\mathrm{poly}$ and $2^{\delta n}\,\mathrm{poly}$.*

**Lemma 4.2** (Binary short-solution sampler)**.** *Fix a rational constant $0 < \beta < 1/2$. Let $n_0 \geq 1$ and $1 \leq m \leq n_0$ be integers, and put*

$$\mathcal{B} = \{\boldsymbol{x} \in \mathbb{F}_2^{n_0} : \mathrm{wt}(\boldsymbol{x}) \leq \lfloor \beta n_0 \rfloor\}.$$

*There is a randomized procedure $\mathcal{S}_2(\mathbf{A}, \boldsymbol{t})$ that, on input $(\mathbf{A}, \boldsymbol{t}) \in \mathbb{F}_2^{m \times n_0} \times \mathbb{F}_2^m$, returns either $\perp$ or $\boldsymbol{x} \in \mathcal{B}$ with $\mathbf{A}\boldsymbol{x} = \boldsymbol{t}$, represented by integer coordinates in $\{0,1\}$. Every execution uses time $(2^{n_0}/|\mathcal{B}|)\,\mathrm{poly}(n_0, m)$, polynomial space, and polynomially many independent random bits. Except for a fraction at most*

$$2^{m-n_0} + 32\,2^m/|\mathcal{B}| \tag{18}$$

*of matrices $\mathbf{A}$, its average success probability over a uniform right-hand side and independent internal randomness is at least $105/128 > 3/4$.*

**Success and mean bounds for every right-hand side.** To control success probabilities and mean correction vectors across right-hand sides, we average over random subset sums. A matrix whose columns belong to an abelian group $\mathcal{G}$ is interpreted as the homomorphism taking an integer vector to the corresponding sum of columns; thus $\mathbf{C}\boldsymbol{u}$, for $\mathbf{C} \in \mathcal{G}^\ell$ and $\boldsymbol{u} \in \{0,1\}^\ell$, is the sum of the columns selected by $\boldsymbol{u}$. The next lemma gives the required second-moment identity.

**Lemma 4.3.** *Let $\mathcal{G}$ be a finite abelian group and let $\ell \geq 1$ be an integer. Let $s$ be uniform in $\mathcal{G}$, and let $\Phi : \mathcal{G} \to \mathbb{R}$ be a fixed function satisfying $\mathbb{E}_s[\Phi(s)] = 0$. Choose $\mathbf{C} \in \mathcal{G}^\ell$ with independent uniform columns, independently of $s$, and let $\boldsymbol{u}$ be independent and uniform in $\{0,1\}^\ell$. Then*

$$\mathbb{E}_{\mathbf{C},s}\Big[\big(\mathbb{E}_{\boldsymbol{u}}[\Phi(s - \mathbf{C}\boldsymbol{u})]\big)^2\Big] = 2^{-\ell}\mathbb{E}_s\Big[\Phi(s)^2\Big]\ .$$

*Proof.* This is the collision calculation of Erdős and Rényi [ER65, Section 1]. Let $\boldsymbol{v}$ be an independent copy of $\boldsymbol{u}$. Expanding the square expresses the left-hand side as

$$\mathbb{E}[\Phi(s-\mathbf{C}\boldsymbol{u})\,\Phi(s-\mathbf{C}\boldsymbol{v})]\,,$$

the expectation being over $\mathbf{C}, s, \boldsymbol{u}, \boldsymbol{v}$. The event $\boldsymbol{u}=\boldsymbol{v}$ has probability $2^{-\ell}$; conditioned on this event, the expectation of the product is $\mathbb{E}_s[\Phi(s)^2]$, since $s-\mathbf{C}\boldsymbol{u}$ is uniform in $\mathcal{G}$ for every fixed $\mathbf{C}$ and $\boldsymbol{u}$. For fixed distinct $\boldsymbol{u},\boldsymbol{v}$, choose $j$ with $u_j - v_j \in \{-1,1\}$ and fix all columns of $\mathbf{C}$ other than the $j$th column $\mathbf{C}\boldsymbol{e}_j$, where $\boldsymbol{e}_j$ is the $j$th unit vector. The pair $(s-\mathbf{C}\boldsymbol{u},\ s-\mathbf{C}\boldsymbol{v})$ is then an affine function of $(s, \mathbf{C}\boldsymbol{e}_j) \in \mathcal{G}^2$ whose linear part is the integer matrix $\left(\begin{smallmatrix}1 & -u_j\\ 1 & -v_j\end{smallmatrix}\right)$. Its determinant $u_j - v_j = \pm 1$ makes this matrix invertible over $\mathbb{Z}$, so the affine function is a bijection of $\mathcal{G}^2$, and the two components of the pair are independent and uniform. The expected product is therefore zero by the mean-zero assumption. Only the event $\boldsymbol{u}=\boldsymbol{v}$ contributes, proving the identity. □

We use this identity to turn average success over right-hand sides into a guarantee for every right-hand side, while controlling the mean output vector.

**Lemma 4.4** (Average success to every right-hand side)**.** *Let $\mathcal{G}$ be a finite abelian group of size $M$, let $n_0 \geq 1$ be an integer, and fix a homomorphism $\mathbf{A}: \mathbb{Z}^{n_0} \to \mathcal{G}$. Fix a real number $B \geq 0$. Suppose a randomized procedure $\mathcal{S}(\mathbf{A}, t)$ returns either $\perp$ or $\boldsymbol{v} \in \mathbb{Z}^{n_0}$ with $\mathbf{A}\boldsymbol{v} = t$ and $\|\boldsymbol{v}\| \leq B$. Assume its average success probability, over uniform $t \in \mathcal{G}$ and independent internal randomness, is at least $p > 0$. Let $\ell, \ell' \geq 1$ be integers, and choose $\mathbf{C}_0 \in \mathcal{G}^{\ell}$ and $\mathbf{C}_1 \in \mathcal{G}^{\ell'}$ with independent uniform columns, independently of $\mathbf{A}$ and the procedure's randomness.*

*For a right-hand side $s \in \mathcal{G}$, an attempt samples independent uniform $\boldsymbol{u} \in \{0,1\}^{\ell}$, $j \in [\ell']$, and $\sigma \in \{-1,1\}$, and calls*

$$\boldsymbol{v} \leftarrow \mathcal{S}(\mathbf{A}, \sigma(s - \mathbf{C}_0\boldsymbol{u} - \mathbf{C}_1\boldsymbol{e}_j))$$

*using independent internal randomness, where $\boldsymbol{e}_j \in \{0,1\}^{\ell'}$ is the $j$th unit vector. On success, it returns $\boldsymbol{w} = (\sigma\boldsymbol{v}^\top, \boldsymbol{u}^\top, \boldsymbol{e}_j^\top)^\top$. Fix $L > 0$, $0 < \eta < p$, $0 < \theta < 1$, and $\alpha > 0$, and put*

$$\xi = \min\{1, \alpha/\eta^2\}, \qquad p_* = (1-\theta)(p-\eta)\ .$$

*Except for a fraction of pairs $(\mathbf{C}_0, \mathbf{C}_1)$ at most*

$$(1 + B^2/L^2)\,\frac{2^{-\ell}}{\alpha} + M\binom{\ell'}{\lceil\theta\ell'\rceil}\xi^{\lceil\theta\ell'\rceil}, \tag{19}$$

*every $s \in \mathcal{G}$ simultaneously satisfies*

$$\Pr[\text{success}] \geq p_*, \qquad \|\mathbb{E}[\boldsymbol{w} \mid \text{success}]\| \leq \frac{\eta L + \theta B + \sqrt{\ell+1}}{p_*}.$$

*Every successful output satisfies $[\mathbf{A}\ \mathbf{C}_0\ \mathbf{C}_1]\boldsymbol{w} = s$ and $\|\boldsymbol{w}\|^2 \leq B^2 + \ell + 1$. An attempt makes one call to $\mathcal{S}$ and otherwise uses polynomially many group operations. If the assumed average-success property of a random $\mathbf{A}$ fails with probability $\zeta$, add $\zeta$ to* (19)*.*

*Proof.* For each $t$, symmetrize the procedure by choosing a uniform sign, solving for $\sigma t$, and multiplying its output by $\sigma$. Let $p(t)$ be the success probability of this symmetrized procedure and let $\boldsymbol{\mu}(t)$ be its expected output, counting failure as the zero vector. Then

$$\bar{p} := \mathbb{E}_t[p(t)] \geq p, \qquad \mathbb{E}_t[\boldsymbol{\mu}(t)] = \mathbf{0}_{n_0}, \qquad 0 \leq p(t) \leq 1, \qquad \|\boldsymbol{\mu}(t)\| \leq B.$$

Indeed, $\boldsymbol{\mu}(t)$ is half the difference between the expected outputs of the original procedure on $t$ and on $-t$, each counting failure as the zero vector; both have norm at most $B$, and $t \mapsto -t$ preserves the uniform distribution on $\mathcal{G}$. The use of unconditional means here is essential, since the success probability may depend on $t$.

Put

$$p_0(s) = \mathbb{E}_{\boldsymbol{u}}[p(s - \mathbf{C}_0\boldsymbol{u})]\,, \qquad \boldsymbol{\mu}_0(s) = \mathbb{E}_{\boldsymbol{u}}[\boldsymbol{\mu}(s - \mathbf{C}_0\boldsymbol{u})]\,.$$

Apply Lemma 4.3 with $\mathbf{C} = \mathbf{C}_0$ to each coordinate of the mean-zero function $t \mapsto (p(t) - \bar{p}, \boldsymbol{\mu}(t)/L) \in \mathbb{R}^{n_0+1}$ and add the resulting identities. The second moment of this function is at most $1 + B^2/L^2$, so

$$\mathbb{E}_{\mathbf{C}_0,s}\Big[(p_0(s) - \bar{p})^2 + \|\boldsymbol{\mu}_0(s)\|^2/L^2\Big] \le (1 + B^2/L^2)2^{-\ell}.$$

Markov's inequality shows that, except for the first fraction in (19), the expectation over $s$ with $\mathbf{C}_0$ fixed is at most $\alpha$. Fix such a $\mathbf{C}_0$. Another application of Markov's inequality gives

$$\frac{|E|}{M} \le \xi, \qquad E = \{s \in \mathcal{G} : |p_0(s) - \bar{p}| > \eta \text{ or } \|\boldsymbol{\mu}_0(s)\| > \eta L\}.$$

For each fixed $s$, the shifts $s - \mathbf{C}_1\boldsymbol{e}_j$, $j \in [\ell']$, are independent and uniform in $\mathcal{G}$. The probability that at least $\lceil\theta\ell'\rceil$ of them belong to $E$ is at most $\binom{\ell'}{\lceil\theta\ell'\rceil}\xi^{\lceil\theta\ell'\rceil}$, by a union bound over subsets of $\lceil\theta\ell'\rceil$ columns. Taking a union bound over all $M$ choices of $s$ yields the second term in (19).

On the complementary event, for every $s$, fewer than a fraction $\theta$ of its shifts are exceptional. The success probability of the attempt is $\frac{1}{\ell'}\sum_{j=1}^{\ell'} p_0(s - \mathbf{C}_1\boldsymbol{e}_j)$ and is at least $p_*$. Its unconditional first-block mean is $\frac{1}{\ell'}\sum_{j=1}^{\ell'} \boldsymbol{\mu}_0(s - \mathbf{C}_1\boldsymbol{e}_j)$, whose norm is at most $\eta L + \theta B$. On success the appended coordinates have norm at most $\sqrt{\ell+1}$, and they are zero on failure for the purpose of taking expectations. Thus the norm of the full unconditional mean is at most $\eta L + \theta B + \sqrt{\ell+1}$. Dividing by the success probability proves the conditional-mean bound. The congruence and squared-norm claims follow directly from the construction. Conditioning first on a good $\mathbf{A}$ and then adding its exceptional probability proves the last assertion. □

**Guarantees for a single correction attempt.** The following consequence of the finite-group averaging lemma applies to either short solver with $0 < \beta < 1$ and any fixed density bound $D$, as specified in the statement. The matrix may have unused columns after the correction block; these columns do not affect the good-matrix event.

**Lemma 4.5.** *Fix rational $0 < \beta < 1$, constants $p \in (0,1)$, $\varepsilon > 0$, $\alpha_0 > 0$, and $D > 0$. Let $q \ge 2$, $m \ge 1$, $n \ge 1$, and $d$ be integers satisfying $\alpha_0 n \le d \le n$ and $\delta n \le Dd$. Put $\ell = \lceil d^{2/3}\rceil$, $\ell' = \lceil \ell/\beta\rceil$, and $n_0 = d - \ell - \ell' > 0$. Suppose a randomized short solver on $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n_0}$ and $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$ returns either $\perp$ or $\boldsymbol{v} \in \mathbb{Z}^{n_0}$ with $\mathbf{A}\boldsymbol{v} = \boldsymbol{t}$ and $\|\boldsymbol{v}\|^2 \le \beta n_0$. Assume that all but a $2^{-\Omega(n_0^{2/3})}$ fraction of matrices have average success at least $p$ over independent uniform $\boldsymbol{t}$ and solver randomness. Partition the first $d$ columns of $\mathbf{H}$ as in Algorithm 3 and use this solver in one correction attempt. For all sufficiently large $n$, all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices $\mathbf{H}$ satisfy, simultaneously for every $\boldsymbol{s} \in (\mathbb{Z}/q\mathbb{Z})^m$,*

$$\Pr[\text{success}] \ge p/2, \qquad \|\mathbb{E}[\boldsymbol{w} \mid \text{success}]\| \le \varepsilon\sqrt{d}.$$

*Every successful output is supported on the first $d$ coordinates and satisfies $\mathbf{H}\boldsymbol{w} = \boldsymbol{s}$ and $\|\boldsymbol{w}\|^2 \le \beta d$.*

*Proof.* We have $n_0 = \Theta(d) = \Theta(n)$ and $\ell = \Theta(n^{2/3})$. Condition on a matrix $\mathbf{A}$ with average success at least $p$ and apply Lemma 4.4 with $\mathcal{G} = (\mathbb{Z}/q\mathbb{Z})^m$, $L = \sqrt{d}$, $B = \sqrt{\beta n_0}$, $\alpha = (1+\beta)2^{-\ell/2}$, and

$\eta = \theta$. Choose a fixed sufficiently small $\theta > 0$ so that

$$(1-\theta)(p-\theta) > p/2, \qquad \frac{2\theta(1+\sqrt{\beta})}{p} < \varepsilon/2.$$

The first exceptional fraction is at most $2^{-\ell/2}$. The fraction of exceptional shifts, in the notation of that lemma, satisfies

$$\xi \le \frac{1+\beta}{\theta^2} 2^{-\ell/2}. \tag{20}$$

Thus the second exceptional fraction is at most

$$q^m \binom{\ell'}{\lceil \theta \ell' \rceil} \xi^{\lceil \theta \ell' \rceil} \le 2^{Dd - \Omega(\ell^2)} = 2^{-\Omega(d^{4/3})}.$$

The conditional mean is bounded by

$$\frac{2}{p}\left(\theta(1+\sqrt{\beta})\sqrt{d} + \sqrt{\ell+1}\right) \le \varepsilon\sqrt{d}$$

for sufficiently large $n$. The squared-length bound follows from

$$\beta n_0 + \ell + 1 \le \beta d - \beta\ell + 1 \le \beta d.$$

For the last inequality, note that $d \le \ell^2$ and $n_0 > 0$ imply

$$\ell^2 \ge d > \ell + \ell' \ge \ell + \ell/\beta,$$

and hence $\beta\ell > 1 + \beta > 1$. The good-$\mathbf{A}$ exception contributes $2^{-\Omega(n^{2/3})}$. Append zeros to the remaining coordinates. The strict margins absorb the $2^{-\Omega(n^2)}$ bounded-bit sampling error for $j$, as in Appendix B.6. This proves the implemented attempt's guarantees. □

**Repeated corrections and the $\gamma$-CVP guarantee.** The small conditional expectation must yield actual candidates at the required distance. We obtain them by repeating the attempt and keeping the closest candidate. The average of the successful corrections is used to prove the distance bound.

**Lemma 4.6.** *Fix constants $0 < \beta < 1$, $p \in (0,1)$, $c > \beta$, and $D > 0$. There is a rational constant $C > 0$, depending only on $p, \beta, c, D$, with the following property. Let $n \ge 1$, $m \ge 0$, and $q \ge 2$ be integers with $\delta \le D$, and fix $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$. Suppose that for every $\boldsymbol{s} \in (\mathbb{Z}/q\mathbb{Z})^m$, a randomized attempt returns either $\perp$ or an integer vector $\boldsymbol{w}$ with*

$$\mathbf{H}\boldsymbol{w} = \boldsymbol{s}, \qquad \|\boldsymbol{w}\|^2 \le \beta n,$$

*and satisfies*

$$\Pr[\text{success}] \ge p/2, \qquad \|\mathbb{E}[\boldsymbol{w} \mid \text{success}]\| \le \frac{c-\beta}{8}\sqrt{n}.$$

*The matrix $\mathbf{H}$ and the vector $\boldsymbol{s}$ are fixed in these probabilities and expectations. Run $r = \lceil Cn \rceil$ independent attempts for each $\boldsymbol{s}$, using the same random bits for corresponding attempts when different $\boldsymbol{s}$ are compared. For sufficiently large $n$, with probability at least $1 - 2^{-\Omega(n)}$ over these bits, every $\boldsymbol{s}$ has a successful attempt, and the average of its successful outputs, denoted $\bar{\boldsymbol{w}}(\boldsymbol{s})$, satisfies*

$$\|\bar{\boldsymbol{w}}(\boldsymbol{s})\| \le (c-\beta)\sqrt{n}.$$

*Proof.* Put $\varepsilon = (c-\beta)/8$ and fix $\boldsymbol{s}$. Let $\mathcal{I} \subseteq \{1,\dots,r\}$ be the indices of the successful attempts. Their success indicators are independent, each with probability at least $p/2$. The lower-tail bound in Lemma 2.4 gives

$$\Pr[|\mathcal{I}| < pr/4] \le e^{-pr/16}.$$

Conditioned on the set $\mathcal{I}$, the successful outputs remain independent, each with the law of one output conditioned on success. After subtracting their common conditional expectation, their norms are at most $(\sqrt{\beta}+\varepsilon)\sqrt{n}$. For $|\mathcal{I}| \ge pr/4$, Lemma 2.5 therefore gives

$$\Pr\left[ \|\bar{\boldsymbol{w}}(\boldsymbol{s}) - \mathbb{E}[\boldsymbol{w} \mid \text{success}]\| > (\sqrt{\beta}+\varepsilon)\sqrt{\frac{n}{|\mathcal{I}|}} + \frac{c-\beta}{2}\sqrt{n} \;\middle|\; \mathcal{I} \right] \le \exp\left( -\frac{|\mathcal{I}|(c-\beta)^2}{8(\sqrt{\beta}+\varepsilon)^2} \right).$$

For sufficiently large $n$, the first term in this threshold is at most $(c-\beta)\sqrt{n}/4$. Together with the conditional-mean bound and the choice of $\varepsilon$, this shows that the probability of no success or of $\|\bar{\boldsymbol{w}}(\boldsymbol{s})\| > (c-\beta)\sqrt{n}$ is at most

$$e^{-pr/16} + \exp\left( -\frac{pr(c-\beta)^2}{32(\sqrt{\beta}+\varepsilon)^2} \right). \tag{21}$$

No sum over possible sets $\mathcal{I}$ is needed: the conditional bound is uniform over every set with $|\mathcal{I}| \ge pr/4$.

For the simultaneous assertion, compare the executions at all $\boldsymbol{s}$ using the same random bits for corresponding attempts. Each individual execution retains its prescribed distribution. Since $q^m \le 2^{Dn}$, summing (21) over all $\boldsymbol{s}$ gives a bound of $2^{-\Omega(n)}$ when $C$ is sufficiently large. Its required size depends only on $p, \beta, c, D$, with $\varepsilon$ chosen in terms of $c-\beta$. This proves the simultaneous assertion for all right-hand sides. □

**Theorem 4.7** (Correctness)**.** *Fix a rational constant $\gamma > 1$ and a constant $p \in (0, 3/4]$. Suppose the parameters of Algorithm 3 are chosen from $n, m, q$ alone. Let $(\beta, c)$ range over a fixed compact subset of $\{(\beta,c) : 0 < \beta < 1,\ c > \beta\}$. On the binary correction branch, additionally require* (11) *with uniform positive margins. For $q \ge 3$, suppose the chosen procedure $\mathcal{S}$ has the following properties whenever the algorithm reaches the correction branch in sufficiently large dimension. On input $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n_0}$ and $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, it returns either $\perp$ or a vector $\boldsymbol{v} \in \{-1,0,1\}^{n_0}$ satisfying*

$$\mathbf{A}\boldsymbol{v} = \boldsymbol{t}, \qquad \|\boldsymbol{v}\|^2 \le \beta n_0.$$

*All but a $2^{-\Omega(n_0^{2/3})}$ fraction of matrices $\mathbf{A}$ have success probability at least $p$, averaged over independent uniform $\boldsymbol{t}$ and internal randomness, with $\mathbf{A}$ fixed. This bound is uniform over the chosen parameters and holds at the actual equation rate $m \log q / n_0$.*

*There is a rational constant $C > 0$, depending only on $p$ and the fixed parameter bounds, such that the following holds for all sufficiently large integers $n$ and all integers $q \ge 2$, and $m \ge 0$ with $\delta \le 3/2$. Run Algorithm 3 with $\mathcal{S}$ for $q \ge 3$, $\mathcal{S}_2$ for $q = 2$, and $r = \lceil Cn \rceil$ attempts. For all but a $2^{-\Omega(n^{2/3})}$ fraction of uniform matrices $\mathbf{H}$, holding $\mathbf{H}$ fixed,*

$$\Pr_{\omega}\left[ \begin{array}{c} \textit{for every } \boldsymbol{y} \in \mathbb{Q}^n, \textit{ the algorithm returns } \boldsymbol{x} \in \Lambda \textit{ with} \\ \|\boldsymbol{x}-\boldsymbol{y}\| \le \gamma \min_{\boldsymbol{a}\in\Lambda} \|\boldsymbol{a}-\boldsymbol{y}\| \end{array} \right] \ge 1 - 2^{-\Omega(n)}.$$

*Here $\omega$ consists of the correction attempts' random bits; corresponding attempts use the same bits when different targets are compared.*

*Proof.* The branches $\delta = 0$ and $0 < \delta \le 1/8$ are exact for every input, by rounding and Lemma 4.1, respectively. For the binary branch $\delta \ge 3/4$, the cosets $\boldsymbol{c} + 2\mathbb{Z}^n$ for $\boldsymbol{c} \in \ker \mathbf{H} \subseteq \mathbb{F}_2^n$ partition $\Lambda$, and the algorithm finds a closest point in each. Hence passing the cutoff makes this branch exact for all targets. Moreover,

$$\mathbb{E}_{\mathbf{H}}[|\ker \mathbf{H}|] = 1 + (2^n - 1)2^{-m} \le 2^{\max\{0,n-m\}+1}.$$

Markov's inequality bounds the fraction rejected by $\dim \ker \mathbf{H} > \max\{0, n-m\} + \lceil n^{2/3} \rceil$ by $2^{1-\lceil n^{2/3} \rceil}$.

On every correction branch,

$$n/8 \le d \le n, \qquad \delta n \le (3/2)d, \qquad n_0 = d - O(d^{2/3}) = \Theta(n),$$

for sufficiently large $n$, uniformly because $\beta$ is bounded away from zero. For $q \ge 3$, the assumed guarantee for $\mathcal{S}$ applies to this actual instance. For $q = 2$, we have $\delta < 3/4$, $d = n$, and $n_0 = n - O(n^{2/3}) > m$. The estimate

$$\log \sum_{j=0}^{\lfloor \beta n_0 \rfloor} \binom{n_0}{j} = h(\beta) n_0 + O(\log n_0)$$

and the uniform gap $h(\beta) > 3/4$ give

$$2^{m-n_0} + \frac{32\, 2^m}{\sum_{j=0}^{\lfloor \beta n_0 \rfloor} \binom{n_0}{j}} = 2^{-\Omega(n)}.$$

Thus Lemma 4.2 supplies the required average success for $\mathcal{S}_2$, since $105/128 > 3/4 \ge p$. Binary outputs are valid integer vectors with the same norm bound.

In either case apply Lemma 4.5 with $\alpha_0 = 1/8$, $D = 3/2$, and $\varepsilon = (c - \beta)/8$. Except for a $2^{-\Omega(n^{2/3})}$ fraction of matrices, every right-hand side has success probability at least $p/2$, successful norm at most $\sqrt{\beta d}$, and conditional-mean norm at most $(c - \beta)\sqrt{d}/8$. Hold one such matrix fixed. Apply Lemma 4.6 to its first $d$ columns in dimension $d$, taking $D = 3/2$. The same concentration calculation with $r = \lceil Cn \rceil \ge Cd$ gives, for sufficiently large fixed $C$, probability $1 - 2^{-\Omega(n)}$ that every right-hand side has a successful correction and their average satisfies

$$\|\bar{\boldsymbol{w}}(\boldsymbol{s})\| \le (c - \beta)\sqrt{d}.$$

Indeed, the bound (21) for a fixed right-hand side depends exponentially on $r$, and there are at most $q^m \le 2^{3n/2}$ right-hand sides. The threshold term involving $d/r$ is smaller than the allowed multiple of $\sqrt{d}$ for sufficiently large $d$. The compact parameter range gives a common positive lower bound on $c - \beta$ and bounds $\beta, c$. Thus the constants in both lemmas, including $C$ and the exceptional-probability exponents, can be chosen uniformly. Continuity of the parameter selection at $\delta = 1$ is not needed.

On this single event, fix any target and let $\boldsymbol{z} = \lfloor \boldsymbol{y} \rceil$, $\boldsymbol{f} = \boldsymbol{z} - \boldsymbol{y}$, $\boldsymbol{s} = -\mathbf{H}\boldsymbol{z}$. All corrections are supported on the first $d$ coordinates. Averaging their squared distances shows that at least one candidate satisfies

$$\|\boldsymbol{z} + \boldsymbol{w} - \boldsymbol{y}\|^2 \le \|\boldsymbol{f}\|^2 + \beta d + 2\|\boldsymbol{f}_{1:d}\|(c - \beta)\sqrt{d} \le \|\boldsymbol{f}\|^2 + cd.$$

Here $\|\boldsymbol{f}_{1:d}\| \le \sqrt{d}/2$ and $\mathbf{H}(\boldsymbol{z} + \boldsymbol{w}) = \mathbf{0}_m$. If exact search succeeds, its output is a closest lattice point. Otherwise the optimum squared distance exceeds (17), and the displayed candidate is a solution to $\gamma$-CVP. The event is common to all right-hand sides and hence to every rational target, even when the target depends on the matrix. Finally, every fallback is exact by Lemma 4.1. □

### 4.3 Computation costs

We first bound the cost of each branch throughout $0 \le \delta \le 3/2$. The parameter computations and feasibility tests take polynomial time and space. Calls run sequentially and reuse their storage.

For the correction branch, fix $a > 0$ satisfying (10) with $\tau = \gamma^{-2}$, that is,

$$\Theta_u(a) \le \Theta(a) e^{-au^2/\gamma^2} \qquad (|u| \le 1/2), \tag{22}$$

with the theta sums of (48). Suppose each call to $\mathcal{S}_q$ in dimension $n_0$ uses time $2^{T_{\mathrm{in}} n_0 + o(n)}$ and space $2^{S_{\mathrm{in}} n_0 + o(n)}$ on every input and execution, uniformly over the chosen parameters. The exponents $T_{\mathrm{in}}, S_{\mathrm{in}}$ may depend on those parameters. Since the adjusted squared radius is

$$\frac{\|\boldsymbol{f}\|^2 + cd}{\gamma^2} - \gamma^{-2}\|\boldsymbol{f}\|^2 = cd/\gamma^2,$$

Lemma 3.5 bounds the exact call, and $O(n)$ short calls add only a polynomial factor. Thus the complete correction branch has bounds

$$\begin{aligned} \text{time:} \quad & 2^{\max\{T_{\mathrm{in}} n_0,\, (acd/\gamma^2 + n \ln \Theta(a))/(2 \ln 2)\} + o(n)}, \\ \text{space:} \quad & 2^{\max\{S_{\mathrm{in}} n_0,\, (acd/\gamma^2 + n \ln \Theta(a))/(4 \ln 2)\} + o(n)}. \end{aligned} \tag{23}$$

For $q \ge 3$, use the representation costs at the actual equation rate. For $q = 2$, Lemma 4.2 gives $T_{\mathrm{in}} = 1 - h(\beta)$ and $S_{\mathrm{in}} = 0$ in this formula. The exact search takes place in dimension $n$, even when the corrections use only $d$ coordinates.

The branch $\delta = 0$ is polynomial. For $0 < \delta \le 1/8$, Lemma 4.1 gives time $2^{2\delta n}\,\mathrm{poly} \le 2^{n/4}\,\mathrm{poly}$ and space $2^{\delta n}\,\mathrm{poly} \le 2^{n/8}\,\mathrm{poly}$. The binary kernel branch processes at most $2^{\max\{0, n-m\} + \lceil n^{2/3} \rceil}$ vectors, one at a time, using polynomial space. Rejected matrices cost less.

All fallbacks occur below a fixed dimension threshold. Indeed, $\beta$ is bounded away from zero, so $n_0 = \Theta(n)$ on the correction branches, and the binary branch has $n_0 > m$ for large $n$. For the fixed parameters below density one, every strict equation, parity, and entropy condition has a positive margin at rate one. The high-density certificate gives uniform positive margins on $[1, 3/2]$. The $O(n^{-1/3})$ rate changes and the $n^{-2}$ evaluation widths are eventually smaller than these margins. The non-strict coordinate conditions are checked exactly. For the remaining bounded dimensions, $q^m \le 2^{3n/2}$ and $q \le q^m$ when $m \ge 1$; hence $\mathcal{A}$ adds only a bounded factor apart from polynomial input-length costs. All bounds above omit such polynomial factors.

For the fixed choice on $0 \le \delta \le 1$, put

$$e(a, c, \gamma) = \frac{ac/\gamma^2 + \ln \Theta(a)}{2 \ln 2}. \tag{24}$$

**Corollary 4.8** (Uniform costs for density at most one)**.** *Use fixed parameters for $\delta \le 1$ satisfying the conditions of Section 4.1 in Algorithm 3, with rational constants $0 < \beta < 1/4$, $\gamma > 1$, $c > \beta$, and $C > 0$, and with $h(\beta) > 3/4$. Suppose $\mathcal{S}$ has time and space exponents $T_{\mathrm{in}}, S_{\mathrm{in}} \ge 0$ per internal coordinate at equation rate $m \log q / n_0 = 1 + o(1)$, uniformly over these inputs. Fix $a > 0$ satisfying (22). For every $q \ge 2$, $n \ge 1$, and $m \ge 0$ with $\delta \le 1$, Algorithm 3 has time and space bounds*

$$2^{\max\{1/4, T_{\mathrm{in}}, e(a,c,\gamma)\} n + o(n)}, \qquad 2^{\max\{1/8, S_{\mathrm{in}}, e(a,c,\gamma)/2\} n + o(n)}.$$

*They hold for every input matrix, rational target, and execution, up to factors polynomial in the binary input length.*

*Proof.* For $q \geq 3$, we have $d = \delta n + O(1)$ and $n_0 = \delta n - O(n^{2/3})$, so the assumed internal costs apply. Using $d, n_0 \leq n$ in (23) gives the $T_{\text{in}}, S_{\text{in}}, e(a, c, \gamma)$ terms. On the binary correction branch, $1 - h(\beta) < 1/4$. The dynamic program has time and space exponents at most $1/4$ and $1/8$, while the binary kernel branch has time exponent $1 - \delta \leq 1/4$ and polynomial space. Taking the largest contributions proves the bounds. □

**Time and space bounds as functions of density.** For the fixed parameters on $0 \leq \delta \leq 1$ satisfying the conditions of Section 4.1, the costs depend explicitly on the density $\delta$. The term $T_{\text{in}}\delta$ below measures correction generation in dimension $n_0 = \delta n - o(n)$. Exact search still takes place in ambient dimension $n$, so its exponent, with $e$ as in (24),

$$e(a, c\delta, \gamma) = \frac{ac}{2\gamma^2 \ln 2}\,\delta + \frac{\ln \Theta(a)}{2 \ln 2}, \tag{25}$$

includes the density-independent term $\ln \Theta(a)/(2 \ln 2)$.

**Corollary 4.9** (Density-dependent costs)**.** *Use fixed parameters for $\delta \leq 1$ satisfying the conditions of Section 4.1 in Algorithm 3, with rational constants $0 < \beta < 1/4$, $\gamma > 1$, $c > \beta$, and $C > 0$, and with $h(\beta) > 3/4$, and fix $a > 0$ satisfying (22). Suppose $\mathcal{S}$ has time and space exponents $T_{\text{in}}, S_{\text{in}} \geq 0$ per internal coordinate at equation rate $m \log q/n_0 = 1 + o(1)$, uniformly over these inputs. For every $q \geq 3$, $n \geq 1$, and $m \geq 0$ with $\delta \leq 1$, the time and space exponents of Algorithm 3 are bounded by*

$$\begin{aligned} T_{\geq 3}(\delta) &= \begin{cases} 2\delta, & 0 \leq \delta \leq 1/8, \\ \max\{T_{\text{in}}\delta, e(a, c\delta, \gamma)\}, & 1/8 < \delta \leq 1, \end{cases} \\ S_{\geq 3}(\delta) &= \begin{cases} \delta, & 0 \leq \delta \leq 1/8, \\ \max\{S_{\text{in}}\delta, e(a, c\delta, \gamma)/2\}, & 1/8 < \delta \leq 1. \end{cases} \end{aligned} \tag{26}$$

*For $q = 2$, they are bounded by*

$$(T_2(\delta), S_2(\delta)) = \begin{cases} (\delta, \delta), & 0 \leq \delta \leq 1/8, \\ (\max\{1 - h(\beta), e(a, c, \gamma)\}, e(a, c, \gamma)/2), & 1/8 < \delta < 3/4, \\ (1 - \delta, 0), & 3/4 \leq \delta \leq 1. \end{cases} \tag{27}$$

*The corresponding bounds are $2^{T(\delta)n + o(n)}$ time and $2^{S(\delta)n + o(n)}$ space, uniformly on the stated ranges, with polynomial input-length factors suppressed. Bounds uniform in the modulus are obtained by taking the maximum of the two time functions and, separately, of the two space functions. The success guarantee remains that of Theorem 4.7, with $C$ chosen as in that theorem.*

*Proof.* For $q \geq 3$ in the correction branch, $d = \delta n + O(1)$ and $n_0 = \delta n - O(n^{2/3})$. Substitution in (23) gives the correction terms in (26). For the small-density branch, the exact dynamic program costs $nq^{m+1}$ poly time and $nq^m$ poly space. Its time exponent is $\delta + \log q/n \leq 2\delta$. In particular, it improves to $\delta + o(1)$ when $\log q = o(n)$.

For $q = 2$, the same dynamic program has exponent $\delta$ in time and space. The middle branch uses $d = n$, so exact search has exponent $e(a, c, \gamma)$, while the binary sampler has exponent $1 - h(\beta)$ and polynomial space. The last branch enumerates at most $2^{n - m + \lceil n^{2/3} \rceil}$ kernel vectors using polynomial space. This proves (27). At $\delta = 1$, zero leading time exponent in this bound means subexponential time. The good-matrix and simultaneous-all-target events are the ones already proved for the corresponding branches. □

**Binary CVP at density at least one.** For $q = 2$ and $1 \le \delta \le 3/2$, the kernel cutoff in Algorithm 3 permits at most $2^{\lceil n^{2/3} \rceil}$ kernel vectors. The analysis of the kernel branch in Theorem 4.7 gives exact CVP for every target on accepted matrices and bounds the rejected fraction by $2^{1-\lceil n^{2/3} \rceil}$. Enumerating the cosets takes time $2^{n^{2/3}+O(1)}$ poly and polynomial space, including at $\delta = 1$. More generally, for fixed $\delta > 1$, the stronger elementary estimate

$$\Pr[\ker \mathbf{H} \neq \{\mathbf{0}\}] \le (2^n - 1)2^{-m} < 2^{-(\delta-1)n}$$

shows that, except for this fraction of matrices, $\Lambda = 2\mathbb{Z}^n$ and coordinatewise rounding to $2\mathbb{Z}^n$ is exact.

### 4.4 Parameter selection

**Density at most one.** For a prescribed rational $\gamma > 1$, we minimize the time bound of the complete algorithm on $0 \le \delta \le 1$:

$$\inf \ \max\{1/4, T_{\text{in}}, e(a, c, \gamma)\}. \tag{28}$$

The variables are the integer depth $k \ge 1$, the rational coordinate proportions $t_0, \dots, t_k$, the rational constants $\eta > 0$ and $c$, and the real parameter $a > 0$, all fixed independently of $n$. The short-solution parameters satisfy Corollary 3.3 with $\rho = 1$, including the even-modulus conditions and (11); in addition, $c > 2t_0$ and (22) must hold. The exponents $T_{\text{in}}, S_{\text{in}}$ are those in (9). The same choice gives the space bound in Corollary 4.8. The strict inequalities leave room for rounding the coordinate counts and controlling the average correction, so the infimum need not be attained with positive margins.

For $\gamma = 5/3$, use three decomposition steps, $k = 3$, and

$$\begin{gathered}(t_0, t_1, t_2, t_3) = (0.1135461, 0.09917397, 0.06690926, 0.03531462),\\ \eta = 10^{-8}, \qquad c = 0.22711, \qquad a = 4.138.\end{gathered} \tag{29}$$

These decimals denote exact rational numbers. Take $p = 3/4$ in the average-success bound of Corollary 3.3. The first coordinate proportion is just above the smaller root of $\mathcal{H}(t, t, 1 - 2t) = 1$, as required by $h_0 > 1$. The remaining coordinate proportions nearly balance the second and third combination costs. The positive $\eta$ and the rational rounding are included when checking every constraint.

For this choice, the third combination determines the time exponent, and the stored list at level two determines the space exponent. The distance margin is $c - \beta = 0.0000178 > 0$. Interval calculations verify all coordinate and congruence conditions, and, by Lemma A.4, the exact-search condition (22) for $a = 4.138$ and $\gamma = 5/3$. They give

$$\beta = 0.2270922, \qquad T_{\text{in}} < 0.26675191, \qquad S_{\text{in}} < 0.262740033, \qquad e(a, c, 5/3) < 0.266706.$$

Moreover, $\beta < 1/4$ and $h(\beta) = h_0 - \beta > 1 - \beta > 3/4$, so (11) holds. The elementary branches have time exponent at most $1/4$ and space exponent at most $1/8$. Thus exact search is strictly cheaper than the representation correction at the worst-case density for this choice. Further reducing its exponent by varying $a$ alone cannot improve the total time bound. Appendix F verifies the length, success, time, and space guarantees simultaneously. By Theorem 4.7 and Corollary 4.8, the resulting algorithm solves 5/3-CVP in time $2^{0.266752n+o(n)}$ and space $2^{0.262741n+o(n)}$, with the stated simultaneous success guarantee.

Within the parameter domain of (28), interval verification gives

$$0.26675186 \le \inf \max\{1/4, T_{\mathrm{in}}, e(a,c,5/3)\} < 0.26675191. \tag{30}$$

Thus the selected rational choice is within $5 \cdot 10^{-8}$ of the infimum of the proved time bound for this construction on $0 \le \delta \le 1$. The lower certificate bounds $T_{\mathrm{in}}$ alone over a larger parameter domain and every fixed depth; it therefore also bounds the complete objective. The selected choice satisfies all the additional constraints, including the binary conditions, and supplies the upper bound. Appendices F.3 to F.5 give the reductions and interval verification.

For $q = 2$, the elementary short solver and kernel branches instead give time $2^{0.266706n+o(n)}$ and space $2^{0.133353n+o(n)}$ at this parameter choice; exact search supplies the largest exponents in that case.

For the selected 5/3-CVP parameters in (29), outward interval arithmetic also gives

$$\frac{ac}{2\gamma^2 \ln 2} < 0.244047177, \qquad \frac{\ln \Theta(a)}{2 \ln 2} < 0.022658287.$$

With the bounds on $T_{\mathrm{in}}, S_{\mathrm{in}}$ above, throughout $1/8 < \delta \le 1$ for $q \ge 3$,

$$\begin{aligned} T_{\ge 3}(\delta) &< \max\{0.26675191\delta,\ 0.244047177\delta + 0.022658287\}, \\ S_{\ge 3}(\delta) &< 0.26274003 3\delta. \end{aligned} \tag{31}$$

For the exact parameter values, $T_{\mathrm{in}}\delta = e(a, c\delta, 5/3)$ at $\delta \simeq 0.9979543221$, and exact search supplies the larger term below this point. The space terms satisfy $S_{\mathrm{in}}\delta = e(a, c\delta, 5/3)/2$ at $\delta \simeq 0.0805104422 < 1/8$, which justifies the simplified space bound. In the binary middle branch, $1 - h(\beta) < 0.227092187 < e(a,c,5/3) < 0.266706$, and its space exponent is less than 0.133353.

**A short-preimage consequence.** The correction procedure also gives an absolute length guarantee without running the exact-search part of the $\gamma$-CVP algorithm.

**Corollary 4.10** (Short preimages)**.** *Fix a constant $0 < \delta_0 \le 1$. Let integers $q \ge 3$ and $m, n \ge 1$ satisfy $\delta_0 \le m \log_2 q / n \le 1$, and put $d = \lceil m \log_2 q \rceil$. There is a randomized algorithm that, given $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ and $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, returns either $\perp$ or $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying*

$$\mathbf{H}\boldsymbol{x} \equiv \boldsymbol{t} \pmod q, \qquad \|\boldsymbol{x}\|^2 \le 0.2271d.$$

*Its time and space bounds are $2^{0.2668d+o(n)}$ and $2^{0.2628d+o(n)}$, respectively, on every input and execution, up to polynomial factors in the binary input length.*

*For sufficiently large $n$, all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices $\mathbf{H}$ have the following property: with probability at least $1 - 2^{-\Omega(n)}$ over the internal randomness, the algorithm succeeds simultaneously for every $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$ when corresponding executions use the same random bits.*

*Proof.* Use the parameters in (29), with $\beta = 0.2270922$ and $p = 3/4$. Put

$$\ell = \lceil d^{2/3} \rceil, \qquad \ell' = \lceil \ell/\beta \rceil, \qquad n_0 = d - \ell - \ell'.$$

Since $d = \Theta(n)$, the actual equation rate satisfies

$$\frac{m \log_2 q}{n_0} = 1 + O(n^{-1/3}).$$

The strict margins verified in Section 4.4 and Appendix F therefore allow the average-success procedure of Corollary 3.3 for sufficiently large $n$.

Apply Lemma 4.5 with $\alpha_0 = \delta_0$, $D = 1$, and any fixed $\varepsilon > 0$. For all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices, one correction attempt succeeds with probability at least $p/2 = 3/8$ for every prescribed value and returns a vector satisfying

$$\|\boldsymbol{x}\|^2 \le \beta d = 0.2270922d < 0.2271d.$$

Run $4n$ independent attempts and return the first successful output, or $\perp$ if all attempts fail. For each such matrix, using the same sequence of random blocks for every prescribed value gives

$$\Pr[\text{some prescribed value fails}] \le q^m (5/8)^{4n} \le 2^n (5/8)^{4n} = 2^{-\Omega(n)}.$$

Each attempt uses one short-solver call in dimension $n_0 = d - O(d^{2/3})$, with

$$T_{\text{in}} < 0.26675191, \qquad S_{\text{in}} < 0.262740033.$$

The polynomially many sequential attempts reuse storage, giving the stated resource bounds. In the bounded dimensions where the parameter checks fail, the algorithm may return $\perp$. □

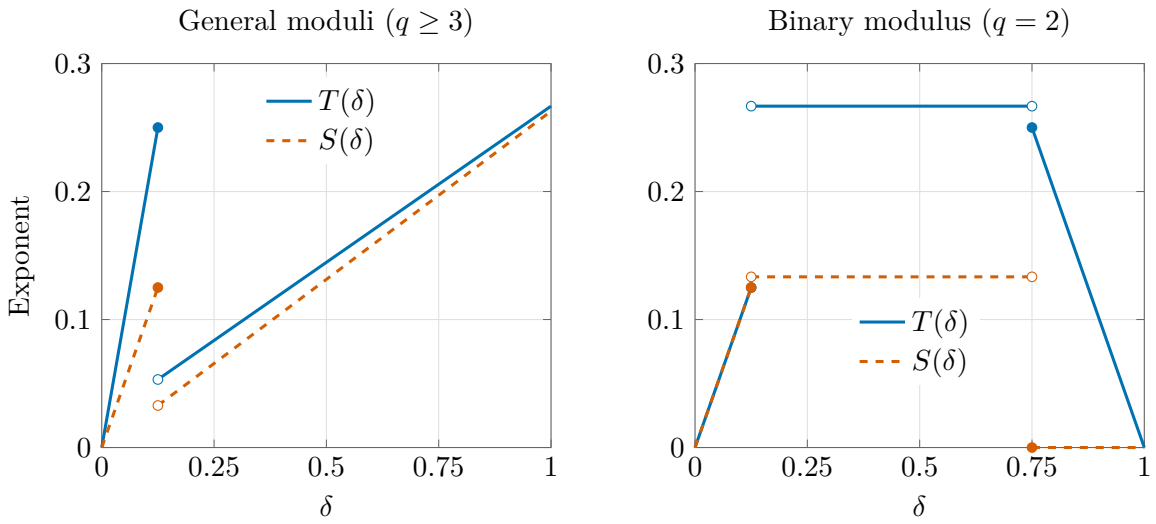


Figure 4: The 5/3-CVP density functions for the fixed parameters. The curves illustrate the entropy formulas; their coefficients are certified by outward interval arithmetic. Jumps mark changes of procedure, not intrinsic complexity thresholds.

These functions describe the selected algorithm. Optimizing the parameters separately at each density is a further optimization problem, subject to the same success and space constraints.

**Density above one.** For $q \ge 3$ and $1 < \delta \le 3/2$, Algorithm 3 uses the proportions and length parameters in Eqs. (13) and (14), with corrections on all $n$ coordinates. We verify their conditions and compute the resulting costs. The parameter curve is certified on the closed interval $[1, 3/2]$; at $\delta = 1$, the algorithm uses the better choice (29).

Evaluate (9) at $\rho = \delta$, $\eta = 10^{-8}$, and the proportions in (13); denote the resulting exponents by $T_{\mathrm{in}}(\delta)$ and $S_{\mathrm{in}}(\delta)$. For $\gamma = 5/3$ and $a = 4.138$, define, with $e$ as in (24),

$$\begin{aligned} T_{>}(\delta) &= \max\{T_{\mathrm{in}}(\delta), e(a, c(\delta), \gamma)\}, \\ S_{>}(\delta) &= \max\{S_{\mathrm{in}}(\delta), e(a, c(\delta), \gamma)/2\} \ . \end{aligned} \tag{32}$$

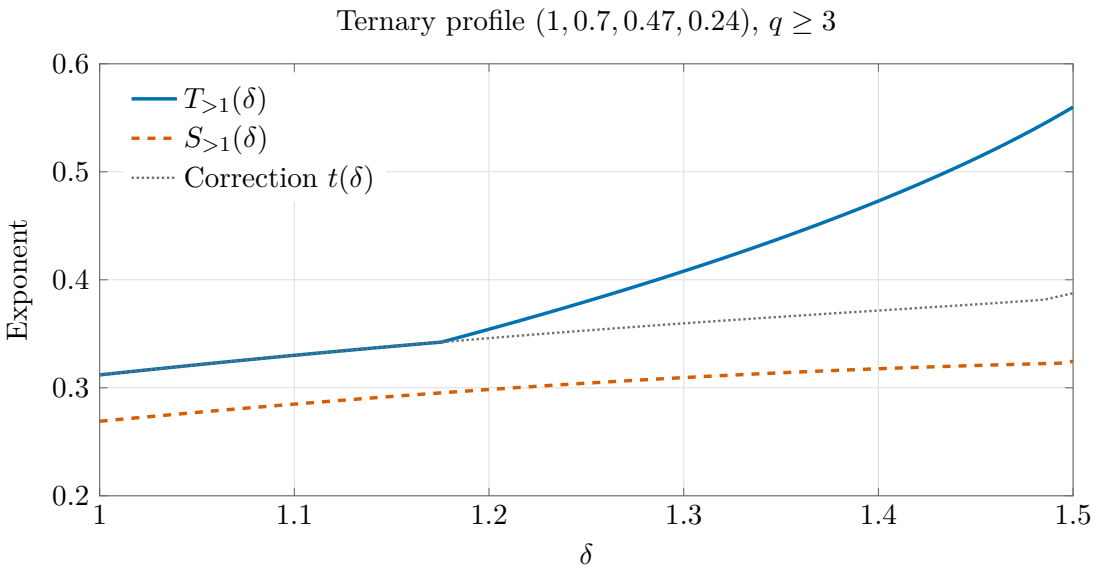


Figure 5: 5/3-CVP cost functions above density one for the continuous profile (13). The time is the maximum of correction generation and exact search; the latter determines the upper endpoint. The curves are floating-point illustrations, and Lemma F.5 certifies the whole interval.

**Corollary 4.11** (Costs for density between one and three halves)**.** *For integers $q \geq 3$, $n \geq 1$, and $m \geq 1$ satisfying $1 < \delta \leq 3/2$, Algorithm 3, with $\gamma = 5/3$ and its parameters for this density range, has time and space bounds*

$$2^{T_{>}(\delta)n + o(n)}, \qquad 2^{S_{>}(\delta)n + o(n)},$$

*uniformly over the density interval, up to factors polynomial in the binary input length. The time and space bounds hold for every input matrix, rational target, and execution. In particular, they are at most $2^{0.56n+o(n)}$ time and $2^{0.325n+o(n)}$ space throughout the interval. Its simultaneous success guarantee for 5/3-CVP is the one in Theorem 4.7.*

*Proof.* Lemma F.5 verifies every strict coordinate, equation, and parity condition for the compact curve (13), together with $h_0 - \delta > 0$ and the allowed $10^{-12}$ root-rounding padding. Their margins have a common positive lower bound. Also, the implemented rational $\beta$ stays in a compact subinterval of $(0, 1)$, so $n_0 = n - O(n^{2/3})$ and

$$\frac{m \log q}{n_0} = \delta + O(n^{-1/3})$$

uniformly. Thus, above one fixed dimension threshold, the same strict conditions, including $h_0 > m \log q / n_0$, hold at the actual equation rate. Corollary 3.3, applied uniformly as specified in Section 3.1.1, supplies a short solver whose average success is at least $p = 3/4$ for all but a $2^{-\Omega(n_0^{2/3})}$

fraction of matrices. Its outputs have squared length at most $\beta n_0$. The bounds on $\beta$ and the fixed gap $c - \beta = 2 \cdot 10^{-5}$ satisfy the uniform hypotheses of Theorem 4.7. That theorem gives the simultaneous approximation guarantee with one fixed $C$ throughout this density interval.

Uniform continuity of the cost formulas, the vanishing rational rounding error, and $n_0 = n - o(n)$ give short-solver time $2^{T_{\mathrm{in}}(\delta)n+o(n)}$ and space $2^{S_{\mathrm{in}}(\delta)n+o(n)}$. The certified weight condition for $a = 4.138$, $\gamma = 5/3$ is independent of the density and correction length. Since $d = n$ and $c = c(\delta) + O(n^{-2})$, the exact-search terms in (23) are $e(a, c(\delta), \gamma)n + o(n)$ in time and $e(a, c(\delta), \gamma)n/2 + o(n)$ in space. Taking the maxima in that bound gives (32). The certificate gives the stated numerical bounds. □

The parameters for $q \geq 3$ are certified through $\delta = 3/2$. The ternary short-solution method cannot be extrapolated to every larger density: there are at most $3^{n_0}$ possible ternary outputs, so the fraction of right-hand sides having any such output is at most $3^{n_0}/q^m$. For fixed $\delta > \log 3$ this is exponentially small. At $\delta = \log 3$, the required fixed positive entropy surplus is also impossible, since $h_0 \leq \log 3$. This restriction concerns this correction method, rather than $\gamma$-CVP itself.

# 5 SVP

We solve the SVP of Definition 2.7 using the two common procedures of Section 3. We first restrict the search to a ball containing a shortest vector with high probability. Within that ball, we specify some coordinates and complete the remaining ones by the short-solution construction. The coordinate choices control both the probability of finding a shortest vector and the total search cost. The main correctness argument uses the common enumeration to bound, for each fixed candidate, the probability of failing to find a nonzero lattice vector no longer than that candidate, conditioned on its lattice membership. Counting these candidates then bounds the probability of failing to find a shortest vector.

Throughout this section, $D \geq 1$ is a fixed constant, and $q \geq 2$, $n \geq 1$, and $m \geq 0$ are integers with density $\delta \leq D$. The correctness and cost formulas below apply whenever the chosen completion parameters satisfy their stated uniform conditions. We give certified choices for $D = 1$ and for several larger bounded density ranges; we do not assert their feasibility for every $D$. The algorithm always returns a nonzero lattice vector. Its shortest-vector guarantee is for a uniformly random $\mathbf{H}$, whose entries are independent and uniform in $\mathbb{Z}/q\mathbb{Z}$; the time and space bounds hold for every input matrix. All congruences in this section are modulo $q$.

### 5.1 The SVP algorithm

**Algorithm 4** SVP

**Input:** $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, where $q \geq 2$, $n \geq 1$, $m \geq 0$, and $\delta \leq D$.
**Output:** A nonzero vector of $\Lambda$.

**if** $m = 0$ **then return** $\boldsymbol{e}_1$
**if** $n = 1$ **then return** $(q/\gcd(q, H_{11}, \ldots, H_{m1}))\boldsymbol{e}_1$
$\hat{\boldsymbol{x}} \leftarrow q\boldsymbol{e}_1$
**if** $q = 2$ or $q^4 \leq n$ **then**
    **for** each nonzero $\boldsymbol{x} \in \mathbb{Z}^n$ with $\|\boldsymbol{x}\|^2 \leq q^2$ **do**
        **if** $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$ and $\|\boldsymbol{x}\| < \|\hat{\boldsymbol{x}}\|$ **then** $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}$
    **end for**
    **return** $\hat{\boldsymbol{x}}$
**end if**
Compute $u, b$ by (33).
**if** $\delta \leq \delta_0$ **then**
    $\boldsymbol{x} \leftarrow \mathcal{E}_0^{\neq 0}(\mathbf{H}, \mathbf{0}_n, b)$
    **if** $\boldsymbol{x} \neq \perp$ **then return** $\boldsymbol{x}$
    **return** $\hat{\boldsymbol{x}}$
**end if**
Compute $n_0, \widehat{p}, t_0, \ldots, t_k, a_0$ and the intervals for (35)–(36).
**if** some $t_i \notin (0, 1/2)$ or (3) fails **then return** $\hat{\boldsymbol{x}}$
**for** each of $n$ rounds **do**
    **for** each $\boldsymbol{z}$ satisfying (35) **do**
        **for** each $i \notin \mathrm{supp}(\boldsymbol{z})$, $\sigma \in \{-1, 1\}$, and $N_+, N_-$ satisfying (36) **do**
            $\mathcal{I} \leftarrow \{1, \ldots, n\} \setminus (\mathrm{supp}(\boldsymbol{z}) \cup \{i\})$
            **if** $N_+ + N_- > |\mathcal{I}|$ or $2a_0 - N_+ - N_- > n_0 - |\mathcal{I}|$ **then continue**
            Form $\boldsymbol{d} \in \{-1, 0, 1\}^{n_0 - |\mathcal{I}|}$ with $a_0 - N_+$ ones, then $a_0 - N_-$ minus ones, then zeros.
            Sample uniform $\mathbf{D} \in (\mathbb{Z}/q\mathbb{Z})^{m \times (n_0 - |\mathcal{I}|)}$; set $\mathbf{A} \leftarrow [\mathbf{H}_{\mathcal{I}}\ \mathbf{D}]$.
            $\boldsymbol{t} \leftarrow -\mathbf{H}(\boldsymbol{z} + \sigma\boldsymbol{e}_i) + \mathbf{D}\boldsymbol{d}$
            **for** each $\boldsymbol{v}$ enumerated by $\mathcal{R}(\mathbf{A}, \boldsymbol{t})$ **do**
                **if** $\boldsymbol{v}_{\{|\mathcal{I}|+1, \ldots, n_0\}} \neq \boldsymbol{d}$ **then continue**
                $\boldsymbol{x} \leftarrow \boldsymbol{z} + \sigma\boldsymbol{e}_i + \widetilde{\boldsymbol{v}}$
                **if** $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$ and $\|\boldsymbol{x}\| < \|\hat{\boldsymbol{x}}\|$ **then** $\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x}$
            **end for**
        **end for**
    **end for**
**end for**
**return** $\hat{\boldsymbol{x}}$

The algorithm enumerates the positions and values of coordinates with absolute value at least two and fixes one further coordinate to 1 or $-1$. For each such choice, it searches for the remaining coordinates of a lattice vector using the short-solution procedure; we call this search completion. The congruence passed to the procedure has slightly more variables than the original lattice dimension. The added variables allow the required numbers of 1 and $-1$ entries to be met without omitting any remaining original coordinate.

**Parameters and the search radius.** Fix rational constants $\delta_0 \in (1/2, 1)$, $\kappa \in (0, 1)$, $\lambda, \varepsilon, \eta > 0$, a fixed number $k \geq 1$ of decomposition steps, rational constants $c_0 = 1, c_1, \ldots, c_k > 0$, and an integer $\nu \geq 1$ for the list limits. Their values are chosen after the complete cost has been derived. A finite collection of such choices may also be used on specified density bands, with the choice determined only by $n, m, q$; the bounds then hold uniformly on each closed band. Both the exact-search and completion branches use the following search radius.

Write $\boldsymbol{e}_i$ for the $i$th standard unit vector in $\mathbb{Z}^n$. For $n \geq 2$ and $m \geq 1$, let $u$ be the least nonnegative integer such that

$$|\mathcal{B}_{n-1}(u)| \geq q^m 2^{\lceil n^{2/3} \rceil}, \qquad b = \min\{u + 1, q^2\}, \tag{33}$$

where $\mathcal{B}_{n-1}(u)$ is the set of integer points of squared norm at most $u$ in $\mathbb{Z}^{n-1}$ (Definition A.1).

Exact search handles $\delta \leq \delta_0$. For $\delta > \delta_0$, completion uses dimension

$$n_0 = \left\lceil \frac{n}{1 - \kappa} \right\rceil.$$

The probability mass functions below determine both the coordinate counts and the range of enumeration. Let $p_a$ and $r_a$ be the probability mass functions on $\mathbb{Z}$ and $\mu(a)$ the second moment of $p_a$ defined in (49) of Appendix A. Choose $a > 0$ so that $\mu(a) = u/(n-1)$; this determines $a$ uniquely, since $\mu$ decreases strictly from $+\infty$ to 0 by Lemma A.2(3) and $u \geq 1$ because $|\mathcal{B}_{n-1}(0)| = 1$. The resulting distributions specify the coordinate restrictions; the algorithm does not sample vectors from them.

Compute a rational upper approximation $\widehat{p}$ satisfying $0 \leq \widehat{p} - p_a(1) \leq \min\{n^{-2}, 10^{-12}\}$, and set

$$t_0 = \widehat{p} + \lambda, \qquad t_i = c_i t_0 \ (1 \leq i \leq k), \qquad a_0 = 2^k \left\lfloor \frac{t_0 n_0}{2^k} \right\rfloor. \tag{34}$$

The equation rate is $\rho = \delta n / n_0$. The constants are chosen so that the coordinate, equation, and parity inequalities of Lemma 3.2 have fixed positive margins throughout the input range. Its bound on the probability of omitting a fixed vector therefore applies uniformly by Section 3.1.1, since these parameters depend only on $n, m, q$. The additional condition $h_0 > \rho$ in Corollary 3.3 is not required here. Let $\mathcal{R}(\mathbf{A}, \boldsymbol{t})$ denote the output list of the procedure of Section 3.1 (Algorithm 1, or Algorithm 7 for a large modulus) in dimension $n_0$, run with the constants $k, \nu, \eta$ and the proportions $t_0, \ldots, t_k$ of (34). Every output has exactly $a_0$ entries of each nonzero sign; enumeration continues after a rejected output.

**Prescribed coordinates and sign counts.** Enumerate $\boldsymbol{z} \in \mathbb{Z}^n$ subject to

$$z_j = 0 \text{ or } |z_j| \geq 2, \qquad \|\boldsymbol{z}\|^2 \leq b, \qquad -\sum_{j=1}^{n} \log r_a(z_j) \leq n\big(\mathcal{H}(r_a) + \varepsilon\big). \tag{35}$$

Let $\operatorname{supp}(\boldsymbol{z}) = \{j : z_j \neq 0\}$. After prescribing $x_i = \sigma \in \{-1, 1\}$ at some $i \notin \operatorname{supp}(\boldsymbol{z})$, let $N_+, N_-$ be the numbers of remaining coordinates equal to 1 and to $-1$. Their permitted nonnegative integer values satisfy

$$\begin{gathered} N_+, N_- \leq a_0, \qquad |N_\pm - n p_a(1)| \leq \varepsilon n + 1, \\ 1 + \|\boldsymbol{z}\|^2 + N_+ + N_- \leq b, \qquad 1 + \|\boldsymbol{z}\|^2 + N_+ + N_- < q^2. \end{gathered} \tag{36}$$

For an index set $\mathcal{I}$ and $\boldsymbol{v} \in \mathbb{Z}^{n_0}$, write $\widetilde{\boldsymbol{v}} \in \mathbb{Z}^n$ for the vector placing the first $|\mathcal{I}|$ entries of $\boldsymbol{v}$ on $\mathcal{I}$ in increasing order and zero elsewhere. All numerical comparisons are made with outward rational intervals, as specified below. The random bits used to sample different added matrices and to run different calls to $\mathcal{R}$ are mutually independent, including across rounds.

**Exact search and completion.** When $\delta \le \delta_0$, the nonzero exact-search procedure is run on the ball of squared radius $b$. Since $\delta_0$ is rational, this test, that is, $q^m \le 2^{\delta_0 n}$, is decided by exact integer arithmetic after raising both sides to the power of the denominator of $\delta_0$.

For $\delta > \delta_0$, the vector $\boldsymbol{z}$ specifies all coordinates outside $\{-1, 0, 1\}$. After specifying $x_i = \sigma$, every remaining coordinate position belongs to $\mathcal{I}$. The added vector $\boldsymbol{d}$ supplies the equal sign counts required by $\mathcal{R}$. Line 22 checks that the prescribed counts fit within the remaining original coordinates and the added coordinates.

For the numerical tests in (35), enclose each side in an interval of width at most one and reject only when the left lower endpoint exceeds the right upper endpoint. For the sign-count window in (36), enclose $np_a(1)$ in an interval of width at most one, subtract $\varepsilon n + 1$ from its lower endpoint, and add $\varepsilon n + 1$ to its upper endpoint. These rules retain every specified case, enlarge the logarithmic budget by at most two, and extend each endpoint of the exact sign-count window by at most one. Section E.6.2 justifies these comparisons. The approximation $\widehat{p}$ is computed separately to the accuracy required in (34), as described in Section E.6.1. The permitted vectors $\boldsymbol{z}$ are generated one at a time by the procedure in Section E.6.3.

## 5.2 Correctness

The analysis uses the following lemmas on the search radius in (33) and on the coordinates of short lattice vectors; their proofs are in Appendix E. We then verify the elementary branches and bound the probability of failing to find a nonzero lattice vector no longer than a fixed candidate, conditioned on its lattice membership. Summing the joint probabilities of membership and failure over these candidates gives a shortest-vector result for most input matrices; independent repetitions strengthen the conditional success probability.

We compute the search radius without using a volume approximation.

**Lemma 5.1.** *Fix $D > 0$. Let $q \ge 2$, $m \ge 1$, and $n \ge 2$ be integers satisfying $\delta \le D$, and let $u$ be as in (33). Then $u = O(n)$, where the hidden constant may depend on $D$, and $u$ can be computed in polynomial time and space from $n, q, m$. Moreover, $u \to \infty$ uniformly over the allowed values of $q, m$. The count at the first qualifying radius satisfies*

$$q^m\, 2^{\lceil n^{2/3} \rceil} \le |\mathcal{B}_{n-1}(u)| < (2n-1) q^m\, 2^{\lceil n^{2/3} \rceil}.$$

The following comparison estimates how many vectors lie inside the radius. It supplies the theta-sum bound used by the exact-search cost analysis in Theorem 5.8.

**Lemma 5.2.** *Fix $D > 0$. Let $q \ge 2$, $m \ge 1$, and $n \ge 2$ be integers with $\delta \le D$, and let $u$ be as in (33). There is a unique $a > 0$ with $\mu(a) = u/(n-1)$, and it satisfies $a_D \le a \le \ln(8(n-1))$ for a constant $a_D \in (0, \ln 4)$ depending only on $D$. With this $a$,*

$$|\mathcal{B}_{n-1}(u)| \le e^{au} \Theta(a)^{n-1} \le |\mathcal{B}_{n-1}(u)|\, e^{O((\log n)^2)}, \tag{37}$$

*where the hidden constant depends only on $D$.*

The computed ball contains a nonzero lattice vector for most input matrices.

**Lemma 5.3.** *Fix $D > 0$. Let $q \ge 3$, $m \ge 1$, and $n \ge 2$ be integers with $\delta \le D$. Let $u$ be as in (33). For a uniformly random matrix $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$,*

$$\Pr_{\mathbf{H}}\Big[\exists \boldsymbol{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}_n\} : \mathbf{H}\boldsymbol{x} = \mathbf{0}_m,\ \|\boldsymbol{x}\|^2 \le u + 1\Big] \ge 1 - 2^{-\Omega(n^{2/3})}.$$

*The hidden constant may depend on $D$.*

On the completion branch, the counting parameter $a$ stays in a fixed compact range.

**Lemma 5.4.** *Fix constants $0 < \delta_0 \le D < \infty$. Let $q \ge 3$, $m \ge 1$, and $n \ge 2$ be integers with $\delta_0 \le \delta \le D$. Let $u$ be as in (33), and let $a > 0$ satisfy $\mu(a) = u/(n-1)$. Then*

$$-\sum_{j\in\mathbb{Z}} p_a(j)\log p_a(j) = \delta + o(1).$$

*In particular, $a$ belongs to a fixed compact subinterval of $(0,\infty)$, depending only on $\delta_0, D$, for all sufficiently large $n$. The error term is uniform over the stated input range.*

The sign-count windows in (36) and the restriction on the separately enumerated coordinates in (35) hold for every relevant lattice vector; the statement does not presume a probability distribution for a shortest vector chosen after the matrix is known.

**Lemma 5.5.** *Fix constants $0 < \delta_0 \le D < \infty$ and $\varepsilon > 0$. Let $q \ge 3$, $m \ge 1$, and $n \ge 2$ be integers with $\delta_0 \le \delta \le D$ and $q^4 > n$. Let $u$ be as in (33), and let $a > 0$ satisfy $\mu(a) = u/(n-1)$. Except for a fraction $2^{-\Omega(n/\log n)}$ of uniformly chosen matrices $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, every vector $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying*

$$\mathbf{H}\boldsymbol{x} = \mathbf{0}_m, \qquad 0 < \|\boldsymbol{x}\|^2 \le u+1, \qquad \|\boldsymbol{x}\| < q$$

*has, for each $\sigma \in \{-1,1\}$,*

$$\bigl|\,|\{j : x_j = \sigma\}| - np_a(1)\,\bigr| \le \varepsilon n, \qquad \prod_{j=1}^{n} r_a(z_j) \ge 2^{-n(\mathcal{H}(r_a)+\varepsilon)}.$$

*Here $z_j = x_j$ if $|x_j| \ge 2$ and $z_j = 0$ otherwise. The hidden constant may depend on $\delta_0, D$ and $\varepsilon$.*

**Elementary branches and exact search.** Every retained vector is a nonzero member of $\Lambda$: the initial candidate is $q\boldsymbol{e}_1$, and each completion has its prescribed nonzero coordinate at $i$. The congruence is checked before retention. If $m = 0$, a unit vector is shortest. If $n = 1$, the congruences define $(q/\gcd(q, H_{11}, \ldots, H_{m1}))\mathbb{Z}$, so the returned generator is shortest; canonical integer representatives may be used to compute this gcd. For the remaining branches, by Lemma 5.3, the surplus factor in (33) makes the chosen ball contain a nonzero lattice vector with high probability. Any vector returned by the nonzero exact-search procedure is shortest, because all vectors outside its searched ball are longer than a returned minimum inside it. This assertion does not apply to the initial candidate returned after a reported failure.

**Finding a vector no longer than a fixed candidate.** On the completion branch, the numerical choice in Section 5.4 ensures, uniformly over the count window, that for sufficiently large $n$

$$a_0 \ge N_+, N_-, \qquad n_0 - 2a_0 + N_+ + N_- \ge n \ge |\mathcal{I}|. \tag{38}$$

The respective limiting margins are the differences between the two sides of

$$\frac{\lambda + \kappa p_a(1)}{1-\kappa} > \varepsilon, \qquad \frac{\kappa(1-2p_a(1)) - 2\lambda}{1-\kappa} > 2\varepsilon. \tag{39}$$

Both have fixed positive lower bounds for the selected constants. The $O(1)$ count rounding and the approximation error in $\widehat{p}$ do not close these margins. Thus all remaining positions fit, and no random choice of a coordinate subset is needed. The strict uniform margins also ensure that the coordinate guard in Algorithm 4 passes for all sufficiently large $n$.

For the completion instance

$$\mathbf{A} = [\mathbf{H}_{\mathcal{I}}\ \mathbf{D}], \qquad \boldsymbol{t} = -\mathbf{H}(\boldsymbol{z} + \sigma \boldsymbol{e}_i) + \mathbf{D}\boldsymbol{d}, \tag{40}$$

the prescribed added coordinates cancel $\mathbf{D}\boldsymbol{d}$. Every accepted vector satisfies the original congruence and has squared length $1 + \|\boldsymbol{z}\|^2 + N_+ + N_-$. The separate coordinate $x_i = \sigma$ also permits the probability analysis: conditioning on a fixed vector's membership in $\Lambda$ determines column $i$ from the others and leaves those other columns uniform. The probability bound in Lemma 3.2 then applies to the completion matrix.

**Lemma 5.6.** *Let $q \geq 3$, $n \geq 2$, and $m \geq 1$ be integers with $m = O(n)$, and sample $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ uniformly. Fix, independently of $\mathbf{H}$, a nonzero $\boldsymbol{x} \in \mathbb{Z}^n$ having a coordinate $x_i = \sigma \in \{-1, 1\}$. Let $\boldsymbol{z} \in \mathbb{Z}^n$ agree with $\boldsymbol{x}$ on the coordinates of absolute value at least two and vanish elsewhere, let $N_\pm = |\{j \neq i : x_j = \pm 1\}|$, and let $\mathcal{I} = [n] \setminus (\operatorname{supp}(\boldsymbol{z}) \cup \{i\})$. Let $n_0 \geq 1$ and $a_0 \geq 0$ be integers with $2a_0 \leq n_0$, $n_0 = O(n)$, and*

$$N_+, N_- \leq a_0, \qquad 2a_0 - N_+ - N_- \leq n_0 - |\mathcal{I}|.$$

*Suppose that, for every fixed $\boldsymbol{v} \in \{-1, 0, 1\}^{n_0}$ with $a_0$ entries of each nonzero sign and a uniform $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n_0}$ independent of $\boldsymbol{v}$,*

$$\Pr_{\mathbf{A}, \omega}[\boldsymbol{v} \notin \mathcal{R}(\mathbf{A}, \mathbf{A}\boldsymbol{v})] \leq \zeta, \qquad 0 \leq \zeta \leq 1,$$

*where $\omega$ denotes the enumeration's random bits, independent of $\mathbf{A}$. For the prescribed $\boldsymbol{z}$, $(i, \sigma)$, and $(N_+, N_-)$, consider one execution of the completion step of Algorithm 4, starting with the construction of $\boldsymbol{d}$ and using fresh uniform added columns. Conditioned on $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$, this execution produces a vector of length at most $\|\boldsymbol{x}\|$ except with probability $\zeta + 2^{-\Omega(n^2)}$.*

*Proof.* Conditioning on $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$ determines

$$\mathbf{H}\boldsymbol{e}_i = -\sigma \sum_{j \neq i} x_j \mathbf{H}\boldsymbol{e}_j.$$

Every assignment to the other columns has exactly one permitted value of column $i$, so those columns remain independent and uniform. The prescribed case, the set $\mathcal{I}$, and the added vector $\boldsymbol{d}$ depend only on $\boldsymbol{x}$ and the dimensions. The fresh matrix $\mathbf{D}$ is independent. Thus $\mathbf{A} = [\mathbf{H}_{\mathcal{I}}\ \mathbf{D}]$ is uniform, and $(\boldsymbol{x}_{\mathcal{I}}^\top, \boldsymbol{d}^\top)^\top$ is fixed independently of it and has the required counts. Moreover,

$$-\mathbf{H}(\boldsymbol{z} + \sigma \boldsymbol{e}_i) + \mathbf{D}\boldsymbol{d} = \mathbf{H}_{\mathcal{I}}\boldsymbol{x}_{\mathcal{I}} + \mathbf{D}\boldsymbol{d}.$$

The assumed probability bound applies to this vector, which passes the added-coordinate check and yields squared length $1 + \|\boldsymbol{z}\|^2 + N_+ + N_- = \|\boldsymbol{x}\|^2$. The finite random choices are implemented with bounded numbers of bits as in Section E.6.4, changing the probability by at most $2^{-\Omega(n^2)}$. □

**Finding a shortest vector.**

**Theorem 5.7** (Correctness)**.** *Fix $D \geq 1$, rational constants $\delta_0 \in (1/2, 1)$, $\kappa \in (0, 1)$, $\lambda, \varepsilon, \eta > 0$, and $c_0 = 1, c_1, \ldots, c_k > 0$, and integers $k, \nu \geq 1$. With $p_a$ as in (49), assume that (39) holds for every $a > 0$ with $\delta_0 \leq \mathcal{H}(p_a) \leq D$. Let $n_0, t_0, \ldots, t_k$ be the parameters specified in Section 5.1, and put $\rho = \delta n / n_0$. Assume that, for all sufficiently large $n$ and all inputs with $\delta_0 < \delta \leq D$, the tuples $(\rho, t_0, \ldots, t_k)$ lie in a fixed compact set satisfying*

$$\rho > 0, \qquad 0 < t_i < \frac{1}{2} \quad (0 \leq i \leq k), \qquad \frac{t_{i-1}}{2} < t_i < \frac{1 - t_{i-1}}{2} \quad (1 \leq i \leq k)\ .$$

*For each tuple, define*

$$r_i = 2t_{i-1} + (1-2t_{i-1})\mathcal{H}\left(\frac{t_i - t_{i-1}/2}{1-2t_{i-1}}, \frac{t_i - t_{i-1}/2}{1-2t_{i-1}}, \frac{1-t_{i-1}-2t_i}{1-2t_{i-1}}\right) \quad (1 \le i \le k) \ ,$$

*and set $d_0 = \rho$ and $d_i = r_i - \eta$ for $1 \le i \le k$. Require throughout the same compact set that*

$$d_0 > d_1 > \cdots > d_k > 0, \quad h(2t_0) > \frac{d_0}{2}, \quad \text{and} \quad h(2t_i) > \frac{d_{i-1}}{2}, \quad r_i > \frac{d_i}{2} + 2t_i - t_{i-1} \quad (1 \le i \le k) \ .$$

*For integers $q \ge 2$, $n \ge 1$, and $m \ge 0$ with $\delta \le D$, Algorithm 4 always returns a nonzero vector of $\Lambda$. For sufficiently large $n$, for all but a $2^{-\Omega(n^{2/3})}$ fraction of uniform input matrices $\mathbf{H}$, the returned vector is a shortest nonzero vector of $\Lambda$ with probability at least $1 - 2^{-\Omega(n)}$ over the algorithm's randomness, with $\mathbf{H}$ held fixed.*

*Proof.* The $m = 0$, $n = 1$, and small-modulus branches are exact on every input. Otherwise, $n \ge 2$ and $m \ge 1$. By Lemma 5.3, the ball of squared radius $u + 1$ contains a nonzero lattice vector outside a $2^{-\Omega(n^{2/3})}$ fraction of matrices. Intersecting with the ball of squared radius $q^2$ preserves this property because $q\boldsymbol{e}_1$ is a lattice vector. This proves the exact-search case.

Consider $\delta > \delta_0$. Compactness gives uniform positive margins in the stated inequalities. The uniform version of Lemma 3.2 therefore gives omission probability at most $2^{-(\nu - o(1))n_0^{2/3}}$ for every fixed vector with the prescribed sign counts, over a uniform input matrix of the short-solution procedure and its independent internal randomness. By Lemma 5.4, $\mathcal{H}(p_a) = \delta + o(1)$, and (58) bounds the number of integer vectors of squared norm at most $u+1$ by $q^m 2^{n^{2/3} + O_D((\log n)^2)}$. Outside a $2^{-\Omega(n/\log n)}$ fraction of matrices, every nonzero lattice vector in this ball of length less than $q$ has each sign count within $\varepsilon n/2$ of $np_a(1)$ and satisfies (35) with tolerance $\varepsilon/2$. Apply Lemma 5.5 with a fixed tolerance smaller than both $\varepsilon/2$ and half the positive minimum of $p_a(1)$ on the compact parameter range. This also ensures a coordinate equal to 1 or $-1$. The proof of this coordinate bound uses Lemma E.1 to exclude short vectors whose coordinates have a common divisor greater than one with $q$. The argument counts all integer vectors with unsuitable coordinates before imposing the random congruences; it does not assume that a shortest lattice vector is sampled from $p_a$.

The strict inequalities in (39) have uniform positive margins on the compact range $\delta_0 \le \mathcal{H}(p_a) \le D$. By continuity, the same holds on a slightly enlarged compact range containing the actual value $\mathcal{H}(p_a) = \delta + o(1)$, including when $\mathcal{H}(p_a) > D$. Together with this estimate, they give (38) for all permitted sign counts when $n$ is sufficiently large. Each candidate just described therefore has a case whose counts fit both the original and added positions. The outward numerical tests retain this case.

Each fixed candidate $\boldsymbol{x}$ has a $\pm 1$ coordinate. Condition on the columns of $\mathbf{H}$ corresponding to all other coordinates. Exactly one value of the remaining uniform column satisfies $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$, so $\Pr_{\mathbf{H}}[\mathbf{H}\boldsymbol{x} = \mathbf{0}_m] = q^{-m}$ even for composite $q$. Summing the conditional omission bounds from Lemma 5.6 over the ball gives

$$2^{n^{2/3} + O_D((\log n)^2)} \left(2^{-(\nu - o(1))n_0^{2/3}} + 2^{-\Omega(n^2)}\right) = 2^{-\Omega(n^{2/3})},$$

since $n_0/n = (1-\kappa)^{-1} + o(1)$ and $\nu(1-\kappa)^{-2/3} > 1$. On the other good events, either $q\boldsymbol{e}_1$ is shortest or a shortest vector is one of these candidates. One round thus fails with probability $2^{-\Omega(n^{2/3})}$ over both the input matrix and the algorithm's randomness.

Markov's inequality shows that all but a $2^{-\Omega(n^{2/3})}$ fraction of matrices have conditional one-round failure probability at most $1/4$. For each such fixed matrix, the $n$ rounds use independent random bits. They all fail with probability at most $4^{-n}$. □

### 5.3 Computation costs

We count the prescribed coordinate assignments and one call to the short-solution enumeration for each assignment. All remaining positions are used, so there is no additional exponential factor from selecting their locations. Successive cases reuse storage.

For the limiting proportions $t_i = c_i(p_a(1)+\lambda)$ and equation rate $\rho = (1-\kappa)\mathcal{H}(p_a)$, let $T_{\mathrm{in}}(a), S_{\mathrm{in}}(a)$ be the exponents (9) of the common enumeration per coordinate. The coordinate and equation inequalities hold with uniform margins, so these are continuous functions on the compact range in use. In the time bound below, the terms $\mathcal{H}(r_a)$, $\varepsilon$, and $T_{\mathrm{in}}(a)/(1-\kappa)$ account for the prescribed coordinate assignments, their allowed surplus, and a completion of dimension $n_0 = n/(1-\kappa)+O(1)$.

**Theorem 5.8** (Complexity)**.** *Fix $D \geq 1$, rational constants $\delta_0 \in (1/2, 1)$, $\kappa \in (0,1)$, $\lambda, \varepsilon, \eta > 0$, and $c_0 = 1, c_1, \ldots, c_k > 0$, and integers $k, \nu \geq 1$. With $p_a, r_a$ as in (49), suppose every completion uses dimension $n_0 = \lceil n/(1-\kappa) \rceil$ and terminates in time $2^{(T_{\mathrm{in}}(a)+o(1))n_0}$ and space $2^{(S_{\mathrm{in}}(a)+o(1))n_0}$, uniformly over the input range, where $T_{\mathrm{in}}, S_{\mathrm{in}}$ are continuous. Then Algorithm 4, on every input matrix with integer $q \geq 2$, $n \geq 1$, and $m \geq 0$ satisfying $\delta \leq D$, has time exponent*

$$\max\left\{\frac{\delta_0}{2},\ \sup_{a>0:\ \delta_0 \leq \mathcal{H}(p_a) \leq D}\left(\mathcal{H}(r_a)+\varepsilon+\frac{T_{\mathrm{in}}(a)}{1-\kappa}\right)\right\} \tag{41}$$

*and space exponent*

$$\max\left\{\frac{\delta_0}{4},\ \sup_{a>0:\ \delta_0 \leq \mathcal{H}(p_a) \leq D}\frac{S_{\mathrm{in}}(a)}{1-\kappa}\right\},$$

*up to $o(1)$ and factors polynomial in the binary input length.*

*Proof.* The $m = 0$ and $n = 1$ branches take polynomial time. The small-modulus branch enumerates supports of size at most $\min\{n, q^2\}$ and assigns nonzero coordinate values from $\{-q, \ldots, -1, 1, \ldots, q\}$, discarding assignments whose squared norm exceeds $q^2$. The number of assignments is at most

$$\sum_{s=1}^{\min\{n,q^2\}} \binom{n}{s}(2q)^s.$$

This is $O(n^4)$ for $q = 2$ and $2^{O(\sqrt{n}\log n)}$ when $q^4 \leq n$. Each assignment requires only polynomial time to check. The count defining $u$ takes polynomial time by Lemma 5.1, and the numerical parameters do so by Section E.6.1.

For exact search, Lemma 5.2 and (58) give

$$\frac{a(u+1)+n\ln\Theta(a)}{\ln 2} = \delta n + n^{2/3} + O((\log n)^2),$$

with $a = \Omega(1)$ and $a = O(\log(n+2))$, and (10) holds for $\tau = 0$ by Lemma A.3. The nonzero exact-search bound of Lemma 3.5 with $b \leq u+1$ gives time $2^{\delta n/2+O(n^{2/3})}$ and space $2^{\delta n/4+O(n^{2/3})}$. The condition $\delta \leq \delta_0$ bounds their exponents by $\delta_0/2, \delta_0/4$. These bounds are uniform even when $\delta \to 0$; this branch requires no positive lower bound on the equation rate.

Under the product distribution $r_a^{\otimes n}$, each vector in (35) has probability at least $2^{-n(\mathcal{H}(r_a)+\varepsilon)}$. Hence the completion branch enumerates at most $2^{n(\mathcal{H}(r_a)+\varepsilon)+O(1)}$ choices of $\boldsymbol{z}$, with polynomial delay and space by Section E.6.3. There are polynomially many choices of $i, \sigma, N_+, N_-$. Each case makes one call, costing $2^{(T_{\mathrm{in}}(a)/(1-\kappa)+o(1))n}$; the $n$ rounds add only a polynomial factor. The actual rate $\rho = \delta n/n_0$ equals $(1-\kappa)\mathcal{H}(p_a)+o(1)$, and the rational proportions differ from their limiting

values by $O(n^{-2})$. Uniformity of the common construction and continuity of its entropy formulas absorb these errors in $o(n)$.

Only one coordinate assignment, one added matrix, the current shortest vector, and one completion workspace are stored at once. The exponential space rate is therefore $S_{\mathrm{in}}(a)/(1-\kappa)$. Taking the suprema over the compact range and the larger bounds over the two branches proves the claim. □

The same two theorems apply to a finite partition of the completion density range into bands with separate parameter choices. Require the hypotheses on the closure of each band, so that continuity provides a neighborhood covering the $o(1)$ difference between $\mathcal{H}(p_a)$ and $\delta$, including at a band endpoint. There are only finitely many choices, so their positive margins and uniform error estimates have a common bound. The total time and space exponents are the maxima of the corresponding band suprema and the exact-search exponents. In particular, $T_{\mathrm{in}}, S_{\mathrm{in}}$ need only be continuous within each profile; profiles need not agree at a switching point.

**Time and space bounds as functions of density.** For $\delta > 0$, let $a(\delta) > 0$ be the unique solution of $\mathcal{H}(p_{a(\delta)}) = \delta$. Existence and uniqueness follow from Eq. (51) and Lemma A.2(4), since $a \mapsto \mathcal{H}(p_a)$ decreases strictly from $+\infty$ to 0. The value $a(\delta)$ can be approximated by bisection. The quantities $p_{a(\delta)}(1) = e^{-a(\delta)}/\Theta(a(\delta))$ and $\mathcal{H}(r_{a(\delta)})$ are evaluated using Eqs. (49) and (52), respectively.

Fix $\kappa, \lambda, \eta, \varepsilon$ and a ratio profile $c = (1, c_1, \ldots, c_k)$. With $T_{\mathrm{in}}(a), S_{\mathrm{in}}(a)$ as defined before Theorem 5.8, that is, (9) evaluated at $t_i = c_i(p_a(1) + \lambda)$ and $\rho = (1-\kappa)\mathcal{H}(p_a)$, the density functions are

$$T_c(\delta) = \mathcal{H}(r_{a(\delta)}) + \varepsilon + \frac{T_{\mathrm{in}}(a(\delta))}{1-\kappa}, \qquad S_c(\delta) = \frac{S_{\mathrm{in}}(a(\delta))}{1-\kappa}. \tag{42}$$

On every compact interval where the profile satisfies the strict conditions with positive uniform margins, these continuous functions give time $2^{T_c(\delta)n+o(n)}$ and space $2^{S_c(\delta)n+o(n)}$. The errors are uniform. In an actual input the counted radius gives $\mathcal{H}(p_a) = \delta + o(1)$, and continuity absorbs this difference and the rational approximation errors. With an exact-search cutoff $\delta_0$, use the pair $(\delta/2, \delta/4)$ for $\delta \le \delta_0$ and $(T_c(\delta), S_c(\delta))$ above the cutoff.

Parameter optimization can also be stated as a function. For the fixed constants and number $k$ of decomposition steps just specified, let $\mathcal{C}_\delta$ contain the profiles with $c_0 = 1$ and $c_i > 0$ for which the proportions $t_i = c_i(p_{a(\delta)}(1) + \lambda)$ satisfy $0 < t_i < 1/2$ and $t_{i-1}/2 < t_i < (1 - t_{i-1})/2$, the strict equation and parity constraints (5) and (8) at $\rho = (1-\kappa)\delta$, and the padding conditions (39) at $a = a(\delta)$. Define

$$\begin{aligned} \mathcal{F}_\delta &= \{(\delta/2, \delta/4)\} \cup \{(T_c(\delta), S_c(\delta)) : c \in \mathcal{C}_\delta\}, \\ T_{\mathrm{opt}}(\delta; s) &= \inf\{T : (T, S) \in \mathcal{F}_\delta,\ S \le s\}. \end{aligned} \tag{43}$$

The infimum of the empty set is $+\infty$. This expresses the best proved time bound in this fixed parameter family at a prescribed space exponent $s$. The exact-search pair is available for every fixed bounded density by the same counted radius argument. Formula (43) is an optimization objective, not a claim that its infimum has been computed or attained. A uniform algorithm uses certified profiles on compact intervals. Minimizing their evaluated time bounds gives a computable upper bound on this objective, with space measured for the same selected profile. Appendix G.5 supplies such profiles and an evaluator for arbitrary densities through 1.5.

Table 2 summarizes the concrete bounds obtained from the parameter choices in Section 5.4. For each listed upper density $D$, the time and space bounds are $2^{Tn+o(n)}$ and $2^{Sn+o(n)}$, respectively, throughout $0 \le \delta \le D$.

| Upper density $D$ | Time exponent $T$ | Space exponent $S$ |
|---|---|---|
| 1 | 0.27400949 | 0.261392024 |
| 1.05 | 0.286 | 0.270 |
| 1.10 | 0.298 | 0.276 |
| 1.20 | 0.326 | 0.286 |
| 1.30 | 0.361 | 0.309 |
| 1.40 | 0.408 | 0.320 |
| 1.50 | 0.468 | 0.329 |

Table 2: Proved SVP bounds holding uniformly over $0 \le \delta \le D$. Each row above density one combines its parameter rule for $1 < \delta \le D$ with the first row's rule for $0 \le \delta \le 1$. These are feasible upper bounds, with no claim of optimality.

### 5.4 Parameter selection

**Density at most one.** We first restrict to $D = 1$ and choose the parameters according to the complete time bound, including exact search and the enumeration of prescribed coordinates. We also compare the choice with a broader family that may use a smaller completion dimension and repeat random choices of the remaining coordinate positions. This comparison determines whether changing that dimension or adding decomposition steps can improve the proved time exponent.

In that broader family, let $\rho > 0$ and $n_0 = (\mathcal{H}(p_a)/\rho)n + o(n)$, and let $T_{\text{in}}, S_{\text{in}}$ be the common time and space exponents for the chosen proportions. The choices may depend on the input density. For $\alpha > 0$ and $0 \le w \le 1$, define

$$G(\alpha, w) = h(w) - \min\{1, \alpha + w\}\, h\!\left(\frac{w}{\min\{1, \alpha + w\}}\right). \tag{44}$$

This is the exponent of the reciprocal probability that a uniform subset of $\min\{1, \alpha + w\}n$ positions among $n$ contains a prescribed set of $wn$ positions. The available space for the added coordinates limits the subset size by $n_0 - 2t_0 n_0 + wn + o(n)$. Replacing the number of remaining original positions by its upper bound $n$ gives the repetition-cost bound $G(\alpha, w)$ with $\alpha = (1 - 2t_0)\mathcal{H}(p_a)/\rho$. Consequently the proved complete time objective in this family is

$$\max\left\{\frac{\delta_0}{2},\ \sup_{a>0:\ \delta_0 \le \mathcal{H}(p_a) \le 1}\left(\mathcal{H}(r_a) + \varepsilon + \max_{\substack{0 \le w \le 1 \\ |w - 2p_a(1)| \le 2\varepsilon}} G\left(\frac{(1-2t_0)\mathcal{H}(p_a)}{\rho}, w\right) + \frac{T_{\text{in}}\mathcal{H}(p_a)}{\rho}\right)\right\}. \tag{45}$$

Every tuple must satisfy the common coordinate, equation, and parity conditions and $t_0\mathcal{H}(p_a)/\rho > p_a(1) + \varepsilon$. The optimization includes every fixed depth $k$, the proportions, the completion dimension, the cutoff, and the positive margins. The condition $h_0 > \rho$ is not imposed, since only the bound on the probability of omitting a fixed vector is used.

Numerical search of this complete objective identifies a choice whose completion dimension approaches $n$ and whose first proportion approaches $p_a(1)$. We leave small positive margins by increasing the dimension and that proportion slightly. Use

$$\begin{gathered}\delta_0 = 0.54, \qquad k = 3, \qquad \kappa = 10^{-8}, \qquad \lambda = 10^{-9}, \qquad \eta = \varepsilon = 10^{-10}, \qquad \nu = 16, \\ (c_0, c_1, c_2, c_3) = (1, 0.874597414466, 0.589775545104, 0.310623936690),\end{gathered} \tag{46}$$

with $n_0 = \lceil n/(1-\kappa) \rceil$ and the rational proportions in (34). The whole count window then has $G = 0$. The coordinate and congruence proportions remain in a compact region with positive margins. The

bound on the probability of omitting a fixed vector in Lemma 3.2 therefore holds uniformly over the input-dependent proportions by Section 3.1.1.

A rational interval calculation verifies these conditions and the complete suprema over $0.54 \le \mathcal{H}(p_a) \le 1$ (Lemma G.1). It includes the rounding error in $\widehat{p}$ and gives time and space

$$2^{0.27400949n+o(n)} \qquad \text{and} \qquad 2^{0.261392024n+o(n)}.$$

Exact search contributes exponents 0.27 and 0.135, respectively. Thus these are the bounds for the full algorithm, with the success guarantee of Theorem 5.7.

The lower-bound argument optimizes over all choices in (45), including parameters that vary with density. It treats $k = 1$ and $k = 2$ separately and bounds every larger $k$ using its first three combinations. The remaining continuous domain is covered by rational interval bounds and a local convexity argument. The result is

$$0.2740094 \;\le\; \inf\{\text{complete time exponent in (45)}\} \;<\; 0.27400949. \tag{47}$$

The full constraints and both certificates are proved in Appendix G. Our three-step choice is therefore within $9 \cdot 10^{-8}$ of this global infimum. Strict positive margins need not permit attainment of the infimum.

**Density above one.** The threshold $\delta = 1$ is not required by the fixed-candidate argument. The equation rate may lie in any fixed compact subset of $(0, \infty)$ on which the strict conditions of Section 3.1.1 hold. The counting estimates extend to $\delta \le D$ for fixed $D$, and the factor $q^m$ in the candidate count cancels the membership probability $q^{-m}$ in the union bound over candidates. Consequently Theorems 5.7 and 5.8 also apply above density one when suitable parameters have been certified.

**Theorem 5.9** (SVP at bounded density)**.** *For each row $(D, T, S)$ of Table 2, there is a parameter choice for Algorithm 4 on all integers $q \ge 2$, $n \ge 1$, and $m \ge 0$ with $0 \le \delta \le D$. Every execution returns a nonzero lattice vector, takes time $2^{Tn+o(n)}$, and uses space $2^{Sn+o(n)}$, omitting factors polynomial in the binary input length. For sufficiently large $n$, for all but a $2^{-\Omega(n^{2/3})}$ fraction of uniform input matrices, the returned vector is a shortest nonzero vector of $\Lambda$ with probability at least $1 - 2^{-\Omega(n)}$ over the algorithm's randomness, with the matrix held fixed.*

*Proof.* Use (46) for $\delta \le 1$ and the profile for the selected row of Table 2 for $1 < \delta \le D$. The latter profiles and their uniform margins are certified in Lemma G.3. Apply Theorems 5.7 and 5.8 on these two closed density bands using the finite-band construction above. The first band has the time and space bounds proved above for (46); taking the larger bounds gives the table. For $D = 1$ only the first band is needed. □

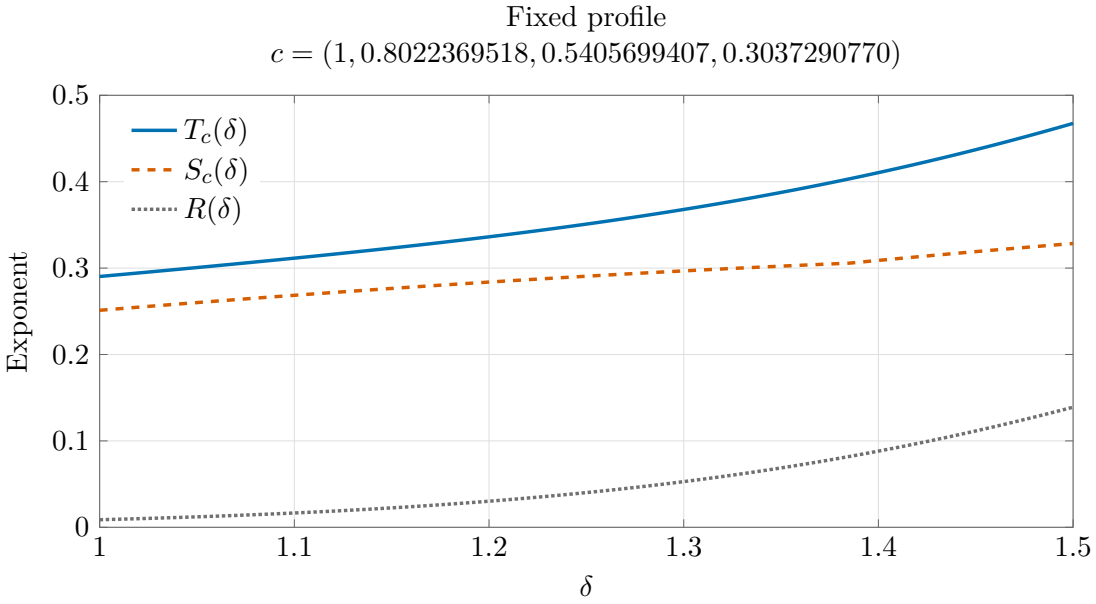


Figure 6: The density functions for the single fixed profile in the last row of Table 3. The curves evaluate (42) and the large-coordinate term $\mathcal{H}(r_{a(\delta)})$; they are sampled illustrations, with uniform guarantees supplied by the interval certificate. They are feasible cost bounds, not optimized complexity curves.

The certificates for the larger density intervals are given in Appendix G.5. These are feasible upper bounds, separate from the optimization certificate for $D = 1$; the time exponent 0.27400949 is not asserted above density one. The additional cost is accounted for by both the prescribed-coordinate entropy $\mathcal{H}(r_a)$ and the completion exponents in (41). All constants in this bounded-density analysis may depend on $D$; no uniform assertion is made when $\delta$ grows without bound with $n$.

# Supplementary material

## A Theta Sums

This appendix collects the definitions and estimates for the one-dimensional theta sums used in the paper: in the cost bound of the exact search (Lemma 3.5), in the condition on the $\gamma$-CVP analysis parameter (Eq. (22)), and in the counting arguments for the SVP search (Appendix E).

**Definition A.1** (Theta sums)**.** For a real number $a > 0$ and a real shift $u$, put

$$\Theta_u(a) = \sum_{j\in\mathbb{Z}} e^{-a(j-u)^2}, \qquad \Theta(a) = \Theta_0(a). \tag{48}$$

Define the probability mass function $p_a\colon \mathbb{Z} \to [0,1]$, its second moment $\mu(a)$, and the probability mass function $r_a\colon \mathbb{Z} \to [0,1]$ by

$$p_a(j) = \frac{e^{-aj^2}}{\Theta(a)}, \qquad \mu(a) = \sum_{j\in\mathbb{Z}} j^2 p_a(j), \qquad r_a(j) = \begin{cases} p_a(0) + 2p_a(1), & j = 0, \\ 0, & |j| = 1, \\ p_a(j), & |j| \geq 2. \end{cases} \tag{49}$$

For an integer $d \geq 0$ and a real number $u \geq 0$, put $\mathcal{B}_d(u) = \{\boldsymbol{z} \in \mathbb{Z}^d : \|\boldsymbol{z}\|^2 \leq u\}$.

The series in (48) converges absolutely, and $r_a$ is obtained from $p_a$ by replacing the values $-1, 0, 1$ by zero. As in Definition 2.1, $\mathcal{H}$ denotes Shannon entropy computed with base-2 logarithms.

**Lemma A.2** (Theta sums, moments, and entropy)**.** *For $a > 0$, let $J$ have distribution $p_a$.*

1. Periodicity and symmetry. *For every $u \in \mathbb{R}$,*

$$\Theta_{u+1}(a) = \Theta_u(a) = \Theta_{-u}(a). \tag{50}$$

2. Entropy identities. *The entropies of $p_a$ and $r_a$ satisfy*

$$\mathcal{H}(p_a) = \frac{a\mu(a) + \ln \Theta(a)}{\ln 2}, \tag{51}$$

$$\mathcal{H}(r_a) = \mathcal{H}(p_a) - \frac{r_a(0)\ln(1 + 2e^{-a}) + 2ap_a(1)}{\ln 2}. \tag{52}$$

3. Monotonicity of $\Theta$ and $\mu$. *Both $\Theta$ and $\mu$ are strictly decreasing on $(0,\infty)$, and $\mu'(a) = -\operatorname{Var}[J^2]$. Moreover, $\mu(a) \sim 1/(2a)$ as $a \downarrow 0$ and $\mu(a) \to 0$ as $a \to \infty$.*

4. Monotonicity of entropy. *As $a$ increases from $0$ to $\infty$, $\mathcal{H}(p_a)$ decreases strictly from $\infty$ to $0$, with derivative $-a\operatorname{Var}[J^2]/\ln 2$.*

5. Bounds for $\Theta$ and the variance. *We have $1 + 2e^{-a} \leq \Theta(a)$, $\ln \Theta(a) \leq \Theta(a) - 1 \leq \mu(a)\Theta(a)$, and $\operatorname{Var}[J^2] \geq 2p_a(0)p_a(1)$.*

6. Bounds for the second moment. *For every integer $d \geq 1$, $\mu(\ln 4) > 1/3$ and $\mu(\ln(8d)) < 1/d$.*

*Proof. (1)* Reindex the sums for $\Theta_{u+1}(a)$ and $\Theta_{-u}(a)$ by $k = j - 1$ and $k = -j$, respectively. Both give $\Theta_u(a)$.

*(2)* Taking the expectation of $-\ln p_a(J) = aJ^2 + \ln \Theta(a)$ gives (51), since $\mu(a)$ is the second moment of $J$. Passing from $p_a$ to $r_a$ combines the masses at $-1, 0, 1$ into a single mass at 0. Thus

$$(\ln 2)(\mathcal{H}(p_a) - \mathcal{H}(r_a)) = p_a(0) \ln \frac{r_a(0)}{p_a(0)} + 2p_a(1) \ln \frac{r_a(0)}{p_a(1)}.$$

Substituting $p_a(1) = e^{-a} p_a(0)$ and $r_a(0) = (1 + 2e^{-a}) p_a(0)$ gives (52).

*(3)* On each interval $a \geq b > 0$, the first two derivatives of $e^{-aj^2}$ are bounded in absolute value by $j^2 e^{-bj^2}$ and $j^4 e^{-bj^2}$, respectively. Both bounds are summable over $\mathbb{Z}$, so termwise differentiation and the quotient rule give

$$\Theta'(a) = -\Theta(a)\mu(a), \qquad \mu'(a) = -\sum_{j \in \mathbb{Z}} j^4 p_a(j) + \mu(a)^2 = -\operatorname{Var}[J^2].$$

Since $p_a(0)$ and $p_a(1)$ are positive, $\mu(a) > 0$ and $J^2$ has positive variance. Both derivatives are therefore negative.

For $a \downarrow 0$, consider the Riemann sums with mesh $\sqrt{a}$ for $e^{-x^2}$ and $x^2 e^{-x^2}$ on $\mathbb{R}$. Both functions are integrable and decrease as $|x|$ increases beyond 1, so the sums converge to the corresponding Gaussian integrals. This gives

$$\sqrt{a}\, \Theta(a) \longrightarrow \sqrt{\pi}, \qquad a^{3/2} \sum_{j \in \mathbb{Z}} j^2 e^{-aj^2} \longrightarrow \frac{\sqrt{\pi}}{2}.$$

Dividing the second limit by the first gives $a\mu(a) \to 1/2$. For $a \geq 1$ and $j \neq 0$, we have $e^{-aj^2} \leq e^{-a} e^{1-j^2}$. Summing this bound with weights 1 and $j^2$ gives $\Theta(a) - 1 = O(e^{-a})$ and $\mu(a) = O(e^{-a})$, since $\Theta(a) \geq 1$.

*(4)* Differentiate (51) and use $\Theta'(a)/\Theta(a) = -\mu(a)$ to obtain

$$(\ln 2) \frac{d}{da} \mathcal{H}(p_a) = \mu(a) + a\mu'(a) + \frac{\Theta'(a)}{\Theta(a)} = a\mu'(a) = -a \operatorname{Var}[J^2] < 0.$$

As $a \downarrow 0$, the estimates in (3) give $a\mu(a) \to 1/2$ and $\Theta(a) \to \infty$. As $a \to \infty$, they give $a\mu(a) \to 0$ and $\Theta(a) \to 1$. The entropy limits now follow from (51).

*(5)* The terms $j = 0, \pm 1$ give $\Theta(a) \geq 1 + 2e^{-a}$. Using $\ln x \leq x - 1$ for $x > 0$ and $j^2 \geq 1$ for $j \neq 0$, we obtain

$$\ln \Theta(a) \leq \Theta(a) - 1 = \sum_{j \neq 0} e^{-aj^2} \leq \sum_{j \in \mathbb{Z}} j^2 e^{-aj^2} = \mu(a)\Theta(a).$$

Expanding the square gives

$$\operatorname{Var}[J^2] = \frac{1}{2} \sum_{j,k \in \mathbb{Z}} p_a(j) p_a(k) (j^2 - k^2)^2.$$

The four ordered pairs $(0, \pm 1)$ and $(\pm 1, 0)$ contribute $2p_a(0)p_a(1)$ in total, and all other terms are nonnegative.

*(6)* At $a = \ln 4$, the terms $j = \pm 1, \pm 2$ give $\sum_{j \in \mathbb{Z}} j^2 4^{-j^2} \geq 17/32$. For $j \geq 2$, we have $j^2 \geq 4 + 3(j-2)$, so

$$\Theta(\ln 4) \leq 1 + \frac{1}{2} + 2 \sum_{k \geq 0} 4^{-4-3k} = \frac{95}{63}.$$

Thus $\mu(\ln 4) \geq (17/32)(63/95) > 1/3$.

For $d \geq 1$, use $\Theta(\ln(8d)) \geq 1$ and sum over all positive integers $k$ to bound the sum over squares $k = j^2$. This gives

$$\mu(\ln(8d)) \leq 2 \sum_{k \geq 1} k(8d)^{-k} = \frac{16d}{(8d-1)^2} \leq \frac{16}{49d} < \frac{1}{d},$$

where $8d - 1 \geq 7d$. □

The next two lemmas bound the shifted sums $\Theta_u(a)$ by $\Theta(a)$.

**Lemma A.3** (Maximum of the shifted theta sum)**.** *For every $a > 0$ and $u \in \mathbb{R}$,*

$$\Theta_u(a) \leq \Theta_0(a) = \Theta(a).$$

*Proof.* Applying the Poisson summation formula to $x \mapsto e^{-a(x-u)^2}$ and pairing the terms at $k$ and $-k$ gives

$$\Theta_u(a) = \sqrt{\frac{\pi}{a}} \left( 1 + 2 \sum_{k \geq 1} e^{-\pi^2 k^2/a} \cos(2\pi k u) \right).$$

Each cosine has a positive coefficient and is at most 1. At $u = 0$ every cosine equals 1, which proves the bound. □

For shifts $|u| \leq 1/2$, the first two factors in Jacobi's triple product give an upper bound that depends on $u$.

**Lemma A.4** (Gaussian upper bound for the shifted theta sum)**.** *For $a > 0$, let*

$$\chi(a) = -8 \ln \left( \tanh \frac{\pi^2}{2a} \tanh \frac{3\pi^2}{2a} \right).$$

*If $\tanh^2(\pi^2/(2a)) \geq 3/5$, then for every $u \in [-1/2, 1/2]$,*

$$\Theta_u(a) \leq \Theta(a) e^{-\chi(a) u^2}.$$

*Proof. Product formula.* For $j \geq 1$, write $t_j = e^{-(2j-1)\pi^2/a}$ and $b_j = (t_j + t_j^{-1})/2 = \cosh((2j-1)\pi^2/a)$. Apply Jacobi's triple product formula [DLMF, Eq. 20.5.3] to the Fourier series in the proof of Lemma A.3. In the ratio with the value at $u = 0$, the factors $1 - e^{-2j\pi^2/a}$ cancel, giving

$$\frac{\Theta_u(a)}{\Theta(a)} = \prod_{j \geq 1} \frac{1 + 2t_j \cos(2\pi u) + t_j^2}{(1 + t_j)^2} = \prod_{j \geq 1} \frac{b_j + \cos(2\pi u)}{b_j + 1}. \tag{53}$$

*Bound for one factor.* For $b \geq 4$ and $0 \leq t \leq 1$, we claim that

$$\frac{b + \cos(\pi t)}{b + 1} \leq \left( \frac{b-1}{b+1} \right)^{t^2}.$$

Let $f(x) = \ln((b+1)/(b+\cos x))$ for $0 \leq x \leq \pi$. Its derivatives satisfy

$$f'(x) = \frac{\sin x}{b + \cos x}, \qquad f'''(x) = \frac{\sin x \, (2 - b^2 + b \cos x)}{(b + \cos x)^3} \leq 0,$$

since $2-b^2+b\cos x \le -(b-2)(b+1)$. Thus $f'$ is concave. As $f'(0)=0$, concavity gives $f'(tx)\ge tf'(x)$. Integrating and using $f(0)=0$ yields

$$f(\pi t) = t\int_0^\pi f'(tx)\,dx \ge t^2 f(\pi).$$

Exponentiating the negatives of both sides gives the claimed bound.

*The first two factors.* The identity

$$\frac{b_j-1}{b_j+1} = \tanh^2\left(\frac{(2j-1)\pi^2}{2a}\right)$$

turns the hypothesis into $(b_1-1)/(b_1+1)\ge 3/5$, or $b_1\ge 4$. Since $b_2\ge b_1$, the bound for one factor applies to $b_1$ and $b_2$ with $t=2|u|$. Every remaining factor in (53) lies in $(0,1]$ because $b_j>1$. Consequently,

$$\frac{\Theta_u(a)}{\Theta(a)} \le \left(\frac{b_1-1}{b_1+1}\frac{b_2-1}{b_2+1}\right)^{4u^2} = e^{-\chi(a)u^2}.$$

□

The counting argument in Appendix E uses the number of integer points in a ball that lie in a given residue class. The next lemma bounds this number and compares $\Theta$ at different arguments.

**Lemma A.5** (Theta sums over lattice cosets)**.** *Let $a>0$, let $d, v\ge 1$ be integers, and let $\boldsymbol{w}\in\mathbb{Z}^d$.*

1. Sum over a coset.
$$\sum_{\boldsymbol{x}\in\boldsymbol{w}+v\mathbb{Z}^d} e^{-a\|\boldsymbol{x}\|^2} \le \Theta(av^2)^d.$$

2. Number of points in a ball. *For every real $u\ge 0$,*
$$\left|\mathcal{B}_d(u)\cap(\boldsymbol{w}+v\mathbb{Z}^d)\right| \le \min\{e^{au/v}\Theta(av)^d,\ e^{au}\Theta(av^2)^d\}. \tag{54}$$

3. Comparison at $av$ and $a$. *For every $a_->0$, there is an integer $v_{a_-}\ge 2$, depending only on $a_-$, such that*
$$\Theta(av)^v \le \Theta(a) \tag{55}$$
*for all $a\ge a_-$ and all integers $v\ge v_{a_-}$. If $a\ge \ln 4$, the same bound holds for every integer $v\ge 2$.*

4. Comparison at $ag^2$ and $a$. *For every integer $g\ge 2$, $\Theta(ag^2)\le\Theta(4a)<\Theta(a)$.*

*Proof. (1)* Write each point of the coset as $\boldsymbol{x}=\boldsymbol{w}+v\boldsymbol{y}$ with $\boldsymbol{y}\in\mathbb{Z}^d$. Factoring the sum over the $d$ coordinates gives

$$\sum_{\boldsymbol{x}\in\boldsymbol{w}+v\mathbb{Z}^d} e^{-a\|\boldsymbol{x}\|^2} = \prod_{i=1}^d \Theta_{-w_i/v}(av^2) \le \Theta(av^2)^d,$$

where the inequality follows from Lemma A.3.

*(2)* For any $t>0$, each $\boldsymbol{x}\in\mathcal{B}_d(u)$ satisfies $1\le e^{t(u-\|\boldsymbol{x}\|^2)}$. Summing over the intersection, extending the sum to the whole coset, and applying (1) with parameter $t$ gives

$$\left|\mathcal{B}_d(u)\cap(\boldsymbol{w}+v\mathbb{Z}^d)\right| \le e^{tu}\sum_{\boldsymbol{x}\in\boldsymbol{w}+v\mathbb{Z}^d} e^{-t\|\boldsymbol{x}\|^2} \le e^{tu}\Theta(tv^2)^d.$$

Taking $t = a/v$ and $t = a$ gives the two bounds in (54).

*(3)* For $v \geq 2$ and $j \neq 0$, the inequality $j^2 \geq 1$ gives $e^{-avj^2} \leq e^{-a(v-1)}e^{-aj^2}$. Summing over $j \neq 0$ yields $\Theta(av) - 1 \leq e^{-a(v-1)}(\Theta(a) - 1)$. Since $1 - 1/x \leq \ln x \leq x - 1$ for $x \geq 1$ and $\Theta(a) > 1$, we obtain

$$\frac{v \ln \Theta(av)}{\ln \Theta(a)} \leq ve^{-a(v-1)}\Theta(a).$$

Fix $a_- > 0$. For all $a \geq a_-$, the right-hand side is at most $\Theta(a_-)\, ve^{-a_-(v-1)}$ by the monotonicity of $\Theta$. This upper bound tends to zero as $v \to \infty$. We can therefore choose $v_{a_-}$ so that the ratio is at most 1 for all $v \geq v_{a_-}$.

If $a \geq \ln 4$, then $\Theta(a) \leq 1 + 2\sum_{j \geq 1} 4^{-j} = 5/3$, since $j^2 \geq j$ for $j \geq 1$. Also, $ve^{-a(v-1)} \leq v/4^{v-1} \leq 1/2$ because $v \leq 2^{v-1}$ for $v \geq 2$. The ratio is then at most $5/6$. In both cases, exponentiating $v \ln \Theta(av) \leq \ln \Theta(a)$ proves (55).

*(4)* For $g \geq 2$ we have $ag^2 \geq 4a > a$. The claim follows because $\Theta$ is strictly decreasing by Lemma A.2(3). □

# B Short Solutions by Representations

This appendix gives the recursive implementation and the probability estimates used in Section 3.1. It also treats large moduli, finite random bits, and parameters chosen from the input dimensions. The construction follows Ozerov's overlapping representations, including the restrictions on intermediate list sizes [Oze16, Algorithm 9 and Theorem 43].

## B.1 Coordinate counts and construction parameters

We use the parameters $k, \nu, \rho, \eta, t_i$, the counts $a_i$ and $a'_i$, the sets $\mathcal{T}_i$ and $\mathcal{D}_i(\boldsymbol{x})$, and the rates (4) of Section 3.1. The choice of the $a_i$ gives $a_i = t_i n + O(1)$, $a_i \leq t_i n$, and $2a'_i \leq n - 2a_{i-1}$, and the factorial formulas give $\log |\mathcal{T}_i| = h_i n + O(\log n)$ and $\log |\mathcal{D}_i(\boldsymbol{x})| = r_i n + O(\log n)$. The algorithm may use decompositions outside $\mathcal{D}_i(\boldsymbol{x})$ as well; it always tests the sum and its coordinate counts over the integers. Since $\mathcal{D}_i(\boldsymbol{x}) \subset \mathcal{T}_i$ and each ordered pair of parts determines its sum,

$$|\mathcal{D}_i(\boldsymbol{x})| \leq |\mathcal{T}_i|, \qquad |\mathcal{T}_{i-1}|\,|\mathcal{D}_i(\boldsymbol{x})| \leq |\mathcal{T}_i|^2,$$
$$r_i \leq h_i, \qquad h_{i-1} + r_i \leq 2h_i.$$

The equation rates $d_0 = \rho$ and $d_i = r_i - \eta$ satisfy (5) with fixed positive margins, since the parameters are independent of $n$.

## B.2 The recursive algorithm

For $q^{\lceil \log(n+2) \rceil} \leq 2^n$, Algorithm 1 retains the last $m_i$ rows of $\mathbf{A}$ with $m_i$ as in (6), so that $\mathcal{G}_i = (\mathbb{Z}/q\mathbb{Z})^{m_i}$, $\mathbf{A}_{i+1} = \pi_i \mathbf{A}_i$, and $\log |\mathcal{G}_i| = d_i n + o(n)$; the parameter conditions ensure that its guards pass for all sufficiently large $n$. This subsection specifies the recursive call. A call combines vectors returned by two smaller calls. We limit both its output list and the number of left vectors paired with any one right vector, so that every execution has bounded cost: for $0 \leq i < k$, $\textsc{Combine}(i, \boldsymbol{t})$ returns a list of at most $u_i$ distinct vectors $\boldsymbol{x} \in \mathcal{T}_i$ satisfying $\mathbf{A}_i \boldsymbol{x} = \boldsymbol{t}$, and for each right vector it inspects at most $v_i$ left vectors with the required image under $\mathbf{A}_i$. Here $w$, $u_i$, and $v_i$ are the limits in (7). The factor $w$ makes exceeding these limits unlikely. Increasing the fixed integer $\nu$ improves the bound on the probability of omitting a fixed vector, while adding only $o(n)$ to the logarithm of the cost.

Algorithm 5 implements COMBINE. For a call with input $(i, \boldsymbol{t})$, order left vectors $\boldsymbol{p}$ by $\mathbf{A}_i\boldsymbol{p}$ and right vectors $\boldsymbol{u}$ by $\boldsymbol{t} - \mathbf{A}_i\boldsymbol{u}$. These are their ordering values; matching values are precisely the pairs whose sum has image $\boldsymbol{t}$. Elements of $\mathcal{G}_i$ are compared lexicographically using canonical residue coordinates. Stored lists use lexicographic order of vectors to break ties.

At the last level, HALVES$(\boldsymbol{t}, \boldsymbol{g}, \varepsilon)$ generates at most $u_k$ distinct vectors $\boldsymbol{x} \in \mathcal{T}_k$ satisfying $\mathbf{A}_k\boldsymbol{x} = \boldsymbol{t}$, ordered by $\boldsymbol{g} + \varepsilon\mathbf{A}_{k-1}\boldsymbol{x}$, where $\boldsymbol{g} \in \mathcal{G}_{k-1}$ and $\varepsilon \in \{\pm 1\}$. It supplies one vector at a time without storing its full output list; the next subsection gives its implementation and tie rule.

**Algorithm 5** The recursive call COMBINE$(i, \boldsymbol{t})$

**Parameters:** The maps $\mathbf{A}_j$, projections $\pi_j$, sets $\mathcal{T}_j$, and limits $u_j, v_j$.
**Input:** $0 \le i < k$ and $\boldsymbol{t} \in \mathcal{G}_i$.
**Output:** A list $\mathcal{L}_i$ of at most $u_i$ distinct $\boldsymbol{x} \in \mathcal{T}_i$ with $\mathbf{A}_i\boldsymbol{x} = \boldsymbol{t}$.

Choose $\boldsymbol{b}$ uniformly in $\mathcal{G}_{i+1}$; set $\boldsymbol{c} \leftarrow \pi_i(\boldsymbol{t}) - \boldsymbol{b}$.
**if** $i < k - 1$ **then**
    $\mathcal{X} \leftarrow$ COMBINE$(i + 1, \boldsymbol{b})$; $\mathcal{Y} \leftarrow$ COMBINE$(i + 1, \boldsymbol{c})$.
    Sort $\mathcal{X}$ by $\mathbf{A}_i\boldsymbol{p}$ and $\mathcal{Y}$ by $\boldsymbol{t} - \mathbf{A}_i\boldsymbol{u}$.
**else**
    Read left vectors from HALVES$(\boldsymbol{b}, \mathbf{0}, +1)$.
    Read right vectors from HALVES$(\boldsymbol{c}, \boldsymbol{t}, -1)$.
    Suspend each generation between successive vectors; store neither output list.
**end if**
$\mathcal{L}_i \leftarrow [\,]$; maintain a separate dictionary for duplicate tests.
**while** both sides have a current vector **do**
    **if** their ordering values differ **then**
        Advance the side with the smaller ordering value by one vector.
    **else**
        $\boldsymbol{d} \leftarrow$ the common ordering value; $\mathcal{B} \leftarrow [\,]$.
        **while** the left side has ordering value $\boldsymbol{d}$ **do**
            **if** $|\mathcal{B}| < v_i$ **then** append its current vector to $\mathcal{B}$.
            Advance the left side by one vector.
        **end while**
        **while** the right side has ordering value $\boldsymbol{d}$ **do**
            $\boldsymbol{u} \leftarrow$ its current vector.
            **for** each $\boldsymbol{p} \in \mathcal{B}$ **do**
                $\boldsymbol{x} \leftarrow \boldsymbol{p} + \boldsymbol{u}$ over $\mathbb{Z}$.
                **if** $\boldsymbol{x} \in \mathcal{T}_i$ and $\boldsymbol{x} \notin \mathcal{L}_i$ **then**
                    Append $\boldsymbol{x}$ to $\mathcal{L}_i$ and add it to the dictionary.
                    **if** $|\mathcal{L}_i| = u_i$ **then return** $\mathcal{L}_i$.
                **end if**
            **end for**
            Advance the right side by one vector.
        **end while**
    **end if**
**end while**
**return** $\mathcal{L}_i$.

Returned lists retain insertion order. Each recursive call samples its element $\boldsymbol{b} \in \mathcal{G}_{i+1}$ indepen-

dently of the preceding draws. Coordinate partitions are mutually independent and independent of these elements. An optional acceptance test is applied only to vectors emitted by the outer algorithm: rejecting one continues the scan and changes no list limit. For $q^{\lceil \log(n+2) \rceil} > 2^n$, we use the extension in Appendix B.5.

### B.3 Generating the last-level vectors

The last recursive call must produce vectors in the order required by its parent without storing their full list. We enumerate vectors on two disjoint coordinate blocks, generate pairs satisfying the required congruence, and merge their ordered outputs. The construction below also specifies when a fixed vector is guaranteed to appear despite the output limit.

Put $(n_1, n_2, n_3) = (a_k, a_k, n - 2a_k)$, the counts of the values $1, -1, 0$, respectively, and let

$$a'_j = \lfloor n_j/2 \rfloor \quad (1 \le j \le 3), \qquad n' = \sum_{j=1}^{3} a'_j.$$

A coordinate partition has a first block of $n'$ positions and its complement as the second block. A vector in $\mathcal{T}_k$ is compatible with the partition if its first block has exactly $a'_j$ coordinates of the $j$th value. For each partition, form two lists of vectors, each vector listed once: all vectors on the first block with exactly $a'_j$ entries of the $j$th value, and all vectors on the second block with exactly $n_j - a'_j$ such entries. Extend their vectors by zero outside their own blocks. Each list has $2^{h_k n/2 + O(\log n)}$ entries.

The parent supplies $\boldsymbol{t} \in \mathcal{G}_k$, a shift $\boldsymbol{g} \in \mathcal{G}_{k-1}$, and a sign $\varepsilon \in \{\pm 1\}$. We retain pairs whose sum $\boldsymbol{x}$ satisfies $\mathbf{A}_k \boldsymbol{x} = \boldsymbol{t}$ and order them by the vector

$$\boldsymbol{g} + \varepsilon \mathbf{A}_{k-1} \boldsymbol{x},$$

using lexicographic order on its canonical residue coordinates. We call this vector the ordering value. Pairs with the same ordering value must all be retained: their integer sums can have different coordinate counts in the parent call. The minimum-cost reduction used in exact search would not preserve these choices.

For example, with $\mathbf{A}_k = \mathbf{A}_{k-1} = \mathbf{0}$, take $\boldsymbol{p}_1 = (1, -1, 0, 0, 0)^\top$, $\boldsymbol{p}_2 = (0, 0, 1, -1, 0)^\top$, and $\boldsymbol{u} = (-1, 1, 0, 0, 0)^\top$. The two left vectors have the same image and squared length. If the parent requires two entries of each nonzero sign, only $\boldsymbol{p}_2 + \boldsymbol{u}$ has the required counts; $\boldsymbol{p}_1 + \boldsymbol{u} = \mathbf{0}$. Discarding $\boldsymbol{p}_2$ because its image agrees with that of $\boldsymbol{p}_1$ would lose this decomposition.

Group the second-block vectors by their $\mathbf{A}_k$-images. Within each group, store their $\varepsilon \mathbf{A}_{k-1}$-images in the prefix tree described in Algorithm 10, retaining all vectors at each terminal node in lexicographic order. For a first-block vector $\boldsymbol{p}$, an iterator traverses the group at $\boldsymbol{t} - \mathbf{A}_k \boldsymbol{p}$, translating its image coordinates by $\boldsymbol{g} + \varepsilon \mathbf{A}_{k-1} \boldsymbol{p}$. At each coordinate, the stored child labels are read in the cyclic order induced by this translation. The iterator stores its current path and its position among the records at the terminal node. Each advance takes polynomial time.

The algorithm uses $\ell = n(n+1)^3$ independent partitions, numbered $1, \ldots, \ell$. A heap merges all iterators by their ordering values; ties are broken by the partition index and then lexicographically by the two vectors. Thus the order is determined by the matrices, inputs, and sampled partitions. Each vector is emitted only from its first compatible partition, which removes duplicates without storing the emitted vectors.

**Algorithm 6** Ordered last-level generation: HALVES$(\boldsymbol{t}, \boldsymbol{g}, \varepsilon)$

**Parameters:** $a_k, u_k, \mathbf{A}_k, \mathbf{A}_{k-1}$, and $q$.
**Input:** $\boldsymbol{t} \in \mathcal{G}_k$, $\boldsymbol{g} \in \mathcal{G}_{k-1}$, and $\varepsilon \in \{\pm 1\}$.
**Output:** At most $u_k$ distinct $\boldsymbol{x} \in \mathcal{T}_k$ with $\mathbf{A}_k \boldsymbol{x} = \boldsymbol{t}$, in nondecreasing ordering value.

$(n_1, n_2, n_3) \leftarrow (a_k, a_k, n - 2a_k)$; $(a'_1, a'_2, a'_3) \leftarrow (\lfloor n_1/2 \rfloor, \lfloor n_2/2 \rfloor, \lfloor n_3/2 \rfloor)$; $n' \leftarrow a'_1 + a'_2 + a'_3$.
$\ell \leftarrow n(n+1)^3$; sample independent uniform $n'$-subsets $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$ of $\{1, \ldots, n\}$.
$\mathcal{H} \leftarrow$ empty min-heap; $f \leftarrow 0$; $f_j \leftarrow 0$ for $1 \le j \le \ell$.
**for** $j = 1, \ldots, \ell$ **do**
    List each vector on $\mathcal{I}_j$ with $a'_1, a'_2, a'_3$ entries of values $1, -1, 0$ exactly once in $\mathcal{X}_j$.
    List each vector on its complement with $n_1 - a'_1, n_2 - a'_2, n_3 - a'_3$ such entries exactly once in $\mathcal{Y}_j$.
    Extend these vectors by zero outside their respective blocks.
    Group $\mathcal{Y}_j$ by $\mathbf{A}_k \boldsymbol{u}$; build the ordered prefix tree in each group.
    **for** each $\boldsymbol{p} \in \mathcal{X}_j$ **do**
        Find the group at $\boldsymbol{t} - \mathbf{A}_k \boldsymbol{p}$; skip this $\boldsymbol{p}$ if it is empty.
        Start its iterator, translated by $\boldsymbol{g} + \varepsilon \mathbf{A}_{k-1} \boldsymbol{p}$.
        Insert its first pair into $\mathcal{H}$, retaining $j, \boldsymbol{p}, \boldsymbol{u}$ and the iterator.
    **end for**
**end for**
**while** $\mathcal{H} \neq \varnothing$ and $f < u_k$ **do**
    Remove the first record $(j, \boldsymbol{p}, \boldsymbol{u}, \text{iterator})$ in the specified heap order.
    **if** $f_j < u_k$ **then**
        $f_j \leftarrow f_j + 1$; $\boldsymbol{x} \leftarrow \boldsymbol{p} + \boldsymbol{u}$.
        **if** $j$ is the first partition compatible with $\boldsymbol{x}$ **then**
            Emit $\boldsymbol{x}$; $f \leftarrow f + 1$.
        **end if**
        **if** $f_j < u_k$ **then**
            Advance this iterator; insert its next pair into $\mathcal{H}$ if one exists.
        **end if**
    **end if**
**end while**

Here $f_j$ counts pairs inspected from partition $j$, and $f$ counts emitted vectors. Once $f_j = u_k$, remaining heap records from that partition are discarded without advancing their iterators. A suspended call retains its lists $\mathcal{X}_j, \mathcal{Y}_j$, prefix trees, heap, and counters; it stores no level-$k$ output list. The next two lemmas justify the ordered generation and its completeness under the stated limit.

**Lemma B.1.** *Fix integers $q \ge 2$, $n \ge 1$, $r, d \ge 0$, and matrices $\mathbf{C} \in (\mathbb{Z}/q\mathbb{Z})^{r \times n}$ and $\mathbf{D} \in (\mathbb{Z}/q\mathbb{Z})^{d \times n}$. Let $\mathcal{X}, \mathcal{Y}$ be lists of integer vectors in $\mathbb{Z}^n$, each record encoded in at most $\lambda \ge 1$ bits, and fix $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^r$, $\boldsymbol{g} \in (\mathbb{Z}/q\mathbb{Z})^d$, and $\varepsilon \in \{\pm 1\}$. A deterministic procedure generates every pair $(\boldsymbol{p}, \boldsymbol{u}) \in \mathcal{X} \times \mathcal{Y}$ satisfying*

$$\mathbf{C}(\boldsymbol{p} + \boldsymbol{u}) = \boldsymbol{t}$$

*in nondecreasing lexicographic order of the canonical residue coordinates of $\boldsymbol{g} + \varepsilon \mathbf{D}(\boldsymbol{p} + \boldsymbol{u})$. Pairs with equal ordering values are all retained. Ties are broken lexicographically by the two vectors and then by their positions in the input lists.*

*For every integer $b \ge 0$, producing the first $b$ pairs, or all pairs if fewer exist, takes time and*

*space, respectively,*

$$(1+|\mathcal{X}|+|\mathcal{Y}|+b)\ \mathrm{poly}\big(n,r,d,\log q,\log(|\mathcal{X}|+|\mathcal{Y}|+2),\log(b+2),\lambda\big) \quad \textit{and}$$
$$(1+|\mathcal{X}|+|\mathcal{Y}|)\ \mathrm{poly}\big(n,r,d,\log q,\log(|\mathcal{X}|+|\mathcal{Y}|+2),\log(b+2),\lambda\big).$$

*Proof.* Group $\mathcal{Y}$ by its images under $\mathbf{C}$. Within each group, build the prefix tree of the $\varepsilon\mathbf{D}$-images, retaining every record at its terminal node. Order terminal records by their vectors and positions in $\mathcal{Y}$. The groups partition the entries of $\mathcal{Y}$, so the total size of these trees is $|\mathcal{Y}|$ poly. For each $\boldsymbol{p}\in\mathcal{X}$, find the group at $\boldsymbol{t}-\mathbf{C}\boldsymbol{p}$ by binary search. Its translated traversal, with shift $\boldsymbol{g}+\varepsilon\mathbf{D}\boldsymbol{p}$, generates precisely the matching pairs for this $\boldsymbol{p}$ in the required order. Merge these iterators with a heap, using the stated tie rule.

Every pair belongs to one iterator and is generated once. Advancing one iterator either reads the next terminal record or follows one path up and down the tree, taking polynomial time. There is at most one iterator and one heap record per entry of $\mathcal{X}$. Thus the initial construction takes $(1+|\mathcal{X}|+|\mathcal{Y}|)\,\mathrm{poly}$ time and space, and each subsequent pair takes polynomial time. □

**Lemma B.2.** *For every input and every choice of the coordinate partitions, Algorithm 6 emits at most $u_k$ distinct vectors in*

$$\{\boldsymbol{x}\in\mathcal{T}_k : \mathbf{A}_k\boldsymbol{x}=\boldsymbol{t}\}$$

*in nondecreasing lexicographic order of the canonical residue coordinates of $\boldsymbol{g}+\varepsilon\mathbf{A}_{k-1}\boldsymbol{x}$, with ties broken by the index of the partition from which a vector is emitted and then lexicographically by its restrictions to the two blocks of that partition. If this solution set has at most $u_k$ vectors, every vector in it compatible with at least one sampled partition is emitted. Every execution takes time $(\sqrt{|\mathcal{T}_k|}+u_k)\,\mathrm{poly}$ and space $\sqrt{|\mathcal{T}_k|}\,\mathrm{poly}$.*

*Proof.* For each partition, Lemma B.1 with $\mathbf{C}=\mathbf{A}_k$ and $\mathbf{D}=\mathbf{A}_{k-1}$ gives the required ordered matching pairs. Their sums lie in $\mathcal{T}_k$ because the two blocks are disjoint and have the prescribed coordinate counts. A sum determines its two parts uniquely, so the pairs from one partition have distinct sums. The heap merges their orders, including the specified ties. Restricting each partition to its first $u_k$ pairs preserves that order. Emitting a vector only from its first compatible partition removes repetitions across partitions without storing earlier outputs.

If the full solution set has at most $u_k$ vectors, every partition has at most $u_k$ matching pairs. No partition's inspection limit then removes a pair. Each compatible vector appears in its first compatible partition, and the overall limit of $u_k$ distinct outputs cannot exclude it. This argument holds for every choice of the matrices, shift, and sign. In particular, it does not require the ordering supplied by the parent to be independent of the full matrix.

The factorial counts give $\sqrt{|\mathcal{T}_k|}\,2^{O(\log n)}$ entries in each of the lists $\mathcal{X}_j,\mathcal{Y}_j$. At most $\ell u_k$ pairs are inspected, and testing the first compatible partition takes at most $\ell$ coordinate-count tests per pair. After a partition reaches its limit, at most one remaining heap record per iterator is discarded. These removals cost only $\sqrt{|\mathcal{T}_k|}$ poly additional operations. Since $\ell=n(n+1)^3$, the construction time and the inspected pairs give the claimed time bound. The lists $\mathcal{X}_j,\mathcal{Y}_j$, trees, iterators, heap, and counters occupy $\sqrt{|\mathcal{T}_k|}$ poly space; the emitted vectors are not stored. □

For a fixed $\boldsymbol{x}\in\mathcal{T}_k$ and a uniformly chosen first block of size $n'$, the compatibility probability is

$$\frac{\prod_{j=1}^{3}\binom{n_j}{a'_j}}{\binom{n}{n'}}\geq(n+1)^{-3}.$$

Indeed, $\binom{n_j}{a'_j} \geq 2^{n_j}/(n_j+1)$ and $\binom{n}{n'} \leq 2^n$. With the $\ell = n(n+1)^3$ independent partitions in the algorithm, the probability that none is compatible is at most $2^{-n}$. Together with Lemma B.2, this bounds the probability of omitting a fixed matching vector whenever its full matching set fits the limit $u_k$.

All coordinate counts and list limits are computed as integers; the real rates are used only in the analysis. The recursive validity argument and the total time and space bounds are given in Sections 3.1.1 and 3.1.2, using the output and cost specifications proved above.

## B.4 Probability estimates for the recursive calls

The estimates below supply the collision counts and distribution comparisons used in the row-retaining induction of Section 3.1.1. Throughout this subsection, $q^{\lceil \log(n+2) \rceil} \leq 2^n$, $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ is uniform, and $\omega$ denotes exact uniform random choices independent of $\mathbf{A}$. For even moduli, assume the parity conditions (8); their strict inequalities have fixed positive margins. These conditions may be omitted when only odd moduli are used. The comparisons retain the entire matrix, since its additional rows determine the ordering used in the last-level calls.

**Lemma B.3.** *Fix integers $q \geq 2$, $n, m \geq 1$, a finite abelian group $\mathcal{G}$, and a homomorphism $\pi : (\mathbb{Z}/q\mathbb{Z})^m \to \mathcal{G}$. Let $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ have any probability distribution, and put $\phi_{\mathbf{A}}(\boldsymbol{x}) = \pi(\mathbf{A}\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{Z}^n$. Let $\mathcal{V} \subset \mathbb{Z}^n$ be a fixed nonempty finite set. Suppose that all distinct $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}$ satisfy*

$$\Pr_{\mathbf{A}}[\phi_{\mathbf{A}}(\boldsymbol{x}) = \phi_{\mathbf{A}}(\boldsymbol{y})] = 1/|\mathcal{G}|.$$

*Compare the following experiments, using the same distribution of $\mathbf{A}$ in each.*

- *Sample $\mathbf{A}$ and an independent uniform $b \in \mathcal{G}$. Choose $\boldsymbol{z}$ uniformly from $\{\boldsymbol{x} \in \mathcal{V} : \phi_{\mathbf{A}}(\boldsymbol{x}) = b\}$, or put $\boldsymbol{z} = \perp$ if this set is empty.*
- *Sample $\mathbf{A}$ and an independent uniform $\boldsymbol{z} \in \mathcal{V}$, and set $b = \phi_{\mathbf{A}}(\boldsymbol{z})$.*

*The joint laws of $(\mathbf{A}, b, \boldsymbol{z})$ have total variation distance at most $\frac{1}{2}\sqrt{(|\mathcal{G}|-1)/|\mathcal{V}|}$.*

*Proof.* For fixed $\mathbf{A}$, put $s_b = |\{\boldsymbol{x} \in \mathcal{V} : \phi_{\mathbf{A}}(\boldsymbol{x}) = b\}|$. The first experiment assigns mass $1/|\mathcal{G}|$ to each $b$, uniformly over its preimages, or to $\perp$ when there are none. The second assigns mass $s_b/|\mathcal{V}|$ to $b$, with the same conditional distribution when $s_b > 0$. Their conditional distance is therefore $\frac{1}{2}\sum_{b \in \mathcal{G}} \left| \frac{s_b}{|\mathcal{V}|} - \frac{1}{|\mathcal{G}|} \right|$. The collision hypothesis gives

$$\mathbb{E}_{\mathbf{A}}\left[ \sum_{b \in \mathcal{G}} \left( s_b - \frac{|\mathcal{V}|}{|\mathcal{G}|} \right)^2 \right] = |\mathcal{V}| \left( 1 - \frac{1}{|\mathcal{G}|} \right).$$

Cauchy–Schwarz bounds the expected distance by $\frac{1}{2}\sqrt{(|\mathcal{G}|-1)/|\mathcal{V}|}$. Conditioning on the full matrix in this calculation proves the joint comparison. □

For even moduli, two ternary vectors with the same coordinate parities can have a larger collision probability. The next result gives the expected number of candidates matching one fixed vector.

**Lemma B.4.** *Fix an even integer $q \geq 4$, integers $n, m \geq 1$ and $0 \leq r \leq m$, and a nonempty set $\mathcal{V} \subset \{-1, 0, 1\}^n$. For $\boldsymbol{v} \in \{0,1\}^n$, put $b_{\boldsymbol{v}} = |\{\boldsymbol{y} \in \mathcal{V} : \boldsymbol{y} \equiv \boldsymbol{v} \pmod 2\}|$. Sample $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ uniformly, and let $\mathbf{J}$ be its last $r$ rows. For every fixed $\boldsymbol{x} \in \mathcal{V}$,*

$$\mathbb{E}_{\mathbf{A}}\left[ \left| \{\boldsymbol{y} \in \mathcal{V} : \mathbf{J}\boldsymbol{y} = \mathbf{J}\boldsymbol{x}\} \right| \right] = 1 + \frac{|\mathcal{V}| - 1}{q^r} + \frac{(b_{\boldsymbol{x} \bmod 2} - 1)(2^r - 1)}{q^r}.$$

*Proof.* Distinct ternary vectors with different parities differ by $\pm 1$ in some coordinate. Those with the same parity differ by twice a nonzero ternary vector. Condition on all columns except one where the vectors differ by $\pm 1$ in the first case or by $\pm 2$ in the second. The difference of their images is uniform in $(\mathbb{Z}/q\mathbb{Z})^r$ in the first case and among vectors of even residues in the second. Hence, for distinct $\boldsymbol{x}, \boldsymbol{y}$,

$$\Pr_{\mathbf{A}}[\mathbf{J}\boldsymbol{x} = \mathbf{J}\boldsymbol{y}] = \begin{cases} q^{-r}, & \boldsymbol{x} \bmod 2 \neq \boldsymbol{y} \bmod 2, \\ (q/2)^{-r}, & \boldsymbol{x} \bmod 2 = \boldsymbol{y} \bmod 2. \end{cases}$$

Summing these probabilities over $\boldsymbol{y} \neq \boldsymbol{x}$, and adding the contribution of $\boldsymbol{x}$, gives the claim. □

The same collision calculation gives a comparison that retains the full matrix for even moduli.

**Lemma B.5.** *Fix an even integer $q \geq 4$, integers $n, m \geq 1$ and $0 \leq r \leq m$, and a nonempty set $\mathcal{V} \subset \{-1, 0, 1\}^n$. For $\boldsymbol{v} \in \{0,1\}^n$, put $b_{\boldsymbol{v}} = |\{\boldsymbol{x} \in \mathcal{V} : \boldsymbol{x} \equiv \boldsymbol{v} \pmod 2\}|$ and $b_{\max} = \max_{\boldsymbol{v}} b_{\boldsymbol{v}}$. Sample $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$ uniformly, and let $\mathbf{J}$ be its last $r$ rows. Compare the following two experiments.*

- *Sample an independent uniform $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^r$, then a uniform $\boldsymbol{z} \in \{\boldsymbol{x} \in \mathcal{V} : \mathbf{J}\boldsymbol{x} = \boldsymbol{t}\}$, or put $\boldsymbol{z} = \perp$ if this set is empty.*
- *Sample an independent uniform $\boldsymbol{z} \in \mathcal{V}$, then set $\boldsymbol{t} = \mathbf{J}\boldsymbol{z}$.*

*The joint laws of $(\mathbf{A}, \boldsymbol{t}, \boldsymbol{z})$ have total variation distance at most*

$$\frac{1}{2}\sqrt{\frac{q^r - 1 + (2^r - 1)(b_{\max} - 1)}{|\mathcal{V}|}}.$$

*Proof.* Put $s_{\boldsymbol{t}} = |\{\boldsymbol{x} \in \mathcal{V} : \mathbf{J}\boldsymbol{x} = \boldsymbol{t}\}|$. Summing the expected counts in Lemma B.4 over $\boldsymbol{x} \in \mathcal{V}$ gives

$$\mathbb{E}_{\mathbf{A}}\left[\sum_{\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^r} \Big(s_{\boldsymbol{t}} - \frac{|\mathcal{V}|}{q^r}\Big)^2\right] = |\mathcal{V}|(1 - q^{-r}) + \frac{2^r - 1}{q^r}\sum_{\boldsymbol{v}} b_{\boldsymbol{v}}(b_{\boldsymbol{v}} - 1).$$

For each fixed full matrix, the conditional distance is $\frac{1}{2}\sum_{\boldsymbol{t}} \left|\frac{s_{\boldsymbol{t}}}{|\mathcal{V}|} - q^{-r}\right|$. Use $\sum_{\boldsymbol{v}} b_{\boldsymbol{v}}(b_{\boldsymbol{v}} - 1) \leq |\mathcal{V}|(b_{\max} - 1)$ and Cauchy–Schwarz to obtain the stated bound. □

We apply these estimates first to the two ways a list limit can discard a required vector. For fixed $\boldsymbol{x} \in \mathcal{T}_i$, the number of vectors in $\mathcal{T}_i$ with image $\mathbf{A}_i\boldsymbol{x}$ must be at most $u_i$. For fixed $\boldsymbol{z} \in \mathcal{T}_{i+1}$, the number of vectors in $\mathcal{T}_{i+1}$ with image $\mathbf{A}_i\boldsymbol{z}$ must be at most $v_i$. If $q$ is odd, distinct ternary vectors differ by a unit modulo $q$ in some coordinate. With $r$ rows, condition on all other columns; the difference of the images remains uniform in $(\mathbb{Z}/q\mathbb{Z})^r$, so the images agree with probability $q^{-r}$. For any level $i$, the expected number of vectors in $\mathcal{T}_i$ matching one fixed vector is therefore $1 + (|\mathcal{T}_i| - 1)/q^r$. Markov's inequality bounds each probability of exceeding the corresponding limit by $1/w$.

For even $q \geq 4$, at most $\binom{2a_i}{a_i} \leq 2^{2t_i n}$ elements of $\mathcal{T}_i$ have any one parity pattern. Since $h_i = h(2t_i) + 2t_i$, Lemma B.4 bounds the additional collision term by $|\mathcal{T}_i|/q^r$ for sufficiently large $n$ whenever $r/n < h(2t_i)$ by a fixed margin. For $\mathcal{T}_0$, the required number of rows is $m$, and $m/n \leq \rho/2 + o(1)$. For $\mathcal{T}_i$ with $i \geq 1$, the required numbers are $m_i$ and $m_{i-1}$, both at most $d_{i-1}n/2 + o(n)$. The conditions in (8) therefore bound each required expectation by $2(1 + |\mathcal{T}_i|/q^r)$. Each probability of exceeding a list or buffer limit is at most $2/w$.

Next fix $\boldsymbol{x} \in \mathcal{T}_{i-1}$ and consider its possible first parts in $\mathcal{D}_i(\boldsymbol{x})$. For odd $q$, Lemma B.3 compares choosing a uniform element of $\mathcal{G}_i$ before a part with choosing a uniform part independently of the matrix. The total variation distance is at most

$$\frac{1}{2}\sqrt{\frac{|\mathcal{G}_i| - 1}{|\mathcal{D}_i(\boldsymbol{x})|}} \le \frac{1}{2}\, 2^{-\lceil \eta n \rceil/2}.$$

For even $q$, the parity of a part determines which half of the parent's support was chosen and which $2a_i'$ zero positions were added. At most $2^{2a_i'}$ assignments of signs remain. The additional term under the square root in Lemma B.5 is thus at most

$$\frac{2^{m_i} 2^{2a_i'}}{|\mathcal{D}_i(\boldsymbol{x})|} \le 2^{(d_i/2 + 2t_i - t_{i-1} - r_i)n + o(n)} = 2^{-\Omega(n)}$$

by (8). The other term is at most $2^{-\lceil \eta n \rceil}$, as before. Consequently, the joint comparison has error $2^{-\Omega(n)}$ at every step for every allowed modulus in this branch. In the second experiment, both $\boldsymbol{z}$ and $\boldsymbol{x} - \boldsymbol{z}$ are individually independent of the entire matrix. The independent randomness used by the two subcalls can be appended to either experiment.

These estimates and the partition bound following Lemma B.2 supply all the probability bounds used in the row-retaining induction of Section 3.1.1. The last-level bound holds for every ordering shift and sign, including choices depending on the full matrix. That induction therefore gives, for every fixed $\boldsymbol{x} \in \mathcal{T}$ independent of $\mathbf{A}$,

$$\Pr_{\mathbf{A},\omega}[\boldsymbol{x} \text{ is not output on input } (\mathbf{A}, \mathbf{A}\boldsymbol{x})] \le O(1/w) + 2^{-\Omega(n)}. \tag{56}$$

This bound concerns the row-retaining construction with exact uniform random choices. The next two subsections extend it to large moduli and to the finite-bit implementation, respectively.

## B.5 Extending the construction to large moduli

When the modulus is too large, retaining whole rows does not give the required intermediate numbers of equations. We reduce the canonical integer entries of the matrix to a power of three, for which we can retain part of a row. We then check the resulting candidates against the original integer equation. This extends the construction without changing its leading time and space exponents or the bound on the probability of omitting a fixed vector.

This branch applies when $q^{\lceil \log(n+2) \rceil} > 2^n$. Recall that $w = 2^{\nu \lceil n^{2/3} \rceil}$. Choose the largest power $q_0 = 3^e$ satisfying $q_0 w \le q$. The guards in the algorithm return no outputs when this choice or the required projections are unavailable. For the asymptotic input families under consideration, they are available for all sufficiently large $n$.

Read each coordinate of $(\mathbb{Z}/q_0\mathbb{Z})^m$ in base three, taking its digits from low to high before moving to the next coordinate. A prefix of this ordered sequence retains some complete coordinates and possibly part of the next coordinate. A complete coordinate has modulus $q_0$; a coordinate with only $r$ retained digits has modulus $3^r$. Thus retaining $j$ digits defines a homomorphism onto a product of cyclic groups of total order $3^j$. It also makes each shorter prefix a projection of every longer one.

The recursive combination uses only group operations, projections, and comparisons of canonical residue coordinates, so it applies to these products. In the pair generation of Lemma B.1, group the right vectors by their $\mathbf{C}$-images as before, and use the modulus of each coordinate when traversing the translated $\varepsilon\mathbf{D}$-images. The time per advance and the storage per record remain polynomial in the sum of the coordinate moduli's bit lengths. Use these coordinate moduli in place of $q$ in Algorithm 6.

The ordering, tie rules, and proof of Lemma B.2 are unchanged: the coordinate partitions still give distinct sums within each partition, and both inspection and output limits are the same. This supplies the last-level generation needed by the recursion below.

The reduction modulo $q_0$ need not preserve the original congruence, because $q_0$ need not divide $q$. Instead, read $\mathbf{A}$ and $\boldsymbol{t}$ as canonical integer representatives and enumerate equations $\mathbf{A}\boldsymbol{x} = \boldsymbol{t} + q\boldsymbol{z}$ over $\mathbb{Z}$. Each such equation can be reduced modulo $q_0$ and then checked exactly after a candidate is found.

**Algorithm 7** Enumerating short solutions for a large modulus

**Parameters:** $k, t_0, \ldots, t_k, \eta, \nu$.
**Input:** $\mathbf{A} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$ and $\boldsymbol{t} \in (\mathbb{Z}/q\mathbb{Z})^m$, where $n, m \geq 1$ and $q \geq 3$ are integers.
**Output:** Distinct vectors $\boldsymbol{x} \in \mathcal{T}$ with $\mathbf{A}\boldsymbol{x} = \boldsymbol{t}$ modulo $q$; possibly none.

Compute the coordinate counts $a_i, a'_i$, cardinalities $|\mathcal{T}_i|$, and $w$.
**if** $q < 3w$ or $\binom{a_{i-1}}{a_{i-1}/2}^2 \frac{(n-2a_{i-1})!}{a'_i!^2(n-2a_{i-1}-2a'_i)!} < 2^{\lceil \eta n \rceil}$ for some $1 \leq i \leq k$ **then**
    **return** no outputs.
**end if**
$e \leftarrow \max\{r \in \mathbb{Z}_{\geq 1} : 3^r w \leq q\}$; $q_0 \leftarrow 3^e$; $m_0 \leftarrow m \cdot e$.
Read $\mathbf{A}, \boldsymbol{t}$ as canonical integer representatives; set $\mathbf{A}_0 \leftarrow \mathbf{A} \bmod q_0$.
$\mathcal{G}_0 \leftarrow (\mathbb{Z}/q_0\mathbb{Z})^m$.
For $1 \leq i \leq k$, set $m_i$ to the largest integer $r \geq 0$ such that
$$3^r 2^{\lceil \eta n \rceil} \leq \binom{a_{i-1}}{a_{i-1}/2}^2 \frac{(n-2a_{i-1})!}{a'_i!^2(n-2a_{i-1}-2a'_i)!}.$$
**if** $m_0 > m_1 > \cdots > m_k \geq 1$ fails **then**
    **return** no outputs.
**end if**
Set $\mathbf{A}_i$ to the prefix projection of $\mathbf{A}_0$ retaining $m_i$ digits, and $\mathcal{G}_i$ to its codomain.
Set $\pi_i : \mathcal{G}_i \to \mathcal{G}_{i+1}$ to the corresponding prefix projection for $0 \leq i < k$.
Recompute all limits $u_i, v_i$ by (7), using the new $|\mathcal{G}_i|$ and including $u_0$.
Sample a tape of independent fair bits, independently of the input, with blocks assigned to the recursive calls and coordinate partitions (Appendix B.6).
**for** each $\boldsymbol{z} \in \{-n, \ldots, n\}^m$ in lexicographic order **do**
    $\boldsymbol{r} \leftarrow \boldsymbol{t} + q\boldsymbol{z}$; reset the tape to its start.
    $\mathcal{L} \leftarrow \textsc{Combine}(0, \boldsymbol{r} \bmod q_0)$.
    **for** each $\boldsymbol{x} \in \mathcal{L}$ in insertion order **do**
        **if** $\mathbf{A}\boldsymbol{x} = \boldsymbol{r}$ over $\mathbb{Z}$ **then**
            Emit $\boldsymbol{x}$.
        **end if**
    **end for**
**end for**

The loop generates one $\boldsymbol{z}$ at a time and reuses the workspace for successive calls. Within a call, random blocks are assigned by the sequence of left and right recursive calls and by the index of each coordinate partition. Resetting the tape restores this same assignment for every lift; early stopping within one call does not change another call's assigned block. The ordering and tie rules of the recursive and last-level procedures are used unchanged. An optional acceptance test is applied only after a vector has passed the integer check; a rejection continues the scan without changing any list

limit.

**Validity of the outputs.** Every candidate lies in $\mathcal{T}$. The integer check therefore ensures that every output satisfies the original congruence and has squared length at most $2t_0n$. Conversely, fix a vector $\boldsymbol{x} \in \mathcal{T}$ satisfying that congruence. Since each matrix entry lies in $\{0, \dots, q-1\}$ and each entry of $\boldsymbol{x}$ lies in $\{-1, 0, 1\}$, every coordinate of $\mathbf{A}\boldsymbol{x}$ lies between $-n(q-1)$ and $n(q-1)$, and each entry of $\boldsymbol{t}$ lies in $\{0, \dots, q-1\}$. Consequently the unique integer vector

$$\boldsymbol{z} = \frac{\mathbf{A}\boldsymbol{x} - \boldsymbol{t}}{q}$$

lies in $\{-n, \dots, n\}^m$ and is included in the loop. In that iteration the reduced target is exactly $\mathbf{A}_0\boldsymbol{x}$, and the integer check retains $\boldsymbol{x}$ if it is found. The uniqueness of this lift also prevents duplicate outputs across different iterations.

**Finding a fixed vector.** The large-modulus condition and $m \log q = \rho n + o(n)$ imply $m = O(\log n)$. Since $n/\lceil \log(n+2) \rceil > \log w + \log 3$ for sufficiently large $n$, this branch has $q > 3w$. The largest permitted power of three then satisfies

$$\log w \le \log(q/q_0) < \log w + \log 3, \qquad m \log(q/q_0) = O(\nu n^{2/3} \log n) = o(n).$$

It follows that $m \log q_0 = \rho n + o(n)$. Moreover, $\log |\mathcal{G}_i| = d_i n + o(n)$: rounding to a power of three loses less than $\log 3$ bits. The strict inequalities on the $d_i$ hence give $m \cdot e > m_1 > \cdots > m_k \ge 1$ for sufficiently large $n$, so the guards eventually pass and the same list-size and cost estimates apply.

For a uniform entry of $\mathbb{Z}/q\mathbb{Z}$, reducing its canonical representative modulo $q_0$ gives a distribution within total variation $q_0/q \le 1/w$ of uniform. The reduced matrix is thus within total variation $mn/w$ of a uniform matrix over $\mathbb{Z}/q_0\mathbb{Z}$. Under the latter law, a nonzero difference of two ternary vectors has a coefficient that is a unit modulo three. Its image under each prefix projection is therefore uniform in the corresponding product of cyclic groups. The collision and distribution comparisons in Appendix B.4 apply with that group's order. With the ordered generation just described, the row-retaining induction in Section 3.1.1 therefore gives the bound (56) for these projections under exact uniform random choices.

For each fixed $\boldsymbol{x} \in \mathcal{T}$, run Algorithm 7 on $(\mathbf{A}, \mathbf{A}\boldsymbol{x} \bmod q)$. Its correct lift may depend on $\mathbf{A}$, but the call at that lift has input $(\mathbf{A}_0, \mathbf{A}_0\boldsymbol{x})$ and uses the independently sampled tape. We apply this bound to the call without conditioning on any earlier iteration. Thus, even with the reused tape, the probability of omitting $\boldsymbol{x}$ is at most

$$O(1/w) + mn/w + 2^{-\Omega(n)} \le 2^{-(\nu - o(1))n^{2/3}}.$$

This is the required bound for the large-modulus branch before the finite-bit implementation of the random choices.

**Costs.** There are $(2n+1)^m = 2^{O((\log n)^2)}$ lifts. Processing them in sequence multiplies the time by $2^{o(n)}$ and reuses the same exponential storage. Thus every run retains time $2^{T_{\mathrm{in}}n + o(n)}$ and space $2^{S_{\mathrm{in}}n + o(n)}$, including runs on which the fixed vector is omitted.

## B.6 Implementing the random choices

The preceding analysis uses exact uniform choices inside the algorithm. We implement them with a polynomial number of independent fair bits and complete the proof of Lemma 3.2. Random input matrices and targets used only in the analysis keep their exact uniform distributions.

To choose among $N \geq 1$ possibilities, put $L = \lceil \log N \rceil + n^2$, read $L$ independent fair bits as an integer $V \in \{0, \ldots, 2^L - 1\}$, and use the index $\lfloor NV/2^L \rfloor$. Each index has either $\lfloor 2^L/N \rfloor$ or $\lceil 2^L/N \rceil$ preimages, so the distribution is within total variation $2^{-n^2}$ of uniform. For a product of cyclic groups, the index determines its canonical coordinates by successive integer division. For coordinate partitions, index the subsets of the required size in lexicographic order; binomial counts allow the indexed subset to be found in polynomial time.

The fixed-depth recursion makes at most $2^k - 1$ choices of elements $\boldsymbol{b}$ and uses $2^k$ last-level generations. Each such generation chooses $n(n+1)^3$ coordinate partitions. There are therefore polynomially many choices, each with an index of polynomial bit length. Assigning an independent fixed block of bits to every possible call and partition gives a polynomial-length tape. The same assigned tape is reused for all integer lifts, so the number of lifts does not increase the required number of random bits.

The total variation between this tape's choices and independent exactly uniform choices is at most a polynomial multiple of $2^{-n^2}$, hence $2^{-\Omega(n^2)}$. This bounds the change in the probability of every event determined by the complete execution, even when some assigned blocks are not read. Sampling and decoding the tape take polynomial time and space for one recursive scan. Repeating the decoding over the integer lifts adds only the factor already included in the large-modulus time bound. Thus the output and time and space bounds proved above remain valid for every execution.

For every fixed $\nu$ and all sufficiently large $n$, adding this error to the omission bounds for the two modulus branches gives, for every fixed $\boldsymbol{x} \in \mathcal{T}$ independent of the uniform matrix,

$$\begin{aligned} &\Pr_{\mathbf{A},\omega}[\boldsymbol{x} \text{ is not output on input } (\mathbf{A}, \mathbf{A}\boldsymbol{x})] \\ &\qquad \leq \frac{\mathrm{poly}(n)}{w} + 2^{-\Omega(n)} + 2^{-\Omega(n^2)} = 2^{-(\nu - o(1))n^{2/3}}. \end{aligned}$$

Here $\omega$ is the finite tape of independent fair bits, independent of $\mathbf{A}$. This proves Lemma 3.2 for the implemented procedure.

## B.7 Parameters chosen from the input dimensions

The SVP and density-dependent $\gamma$-CVP algorithms choose the coordinate proportions from their input dimensions. We show that the probability bound in Lemma 3.2 and the time and space bounds remain uniform under such choices. With a uniform positive entropy surplus, the average-success guarantee in Corollary 3.3 for uniform right-hand sides also holds uniformly.

Keep $k, \nu, \eta$ fixed, set $\rho = m \log q/n$, and let the rational $t_i$ be deterministic functions of $n, m, q$, computable in polynomial time. Assume that $(\rho, t_0, \ldots, t_k)$ belongs to a fixed compact set on which $\rho > 0$, $0 < t_i < 1/2$,

$$\frac{t_{i-1}}{2} < t_i < \frac{1 - t_{i-1}}{2} \qquad (1 \leq i \leq k),$$

and the strict equation conditions (5) hold. If even moduli are allowed, assume the parity conditions (8) on the same set. Compactness and continuity give one positive lower bound for every strict margin. In particular, the proportions of each sign, of zero entries, of added signs, and of zero entries left after a decomposition stay bounded away from zero. These assumptions impose no surplus $h_0 > \rho$.

The coordinate rounding errors are bounded by a constant depending only on $k$. The factorial formulas for $|\mathcal{T}_i|$ and $|\mathcal{D}_i(\boldsymbol{x})|$ therefore give

$$\log|\mathcal{T}_i| = h_i n + O(\log n) \quad (0 \le i \le k),$$
$$\log|\mathcal{D}_i(\boldsymbol{x})| = r_i n + O(\log n) \quad (1 \le i \le k,\ \boldsymbol{x} \in \mathcal{T}_{i-1})$$

with constants independent of the selected tuple. In the branch $q^{\lceil \log(n+2) \rceil} \le 2^n$, retaining rows loses at most $\log q \le n/\lceil \log(n+2) \rceil = o(n)$ bits. In the other branch, $\rho$ is bounded above, so $m = O(\log n)$ uniformly. Replacing the modulus loses $O(\nu n^{2/3} \log n) = o(n)$ bits, and retaining a base-three prefix loses less than $\log 3$ further bits. Thus $\log|\mathcal{G}_i| = d_i n + o(n)$ uniformly in either branch. The fixed positive margins between successive $d_i$ ensure that all guards pass above one threshold for $n$, independent of the tuple.

The positive parts and maxima in the cost formulas preserve these uniform errors. Hence every execution has time $2^{T_{\text{in}} n + o(n)}$ and space $2^{S_{\text{in}} n + o(n)}$, with $T_{\text{in}}, S_{\text{in}}$ evaluated at the current tuple and with uniform $o(n)$ terms. Computing the rational proportions and their integer coordinate counts adds only a polynomial factor. The large-modulus branch has $2^{O((\log n)^2)}$ lifts with a uniform constant, so it preserves these bounds as well.

For the omission probability, the decomposition comparison has the same fixed margin $\eta$. The finitely many additional parity gaps have a common positive lower bound on the compact set. After the uniform rounding errors above, every comparison error is therefore at most $2^{-cn}$ for a single constant $c > 0$. The list-limit estimates and the last-level partition probability are unchanged. With $w = 2^{\nu \lceil n^{2/3} \rceil}$ and fixed recursion depth, the omission recurrence, modulus reduction, and the bounded-bit implementation give, uniformly over every fixed $\boldsymbol{x} \in \mathcal{T}$ independent of the uniform matrix,

$$\Pr_{\mathbf{A},\omega}[\boldsymbol{x} \text{ is omitted on input } (\mathbf{A}, \mathbf{A}\boldsymbol{x})] \le \frac{\text{poly}(n)}{w} + 2^{-cn} + 2^{-\Omega(n^2)} = 2^{-(\nu - o(1))n^{2/3}}.$$

Choosing the proportions from dimensions alone leaves the matrix distribution unchanged and keeps the fixed vector independent of the matrix. Thus the same conditioning argument applies to every such choice, with a uniform error term.

For the average-success guarantee, assume additionally that $h_0 > \rho$ throughout the same compact set. There is then a constant $\sigma > 0$ such that $h_0 - \rho \ge \sigma$ everywhere. Repeat the joint comparison in the proof of Corollary 3.3. Its squared error is bounded, up to a constant factor, by

$$2^{(\rho - h_0)n + o(n)} + 2^{(\rho/2 + 2t_0 - h_0)n + o(n)},$$

where the second term is needed only for even $q \ge 4$. The first exponent is uniformly negative by the surplus, and the second is uniformly negative by $h_0 - 2t_0 = h(2t_0) > \rho/2$ and compactness. Thus this comparison has error $2^{-\Omega(n)}$ uniformly. Averaging the uniform omission bound over $\mathcal{T}$ gives joint failure $2^{-\Omega(n^{2/3})}$. For every fixed $p \in (0,1)$, Markov's inequality bounds the fraction of matrices whose conditional failure exceeds $1 - p$ by that joint failure divided by $1 - p$. Thus Corollary 3.3 holds uniformly under the additional entropy-surplus assumption.

# C Implementing the Exact Search

This appendix implements the subroutines of $\mathcal{E}_\tau$ (Algorithm 2) and proves their correctness and cost bounds. It also establishes the coordinate-balancing lemma and gives the nonzero variant $\mathcal{E}_\tau^{\neq 0}$ used for SVP. These supply the details used in the proof of Lemma 3.5 in Section 3.2.

Section 3.2.2 uses the shifted-theta condition (10). Lemma A.3 verifies this condition for $\tau = 0$, and Lemma A.4 gives a sufficient condition for $\tau > 0$, which Appendix F checks for the 5/3-CVP parameters.

## C.1 Subroutines

We describe the subroutines used by Algorithm 2 and establish their correctness and time and space bounds. We first divide coordinate lists and enumerate candidate assignments, then generate ordered pairs and combine four lists.

### C.1.1 Cyclic coordinate splits

The procedure $\mathcal{C}$ in Algorithm 8 takes a list $\mathcal{K}$ of coordinate indices and outputs, for each cyclic rotation of $\mathcal{K}$, the pair of its two halves; Algorithm 2 calls it three times to split $[n] \setminus \mathcal{J}$ into the four parts $\mathcal{I}_1, \ldots, \mathcal{I}_4$. Trying every rotation guarantees that some choice of splits balances both the sizes and the costs of the four parts (Lemma C.4). If $|\mathcal{K}|$ is odd, the list is first padded with the entry $0 \notin [n]$ so that it has even length; the padding entry is removed from the halves before they are output. The procedure suspends after each output pair until the next one is requested, so the pairs are never stored simultaneously.

**Algorithm 8** Cyclic coordinate splits: $\mathcal{C}$

**Input:** A list $\mathcal{K}$ over $[n]$ with distinct entries $k_i = \mathcal{K}(i)$.

**Output:** With $\mathcal{K}' = (k'_1, \ldots, k'_{|\mathcal{K}'|})$ denoting $\mathcal{K}$ if $|\mathcal{K}|$ is even, and $\mathcal{K}$ followed by the padding entry 0 if $|\mathcal{K}|$ is odd: for each $t \in \{0, \ldots, |\mathcal{K}'|-1\}$, a copy of $\big((k'_{t+1}, \ldots, k'_{t+|\mathcal{K}'|/2}), (k'_{t+|\mathcal{K}'|/2+1}, \ldots, k'_{t+|\mathcal{K}'|})\big)$ with the padding entry removed, indices taken modulo $|\mathcal{K}'|$ in $[|\mathcal{K}'|]$, exactly once; for $|\mathcal{K}| = 0$, the pair $([\,], [\,])$ exactly once.

```
if |K| = 0 then
    Output ([], []); return
end if
K' ← a copy of K
if |K| is odd then
    Append 0 to K'
end if
for t = 0, ..., |K'| − 1 do
    for j = 1, ..., |K'| do
        a_j ← the entry of K' at position ((t + j − 1) mod |K'|) + 1
    end for
    Delete the entry 0, if present, from (a_1, ..., a_{|K'|/2}) and from (a_{|K'|/2+1}, ..., a_{|K'|})
    Output copies of the resulting pair of lists
end for
```

Each output list again has distinct entries in $[n]$, so it can be divided again.

**Lemma C.1.** *Let $\mathcal{K}$ be a list over $[n]$ with distinct entries, each encoded using at most $\lambda \geq 1$ bits. For $|\mathcal{K}| > 0$, Algorithm 8 outputs, for each of the $|\mathcal{K}'| = 2\lceil |\mathcal{K}|/2 \rceil$ cyclic rotations of the padded list $\mathcal{K}'$, its two halves with the padding entry removed, exactly once; the two output lists have lengths $\lceil |\mathcal{K}|/2 \rceil$ and $\lfloor |\mathcal{K}|/2 \rfloor$ in some order, and together contain every entry of $\mathcal{K}$ exactly once. For $|\mathcal{K}| = 0$, it outputs the pair of empty lists exactly once. When its outputs are consumed successively, its total*

*time and working space are, respectively,*

$$O((|\mathcal{K}|+1)^2(\lambda+\log(|\mathcal{K}|+2))) \quad \textit{and} \quad O((|\mathcal{K}|+1)(\lambda+\log(|\mathcal{K}|+2))).$$

*Proof.* The empty-input branch gives the stated output immediately. Otherwise, the padded list $\mathcal{K}'$ has even length $|\mathcal{K}'| = 2\lceil|\mathcal{K}|/2\rceil$ and distinct entries, since $0 \notin [n]$. For each $t \in \{0,\dots,|\mathcal{K}'|-1\}$, the position $((t+j-1) \bmod |\mathcal{K}'|)+1$ lists the entries of $\mathcal{K}'$ in the cyclic order starting at position $t+1$, so the two lists formed in the loop are the halves of that rotation. Deleting the entry 0 removes nothing when $|\mathcal{K}|$ is even and exactly one entry from one of the two halves when $|\mathcal{K}|$ is odd, which gives the stated lengths, and every entry of $\mathcal{K}$ remains in exactly one of the two lists. The loop visits every starting position exactly once.

Each entry and index requires $O(\lambda+\log(|\mathcal{K}|+2))$ bits. Since $|\mathcal{K}'| = O(|\mathcal{K}|+1)$, generating each rotation and its two output lists takes $O((|\mathcal{K}|+1)(\lambda+\log(|\mathcal{K}|+2)))$ bit operations. There are $O(|\mathcal{K}|+1)$ rotations, giving the stated time bound. Only the padded list, the current rotation and output pair, and the loop indices are stored, giving the stated space bound. □

### C.1.2 Listing the integer points in a ball

The procedure $\mathcal{L}$ in Algorithm 9 lists all integer points in a ball specified by a target and a squared radius. The exact search uses it to build the candidate lists on its four coordinate parts.

**Algorithm 9** Listing the integer points in a ball: $\mathcal{L}$

**Input:** $\boldsymbol{y} \in \mathbb{Q}^d$, $b \in \mathbb{Q}_{\geq 0}$.
**Output:** Every $\boldsymbol{x} \in \mathbb{Z}^d$ with $\|\boldsymbol{x}-\boldsymbol{y}\|^2 \leq b$, exactly once.

$\boldsymbol{f} \leftarrow \lfloor \boldsymbol{y} \rceil - \boldsymbol{y}$; $\boldsymbol{x} \leftarrow \boldsymbol{0}_d$
$e_i \leftarrow \sum_{j=i}^{d} f_j^2$ for $i = 1,\dots,d+1$
**procedure** ENUMERATE($i, c$)
    **if** $c + e_i > b$ **then**
        **return**
    **end if**
    **if** $i = d+1$ **then**
        Output a copy of $\boldsymbol{x}$; **return**
    **end if**
    **for** each $z \in \mathbb{Z}$ with $(z-y_i)^2 \leq b - c - e_{i+1}$ **do**
        $x_i \leftarrow z$
        ENUMERATE($i+1, c+(z-y_i)^2$)
    **end for**
**end procedure**
ENUMERATE($1, 0$)

The time bound below uses the functions $\Theta_u(a)$ and $\Theta(a)$ defined in (48).

**Lemma C.2.** *For every integer $n \geq 1$ and every integer $0 \leq d \leq n$, a deterministic algorithm takes a rational target $\boldsymbol{y} \in \mathbb{Q}^d$ and a rational squared radius $b \geq 0$ with $b = O(n)$. It outputs every $\boldsymbol{x} \in \mathbb{Z}^d$ satisfying $\|\boldsymbol{x}-\boldsymbol{y}\|^2 \leq b$ exactly once, and outputs nothing if there is no such point. For every real $a > 0$, every execution uses polynomial space and time*

$$\Big(1 + e^{ab} \prod_{i=1}^{d} \Theta_{y_i}(a)\Big) \operatorname{poly}.$$

*The polynomial factor is in $n$ and the binary encoding length of $(\boldsymbol{y}, b)$. The algorithm does not use $a$.*

*Proof.* Algorithm 9 assigns each coordinate once along any branch. A discarded prefix has no completion in the ball, since rounding gives the least possible error on every unassigned coordinate. Conversely, every integer point in the ball follows a unique branch that is never discarded. This proves the output claim. Only the current partial vector and the recursion stack are stored, so the space is polynomial.

For the time bound, the algorithm is a depth-first search that assigns the coordinates in order and discards a prefix as soon as its squared cost plus $\sum f_i^2$ over the unassigned coordinates exceeds $b$; in particular the root is discarded if $\sum_i f_i^2 > b$. A surviving prefix of length $j$ and cost $c$ satisfies $b - c - \sum_{i>j} f_i^2 \geq 0$, so the number of such prefixes is at most

$$e^{a(b-\sum_{i>j} f_i^2)} \prod_{i \leq j} \Theta_{y_i}(a) \leq e^{ab} \prod_{i=1}^{d} \Theta_{y_i}(a),$$

where the inequality uses $\Theta_{y_i}(a) \geq e^{-af_i^2}$ for the unassigned coordinates. Summing over the $d+1$ depths, and noting that each node tries $O(\sqrt{b}+1)$ values for its next coordinate with exact rational comparisons, proves the bound. □

### C.1.3 Generating pair sums in order

The procedure $\mathcal{Q}$ in Algorithm 10 generates all pairs from two lists in nondecreasing order of the element $\boldsymbol{g}_W \in \mathcal{G}$ defined below. These ordered outputs are used by the four-list procedure $\mathcal{M}$. We define the records and the ordering before giving the procedure and its correctness and complexity analysis.

In the exact search, an assignment $\boldsymbol{v} \in \mathbb{Z}^{\mathcal{I}}$ on a coordinate set $\mathcal{I} \subseteq [n]$ is stored as a record

$$X = (\boldsymbol{g}_X, c_X, \boldsymbol{v}_X) = (\mathbf{H}_{\mathcal{I}}\boldsymbol{v}, \|\boldsymbol{v} - \boldsymbol{y}_{\mathcal{I}}\|^2, \boldsymbol{v}).$$

The field $c_X$ is called the cost. An empty coordinate part has the single record $(\mathbf{0}_m, 0, \mathbf{0}_0)$.

We use the ordered pair generation of Schroeppel and Shamir [SS81], as described by Shallue [Sha12, Section 4]. The elements $\boldsymbol{g}_X$ may lie in any product $\mathcal{G} = \prod_{i=1}^{m} \mathbb{Z}/q_i\mathbb{Z}$, where $m \geq 0$ and each $q_i \geq 2$ is an integer; for $m = 0$, this is the one-element group. This includes the case $q_i = q$ needed for the exact search.

Before matching four records, we specify how to generate pair sums without storing all pairs. Let $\mathcal{X}, \mathcal{Y}$ be two lists of records. For $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, a pair record has the four fields

$$W = (\boldsymbol{g}_W, c_W, X_W, Y_W).$$

Its first two fields are an element of $\mathcal{G}$ and the total cost; its last two fields retain the two original records. For a shift $\boldsymbol{t} \in \mathcal{G}$ and a sign $\varepsilon \in \{-1, 1\}$, we set

$$\boldsymbol{g}_W = \boldsymbol{t} + \varepsilon(\boldsymbol{g}_X + \boldsymbol{g}_Y), \qquad c_W = c_X + c_Y, \qquad (X_W, Y_W) = (X, Y).$$

The generator $\mathcal{Q}(\mathcal{X}, \mathcal{Y}; \boldsymbol{t}, \varepsilon)$ outputs all pair records in nondecreasing order of $\boldsymbol{g}_W$. $\mathcal{Q}$ requires distinct elements $\boldsymbol{g}_X$ within each input list. All comparisons of elements of $\mathcal{G}$ use lexicographic order of their canonical residue coordinates. Calling next consumes one output and returns $\perp$ after no outputs remain. Each output suspends the generator until its next call, so the list of all pairs is never stored.

The ordered traversal extends the binary-coordinate construction of Dietzfelbinger, Schlag, and Walzer [DSW18, Algorithm 3] by sorting the occurring coordinate values. We implement it as follows.

For each $Y \in \mathcal{Y}$, store the vector $\varepsilon \boldsymbol{g}_Y$ in a rooted prefix tree: a path records successive residue coordinates, common prefixes share nodes, and each node stores its occurring child digits in sorted order. A leaf points to the original record $Y$. For a fixed shift $\boldsymbol{t} \in \mathcal{G}$ (the iterator of a left record $X$ uses the shift $\boldsymbol{t} + \varepsilon \boldsymbol{g}_X$), visit the children at coordinate $i$, with stored digit $z$, in increasing order of $(z + t_i) \bmod q_i$. This is the cyclic rotation starting at the first stored digit $z \geq q_i - t_i$; if no such digit occurs, start at the first stored digit. An iterator stores the current path and the next child position at each node. Its next operation descends to the first unvisited leaf in this order, backtracking past exhausted nodes, and returns the leaf's record. It returns $\perp$ when the root is exhausted. For $m = 0$, the root is the single leaf. A min-heap below stores at most one pair per left record, ordered by $\boldsymbol{g}_W$; popping it removes a record with the smallest such vector, breaking ties by the input order of the left records in $\mathcal{X}$.

**Algorithm 10** Generating pair sums in order: $\mathcal{Q}$

**Input:** Lists $\mathcal{X}, \mathcal{Y}$, each with distinct elements of $\mathcal{G}$, $\boldsymbol{t} \in \mathcal{G}$, $\varepsilon \in \{-1, 1\}$.
**Output:** All pair records $W$ defined above, in nondecreasing order of $\boldsymbol{g}_W$.

**if** $|\mathcal{X}| = 0$ or $|\mathcal{Y}| = 0$ **then**
    **return**
**end if**
Build the shared prefix tree of $\{\varepsilon \boldsymbol{g}_Y : Y \in \mathcal{Y}\}$; $\mathcal{U} \leftarrow$ empty min-heap
**for** each $X \in \mathcal{X}$ **do**
    Create iterator $\mathcal{Y}_X$ at the root, with child order shifted by $\boldsymbol{t} + \varepsilon \boldsymbol{g}_X$
    $Y \leftarrow \text{next}(\mathcal{Y}_X)$
    Insert $(\boldsymbol{t} + \varepsilon(\boldsymbol{g}_X + \boldsymbol{g}_Y), c_X + c_Y, X, Y)$ into $\mathcal{U}$
**end for**
**while** $\mathcal{U} \neq \varnothing$ **do**
    Pop a record $W$ of minimum $\boldsymbol{g}_W$ from $\mathcal{U}$; output $W$
    $X \leftarrow X_W$; $Y \leftarrow \text{next}(\mathcal{Y}_X)$
    **if** $Y \neq \perp$ **then**
        Insert $(\boldsymbol{t} + \varepsilon(\boldsymbol{g}_X + \boldsymbol{g}_Y), c_X + c_Y, X, Y)$ into $\mathcal{U}$
    **end if**
**end while**

**Correctness and complexity.** For a fixed left record, the iterator visits every right record in increasing order of the element $\boldsymbol{g}_W$: all leaves under a child share the same prefix, and the cyclic child order makes the next digits increase. The heap in Algorithm 10 merges these ordered outputs. Thus every pair is emitted exactly once, in nondecreasing order of $\boldsymbol{g}_W$. Empty input lists give no output; for $m = 0$, the same argument applies to the single-leaf tree.

Suppose each list has at most $\ell \geq 1$ records and each record uses at most $\lambda \geq 1$ bits. A prefix tree for $\mathcal{Y}$ has at most $1 + m|\mathcal{Y}|$ nodes, and its child arrays contain only occurring digits. An iterator needs at most $m + 1$ path entries, and binary search locates each cyclic rotation start. It neither copies the tree nor retains prior outputs. The heap has at most $|\mathcal{X}|$ entries. Over its complete traversal, each iterator makes $O(1 + m|\mathcal{Y}|)$ tree-edge traversals. Consequently, generating all $|\mathcal{X}||\mathcal{Y}|$ pairs takes time

$$\ell^2 \operatorname{poly}\left(m, \log \ell, \sum_i \log q_i, \lambda\right)$$

and space

$$\ell \ \mathrm{poly}\left(m, \log \ell, \sum_i \log q_i, \lambda\right).$$

These polynomial factors include tree construction, heap operations, and exact modular arithmetic and rational additions.

### C.1.4 Combining four lists at minimum total cost

The procedure $\mathcal{M}$ in Algorithm 11 selects one record from each of four lists to satisfy the required congruence at minimum total cost. We implement it using the ordered outputs of $\mathcal{Q}$ and prove its correctness and complexity bounds.

The generator $\mathcal{P}$ reads consecutive outputs of $\mathcal{Q}$ with the same $\boldsymbol{g}_W$ and retains a pair of minimum cost, keeping the first such pair when costs tie. On encountering a new element, it emits the retained pair and keeps the first pair of the new group for the next call. At the end it emits the last retained pair. Empty input lists yield no outputs. This needs only a constant number of additional records. Algorithm 11 applies this operation to the pairs from lists one and two, and to the required sum minus the pairs from lists three and four. Equal output values give precisely the compatible choices.

**Algorithm 11** Combining four lists at minimum squared distance: $\mathcal{M}$

**Input:** Lists $\mathcal{X}_1, \ldots, \mathcal{X}_4$ of records defined above; required sum $\boldsymbol{s}' \in \mathcal{G}$.
**Output:** Four records $X_1, \ldots, X_4$ of minimum total cost with $\sum_j \boldsymbol{g}_{X_j} = \boldsymbol{s}'$, or $\perp$.

**for** $j = 1, \ldots, 4$ **do**
    Sort $\mathcal{X}_j$ by $\boldsymbol{g}_X$; retain a minimum-cost record for each value, breaking ties by input order
**end for**
$\mathcal{P}_1 \leftarrow \mathcal{P}(\mathcal{X}_1, \mathcal{X}_2; \mathbf{0}_m, 1)$; $\mathcal{P}_2 \leftarrow \mathcal{P}(\mathcal{X}_3, \mathcal{X}_4; \boldsymbol{s}', -1)$
$W_1 \leftarrow \mathrm{next}(\mathcal{P}_1)$; $W_2 \leftarrow \mathrm{next}(\mathcal{P}_2)$; $\mathcal{R} \leftarrow \perp$; $b_{\mathrm{best}} \leftarrow +\infty$
**while** $W_1 \neq \perp$ and $W_2 \neq \perp$ **do**
    **if** $\boldsymbol{g}_{W_1} < \boldsymbol{g}_{W_2}$ **then**
        $W_1 \leftarrow \mathrm{next}(\mathcal{P}_1)$
    **else if** $\boldsymbol{g}_{W_2} < \boldsymbol{g}_{W_1}$ **then**
        $W_2 \leftarrow \mathrm{next}(\mathcal{P}_2)$
    **else**
        **if** $c_{W_1} + c_{W_2} < b_{\mathrm{best}}$ **then**
            $\mathcal{R} \leftarrow (X_{W_1}, Y_{W_1}, X_{W_2}, Y_{W_2})$; $b_{\mathrm{best}} \leftarrow c_{W_1} + c_{W_2}$
        **end if**
        $W_1 \leftarrow \mathrm{next}(\mathcal{P}_1)$; $W_2 \leftarrow \mathrm{next}(\mathcal{P}_2)$
    **end if**
**end while**
**return** $\mathcal{R}$

**Lemma C.3.** *Let $m \geq 0$, $\ell \geq 1$ and $\lambda \geq 1$ be integers, and let $q_1, \ldots, q_m \geq 2$ be integers given in binary. Write $\mathcal{G} = \prod_{i=1}^m \mathbb{Z}/q_i\mathbb{Z}$; for $m = 0$, this is the one-element group. A deterministic algorithm takes four lists $\mathcal{X}_1, \ldots, \mathcal{X}_4$, each of at most $\ell$ records, and a required sum $\boldsymbol{s}' \in \mathcal{G}$. Each record stores an element of $\mathcal{G}$, a rational cost, and the corresponding partial solution, using at most $\lambda$ bits in total.*

*The algorithm returns records $X_1, \ldots, X_4$, one from each list, with $\sum_j \boldsymbol{g}_{X_j} = \boldsymbol{s}'$ and total cost minimum among all such choices. It returns $\perp$ exactly when no such choice exists. Every execution*

*uses time*

$$\ell^2 \operatorname{poly}\left(m, \log \ell, \sum_i \log q_i, \lambda\right)$$

*and space*

$$\ell \operatorname{poly}\left(m, \log \ell, \sum_i \log q_i, \lambda\right).$$

*Proof.* Discarding all but the cheapest record of each element of $\mathcal{G}$ preserves the optimum, since costs add and the four choices come from separate lists; cost ties are resolved by input order as in Algorithm 11. By Section C.1.3, $\mathcal{Q}$ produces every pair in nondecreasing order of $\boldsymbol{g}_W$. The generator $\mathcal{P}$ groups consecutive equal elements and retains their minimum-cost pair. Thus Algorithm 11 encounters every common element of $\mathcal{G}$ with a cheapest pair from each side. These pairs give the cheapest four records at that element; minimizing over all common elements proves correctness. Empty lists produce $\perp$. When $m = 0$, deduplication leaves the cheapest record of each list, and the same code returns those four records.

The two calls to $\mathcal{Q}$ satisfy the time and space bounds proved in Section C.1.3. Pairs with the same $\boldsymbol{g}_W$ are consumed successively; $\mathcal{P}$ and the four-list merge retain only a constant number of additional records. Sorting and deduplication take $\ell$ poly time and space, and all cost comparisons are exact. The claimed $\ell^2$ poly time and $\ell$ poly space bounds follow. □

## C.2 Balancing the coordinate parts

The three cyclic splits used in Algorithm 2 include a partition that balances both the sizes and the costs of the four coordinate parts.

**Lemma C.4.** *Let $d \geq 0$ be an integer and let $\Delta \geq 0$. Let $\mathcal{K} = (k_1, \ldots, k_d)$ be a list of distinct coordinate indices. Assign cost $u_i \in [0, \Delta]$ to coordinate $k_i$ for $1 \leq i \leq d$. Choose one output of Algorithm 8 on $\mathcal{K}$, and then one output on each of its two halves. Among the $O((d+1)^3)$ choices of these three rotations, at least one gives four parts, each of size at most $\lceil d/4 \rceil$ and cost at most*

$$\frac{1}{4}\sum_{i=1}^{d} u_i + \frac{3}{2}\Delta.$$

*Proof.* The claim is immediate for the empty list. For an even list, two opposite windows of half its length have complementary costs. Consecutive windows differ by removing one cost and adding another, so their costs differ by at most $\Delta$. Since opposite windows lie on opposite sides of half the total cost, some window has cost within $\Delta$ of that value. Its complement satisfies the same bound. For an odd list, Algorithm 8 first appends a padding entry; assigning it cost zero preserves these facts, and removing it from the output halves changes no cost.

Choose such a division, and apply the same argument inside each half. Each of the resulting four parts has cost at most one quarter of the original total plus $\Delta/2 + \Delta$. The first division gives halves of length at most $\lceil d/2 \rceil$. After the second division, each part has at most $\lceil \lceil d/2 \rceil / 2 \rceil = \lceil d/4 \rceil$ coordinates. Each of the three rotations has $O(d+1)$ choices, proving the count. □

Section 3.2.1 applies this lemma to the adjusted squared errors of a closest feasible vector and uses Lemmas C.2 and C.3 to complete the correctness proof.

### C.3 Excluding the zero vector

Algorithm 12 implements the nonzero variant used for SVP. It fixes one nonzero coordinate before running the search: removing zero only after minimum-cost combination may lose the closest nonzero solution.

Fix a rational constant $\tau \in [0, 1]$. For the input matrix $\mathbf{H}$, write $\mathbf{H}_{\mathcal{J}}$ for the columns indexed by $\mathcal{J}$ in increasing order. The procedure is denoted $\mathcal{E}_{\tau}^{\neq 0}$.

Fixing $x_i = z \neq 0$ leaves an auxiliary problem on $\mathcal{J} = [n] \setminus \{i\}$:

$$\mathbf{H}_{\mathcal{J}}\boldsymbol{u} = -z\mathbf{H}\boldsymbol{e}_i, \qquad \|\boldsymbol{u} - \boldsymbol{y}_{\mathcal{J}}\|^2 \leq b - (z - y_i)^2.$$

To solve such problems, write $\mathcal{E}_{\tau}^{\mathrm{aux}}(\mathbf{H}, \boldsymbol{s}, \boldsymbol{y}, b)$ for Algorithm 2 with the residual assignment $\boldsymbol{s}' \leftarrow -\mathbf{H}_{\mathcal{J}}\boldsymbol{u}$ replaced by $\boldsymbol{s}' \leftarrow \boldsymbol{s} - \mathbf{H}_{\mathcal{J}}\boldsymbol{u}$. Here $\boldsymbol{s} \in (\mathbb{Z}/q\mathbb{Z})^m$ is the right-hand side of the auxiliary congruence. Only the required sum in the four-list combination changes; the lists and their construction are unchanged.

**Algorithm 12** Closest nonzero lattice vector within a ball: $\mathcal{E}_{\tau}^{\neq 0}$

**Input:** $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m \times n}$, $\boldsymbol{y} \in \mathbb{Q}^n$, $b \in \mathbb{Q}_{\geq 0}$, where $n \geq 1$, $m \geq 0$, $q \geq 2$ are integers.
**Output:** A closest $\boldsymbol{x} \in \mathbb{Z}^n \setminus \{\mathbf{0}_n\}$ to $\boldsymbol{y}$ with $\mathbf{H}\boldsymbol{x} = \mathbf{0}_m$ and $\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq b$, or $\perp$ if none exists.

**if** $b < \|\lfloor \boldsymbol{y} \rceil - \boldsymbol{y}\|^2$ **then return** $\perp$
$\boldsymbol{x} \leftarrow \perp$; $b_{\mathrm{best}} \leftarrow b$
**for** $i = 1, \ldots, n$ **do**
    $\mathcal{J} \leftarrow [n] \setminus \{i\}$
    **for** each $z \in \mathbb{Z} \setminus \{0\}$ with $(z - y_i)^2 \leq b$ **do**
        **if** $n = 1$ **then**
            **if** $z\mathbf{H}\boldsymbol{e}_i \neq \mathbf{0}_m$ **then continue**
            $\boldsymbol{v} \leftarrow (z)$
        **else**
            $\boldsymbol{u} \leftarrow \mathcal{E}_{\tau}^{\mathrm{aux}}(\mathbf{H}_{\mathcal{J}}, -z\mathbf{H}\boldsymbol{e}_i, \boldsymbol{y}_{\mathcal{J}}, b - (z - y_i)^2)$
            **if** $\boldsymbol{u} = \perp$ **then continue**
            $v_i \leftarrow z$; $\boldsymbol{v}_{\mathcal{J}} \leftarrow \boldsymbol{u}$
        **end if**
        **if** $\|\boldsymbol{v} - \boldsymbol{y}\|^2 \leq b_{\mathrm{best}}$ **then**
            $\boldsymbol{x} \leftarrow \boldsymbol{v}$; $b_{\mathrm{best}} \leftarrow \|\boldsymbol{v} - \boldsymbol{y}\|^2$
        **end if**
    **end for**
**end for**
**return** $\boldsymbol{x}$

As shown in Sections 3.2.1 and 3.2.2, this variant returns a closest nonzero solution in the ball, or $\perp$ when none exists, and has the time and space bounds of Lemma 3.5, up to polynomial factors. Setting $\boldsymbol{y} = \mathbf{0}_n$ gives the search used for SVP.

# D Auxiliary Procedures for $\gamma$-CVP

This appendix gives the implementations and proofs of the two auxiliary procedures used by Algorithm 3: the exact dynamic program of Lemma 4.1 and the binary short-solution sampler of Lemma 4.2. We use the notation of those statements.

## D.1 The exact dynamic program

*Proof of Lemma 4.1.* Use a state for each $\boldsymbol{s} \in (\mathbb{Z}/q\mathbb{Z})^m$. Initialize $F_0(\mathbf{0}_m) = 0$ and $F_0(\boldsymbol{s}) = +\infty$ for $\boldsymbol{s} \neq \mathbf{0}_m$, and compute

$$F_i(\boldsymbol{s}) = \min\{F_{i-1}(\boldsymbol{s} - x_i\mathbf{H}\boldsymbol{e}_i) + (x_i - y_i)^2 : x_i \in \mathbb{Z},\ |x_i - y_i| \le q/2\} \qquad (1 \le i \le n).$$

Induction on $i$ shows that $F_i(\boldsymbol{s})$ is the minimum of $\sum_{j=1}^{i}(x_j - y_j)^2$ over all integers $x_1, \dots, x_i$ with $\sum_{j=1}^{i} x_j\mathbf{H}\boldsymbol{e}_j = \boldsymbol{s}$: replacing a coordinate $x_j$ by the integer closest to $y_j$ in its residue class modulo $q$ preserves this sum and does not increase the cost, and that integer satisfies $|x_j - y_j| \le q/2$. The final state $F_n(\mathbf{0}_m)$ is finite, since $q\mathbb{Z}^n \subseteq \Lambda$, and equals the optimum squared distance. For every state and stage, store one minimizing $x_i$ and its predecessor. Backtracking then returns a closest vector.

There are $nq^m$ states and at most $q + 1 \le 2q$ transitions per state. Each cost is a sum of squares of input rationals and integers of bit length polynomial in the input length. Exact arithmetic and comparisons therefore have polynomial bit cost. Finally, $m \ge 1$ implies $q \le q^m = 2^{\delta n}$, which proves the last bounds. □

## D.2 The binary short-solution sampler

*Proof of Lemma 4.2.* First choose $\mathbf{A}$ and $\boldsymbol{t}$ independently and uniformly, and let $N(\mathbf{A}, \boldsymbol{t}) = |\mathcal{B} \cap \mathbf{A}^{-1}(\boldsymbol{t})|$. For each $\boldsymbol{x} \in \mathcal{B}$, the indicator of $\mathbf{A}\boldsymbol{x} = \boldsymbol{t}$ has mean $2^{-m}$. For distinct $\boldsymbol{x}, \boldsymbol{x}'$, these indicators are independent: the vectors $(\boldsymbol{x}^\top, 1)^\top$ and $(\boldsymbol{x}'^\top, 1)^\top$ are distinct and nonzero, hence linearly independent over $\mathbb{F}_2$, so each uniform row of $[\mathbf{A}\ \boldsymbol{t}]$ produces two independent bits. Consequently,

$$\mathbb{E}[N] = |\mathcal{B}|2^{-m}, \qquad \mathrm{Var}(N) = \mathbb{E}[N]\,(1 - 2^{-m}), \qquad \Pr_{\mathbf{A},\boldsymbol{t}}[N < \mathbb{E}[N]\,/2] \le \frac{4}{\mathbb{E}[N]} = \frac{4\,2^m}{|\mathcal{B}|},$$

the last bound by Chebyshev's inequality. Markov's inequality now shows that at most a $32\,2^m/|\mathcal{B}|$ fraction of matrices have more than $1/8$ of their right-hand sides violating $N \ge \mathbb{E}[N]\,/2$. Separately, a union bound over nonzero vectors in the left kernel gives

$$\Pr_{\mathbf{A}}[\operatorname{rank}\mathbf{A} < m] \le (2^m - 1)2^{-n_0} < 2^{m-n_0}.$$

Call a matrix good when it lies outside both exceptional sets.

The procedure computes the rank and returns $\perp$ unless it equals $m$. For a full-row-rank matrix, Gaussian elimination gives a particular solution $\boldsymbol{x}_{\boldsymbol{t}}$ of $\mathbf{A}\boldsymbol{x} = \boldsymbol{t}$ and a kernel basis matrix $\mathbf{B} \in \mathbb{F}_2^{n_0 \times (n_0 - m)}$. A uniform $\boldsymbol{u} \in \mathbb{F}_2^{n_0 - m}$ makes $\boldsymbol{x}_{\boldsymbol{t}} + \mathbf{B}\boldsymbol{u}$ uniform in the fiber $\mathbf{A}^{-1}(\boldsymbol{t})$. For a good matrix and at least $7/8$ of its right-hand sides, the probability that a uniform element of the fiber lies in $\mathcal{B}$ is

$$\frac{N(\mathbf{A}, \boldsymbol{t})}{2^{n_0 - m}} \ge \frac{|\mathcal{B}|}{2^{n_0+1}}.$$

Set $J = \lceil 32\,2^{n_0}/|\mathcal{B}| \rceil$ and choose a uniform $\mathbf{U} \in \mathbb{F}_2^{(n_0-m)\times(\lceil \log J \rceil + 1)}$. For $i = 1, \dots, J$, let $\boldsymbol{\iota}_i \in \mathbb{F}_2^{\lceil \log J \rceil}$ be the binary encoding of $i - 1$, and generate

$$\boldsymbol{u}_i = \mathbf{U}(\boldsymbol{\iota}_i^\top, 1)^\top, \qquad \boldsymbol{x}_i = \boldsymbol{x}_{\boldsymbol{t}} + \mathbf{B}\boldsymbol{u}_i.$$

Return the first $\boldsymbol{x}_i \in \mathcal{B}$, or $\perp$ if no such sample is found. For $i \neq j$, the vectors $(\boldsymbol{\iota}_i^\top, 1)^\top$ and $(\boldsymbol{\iota}_j^\top, 1)^\top$ are distinct and nonzero, hence linearly independent over $\mathbb{F}_2$, so each uniform row of $\mathbf{U}$ produces two independent bits and $\boldsymbol{u}_i, \boldsymbol{u}_j$ are independent and uniform. Thus every sample is uniform in the fiber and the samples are pairwise independent, so the number of short samples has mean at least

$J|\mathcal{B}|/2^{n_0+1} \geq 16$ and variance at most its mean. Chebyshev's inequality gives probability at most $1/16$ that no sample is short. For every good matrix, the average success probability is therefore at least $(7/8)(15/16) = 105/128$.

The binomial sum $|\mathcal{B}| = \sum_{j=0}^{\lfloor \beta n_0 \rfloor} \binom{n_0}{j}$ is computable in polynomial time, and the bit lengths of $|\mathcal{B}|$ and $J$ are $O(n_0)$. The seed $\mathbf{U}$ uses $(n_0 - m)(\lceil \log J \rceil + 1) = O(n_0^2)$ independent random bits. Each sample can be generated and checked in polynomial time and discarded before the next is produced. Since $J = O(2^{n_0}/|\mathcal{B}|)$, this proves the claimed time and space bounds, including their bit costs. □

# E Auxiliary Lemmas and Procedures for SVP

This appendix proves the counting lemmas stated in Section 5.2 and implements the numerical steps of Algorithm 4. First we compute the radius in (33) by counting integer points (Lemma 5.1). A comparison with a theta sum (Lemma 5.2) bounds both the exact-search cost and the number of candidates in the correctness proof. We then show that the chosen ball contains a nonzero lattice vector for most input matrices (Lemma 5.3).

For the randomized search, we first exclude short vectors whose coordinates have a common divisor greater than one with the modulus. This permits a counting argument for the sign and large-coordinate restrictions in (36) and (35) (Lemma 5.5). The final subsection implements the numerical tests and enumerates the permitted coordinates. The arguments allow composite moduli.

## E.1 Computing the radius

We compute the search radius without using a volume approximation. Throughout, $\mathcal{B}_d(u)$ denotes the set of integer points in $\mathbb{Z}^d$ of squared norm at most $u$, as in Definition A.1.

Throughout this appendix, for integers $n \geq 2$, $q \geq 2$, and $m \geq 1$, $u$ denotes the least nonnegative integer satisfying (33).

*Proof of Lemma 5.1.* Put $k = \lceil (m \log q)/(n-1) \rceil + 1$, which is bounded in terms of $D$. The box $\{-2^k, \ldots, 2^k\}^{n-1}$ has at least $q^m 2^{2(n-1)} \geq q^m 2^{\lceil n^{2/3} \rceil}$ points and squared radius $(n-1)4^k = O(n)$. Hence $u = O(n)$. For any fixed squared radius, the number of nonzero coordinates is bounded, so the number of points grows only polynomially in $n$. It cannot reach $q^m 2^{\lceil n^{2/3} \rceil} \geq 2^{1+\lceil n^{2/3} \rceil}$; therefore $u \to \infty$ uniformly in $q, m$.

For $s \geq 1$, move the first nonzero coordinate of a nonzero vector in $\mathcal{B}_{n-1}(s)$ one step toward zero. Its squared norm decreases by at least one, and every resulting vector has at most $2(n-1)$ preimages. Including the zero vector gives

$$|\mathcal{B}_{n-1}(s)| \leq (2n-1)|\mathcal{B}_{n-1}(s-1)|.$$

Minimality of $u$ proves the stated bounds on $|\mathcal{B}_{n-1}(u)|$.

To compute $u$ exactly, use the following recurrence. For integers $j, s \geq 0$, let $N_j(s)$ count the vectors of squared norm exactly $s$ in $\mathbb{Z}^j$. Start with $N_0(0) = 1$ and $N_0(s) = 0$ for $s > 0$, and use

$$N_{j+1}(s) = N_j(s) + 2 \sum_{v=1}^{\lfloor \sqrt{s} \rfloor} N_j(s - v^2), \qquad |\mathcal{B}_j(s)| = \sum_{r=0}^{s} N_j(r). \tag{57}$$

The recurrence separates the last coordinate according to whether it is zero, positive, or negative. Computing the table up to a cap on $s$ and doubling that cap until the threshold is reached determines

$u$; one then takes the first qualifying entry of the final table. Each cap is $O(n)$. Since $|v| \le v^2$ for integer $v$, taking absolute values and then assigning signs gives

$$|\mathcal{B}_j(s)| \le 2^j \binom{s+j}{j} \le 2^{s+2j} = 2^{O(n)}.$$

Thus there are polynomially many entries of $O(n)$ bits, and the exact computation has polynomial bit cost and space. □

## E.2 Comparing the point count with a theta sum

The recurrence computes the radius; the following comparison estimates how many vectors lie inside it. It supplies the theta-sum bound used by the exact-search cost analysis in Theorem 5.8.

*Proof of Lemma 5.2.* By Lemma 5.1, $u/(n-1)$ is bounded in terms of $D$, and $u \ge 1$ because $|\mathcal{B}_{n-1}(0)| = 1 < q^m 2^{\lceil n^{2/3} \rceil}$. By Lemma A.2, $\mu$ decreases strictly from $+\infty$ to 0, so $\mu(a) = u/(n-1) > 0$ has a unique solution. Since $u/(n-1)$ is bounded in terms of $D$, the solution is bounded below by a positive constant depending only on $D$, which we may take less than $\ln 4$; call it $a_D$. Since $\mu(\ln(8(n-1))) < 1/(n-1) \le u/(n-1)$ by Lemma A.2, strict decrease gives $a < \ln(8(n-1))$.

Every point of $\mathcal{B}_{n-1}(u)$ contributes at least $e^{-au}$ to $\Theta(a)^{n-1} = \sum_{\boldsymbol{z} \in \mathbb{Z}^{n-1}} e^{-a\|\boldsymbol{z}\|^2}$, so $|\mathcal{B}_{n-1}(u)| \le e^{au}\Theta(a)^{n-1}$. For the reverse bound, put $L = \lceil \log n \rceil$. For each nonzero integer $j$ with $|j| \le L$, set

$$c_j = \lfloor (n-1)p_a(j) \rfloor, \quad c_0 = n - 1 - \sum_{0<|j|\le L} c_j \ .$$

Consider the integer vectors having exactly $c_j$ coordinates equal to $j$ for each $|j| \le L$, with all other coordinates absent. Their squared norms are at most

$$\sum_{0<|j|\le L} j^2 c_j \le (n-1)\mu(a) = u \ .$$

Consequently,

$$|\mathcal{B}_{n-1}(u)| \ge \frac{(n-1)!}{c_0! \prod_{0<|j|\le L} c_j!} \ .$$

By Stirling's formula, the logarithm of this multinomial coefficient equals

$$-\sum_{|j|\le L} c_j \ln \frac{c_j}{n-1} + O((\log n)^2) \ ,$$

where $0 \ln 0 = 0$. The error is uniform because there are $O(\log n)$ terms.

We compare this expression with the entropy of $p_a$. Since $a \ge a_D > 0$, the Gaussian tails beyond $|j| > L$, including their entropy contributions, are smaller than every inverse power of $n$.

For each retained nonzero $j$, rounding $(n-1)p_a(j)$ down to $c_j$ changes its entropy contribution by $O(\log n)$. Indeed, when $(n-1)p_a(j) \ge 1$, this follows from the derivative of $-t \ln t$ on $t \ge 1/(n-1)$; otherwise,

$$(n-1)p_a(j) \ln \frac{1}{p_a(j)} = O(\log n) \ .$$

There are $O(\log n)$ retained values. The zero count differs from $(n-1)p_a(0)$ by $O(\log n)$, and its entropy contribution changes by the same order because $p_a(0) \ge 1/\Theta(a_D) > 0$ .

It follows that

$$\ln \frac{(n-1)!}{c_0! \prod_{0<|j|\le L} c_j!} = -(n-1) \sum_{j\in\mathbb{Z}} p_a(j) \ln p_a(j) + O((\log n)^2) \ .$$

Using $p_a(j) = e^{-aj^2}/\Theta(a)$ and $(n-1)\mu(a) = u$, the right-hand side is

$$au + (n-1) \ln \Theta(a) + O((\log n)^2) \ .$$

Therefore,

$$|\mathcal{B}_{n-1}(u)| \ge e^{au} \Theta(a)^{n-1} e^{-O((\log n)^2)},$$

as required. □

Since $a \le \ln(8(n-1))$ and $\Theta(a) \le \Theta(a_D)$, we have $e^a \Theta(a) = O(n)$. By (37) and Lemma 5.1, we obtain

$$|\mathcal{B}_n(u+1)| \le e^{a(u+1)} \Theta(a)^n = q^m \, 2^{n^{2/3} + O((\log n)^2)}. \tag{58}$$

### E.3 A nonzero lattice vector inside the radius

We show that the computed ball contains a nonzero lattice vector for most input matrices. For a composite modulus, pairs of candidate vectors can have different collision probabilities. We account for this by separating the divisors of the modulus.

*Proof of Lemma 5.3.* If $q^2 \le u+1$, the vector $q\boldsymbol{e}_1$ proves the claim for every matrix, where $\boldsymbol{e}_1$ is the first standard unit vector. Suppose instead that $q^2 > u+1$. Take the number $a$ with $\mu(a) = u/(n-1)$. By Lemma 5.1 and (37),

$$\begin{aligned} |\mathcal{B}_{n-1}(u)| &\ge q^m 2^{\lceil n^{2/3} \rceil}, \\ au + (n-1) \ln \Theta(a) - \ln |\mathcal{B}_{n-1}(u)| &= O((\log n)^2), \\ au + (n-1) \ln \Theta(a) &= O(n). \end{aligned}$$

For an integer $v \ge 2$, let $b(v)$ be the maximum number of points of $\mathcal{B}_{n-1}(u)$ in a single class modulo $v\mathbb{Z}^{n-1}$. By Lemma A.5 with $a_- = a_D$,

$$b(v) \le e^{au/v} \Theta(av)^{n-1}, \qquad b(v) \le e^{au} \Theta(av^2)^{n-1}, \tag{59}$$

and there is an integer $v_D \ge 2$, depending only on $D$ through $a_D$, such that $\Theta(av)^v \le \Theta(a)$ whenever $v \ge v_D$ or $a \ge \ln 4$. Hence $b(v) \le e^{au/v} \Theta(a)^{(n-1)/v}$ whenever $v \ge v_D$ or $a \ge \ln 4$.

Write $\mathbf{H} = [\boldsymbol{c} \ \mathbf{A}]$, and let

$$z = |\{\boldsymbol{v} \in \mathcal{B}_{n-1}(u) : \mathbf{A}\boldsymbol{v} = -\boldsymbol{c}\}|.$$

Each such $\boldsymbol{v}$ gives the nonzero lattice vector $(1, \boldsymbol{v}^\top)^\top$, of squared norm at most $u+1$. Moreover, $\mathbb{E}[z] = |\mathcal{B}_{n-1}(u)|/q^m$. For $\boldsymbol{v}, \boldsymbol{v}' \in \mathcal{B}_{n-1}(u)$, put $g = \gcd(q, v_1 - v_1', \ldots, v_{n-1} - v_{n-1}')$. A uniform row of $\mathbf{A}$ maps $\boldsymbol{v} - \boldsymbol{v}'$ uniformly onto the multiples of $g$ modulo $q$. Independence of $\boldsymbol{c}$ gives

$$\Pr[\mathbf{A}\boldsymbol{v} = \mathbf{A}\boldsymbol{v}' = -\boldsymbol{c}] = \frac{g^m}{q^{2m}}.$$

Separating these collisions by their divisors, as in the subgroup-based second-moment argument of Sahebi and Pradhan [SP15], yields

$$\frac{\mathrm{Var}[z]}{\mathbb{E}[z]^2} \le \frac{q^m b(q)}{|\mathcal{B}_{n-1}(u)|} + \sum_{\substack{v \mid q, \ 2 \le v < q \\ v \le 2\sqrt{u}}} \frac{v^m b(v)}{|\mathcal{B}_{n-1}(u)|}. \tag{60}$$

The first term includes identical vectors and all pairs congruent modulo $q$. For every other pair, a divisor of its nonzero difference is at most $2\sqrt{u}$; for each first vector there are at most $b(v)$ choices of the second vector. This proves the bound, with $O(\sqrt{n})$ terms.

For a proper divisor $v$ of $q$, write $q = vh$ with $h \geq 2$. Since $h \geq 2$ and $v \geq 2$ give $h^{v-1} \geq v$, we have $h^v \geq q$ and hence $v^m \leq q^{m(1-1/v)}$. If $v \geq v_D$ or $a \geq \ln 4$, its term in (60) is therefore at most

$$\begin{aligned}
&\frac{q^{m(1-1/v)} e^{au/v} \Theta(a)^{(n-1)/v}}{|\mathcal{B}_{n-1}(u)|} \\
&\quad = \left(\frac{q^m}{|\mathcal{B}_{n-1}(u)|}\right)^{1-1/v} \exp\left(\frac{au + (n-1)\ln\Theta(a) - \ln|\mathcal{B}_{n-1}(u)|}{v}\right) \\
&\quad \leq 2^{-\lceil n^{2/3}\rceil/2} e^{O((\log n)^2)}.
\end{aligned}$$

For the finitely many $2 \leq v < v_D$ with $a \in [a_D, \ln 4)$, compactness gives a constant $\xi > 0$, depending only on $D$, such that $\ln\Theta(a) - \ln\Theta(av^2) \geq \xi$. Also $\mu(a) \geq \mu(\ln 4) > 1/3$ by Lemma A.2. Thus $u > (n-1)/3$, and $q > \sqrt{u+1}$ implies $m = O(n/\log n)$. The second bound in (59) now gives

$$\frac{v^m b(v)}{|\mathcal{B}_{n-1}(u)|} \leq \exp\bigl(m\ln v - \xi(n-1) + O((\log n)^2)\bigr) = 2^{-\Omega(n)}.$$

Finally, $u \to \infty$ and $q > \sqrt{u+1}$ ensure $q \geq v_D$ for large $n$, so $b(q) \leq e^{au/q}\Theta(a)^{(n-1)/q}$. By Lemma A.2, $\ln\Theta(a) \leq \mu(a)\Theta(a)$, so the equation $\mu(a) = u/(n-1)$ implies

$$\frac{au + (n-1)\ln\Theta(a)}{u} \leq a + \Theta(a_D) = O(\log n).$$

Together with $au + (n-1)\ln\Theta(a) = O(n)$, this gives

$$\begin{aligned}
\frac{au + (n-1)\ln\Theta(a)}{q} &\leq \frac{au + (n-1)\ln\Theta(a)}{\sqrt{u}} = \sqrt{(au + (n-1)\ln\Theta(a)) \cdot \frac{au + (n-1)\ln\Theta(a)}{u}} \\
&= O(\sqrt{n\log n}).
\end{aligned}$$

Consequently,

$$\frac{q^m b(q)}{|\mathcal{B}_{n-1}(u)|} \leq 2^{-\lceil n^{2/3}\rceil} e^{O(\sqrt{n\log n})}.$$

Every contribution in (60), including the number of divisors summed, is absorbed in $2^{-\Omega(n^{2/3})}$. Applying Markov's inequality to $(z - \mathbb{E}[z])^2$ bounds $\Pr[z = 0]$ by that expression. □

### E.4 Excluding short vectors with a common divisor

To justify the coordinate restrictions in the randomized search, we must control the probability that a fixed integer vector belongs to the lattice. For composite moduli, that probability depends on the common divisor of its coordinates and the modulus. We first show that the relevant short vectors have common divisor one on most matrices. This step uses a positive lower bound on the density. We begin by locating the counting parameter $a$ in this range.

*Proof of Lemma 5.4.* By (51), the entropy in natural logarithms equals

$$a\mu(a) + \ln\Theta(a) = \frac{au + (n-1)\ln\Theta(a)}{n-1} = \delta\ln 2 + o(1),$$

uniformly over the input range, by Lemma 5.1 and (37). By Lemma A.2, this entropy decreases strictly with $a$, from infinity to zero. The asserted compact interval follows from $\delta_0 \leq \delta \leq D$. □

We can now exclude the vectors for which the common divisor would increase the probability of satisfying the congruence.

**Lemma E.1.** *Fix constants $0 < \delta_0 \le D < \infty$. Let $q \ge 3$, $m \ge 1$, and $n \ge 2$ be integers with $\delta_0 \le \delta \le D$ and $q^4 > n$. Let $u$ be as in (33). Except for a fraction $2^{-\Omega(n/\log n)}$ of uniformly chosen matrices $\mathbf{H} \in (\mathbb{Z}/q\mathbb{Z})^{m\times n}$, every vector $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying*

$$\mathbf{H}\boldsymbol{x} = \mathbf{0}_m, \qquad 0 < \|\boldsymbol{x}\|^2 \le u + 1, \qquad \|\boldsymbol{x}\| < q$$

*has $\gcd(q, x_1, \dots, x_n) = 1$. The hidden constant may depend on $\delta_0, D$.*

*Proof.* For a fixed nonzero integer vector $\boldsymbol{x}$, write $g = \gcd(q, x_1, \dots, x_n)$. A uniform row maps $\boldsymbol{x}$ uniformly onto the multiples of $g$ modulo $q$, so

$$\Pr_{\mathbf{H}}[\mathbf{H}\boldsymbol{x} = \mathbf{0}_m] = (g/q)^m.$$

If $\|\boldsymbol{x}\| < q$, then $g < q$. For $g > 1$, write $\boldsymbol{x} = g\boldsymbol{v}$, where $\|\boldsymbol{v}\|^2 \le (u+1)/g^2$ and $g \le \sqrt{u+1} = O_D(\sqrt{n})$ by Lemma 5.1. For a fixed divisor $g$ of $q$, the vectors whose exact common divisor is $g$ are a subset of these multiples, and every such vector has the probability $(g/q)^m$ just computed.

Take $a$ with $\mu(a) = u/(n-1)$. For each proper divisor $g > 1$ of $q$, summing $e^{-ag^2\|\boldsymbol{v}\|^2}$ over $\boldsymbol{v} \in \mathbb{Z}^n$ bounds the expected number of vectors with exact common divisor $g$ by

$$(g/q)^m e^{a(u+1)}\Theta(ag^2)^n \le 2^{n^{2/3}+O_D((\log n)^2)} g^m \left(\frac{\Theta(ag^2)}{\Theta(a)}\right)^n. \tag{61}$$

The last inequality is (58). By Lemma 5.4, $a$ lies in a fixed compact subinterval of $(0, \infty)$. By Lemma A.5, $\Theta(ag^2)/\Theta(a) \le \Theta(4a)/\Theta(a) < 1$ for $g \ge 2$, and by compactness there is a constant $\xi > 0$ such that $\Theta(4a)/\Theta(a) \le 2^{-\xi}$ uniformly on this interval. For $2 \le g \le \lceil \log n \rceil^2$, the condition $q^4 > n$ gives $m < 4Dn/\log n$, and hence

$$m \log g = O_D(n \log\log n/\log n) = o(n).$$

The preceding expectation bound is therefore $2^{-\Omega(n)}$, uniformly over these divisors.

If $g > \lceil \log n \rceil^2$, then $\|\boldsymbol{v}\|^2 = O_D(n/\log^4 n)$. Such vectors have at most $O_D(n/\log^4 n)$ nonzero coordinates, and specifying their positions and values gives $2^{O_D(n/\log^3 n)}$ possibilities. If $q \le n$, the facts $q/g \ge 2$ and $m \ge \delta_0 n/\log n$ bound the probability of each by $2^{-\delta_0 n/\log n}$. If $q > n$, the bound $u + 1 = O_D(n)$ gives, for sufficiently large $n$ and uniformly over these moduli,

$$g \le \sqrt{u+1} \le n^{2/3} < q^{2/3}, \qquad (g/q)^m \le q^{-m/3} \le 2^{-\delta_0 n/3}.$$

After multiplication by the candidate count, these two cases give $2^{-\Omega(n/\log n)}$ and $2^{-\Omega(n)}$, respectively. Summing over $O_D(\sqrt{n})$ possible divisors shows that the probability of any such vector is $2^{-\Omega(n/\log n)}$. □

## E.5 The coordinate restrictions used by the randomized search

We now prove the sign-count windows in (36) and the restriction on the separately enumerated coordinates in (35). The claim concerns every relevant vector in the lattice; it does not presume a probability distribution for a shortest vector chosen after the matrix is known.

The distribution $r_a$ of (49) is obtained from $p_a$ by replacing $-1, 0, 1$ by zero. Its entropy $\mathcal{H}(r_a)$ therefore describes the coordinates retained by this replacement. For an integer vector $\boldsymbol{x}$, write $\boldsymbol{z}$ for the vector with $z_j = x_j$ when $|x_j| \ge 2$ and $z_j = 0$ otherwise. Thus $\boldsymbol{z}$ records precisely the coordinates enumerated outside the short congruence search.

*Proof of Lemma 5.5.* First consider fixed integer vectors with $\gcd(q, x_1, \ldots, x_n) = 1$. Each belongs to the lattice with probability exactly $q^{-m}$ under a uniform matrix. We count the vectors violating the stated coordinate conditions by sampling independent coordinates with distribution $p_a$; this sampling is only a device for counting integer vectors. The number of entries equal to either fixed sign has mean $np_a(1)$. For a Bernoulli variable $v$ of mean $p$, $\ln \mathbb{E}\left[e^{t(v-p)}\right] \le t^2/8$: its second derivative is at most $1/4$, and its value and first derivative at zero vanish. Applying this bound independently to the coordinate indicators gives a probability at most $2e^{-2\varepsilon^2 n}$ that the count differs from its mean by more than $\varepsilon n$.

The coordinates $z_j$ then have distribution $r_a$. The independent variables $y_j = -\log r_a(z_j)$ have mean $\mathcal{H}(r_a)$. The compact range of $a$ from Lemma 5.4 implies uniform Gaussian tails and, in particular,

$$\mathbb{E}\left[2^{y_j/2}\right] = \sum_{v\in\mathbb{Z}} \sqrt{r_a(v)} = O(1).$$

Their centered logarithmic moment generating functions are bounded by a constant times the square of the argument near zero, uniformly in $a$. Applying the exponential form of Markov's inequality with a sufficiently small positive argument therefore gives

$$\Pr\left[\sum_{j=1}^{n} y_j > n(\mathcal{H}(r_a) + \varepsilon)\right] \le 2^{-c_\varepsilon n}$$

for a constant $c_\varepsilon > 0$.

For every $\boldsymbol{x} \in \mathcal{B}_n(u+1)$,

$$\prod_{j=1}^{n} p_a(x_j) = \frac{e^{-a\|\boldsymbol{x}\|^2}}{\Theta(a)^n} \ge \frac{e^{-a(u+1)}}{\Theta(a)^n}.$$

Hence the number of points of this ball violating either coordinate condition is at most $e^{a(u+1)}\Theta(a)^n$ times their probability under the independent-coordinate distribution. Multiplying by $q^{-m}$ for the vectors with common divisor one, and using (58), bounds their expected number in the lattice by $2^{-\Omega(n)}$. Markov's inequality bounds the probability of any such vector by $2^{-\Omega(n)}$. Adding the exceptional probability from Lemma E.1 proves the claim simultaneously for all the stated lattice vectors. □

The product condition also bounds the number of separately enumerated vectors $\boldsymbol{z}$. Since their probabilities under $r_a^{\otimes n}$ sum to at most one, at most $2^{n(\mathcal{H}(r_a)+\varepsilon)}$ vectors can satisfy it. This is the exponential count used in Theorem 5.8; the next subsection implements the test and enumeration with polynomial delay and space.

## E.6 Implementing the SVP search

We now implement the parameter calculations and coordinate tests in Algorithm 4. Rational intervals retain every vector required by the correctness proof, and the enumeration generates the permitted vectors one at a time with polynomial delay and space. The last part implements the algorithm's random choices with finitely many fair bits.

### E.6.1 Computing the parameters

The radius $u$ is computed by Lemma 5.1. Throughout this implementation, $b = \min\{u+1, q^2\}$ is the squared-radius bound in (33), so $b = O(n)$.

Comparisons of the density $\delta$ with rational thresholds need no numerical logarithms. For positive integers $r_1, r_2$, the test $\delta \le r_1/r_2$ is equivalent to $q^{mr_2} \le 2^{r_1 n}$. The completion dimension $n_0 = \lceil n/(1-\kappa) \rceil$ in Section 5.1 is computed directly by integer division, since $\kappa$ is rational. Any fixed rational upper bound on the density is checked by the same rule; $m$ is supplied independently of $n, q$. All these integers have polynomial encoding length on the input range in question.

The following lemma records the numerical evaluations needed below.

**Lemma E.2.** *Fix rational constants $0 < a_- < a_+$. Given a rational $a \in [a_-, a_+]$, an integer $N \ge 2$, and an integer $j$, one can compute rational intervals of width at most $1/N$ containing $\Theta(a)$, $\Theta'(a)$, $\Theta''(a)$, $\mu(a)$, $\mathcal{H}(p_a)$, $\mathcal{H}(r_a)$, and $p_a(j)$. If a rational interval $I \subseteq [a_-, a_+]$ of width at most $1/N$ is given instead of $a$, one can compute rational intervals of width $O(1/N)$ containing the values of the same functions at every point of $I$. Given a rational $\theta$ with $\mu(a_+) \le \theta \le \mu(a_-)$ and an integer $N \ge 2$, one can compute a rational interval of width at most $1/N$ containing the unique $a \in [a_-, a_+]$ satisfying $\mu(a) = \theta$. All these computations take time polynomial in the total binary encoding length of the supplied inputs. The polynomial and the implicit constant depend only on $a_-$ and $a_+$.*

*Proof.* Each of the series defining $\Theta(a)$, its first two derivatives, $\mu(a)$, and the two entropies has terms bounded by a polynomial in $j$ times $e^{-a_- j^2}$, so truncating at $|j| \le \ell$ with $\ell = O(\sqrt{\log N})$ leaves a tail smaller than $1/(2N)$, uniformly for $a \in [a_-, a_+]$. For the evaluation of $p_a(j)$, if $|j| > \ell$, the tail bound and $\Theta(a) \ge 1$ give $p_a(j) < 1/(2N)$, so $[0, 1/N]$ is a valid enclosure. Otherwise, $p_a(j)$ is evaluated using the retained terms. The retained terms involve $e^{-aj^2} = (e^{-a})^{j^2}$ with $j^2 = O(\log N)$, and logarithms of quantities in a fixed positive compact interval. Exponential values are enclosed by the Taylor series with its factorial remainder, and logarithms by

$$\left| \ln x - 2 \sum_{j=0}^{r-1} \frac{1}{2j+1} \left( \frac{x-1}{x+1} \right)^{2j+1} \right| \le \frac{2 \left| \frac{x-1}{x+1} \right|^{2r+1}}{(2r+1) \left( 1 - \left( \frac{x-1}{x+1} \right)^2 \right)},$$

with $r = O(\log N)$ terms. Rational interval arithmetic on these enclosures gives the stated widths with polynomially many bits. All these functions have derivatives bounded on $[a_-, a_+]$, so evaluating the same enclosures over an interval $I$ of width at most $1/N$ enlarges the widths by at most $O(1/N)$.

To enclose $a$ with $\mu(a) = \theta$, bisect the bracket $[a_-, a_+]$. On this bracket, $-\mu'(a) = \mathrm{Var}_{p_a}[j^2] \ge 2p_a(0)p_a(1) \ge 2e^{-a_+}/\Theta(a_-)^2$ by Lemma A.2. Choose an integer $N' \ge N\Theta(a_-)^2 e^{a_+}$ by rounding up the product of $N$ and fixed rational upper bounds for $\Theta(a_-)^2$ and $e^{a_+}$. Then $N' = O(N)$. At each midpoint, enclose $\mu$ in an interval of width at most $1/N'$. If the enclosure contains $\theta$, the derivative bound confines $a$ to an interval of width at most $1/N$ around the midpoint; otherwise the sign of the difference determines the half containing $a$. After $O(\log(N(a_+ - a_-) + 2))$ steps the bracket has width at most $1/N$. □

For the completion branch, the parameter $a$ lies in a fixed compact subinterval $[a_-, a_+]$ of $(0, \infty)$ by Lemma 5.4, and $u/(n-1)$ is a rational number of polynomial encoding length. By Lemma E.2, a rational interval containing $a$, and rational intervals containing $p_a(1)$, $\mathcal{H}(r_a)$, and the coefficients used below, all of any prescribed inverse-polynomial width, are computed in polynomial time and bits. In particular, enclose $p_a(1) = e^{-a}/\Theta(a)$ in an interval of width at most $\min\{n^{-2}, 10^{-12}\}$ and take its upper endpoint as $\widehat{p}$. This gives $0 \le \widehat{p} - p_a(1) \le \min\{n^{-2}, 10^{-12}\}$ and hence rational proportions $t_0 = \widehat{p} + \lambda$, $t_i = c_i t_0$ as in (34), with polynomial encoding length. The integer $a_0$ is then obtained by the rounding rule in that equation. These computations use only $n, m, q$ and the counted radius, so the proportions are independent of the random matrix.

The entropy needed for (35) is evaluated from (52).

### E.6.2 Implementing the coordinate comparisons

For a vector $\boldsymbol{z}$ with entries zero or of absolute value at least two, the logarithm in (35) is

$$-\sum_{j=1}^{n} \log r_a(z_j) = n \log \frac{\Theta(a)}{1+2e^{-a}} + |\operatorname{supp}(\boldsymbol{z})| \log(1+2e^{-a}) + \frac{a}{\ln 2}\|\boldsymbol{z}\|^2. \tag{62}$$

The three coefficients are nonnegative, since $\Theta(a) \geq 1+2e^{-a}$ and $a > 0$. Compute rational enclosing intervals of width at most $1/(4(n+b+1))$ for them and replace any negative lower endpoint by zero. Denote these nonnegative rational lower bounds by $\underline{\alpha}_0, \underline{\alpha}_1, \underline{\alpha}_2$, respectively. Thus

$$\begin{aligned}
0 &\leq \log \frac{\Theta(a)}{1+2e^{-a}} - \underline{\alpha}_0 \leq \frac{1}{4(n+b+1)}, \\
0 &\leq \log(1+2e^{-a}) - \underline{\alpha}_1 \leq \frac{1}{4(n+b+1)}, \\
0 &\leq \frac{a}{\ln 2} - \underline{\alpha}_2 \leq \frac{1}{4(n+b+1)}.
\end{aligned}$$

Independently enclose $n(\mathcal{H}(r_a)+\varepsilon)$ in an interval of width at most one and take its upper endpoint $\bar{t}$, so $0 \leq \bar{t} - n(\mathcal{H}(r_a)+\varepsilon) \leq 1$. For $\|\boldsymbol{z}\|^2 \leq b$, the resulting error satisfies

$$\begin{aligned}
0 \leq\ & -\sum_{j=1}^{n} \log r_a(z_j) - \left(n\underline{\alpha}_0 + |\operatorname{supp}(\boldsymbol{z})|\underline{\alpha}_1 + \|\boldsymbol{z}\|^2\underline{\alpha}_2\right) \\
&\leq \frac{2n+b}{4(n+b+1)} \leq 1.
\end{aligned}$$

Use the exact rational comparison

$$n\underline{\alpha}_0 + |\operatorname{supp}(\boldsymbol{z})|\underline{\alpha}_1 + \|\boldsymbol{z}\|^2\underline{\alpha}_2 \leq \bar{t}. \tag{63}$$

Every vector satisfying (35) passes this test. Among vectors with $\|\boldsymbol{z}\|^2 \leq b$, every vector passing it has $-\sum_{j=1}^{n} \log r_a(z_j) \leq n(\mathcal{H}(r_a)+\varepsilon)+2$. Thus at most $2^{n(\mathcal{H}(r_a)+\varepsilon)+2}$ vectors can pass: each has probability at least $2^{-n(\mathcal{H}(r_a)+\varepsilon)-2}$ under $r_a^{\otimes n}$, whose total mass is one.

For the sign-count windows in (36), enclose $np_a(1)$ in an interval of width at most one and expand both ends by $\varepsilon n+1$. Every count in the exact window is retained, and each endpoint extends by at most one beyond that window. Enumerate nonnegative integers in this interval, applying $N_+, N_- \leq a_0$ and the squared-length conditions in (36) by exact integer comparisons.

### E.6.3 Enumerating the coordinates

**Definition E.3** (Enumerating the large coordinates)**.** Given integers $n \geq 1$, $b \geq 0$, nonnegative rational numbers $\underline{\alpha}_0, \underline{\alpha}_1, \underline{\alpha}_2$, and a rational threshold $\bar{t}$, output exactly once each vector

$$\boldsymbol{z} \in (\{0\} \cup \{j \in \mathbb{Z} : |j| \geq 2\})^n$$

with $\|\boldsymbol{z}\|^2 \leq b$ satisfying (63). If no such vector exists, output nothing.

We apply this enumeration to the rational bounds constructed in Section E.6.2. Algorithm 13 solves the problem in Definition E.3 by maintaining the current vector $\boldsymbol{z}$. On entry to a call at depth $i$, its first $i$ coordinates have been assigned and the remaining coordinates are zero. Both the support size and squared norm in the test are computed directly from this vector.

**Algorithm 13** Enumerating the prescribed large coordinates

**Input:** Integers $n \geq 1$, $b \geq 0$, nonnegative rational numbers $\underline{\alpha}_0, \underline{\alpha}_1, \underline{\alpha}_2$, and a rational threshold $\bar{t}$.
**Output:** All vectors required by Definition E.3, exactly once each.

```
Set z ← 0_n and call V(0).
procedure V(i)
    if n·α_0 + |supp(z)|·α_1 + ‖z‖^2·α_2 > t̄ then return
    if i = n then output z and return
    for j ∈ {0} ∪ {±r : 2 ≤ r ≤ ⌊√(b − ‖z‖^2)⌋}, with j = 0 first do
        z_{i+1} ← j.
        Call V(i + 1).
        z_{i+1} ← 0.
    end for
end procedure
```

The squared norm never exceeds $b$, by the choice of the next coordinate. Each call preserves its first $i$ coordinates and returns with all later coordinates zero: after a child returns, its newly assigned coordinate is reset. This also holds when the child returns at the initial test or after an output.

Nonnegativity of the coefficients ensures that a rejected prefix has no passing extension. Every passing prefix has a passing completion obtained by setting the remaining coordinates to zero. The constant term in the test remains $n\underline{\alpha}_0$, including the contributions of those zero coordinates. Every vector satisfying the length and rational tests therefore appears, and its coordinate choices determine a unique path, so it appears exactly once.

Thus no retained prefix leads to a subtree without an output. The depth is $n$ and the number of possible children is $O(\sqrt{b+1})$, so the work between consecutive outputs, and before the first or after the last, is polynomial in $n, b$ and the encoding lengths of the rational inputs. Computing the support and squared norm at each call adds only a polynomial factor. The current vector and the recursion stack require polynomial space. In the SVP application, $b = O(n)$, giving the delay and space bounds used in Theorem 5.8.

### E.6.4 Implementing the random choices

The SVP algorithm samples the added matrices $\mathbf{D}$ and calls the randomized common enumeration. To sample one entry of $\mathbf{D}$, read $\lceil \log q \rceil + n^2$ independent fair bits as an integer $v \in \{0, \ldots, 2^{\lceil \log q \rceil + n^2} - 1\}$ and return

$$\left\lfloor \frac{qv}{2^{\lceil \log q \rceil + n^2}} \right\rfloor \in \{0, \ldots, q-1\}.$$

Each value has either the floor or ceiling of the average number of preimages, so this distribution is within total variation $2^{-n^2}$ of uniform. The choices inside the common enumeration use Appendix B.6 in dimension $n_0$. All bit blocks for different entries and calls are independent of the input and of one another, including across rounds.

There are at most $2^{O(n)}$ draws in the stated algorithm. For every fixed input matrix, the total change in any success probability is at most $2^{-\Omega(n^2)}$, including the errors of the common enumeration. Generate the blocks as needed for each entry or call and reuse their storage afterwards. Sampling and decoding thus adds only a polynomial factor to the time and uses polynomial extra space. The error is smaller than the failure bounds in Theorem 5.7.

# F Verification and Optimization of the 5/3-CVP Parameters

This appendix verifies the parameters of our 5/3-CVP construction in both density ranges. For $0 \le \delta \le 1$, it places the infimum of the proved uniform time exponent in $[0.26675186, 0.26675191)$. An explicit parameter choice gives the upper bound, and the lower bound covers every allowed choice and every fixed number of decomposition steps. We state this optimization problem first, then verify the chosen parameters and prove the lower bound. The final subsection verifies the parameter choices and cost upper bounds for $1 < \delta \le 3/2$.

The upper bounds used by the 5/3-CVP theorems are verified in Appendices F.2 and F.7. The remaining numerical lower bounds concern optimization within the parameter family specified below. The accompanying archive `numerical-certificates.zip` contains the verification programs, recorded outputs, and execution instructions in `scripts/README.md`.

## F.1 The optimization problem for density at most one

We specify the full cost to be minimized and all constraints on its parameters for the uniform bound in Corollary 4.8 on $0 \le \delta \le 1$. The space bound is evaluated for the same parameter choice.

For $0 < x < 1/2$ and $x/2 \le y \le (1-x)/2$, define

$$\begin{aligned} g(x) &= \mathcal{H}(x, x, 1-2x), \\ r(x,y) &= 2x + (1-2x)\mathcal{H}\left(\frac{y-x/2}{1-2x}, \frac{y-x/2}{1-2x}, \frac{1-x-2y}{1-2x}\right), \\ \ell(x,y) &= g(y) - r(x,y). \end{aligned}$$

Here $g$ is the exponential rate of the candidate count, $r$ is the rate of the decomposition count, and $\ell$ is their difference. The entropy convention $0 \log 0 = 0$ includes the endpoints of the allowed interval for $y$. The same convention extends $g$ to $[0, 1/2]$.

The variables are an integer $k \ge 1$, rational numbers $0 < t_i < 1/2$ for $0 \le i \le k$, rational numbers $\eta > 0$ and $c$, and a real number $a > 0$, all fixed independently of the dimension. The coordinate proportions must satisfy

$$\frac{t_{i-1}}{2} \le t_i \le \frac{1-t_{i-1}}{2} \qquad (1 \le i \le k).$$

Put

$$h_i = g(t_i) \quad (0 \le i \le k), \qquad r_i = r(t_{i-1}, t_i), \quad d_i = r_i - \eta \quad (1 \le i \le k), \qquad d_0 = 1.$$

The value $d_0 = 1$ is the limiting equation rate of the correction branch in this density range: for $q \ge 3$, it uses $n_0 = \delta n - O(n^{2/3})$ coordinates. The short-solution construction requires

$$1 > d_1 > \cdots > d_k > 0, \qquad h_0 > 1,$$

and, to include even moduli,

$$h(2t_0) > \frac{1}{2}, \qquad h(2t_i) > \frac{d_{i-1}}{2}, \qquad r_i > \frac{d_i}{2} + 2t_i - t_{i-1} \quad (1 \le i \le k).$$

Its time and space exponents $T_{\mathrm{in}}$ and $S_{\mathrm{in}}$ are given by (9) with these $h_i$ and $d_i$.

For exact search, use the functions in (48). At $\gamma = 5/3$, (24) becomes

$$e(a, c, 5/3) = \frac{9ac/25 + \ln \Theta(a)}{2 \ln 2}.$$

For the binary branch we additionally require

$$0 < 2t_0 < 1/4, \qquad h(2t_0) > 3/4.$$

The remaining constraints are

$$c > 2t_0, \qquad \Theta_u(a) \le \Theta(a)e^{-9au^2/25} \quad \text{for every } |u| \le 1/2.$$

The complete 5/3-CVP algorithm has time and space exponents

$$\max\{1/4, T_{\text{in}}, e(a,c,5/3)\} \quad \text{and} \quad \max\{1/8, S_{\text{in}}, e(a,c,5/3)/2\},$$

respectively. We prove

$$0.26675186 \le \inf \max\{1/4, T_{\text{in}}, e(a,c,5/3)\} < 0.26675191, \tag{64}$$

where the infimum ranges over exactly the choices specified above. The explicit choice below is therefore within $5 \cdot 10^{-8}$ of this infimum. The strict constraints need not permit attainment of the infimum. The lower bound will follow from $T_{\text{in}}$ alone and hence applies to every allowed choice of the exact-search parameters. The lower-bound reductions below permit the larger domain obtained by omitting the two binary constraints, so they remain valid here. The additional $1/4$ term is smaller than the certified lower bound on $T_{\text{in}}$ and therefore does not change the infimum.

Every finite decimal used in the verification denotes an exact rational number. Each nonrational function value is enclosed in an interval whose lower endpoint is rounded downward and upper endpoint upward. A strict lower bound is accepted only when the lower endpoint exceeds the claimed bound; strict upper bounds use the upper endpoint. The elementary enclosures and precisions are specified in Appendix F.6.

## F.2 Verifying the selected parameters

We establish the upper bound by checking one choice against all the constraints, then evaluating the complete time and space bounds. The first coordinate proportion is chosen close to the boundary $g(t_0) = 1$. The second and third combinations have nearly equal costs; exact search has a smaller exponent. The lower-bound argument below explains why additional decomposition steps cannot improve this choice by as much as $5 \cdot 10^{-8}$.

Take $k = 3$ and

$$(t_0, t_1, t_2, t_3) = (0.1135461, 0.09917397, 0.06690926, 0.03531462),$$
$$\eta = 10^{-8}, \qquad c = 0.22711, \qquad a = 4.138.$$

Rational arithmetic verifies $0 < t_i < 1/2$ and $t_{i-1}/2 < t_i < (1 - t_{i-1})/2$. The following interval bounds check the remaining short-solution conditions and the distance margin.

| Quantity | Certified positive lower bound |
|---|---|
| $h_0 - 1$ | $1.3228 \cdot 10^{-8}$ |
| $\min\{d_3, 1 - d_1, d_1 - d_2, d_2 - d_3\}$ | 0.1721 |
| $h(2t_0) - 1/2$ | 0.2729 |
| $\min_{1 \le i \le 3}(h(2t_i) - d_{i-1}/2)$ | 0.1488 |
| $\min_{1 \le i \le 3}(r_i - d_i/2 - 2t_i + t_{i-1})$ | 0.0823 |

Also $c-2t_0 = 0.0000178 > 0$ and $\eta = 10^{-8} > 0$. Thus the coordinate, equation, and parity conditions all hold with positive margins. The output-length coefficient is $\beta = 2t_0 = 0.2270922$.

To check the exact-search condition, use $\chi(a)$ from Lemma A.4. The interval evaluations give

$$\tanh^2 \frac{\pi^2}{2a} > 0.691176 > \frac{3}{5}, \qquad \chi(a) - 9a/25 > 0.0002506.$$

By Lemma A.4, these inequalities imply $\Theta_u(a) \le \Theta(a)e^{-9au^2/25}$ for every $|u| \le 1/2$. The selected parameters therefore satisfy all constraints of the optimization problem.

The individual contributions to time have the following upper bounds.

| Operation | Exponent | Certified upper bound |
|---|---|---|
| Last-level coordinate enumeration | $h_3/2$ | 0.219448358 |
| First combination | $[h_1 - d_1]_+ + [h_1 - 1]_+$ | 0.219334673 |
| Second combination | $[h_2 - d_2]_+ + [h_2 - d_1]_+$ | 0.266751897 |
| Third combination | $[h_3 - d_3]_+ + [h_3 - d_2]_+$ | 0.266751910 |
| Exact search | $e(a, c, 5/3)$ | 0.266706 |

The enclosing intervals also show that the third combination has the largest time exponent, while $[h_2 - d_2]_+$ is the largest space term. The binary conditions also follow: $\beta = 2t_0 = 0.2270922 < 1/4$ and $h(\beta) = h_0 - \beta > 1 - \beta > 3/4$. The elementary branches' exponents $1/4$ in time and $1/8$ in space are smaller than the displayed representation costs. In particular,

$$\max\{1/4, T_{\text{in}}, e(a, c, 5/3)\} < 0.26675191, \qquad \max\{1/8, S_{\text{in}}, e(a, c, 5/3)/2\} < 0.262740033.$$

The short-solution and exact-search guarantees now apply with the same parameters. By Theorem 4.7 and Corollary 4.8, this gives the stated guarantees for 5/3-CVP, with time $2^{0.266752n+o(n)}$ and space $2^{0.262741n+o(n)}$.

For the same parameter choice, the coefficients of the exact-search exponent in (25) satisfy

$$0.244047176623220090 < \frac{ac}{2\gamma^2 \ln 2} < 0.244047176623220091,$$
$$0.022658286239231720 < \frac{\ln \Theta(a)}{2 \ln 2} < 0.022658286239231722.$$

Together with the intervals for $T_{\text{in}}, S_{\text{in}}$, these bounds certify the branch comparisons and crossings in Section 4.4 throughout $0 \le \delta \le 1$.

### F.3 Reducing all fixed depths to two variables

We derive a lower bound that applies to every fixed recursion depth. Depths one and two can be excluded directly. At every larger depth, the first three combinations suffice: their costs reduce the problem to two coordinate proportions, regardless of the later choices.

Let $\alpha$ be the unique solution of $g(\alpha) = 1$ in $(0, 1/3)$. The numerical bounds proved below give

$$0.11354609760967409 < \alpha < 0.11354609760967411. \tag{65}$$

For $0 < x < 1/2$ and $d \ge 0$, define

$$\psi(x, d) = \min_{x/2 \le y \le (1-x)/2} \{\ell(x, y) + [g(y) - d]_+\}.$$

This is the smallest combination cost when the preceding coordinate proportion is $x$ and the preceding congruence exponent is $d$. The polynomial

$$f(x,y) = y(1-2y)(1-3y) - \frac{x}{2}(1-4y+2y^2)$$

will determine the stationary points needed in the reduction and in the evaluation of $\psi$.

**Lemma F.1.** *Let $k \geq 1$ be an integer, let $\eta \geq 0$, and let $0 < t_i < 1/2$ for $0 \leq i \leq k$ satisfy*

$$g(t_0) \geq 1, \qquad \frac{t_{i-1}}{2} \leq t_i \leq \frac{1-t_{i-1}}{2} \quad (1 \leq i \leq k).$$

*Set $h_i = g(t_i)$ for $0 \leq i \leq k$, $d_0 = 1$, and $r_i = r(t_{i-1}, t_i)$, $d_i = r_i - \eta$ for $1 \leq i \leq k$. Put*

$$T_{\mathrm{in}} = \max\left\{ \frac{h_k}{2},\ \max_{1 \leq i \leq k} \left( [h_i - d_i]_+ + [h_i - d_{i-1}]_+ \right) \right\}.$$

*Then*

$$T_{\mathrm{in}} \geq \min\left\{ 0.267,\ \inf_{\substack{0.08 < x < 0.14 \\ x/2 \leq y < 0.14}} \max\left\{ \ell(x,y) + [g(y) - r(\alpha, x)]_+,\ \psi(y, r(x,y)) \right\} \right\}.$$

*Proof.* Positive $\eta$ can only increase the displayed costs, so it suffices to consider $\eta = 0$. No restriction on nesting or parity is needed for this lower bound. The first combination then costs $\ell(t_0, t_1) + [g(t_1) - 1]_+$, and each later combination costs

$$\ell(t_{i-1}, t_i) + [g(t_i) - r(t_{i-2}, t_{i-1})]_+.$$

The positive part around $\ell$ is unnecessary because $\ell(x,y) \geq x > 0$, as we now verify.

In the interior of $x/2 \leq y \leq (1-x)/2$, differentiation gives

$$\frac{\partial r}{\partial x}(x,y) = 2 + \log \frac{(y - x/2)(1-x-2y)}{(1-2x)^2} \leq -1, \qquad \frac{\partial r}{\partial y}(x,y) = 2 \log \frac{1-x-2y}{y - x/2}.$$

For the first inequality, the product in the numerator is at most $(1-2x)^2/8$. Also,

$$\frac{\partial \ell}{\partial y}(x,y) = 2 \log \frac{(1-2y)(y-x/2)}{y(1-x-2y)}.$$

Its sign is the sign of $y - 1/4$. Thus $\ell$ increases in its first argument and decreases in its second argument up to $1/4$, and continuity at the endpoints gives

$$\ell(x,y) \geq \ell(x, 1/4) = x. \tag{66}$$

The hypothesis $g(t_0) \geq 1$ implies $t_0 \geq \alpha$. Decreasing $t_0$ to $\alpha$ decreases the first two combination costs, leaves all later costs unchanged, and enlarges the permitted interval for $t_1$. We may therefore assume $t_0 = \alpha$.

**Depths one and two.** The counting inequality $h_0 + r_1 \leq 2h_1$ gives, at depth one,

$$T_{\mathrm{in}} \geq \max\{h_1/2, 1 - h_1\} \geq 1/3.$$

At depth two, suppose $T_{\mathrm{in}} \leq 0.267$. Then $g(t_2) \leq 0.534$. Concavity, $g(1/2) = 1$, and $g(0.046) > 0.535110$ imply $t_2 < 0.046$, and hence $t_1 \leq 2t_2 < 0.092$. If $t_1 \leq 0.08$, the first combination costs at least

$$\ell(0.1135, 0.08) > 0.267218.$$

If $t_1 > 0.08$, the second combination instead costs at least

$$\ell(0.08, 0.046) > 0.272371.$$

Both contradict $T_{\mathrm{in}} \leq 0.267$.

**Every larger depth.** Suppose $k \geq 3$ and $T_{\text{in}} \leq 0.267$, and put $x = t_1$, $y = t_2$. Equation (66) gives $x, y \leq 0.267$. The preceding first-combination comparison also gives $x > 0.08$. For $x \in [0.14, 0.267]$, the first cost is $2g(x) - r(\alpha, x) - 1$, whose derivative has the sign of $f(\alpha, x)$. Since $\alpha < 0.114$ and $1 - 4x + 2x^2 > 0$ on this interval, $f(\alpha, x) \geq f(0.114, x)$. The latter polynomial is concave there and positive at both endpoints, with values 0.0311496 and 0.020509032. Thus the first cost increases on this interval; at $x = 0.14$ it already exceeds 0.296361. Consequently $x < 0.14$.

For $0.08 < x < 0.14$, the function $r(\alpha, x) - x$ increases with $x$ and is bounded above by

$$r(0.1135, 0.14) - 0.14 < 0.834540.$$

Indeed, $\partial r/\partial y(\alpha, x) - 1$ is positive up to $x = 0.14$, and $\partial r/\partial x < 0$. The second cost is at least $x + g(y) - r(\alpha, x)$. If $0.14 \leq y \leq 0.267$, this exceeds

$$g(0.14) - 0.834540 > 0.300910,$$

again a contradiction. We have proved

$$0.08 < x < 0.14, \qquad x/2 \leq y < 0.14. \tag{67}$$

The second and third combination costs now imply

$$T_{\text{in}} \geq \max\left\{\ell(x, y) + [g(y) - r(\alpha, x)]_+,\, \psi(y, r(x, y))\right\}. \tag{68}$$

This proves the claimed bound. All strict numerical comparisons used in the proof are enclosed by the rational arithmetic bounds given below. □

The reduction leaves a minimum over the third coordinate proportion inside $\psi$. The following formula evaluates that minimum using one polynomial root and one entropy inverse. Let $\xi(x)$ denote the unique root of $f(x, y) = 0$ in $(x/2, 1/4)$; its existence and uniqueness are proved next. In the notation $g^{-1}$ below, $g$ is restricted to $[0, 1/4]$, where it increases from 0 to $3/2$.

**Lemma F.2.** *For every $0 < x < 1/2$ and $d \geq 0$, the minimum*

$$\psi(x, d) = \min_{x/2 \leq y \leq (1-x)/2} \left\{\ell(x, y) + [g(y) - d]_+\right\}$$

*is attained at*

$$y = \max\{\xi(x), g^{-1}(\min\{d, 3/2\})\}. \tag{69}$$

*Proof.* The identities

$$\ell(x, y) = \ell(x, 1/2 - y), \qquad g(y) - g(1/2 - y) = 4y - 1$$

allow us to restrict the minimum to $y \leq 1/4$. On $(x/2, 1/4)$, the derivative of $2g(y) - r(x, y)$ starts at $-\infty$ and equals 2 at $y = 1/4$. Its second derivative has the sign of

$$2x(1 - x) - (1 + 2x)(2y)(1 - 2y),$$

which changes at most once from positive to negative on this interval. The first derivative therefore crosses zero exactly once. Its sign is the sign of $f(x, y)$, proving the existence and uniqueness of $\xi(x)$.

Where $g(y) \leq d$, the objective is $\ell(x, y)$, which decreases up to $1/4$. Where $g(y) \geq d$, the objective is $2g(y) - r(x, y) - d$, which decreases up to $\xi(x)$ and then increases. If $d \geq 3/2$, the first case covers the entire interval and the minimum is at $1/4$. Otherwise their meeting point is $g^{-1}(d)$. In either case the minimizing coordinate proportion is (69). □

We will also use the monotonicity of $\psi$: it is nondecreasing in its first argument and nonincreasing in its second. Increasing $x$ shrinks the allowed interval and increases $\ell(x, y)$ for each remaining $y$; increasing $d$ decreases the positive-part term. The next two subsections certify that the expression in (68) is at least 0.26675186 on (67), first on a small rectangle and then on the remaining domain. Since this is below 0.267, the all-depth bound then gives the same lower bound for every permitted fixed depth and hence for the complete 5/3-CVP time objective.

### F.4 A lower bound on a small rectangle

We first prove the lower bound on a small rectangle around the selected coordinate proportions; the next subsection treats the rest of the domain. For each pair in this rectangle, choose a third coordinate proportion minimizing its combination cost. We combine three expressions bounded above by the two relevant costs, using positive coefficients that sum to one. The resulting function is a lower bound for the larger cost. We show that it exceeds the required threshold throughout a box containing all these choices.

**Lemma F.3.** *Let $\alpha \in (0, 1/3)$ be the root of $g(\alpha) = 1$. For every*

$$(x, y) \in [0.098674, 0.099674] \times [0.066409, 0.067409], \tag{70}$$

*we have*

$$\max\{\ell(x, y) + [g(y) - r(\alpha, x)]_+,\ \psi(y, r(x, y))\} > 0.26675186.$$

*Proof.* Fix $(x, y)$ in the stated rectangle, and choose $z$ attaining $\psi(y, r(x, y))$. Then the expression to be bounded is

$$\max\{\ell(x, y) + [g(y) - r(\alpha, x)]_+,\ \ell(y, z) + [g(z) - r(x, y)]_+\}.$$

The first term is at least $2g(y) - r(x, y) - r(\alpha, x)$. The second is at least each of $g(z) - r(y, z)$ and $2g(z) - r(y, z) - r(x, y)$. Consequently, any convex combination of these three expressions is a lower bound for the displayed maximum. Use the exact rational coefficients

$$\lambda_1 = 0.415123714084, \qquad \lambda_2 = 0.513121169111,$$

and define

$$\begin{aligned}\varphi(x, y, z) = \lambda_1&\big(2g(y) - r(x, y) - r(\alpha, x)\big)\\ &+ (1 - \lambda_1 - \lambda_2)\big(g(z) - r(y, z)\big)\\ &+ \lambda_2\big(2g(z) - r(y, z) - r(x, y)\big).\end{aligned}$$

The coefficients $\lambda_1$, $1 - \lambda_1 - \lambda_2$, and $\lambda_2$ are positive and sum to one, so it suffices to bound $\varphi(x, y, z)$ from below.

**An interval for a minimizing third coordinate proportion.** Take the rational center

$$(\bar{x}, \bar{y}, \bar{z}) = (0.09917396755431343, 0.06690925652273322, 0.03531462090790641)$$

and the box

$$\mathcal{W} = \{(x, y, z) : \max\{|x - \bar{x}|, |y - \bar{y}|, |z - \bar{z}|\} \le 0.001\}.$$

The rectangle in (70) lies within the first two coordinate intervals of this box. Put $z_- = \bar{z} - 0.001$ and $z_+ = \bar{z} + 0.001$. The minimizing-coordinate formula (69) permits the choice

$$z = \max\{\xi(y), g^{-1}(\min\{r(x,y), 3/2\})\},$$

where the inverse is taken on $[0, 1/4]$. On the stated rectangle, $r(x,y)$ decreases with $x$ and increases with $y$. Rational interval evaluation gives

$$r(0.099674, 0.066409) - g(z_-) > 0.002,$$

$$g(z_+) - r(0.098674, 0.067409) > 0.002.$$

Since $g$ increases on $[0, 1/4]$, these inequalities place the inverse term strictly between $z_-$ and $z_+$. Moreover, $f(y, z_+)$ decreases with $y$ in this interval, and

$$f(0.067409, z_+) > 0.001.$$

The unique zero $\xi(y)$ of $f(y, \cdot)$ is therefore below $z_+$. Thus a minimizing $z$ belongs to $[z_-, z_+]$, and the chosen triple belongs to $\mathcal{W}$.

**Certifying the lower bound throughout the box.** We first prove convexity of $\varphi$ on $\mathcal{W}$ by bounding its Hessian from below by a positive-definite matrix. Define the column vectors

$$\boldsymbol{u}_2 = (-1/2, 1, 0)^\top, \quad \boldsymbol{v}_2 = (-1, -2, 0)^\top,$$
$$\boldsymbol{u}_3 = (0, -1/2, 1)^\top, \quad \boldsymbol{v}_3 = (0, -1, -2)^\top.$$

Direct differentiation gives

$$\begin{aligned}(\ln 2)\nabla^2\varphi = \operatorname{diag}(\zeta_1, \zeta_2, \zeta_3) &+ \frac{2(\lambda_1+\lambda_2)}{y - x/2}\boldsymbol{u}_2\boldsymbol{u}_2^\top + \frac{\lambda_1+\lambda_2}{1-x-2y}\boldsymbol{v}_2\boldsymbol{v}_2^\top \\ &+ \frac{2(1-\lambda_1)}{z - y/2}\boldsymbol{u}_3\boldsymbol{u}_3^\top + \frac{1-\lambda_1}{1-y-2z}\boldsymbol{v}_3\boldsymbol{v}_3^\top,\end{aligned}$$

where

$$\begin{aligned}\zeta_1 &= \lambda_1\left(\frac{2}{x - \alpha/2} + \frac{4}{1-\alpha-2x}\right) - \frac{4(\lambda_1+\lambda_2)}{1-2x}, \\ \zeta_2 &= -\frac{4\lambda_1}{y(1-2y)} - \frac{4(1-\lambda_1)}{1-2y}, \\ \zeta_3 &= -\frac{2(1-\lambda_1+\lambda_2)}{z(1-2z)}.\end{aligned}$$

Every denominator is positive on $\mathcal{W}$. Rational interval evaluation gives the following lower bounds on the coefficients:

| Coefficient | Lower bound |
|---|---|
| $\zeta_1$ | 16.8926 |
| $\zeta_2$ | −31.8604 |
| $\zeta_3$ | −69.0079 |
| $2(\lambda_1+\lambda_2)/(y-x/2)$ | 98.6326 |
| $(\lambda_1+\lambda_2)/(1-x-2y)$ | 1.2055 |
| $2(1-\lambda_1)/(z-y/2)$ | 348.1414 |
| $(1-\lambda_1)/(1-y-2z)$ | 0.6757 |

Replacing each coefficient by its lower bound gives the matrix

$$\mathbf{M} = \begin{pmatrix} 42.75625 & -46.9053 & 0 \\ -46.9053 & 159.30525 & -172.7193 \\ 0 & -172.7193 & 281.8363 \end{pmatrix}.$$

The difference $(\ln 2)\nabla^2\varphi - \mathbf{M}$ is positive semidefinite: each lowered diagonal coefficient multiplies a coordinate projection, and each remaining coefficient multiplies a positive-semidefinite rank-one matrix. The successive pivots of $\mathbf{M}$ in symmetric elimination exceed 42.756, 107.848, and 5.225, respectively. Hence $\mathbf{M}$ is positive definite, and $\varphi$ is convex on $\mathcal{W}$.

At the rational center, interval evaluation gives

$$\varphi(\bar{x}, \bar{y}, \bar{z}) > 0.266751868028863654$$

and

$$|\varphi_x| < 4.1 \cdot 10^{-14}, \qquad |\varphi_y| < 3.73958224 \cdot 10^{-8}, \qquad |\varphi_z| < 2.25793 \cdot 10^{-12},$$

where the subscripts denote partial derivatives evaluated at that center. The supporting-plane inequality therefore gives, at every point of $\mathcal{W}$,

$$\begin{aligned} \varphi(x, y, z) &\geq \varphi(\bar{x}, \bar{y}, \bar{z}) - 0.001\,(|\varphi_x| + |\varphi_y| + |\varphi_z|) \\ &> 0.2667518679914655 > 0.26675186. \end{aligned}$$

This applies to the minimizing choice of $z$ already placed in $\mathcal{W}$. Since the two-variable maximum is at least $\varphi(x, y, z)$, the claimed lower bound follows. □

### F.5 Verifying the remaining rectangles

We complete the lower bound outside the rectangle covered by the local argument. The verification divides a fixed rational rectangle into smaller rectangles. Each is either outside the allowed coordinate proportion range, covered by the local argument, or certified directly by a lower bound on its entire allowed part.

Enlarge the domain in (67) to

$$\mathcal{R} = [5/64, 9/64] \times [5/128, 9/64],$$

and write $\mathcal{Q}$ for the rectangle in (70). Consider a rectangle $[x_-, x_+] \times [y_-, y_+] \subseteq \mathcal{R}$. It contains no allowed pair when $y_+ < x_-/2$. Otherwise, put $\widehat{y} = \max\{y_-, x_-/2\}$ for evaluating its lower bound. This does not change the rectangle recorded in the partition. On the allowed part of $\mathcal{R}$, $g$ increases, while $r$ decreases in its first argument and increases in its second. With $\alpha_- = 0.11354609760967409$, the reduced objective is therefore at least

$$\max\left\{g(\widehat{y}) - r(x_-, y_+) + [g(\widehat{y}) - r(\alpha_-, x_+)]_+,\ \psi(\widehat{y}, r(x_-, y_+))\right\}. \tag{71}$$

Indeed, $\alpha_- < \alpha$, and increasing the second argument of $\psi$ or decreasing its first argument can only decrease its value. Every occurrence of $r$ in this bound is enclosed from above. For a prescribed rational threshold $\vartheta$, first evaluate the first term of the maximum. If its lower endpoint exceeds $\vartheta$, no evaluation of the second term is needed. Otherwise use the following procedure.

To bound $\psi(u, d)$, take $d$ to be the rational upper endpoint of the enclosure for $r(x_-, y_+)$ and set $u = \widehat{y}$. The evaluations below have $0 < d < 3/2$. Bisection gives candidate rational intervals of width at most $2^{-38}$ containing $\xi(u)$ and $g^{-1}(d)$, respectively. The endpoints for $\xi(u)$ are checked by

opposite signs of $f(u,\cdot)$; the endpoints for $g^{-1}(d)$ are checked by entropy enclosures on opposite sides of $d$. The intervals are accepted only after these sign checks. Their endpointwise maximum gives an interval $[z_-, z_+]$ containing the minimizing child in (69). Check that $u/2 \le z_- \le z_+ \le 1/4$. Since $\ell(u,\cdot)$ decreases and $g$ increases on this interval,

$$\psi(u,d) \ge \ell(u, z_+) + [g(z_-) - d]_+.$$

Together with outward evaluation of (71), this supplies a rational lower bound for each rectangle. If these brackets or their containment in $[u/2, 1/4]$ cannot be verified, the verification reports failure.

**Algorithm 14** Verification of a two-variable lower bound

**Input:** A rational rectangle $\mathcal{R} \subseteq [5/64, 9/64] \times [5/128, 9/64]$, a rational threshold $\vartheta$, and a rectangle $\mathcal{Q}$ where the objective is already known to exceed $\vartheta$.

**Output:** If verification completes, a partition into rectangles certified as infeasible, directly bounded, or contained in $\mathcal{Q}$, or a report that a root bracket failed verification.

Put $\mathcal{R}$ on a stack; start with an empty partition.
**while** the stack is nonempty **do**
  Remove $\mathcal{D} = [x_-, x_+] \times [y_-, y_+]$ from the stack.
  **if** $y_+ < x_-/2$ **then**
    Record $\mathcal{D}$ as infeasible; **continue**.
  **end if**
  Let $v$ be a rational lower bound for the first term in (71).
  **if** $v \le \vartheta$ **then**
    Bound the second term using the verified root brackets.
    **if** a required bracket fails verification **then return** failure.
    Replace $v$ by the larger of the two lower bounds.
  **end if**
  **if** $v > \vartheta$ **then**
    Record $\mathcal{D}$ as directly bounded; **continue**.
  **end if**
  **if** $\mathcal{D} \subseteq \mathcal{Q}$ **then**
    Record $\mathcal{D}$ as contained in $\mathcal{Q}$; **continue**.
  **end if**
  Bisect the longer side at its exact rational midpoint; bisect the $x$-side in a tie.
  Push both children, with the left or lower child processed first.
**end while**
**return** the recorded partition.

No part of a rectangle is removed when forming its children. Thus the recorded rectangles retain an exact partition of the input rectangle, including those with no allowed pair.

**Lemma F.4.** *Let $\alpha$ be the root of $g(\alpha) = 1$ in $(0, 1/3)$. For every*

$$(x,y) \in [5/64, 9/64] \times [5/128, 9/64], \qquad y \ge x/2,$$

*one has*

$$\max\{\ell(x,y) + [g(y) - r(\alpha,x)]_+,\ \psi(y, r(x,y))\} > 0.26675186.$$

*Proof.* Apply Algorithm 14 with the displayed $\mathcal{R}$, threshold $\vartheta = 0.26675186$, and the rectangle $\mathcal{Q}$ from (70). The outward interval calculations specified below terminate after 17053 midpoint subdivisions. The resulting 17054 rectangles have the following disjoint classifications:

| | |
|---|---|
| no allowed pair | 1 |
| direct lower bound greater than 0.2667518635 | 17004 |
| contained in $\mathcal{Q}$ | 49. |

All endpoints have denominators dividing $2^{24}$. An exact rational sweep checks that the interiors have multiplicity one and that the rectangles cover $\mathcal{R}$; their total area is $13/2048$. Every allowed pair therefore lies either in a directly bounded rectangle or in $\mathcal{Q}$. The latter case follows from Lemma F.3. □

By Lemmas F.1 and F.4, the short-solution time exponent is at least 0.26675186 at every fixed recursion depth. The complete 5/3-CVP time exponent is at least this short-solution exponent. The selected rational parameters in Appendix F.2 give a complete 5/3-CVP time exponent below 0.26675191. These conclusions establish (64).

### F.6 Bounds used in the numerical verification

We provide the elementary-function bounds used in the selected parameter checks, the global lower bound, and the density-dependent cost certificates. They reduce every strict numerical comparison to rational arithmetic. In particular, the unknown value of $\alpha$ is always enclosed by (65); it is never replaced by a rounded point.

For a positive rational $x$, write $x = 2^s u$ with an integer $s$ and $1 \le u \le 2$, and put $z = (u-1)/(u+1)$. For an integer $N \ge 1$,

$$2\sum_{j=0}^{N-1} \frac{z^{2j+1}}{2j+1} \le \ln u \le 2\sum_{j=0}^{N-1} \frac{z^{2j+1}}{2j+1} + \frac{2z^{2N+1}}{(2N+1)(1-z^2)}.$$

The same formula at $u = 2$ encloses $\ln 2$. Using $\ln x = s\ln 2 + \ln u$ and dividing by the positive enclosure for $\ln 2$ gives bounds on $\log x$. Substitution gives the entropy bounds, with a zero summand evaluated exactly as $0 \log 0 = 0$.

For a rational $0 \le x < 66$,

$$\sum_{j=0}^{64} \frac{x^j}{j!} \le e^x \le \sum_{j=0}^{64} \frac{x^j}{j!} + \frac{x^{65}/65!}{1 - x/66}.$$

To bound $\pi$, use $\pi = 16\arctan(1/5) - 4\arctan(1/239)$, retaining 80 and 20 terms, respectively, and enclosing each alternating-series remainder between zero and its first omitted term. These bounds give the evaluations of $\tanh(\pi^2/(2a))$ and $\chi(a)$ in the exact-search conditions.

For the theta sum, enclose $x = e^{-a}$ by reciprocating the positive enclosure for $e^a$. Then

$$1 + 2\sum_{j=1}^{5} x^{j^2} \le \Theta(a) \le 1 + 2\sum_{j=1}^{5} x^{j^2} + \frac{2x^{36}}{1 - x^{13}}.$$

The remainder follows from $j^2 \ge 36 + 13(j-6)$ for every integer $j \ge 6$.

After each arithmetic operation, round the lower endpoint downward and the upper endpoint upward to the indicated rational grid:

| verification | logarithm terms | grid spacing |
|---|---|---|
| selected parameters and local convexity | 64 | $10^{-50}$ |
| rectangle and density-interval bounds | 40 | $10^{-35}$. |

The analytic remainder bounds and outward rounding account for truncation and arithmetic errors. Exact rational midpoint subdivision and coverage checks require no rounding. These rules verify the strict inequalities used in the parameter, local, and rectangle calculations; displayed decimal rounding is not used to decide their signs.

### F.7 Verifying the parameters above density one

For $q \geq 3$ and $1 < \delta \leq 3/2$, the correction branch in Algorithm 3 uses $d = n$ and $n_0 = n - O(n^{2/3})$, so its limiting equation rate is $\delta$. We therefore set $d_0 = \rho = \delta$ in (9), with the entropy and decomposition rates from Appendix F.1. The required success conditions are those of Corollary 3.3 at this rate, including $h_0 > \delta$. The binary case uses the kernel procedure in Algorithm 3; the binary-sampler restrictions (11) do not apply here.

The following certificate verifies the continuous parameter choice in (13) and its cost upper bounds on the closed interval $[1, 3/2]$. It does not optimize those bounds.

**Lemma F.5.** *For $1 \leq \delta \leq 3/2$, let $u_\delta \in (0, 1/3)$ satisfy $\mathcal{H}(u_\delta, u_\delta, 1 - 2u_\delta) = \delta$. Set*

$$\rho = \delta, \qquad \eta = 10^{-8}, \qquad (t_0, t_1, t_2, t_3) = (u_\delta + 10^{-7} + \theta)(1, 0.7, 0.47, 0.24), \qquad 0 \leq \theta \leq 10^{-12}.$$

*These parameters satisfy all strict coordinate, nested-equation, and even-modulus conditions of the common construction. They also satisfy the uniform entropy surplus*

$$\mathcal{H}(t_0, t_0, 1 - 2t_0) - \rho > 1.99 \cdot 10^{-7}.$$

*Let $\beta = 2t_0$, $c = \beta + 2 \cdot 10^{-5}$, and take $a = 4.138$, $\gamma = 5/3$, and the parameter $\nu = 16$ of Section 3.1. The shifted-theta condition (22) holds. With these perturbed proportions and radius, the complete time and space exponents in (32) have the following strict upper bounds, respectively:*

$$0.559968479824129 < 0.56, \qquad 0.324999937051930 < 0.325.$$

*These bounds hold throughout the density interval.*

*Proof.* For the entropy function $g$ defined in Appendix F.1,

$$g'(u) = 2 \log \frac{1 - 2u}{u} > 0 \quad (0 < u < 1/3).$$

This derivative is strictly decreasing. Rational bisection with certified entropy comparisons provides an inverse bracket at density one whose left endpoint is

$$L = \frac{13405157140314251972633}{11805916207174113034240 0}.$$

At density $3/2$, the root is exactly $1/4$. Set $U = 1/4 + 10^{-24}$; the entropy comparisons verify $g(L) < 1$ and $g(U) > 3/2$. Thus every required root belongs to the rational interval $[L, U]$.

The two nearby entropy terms must retain their dependence on the same root. By the mean value theorem, for every $u \in [L, U]$,

$$g(u + 10^{-7} + \theta) - g(u) \geq 10^{-7} g'(U + 10^{-7} + 10^{-12}) > 1.99999769 \cdot 10^{-7}.$$

The final strict inequality is verified by outward logarithm intervals. This proves the root-surplus condition uniformly, including at the endpoints.

The representation conditions and costs use the interval arithmetic of Appendix F.6, with 40 logarithm terms and outward rounding to multiples of $10^{-35}$. Enclose $\rho = g(u)$ and $t_i = c_i(u + 10^{-7} + [0, 10^{-12}])$ on each rational subinterval. Evaluate all coordinate, equation, and parity margins and the representation costs by outward arithmetic. Evaluate the exact-search cost in (32) at $a = 4.138$, $\gamma = 5/3$, and $c = 2t_0 + 2 \cdot 10^{-5}$. The certificate in Appendix F.2 verifies the shifted-theta condition independently of $c$; its positive weight margin therefore remains valid for every correction radius here. The fixed theta and logarithm values use the selected-parameter enclosures in Appendix F.6, with 64 logarithm terms and grid spacing $10^{-50}$, and are then rounded outward to multiples of $10^{-35}$.

Starting with $[L, U]$, bisect at rational midpoints. Accept an interval only after all margins are strictly positive and the complete time and space costs in (32) are below 0.56 and 0.325, respectively; otherwise subdivide it. The completed calculation visits 829 intervals and accepts 415. An exact endpoint-adjacency check verifies that their union is precisely $[L, U]$. The largest outward upper endpoints are below 0.559968479824129 and 0.324999937051930, respectively. For example, the smallest added-sign margin exceeds $6.66 \cdot 10^{-5}$, the smallest nested-equation gap exceeds 0.1143, the root parity margin exceeds 0.2495, and all remaining coordinate and parity margins are also positive. The equation margin is $10^{-8}$. Finally, $0 < \beta < 1$ and $c - \beta = 2 \cdot 10^{-5}$ uniformly. These checks prove the lemma. The strict margins and continuity give an open neighborhood of the certified curve, as required to replace the limiting rate $\delta$ by the implemented $\rho_n = \delta + O(n^{-1/3})$ in Corollary 4.11. □

The full certificate records the exact rational interval, all smallest margins, the root-surplus enclosure, and the largest cost upper endpoints. At any rational density $\delta \in [0, 3/2]$, time and space enclosures follow by selecting the corresponding proved density regime. At $\delta = 1$, we retain the better parameters of Section 4.4; above one, we use the continuous profile just certified. For binary matrices above one, the leading exponents are zero, corresponding to the subexponential kernel bound proved in Section 4.3. The curves in Figs. 4 and 5 are sampled visualizations of these formulas. None of these certificates asserts optimality over parameter choices or extends the nonbinary construction beyond $\delta = 3/2$.

# G Verification and Optimization of the SVP Parameters

This appendix verifies the SVP parameters after accounting for every part of the algorithm. On $0 \leq \delta \leq 1$, we allow a different choice for each input density and include every fixed number of decomposition steps. The selected rule has time $2^{0.27400949n+o(n)}$ and space $2^{0.261392024n+o(n)}$. Its time exponent is within $9 \cdot 10^{-8}$ of the global infimum of the proved complete cost. The lower bound concerns this parameterized construction, not arbitrary SVP algorithms. The final subsection verifies feasible time and space upper bounds for $1 < \delta \leq 1.5$ using the same algorithm with different coordinate proportions; it makes no optimality claim on that larger range.

The upper bounds used by the SVP theorems are verified in Lemma G.1 and Appendix G.5. The lower-bound certificate additionally establishes near-optimality within the stated parameter family. The accompanying archive `numerical-certificates.zip` contains the verification programs, recorded outputs, and execution instructions in `scripts/README.md`.

## G.1 The complete cost and its constraints

We specify the full objective and all parameter constraints before selecting their values. The auxiliary dimension affects both the cost of solving a congruence and the number of coordinate subsets that must be tried; both contributions belong in the optimization.

For $a > 0$, let $p_a$ and $r_a$ be the distributions of (49) in Appendix A. The counted radius determines $a$ with $\mathcal{H}(p_a) = \delta + o(1)$. The entropy $\mathcal{H}(r_a)$ bounds the cost of enumerating coordinates whose absolute values are at least two.

Fix a depth $k \geq 1$, proportions $0 < t_i < 1/2$ for $0 \leq i \leq k$, an equation margin $\eta > 0$, and an equation rate $\rho > 0$. The auxiliary dimension satisfies $n_0 = (\delta/\rho)n + o(n)$. Let $h_i$ and $r_i$ be the rates (4), put $d_0 = \rho$ and $d_i = r_i - \eta$ for $1 \leq i \leq k$, and let $T_{\mathrm{in}}$ and $S_{\mathrm{in}}$ be the exponents (9); they measure the time and space of a complete congruence enumeration per auxiliary coordinate. The conditions of Lemma 3.2, including for even moduli, are (3), (5), and (8). There is no constraint $h_0 > \rho$: that surplus is needed for the average-success bound in Corollary 3.3, whereas SVP uses the bound on the probability of omitting a fixed vector.

Let $G(\alpha, w)$ be the repetition-cost bound (44); it vanishes when $\alpha + w \geq 1$, because all remaining positions can then be used. With a coordinate tolerance $\varepsilon > 0$, the complete time exponent is (45). Write $T(a)$ for the expression inside its supremum; its middle term, the maximum of $G((1 - 2t_0)\mathcal{H}(p_a)/\rho, w)$ over $|w - 2p_a(1)| \leq 2\varepsilon$, is called the subset term. The added coordinates require the padding condition

$$\frac{t_0 \mathcal{H}(p_a)}{\rho} > p_a(1) + \varepsilon, \tag{72}$$

the limiting form of $a_0 \geq N_+, N_-$ in (38). If exact search is used below the cutoff $\delta_0 \in (1/2, 1)$, the complete space exponent is

$$\max\left\{\frac{\delta_0}{4}, \sup_{\delta_0 \leq \mathcal{H}(p_a) \leq 1} \frac{S_{\mathrm{in}} \mathcal{H}(p_a)}{\rho}\right\}. \tag{73}$$

The proportions and $\rho$ may depend on $a$; they are chosen from the dimensions and counted radius, independently of the random matrix. The depth is fixed independently of the dimension. Strict conditions must have uniform positive margins over the chosen parameter range so that the asymptotic success bounds are uniform. The integer $\nu \geq 1$ in (7) may be increased to satisfy $\nu(\mathcal{H}(p_a)/\rho)^{2/3} > 1$ uniformly, without changing these leading time and space exponents.

## G.2 A parameter rule and its uniform guarantees

We give one rule that retains all remaining coordinates and verify its feasibility and total cost on the entire input range. We then explain why the fixed-vector success proof remains uniform when the parameters vary with the input.

Use the rational constants

$$\begin{gathered} \delta_0 = 0.54, \qquad k = 3, \qquad \nu = 16, \qquad \kappa = 10^{-8}, \qquad \lambda = 10^{-9}, \qquad \varepsilon = \eta = 10^{-10}, \\ c_0 = 1, \qquad c_1 = 0.874597414466, \qquad c_2 = 0.589775545104, \qquad c_3 = 0.310623936690. \end{gathered} \tag{74}$$

Above the exact-search cutoff, set

$$n_0 = \left\lceil \frac{n}{1 - \kappa} \right\rceil, \qquad t_i = c_i(\widehat{p} + \lambda) \quad (0 \leq i \leq 3), \tag{75}$$

where $\widehat{p}$ is a rational upper approximation satisfying

$$0 \leq \widehat{p} - p_a(1) \leq \min\{n^{-2}, 10^{-12}\}.$$

All coordinate proportions passed to the integer construction are rational. The actual equation rate is

$$\rho = \frac{\delta n}{n_0} = (1-\kappa)\mathcal{H}(p_a) + o(1).$$

The implementation need not represent this rate exactly: it chooses the integer dimension and the numbers of retained equations directly. The radius is computed by the integer-point recurrence in Lemma 5.1. The subsequent theta evaluations compute $\widehat{p}$ with polynomially many bits, as described in Section E.6.1.

**Using every remaining position.** Let $N_+, N_-$ be a pair of sign counts considered by the algorithm. The outward comparisons in Section E.6.2 give $|N_\pm - np_a(1)| \le \varepsilon n + 2$. The root has $a_0 = 8\lfloor t_0 n_0/8 \rfloor$ entries of each sign. After division by $n$, the constant enlargement and count rounding contribute only $o(1)$, and the rounding $0 \le \widehat{p} - p_a(1) \le 10^{-12}$ only increases $a_0$. The two padding conditions (39), with $\lambda$ replaced by $\lambda + 10^{-12}$ in the second, give uniformly

$$\frac{a_0 - N_\pm}{n} \ge \frac{\lambda + \kappa p_a(1)}{1-\kappa} - \varepsilon + o(1) > 9 \cdot 10^{-10} + o(1),$$

so the prescribed added-coordinate counts are nonnegative for all sufficiently large $n$, and, since at most $n$ positions remain after removing the large coordinates and the selected unit coordinate,

$$\frac{n_0 - 2a_0 + N_+ + N_- - n}{n} \ge \frac{\kappa(1-2p_a(1)) - 2(\lambda + 10^{-12})}{1-\kappa} - 2\varepsilon + o(1).$$

The numerical enclosure below gives $p_a(1) < 0.11202$ on the selected range and hence the lower bound $5.55 \cdot 10^{-9}$ for the constant term. Consequently $n_0 - 2a_0 + N_+ + N_- \ge n$. Every remaining position can be retained, and the subset term in $T(a)$ is zero throughout the allowed count window. The limiting time and space bounds above the cutoff reduce to

$$\mathcal{H}(r_a) + \varepsilon + \frac{T_{\mathrm{in}}(a)}{1-\kappa}, \qquad \frac{S_{\mathrm{in}}(a)}{1-\kappa}. \tag{76}$$

Here $T_{\mathrm{in}}(a), S_{\mathrm{in}}(a)$ are evaluated with $t_i = c_i(p_a(1) + \lambda)$ and $\rho = (1-\kappa)\mathcal{H}(p_a)$. The factor $1/(1-\kappa)$ arises from $n_0/n = 1/(1-\kappa) + O(1/n)$, since $T_{\mathrm{in}}(a)$ and $S_{\mathrm{in}}(a)$ are measured per coordinate of the auxiliary instance.

**Lemma G.1.** *Let $a > 0$ satisfy $0.54 \le \mathcal{H}(p_a) \le 1$, where $p_a$ and $r_a$ are the distributions of (49). Set $\rho = (1-10^{-8})\mathcal{H}(p_a)$ and $\eta = \varepsilon = 10^{-10}$. For any $0 \le \theta \le 10^{-12}$, set*

$$(t_0, t_1, t_2, t_3) = (p_a(1) + 10^{-9} + \theta)(1, 0.874597414466, 0.589775545104, 0.310623936690),$$

*and let $h_i$ and $r_i$ be the rates (4) with $k = 3$, $d_0 = \rho$, $d_i = r_i - \eta$, and $T_{\mathrm{in}}$ and $S_{\mathrm{in}}$ the exponents (9). Then*

$$\frac{t_{i-1}}{2} < t_i < \frac{1-t_{i-1}}{2}, \qquad \rho > d_1 > d_2 > d_3 > 0, \qquad h(2t_0) > \rho/2,$$
$$h(2t_i) > d_{i-1}/2, \qquad r_i > d_i/2 + 2t_i - t_{i-1} \quad (1 \le i \le 3),$$

*and*

$$\mathcal{H}(r_a) + 10^{-10} + \frac{T_{\mathrm{in}}}{1-10^{-8}} < 0.274009489691249, \qquad \frac{S_{\mathrm{in}}}{1-10^{-8}} < 0.261392023934394.$$

*Proof.* Put $x = e^{-a}$. The interval with rational endpoints

$$\frac{48470255300856487442837}{94447329657392904273920 0} \le x \le \frac{13645599416629146239204 9}{94447329657392904273920 0} \tag{77}$$

contains the full entropy range in the statement. Its endpoints are approximately 0.05132 and 0.14448. Their entropies are strictly below 0.54 and above 1, respectively, as certified by rational intervals. Monotonicity of $x \mapsto \mathcal{H}(p_{-\ln x})$ follows from Lemma A.2(4). The quantity $p_{-\ln x}(1)$ is increasing on this interval too: the numerator of its derivative is $1 - 2\sum_{j\ge 2}(j^2-1)x^{j^2} > 0$ for $0 < x \le 1/4$.

Starting with this interval, bisect at exact rational midpoints. On each interval, enclose $\mathcal{H}(p_a)$, $\mathcal{H}(r_a)$, and $p_a(1)$, allow the full error $0 \le \theta \le 10^{-12}$, and evaluate every constraint and both costs by outward rational arithmetic. Accept an interval only after proving positive margins, time below 0.27400949, and space below 0.26139204; otherwise subdivide it. The completed calculation visits 2139 intervals and accepts 1070 intervals whose union is exactly (77). The largest certified costs are less than 0.274009489691249 and 0.261392023934394. The smallest margins, rounded downward, are

| Quantity | Positive lower bound |
|---|---|
| $\min_{0\le i\le 3} t_i$ | 0.0144 |
| $\min_{0\le i\le 3}(1/2 - t_i)$ | 0.3879 |
| $\min_{1\le i\le 3}(t_i - t_{i-1}/2)$ | 0.00030 |
| $\min_{1\le i\le 3}(1 - t_{i-1} - 2t_i)$ | 0.6920 |
| $d_3$ | 0.0600 |
| $\min_{1\le i\le 3}(d_{i-1} - d_i)$ | 0.0970 |
| $h(2t_0) - \rho/2$ | 0.1637 |
| $\min_{1\le i\le 3}(h(2t_i) - d_{i-1}/2)$ | 0.0760 |
| $\min_{1\le i\le 3}(r_i - d_i/2 - 2t_i + t_{i-1})$ | 0.0133 |

The equation margin is $\eta = 10^{-10}$. The same calculation gives padding margins greater than $9 \cdot 10^{-10}$ and $5.55 \cdot 10^{-9}$ for the two conditions (39). The numerical enclosures for this certificate are specified in Appendix G.4. □

**Uniformity of the probability bound for a fixed vector.** By Lemma 5.4, $\mathcal{H}(p_a) = \delta + o(1)$ uniformly. Moreover, $\widehat{p} - p_a(1) = O(n^{-2})$ and $\rho = \delta n/n_0 = (1-\kappa)\mathcal{H}(p_a) + o(1)$. By continuity and the uniform positive margins established in Lemma G.1, the actual parameter tuples lie, for all sufficiently large $n$, in a fixed compact set on which the strict conditions of Section 3.1.1 hold.

The parameters are computed deterministically in polynomial time from $n, m, q$ alone. Hence the uniform version of Lemma 3.2 bounds the probability of omitting a fixed vector by $2^{-(\nu - o(1))n_0^{2/3}}$ for the implemented procedure. Together with the conditions in Eq. (39), verified above, and $\nu(1-\kappa)^{-2/3} > 1$, this establishes the hypotheses of Theorem 5.7.

The exact-search branch has time and space exponents $\delta_0/2 = 0.27$ and $\delta_0/4 = 0.135$, respectively. Continuity of the cost formulas and the bounds in Lemma G.1, combined with Theorem 5.8, give time $2^{0.27400949n + o(n)}$ and space $2^{0.261392024n + o(n)}$.

## G.3 Reducing the lower bound to three variables

We reduce the lower-bound problem to a compact domain of at most three variables. Depths one and two are treated separately; the first three combinations give a lower bound for every larger

depth. A single input density suffices for this argument even when different input dimensions use different parameters.

Fix $x_* = 0.144478403$, set $a_* = -\ln x_*$, and put

$$\delta_* = \mathcal{H}(p_{a_*}), \qquad w_* = 2p_{a_*}(1), \qquad e_* = \mathcal{H}(r_{a_*}), \qquad c_* = 0.11201367, \qquad \vartheta = 0.2740094.$$

Outward rational intervals give

$$0.99 < \delta_* < 1, \qquad \frac{w_*}{2\delta_*} > c_*, \qquad \delta_* > 2\vartheta.$$

These are values of the distribution $p_{a_*}$ of (49); the rational choice of $x_*$ allows direct power-series enclosures without solving an equation for the entropy-one distribution. Even moduli nearest to $2^{\delta_* n}$ realize this limiting density with $m = 1$, so the even-modulus conditions are required here. If the complete exponent were below $\vartheta$, its exact-search cutoff would satisfy $\delta_0 < 2\vartheta < \delta_*$. The completion branch must therefore handle density $\delta_*$. Its padding condition (72) implies $t_0 > c_*\rho$.

Let $g$, $r$, $\ell$, and $\psi$ be the functions of Appendices F.1 and F.3, so that $h_i = g(t_i)$ and $r_i = r(t_{i-1}, t_i)$. By the derivatives computed in the proof of Lemma F.1, $r$ decreases in its first argument and $\ell$ increases in its first argument, and $\ell(x, y) \geq x$ by (66). Decreasing $t_0$ to $c_*\rho$ decreases the first two combination costs, decreases the subset term, and enlarges the permitted interval for $t_1$. Setting $\eta = 0$ can only reduce the cost. We may make both changes for a lower bound, without asserting that the relaxed parameters satisfy the original strict conditions.

The root parity condition in (8) and $h(2t_0) \leq 1$ imply $\rho < 2$. Counting pairs of parts gives $g(x) + r(x, y) \leq 2g(y)$: a parent and one of its decompositions are determined by the ordered pair of parts. Apply this inequality to the relaxed root $c_*\rho$: its first combination costs at least $g(c_*\rho) - \rho$. Thus

$$\frac{T_{\text{in}}}{\rho} \geq \frac{g(c_*\rho)}{\rho} - 1.$$

Concavity of $g$ and $g(0) = 0$ make the right-hand side nonincreasing in $\rho$. An interval evaluation at $\rho = 0.4$ proves

$$e_* + \delta_* \left( \frac{g(0.4c_*)}{0.4} - 1 \right) > \vartheta.$$

Thus a counterexample must lie in $0.4 < \rho < 2$.

Put $x = t_1$ and $y = t_2$. The following lower bounds depend on only the displayed variables:

$$\begin{aligned} T_{\text{in}} &\geq \ell(c_*\rho, x) + [g(x) - \rho]_+ \quad (k \geq 1), \\ T_{\text{in}} &\geq \ell(x, y) + [g(y) - r(c_*\rho, x)]_+ \quad (k \geq 2), \\ T_{\text{in}} &\geq \begin{cases} g(x)/2, & k = 1, \\ g(y)/2, & k = 2, \\ \psi(y, r(x, y)), & k \geq 3. \end{cases} \end{aligned} \tag{78}$$

Every permitted later choice is covered by the last case. Substitute the corresponding lower bound for $T_{\text{in}}$ into

$$e_* + G\left( \frac{\delta_*}{\rho} - 2c_*\delta_*, w_* \right) + \frac{\delta_* T_{\text{in}}}{\rho}.$$

The resulting expression bounds the complete exponent from below. It remains to bound it on $[0.4, 2] \times [0, 1/2]$ for $k = 1$ and on $[0.4, 2] \times [0, 1/2]^2$ for the other cases, subject only to the coordinate inequalities. Nesting and the other parity inequalities have been discarded, which enlarges the domain and preserves a lower bound.

## G.4 Certifying the global lower bound

We explain the inequalities used to exclude every box in this domain. All accepted exclusions use rational intervals. Floating-point calculations only prioritize which box to process next.

**Direct bounds on a box.** The minimum of $g$ on an interval is at an endpoint by concavity. For fixed parent $x$, the minimum of $\ell(x,y)$ is obtained by clamping $1/4$ to the child interval, and the maximum of $r(x,y)$ by clamping $(2-x)/6$ to that interval; both follow from the derivatives in the proof of Lemma F.1. Increasing the parent decreases $r$ and increases $\ell$. These facts give independent rigorous lower bounds for the first two bounds in (78) on every box.

The function $\psi$ is nondecreasing in its first argument, because $\ell$ increases in its first argument and the range of the minimum shrinks, and nonincreasing in its second. By Lemma F.2, the minimum $\psi(y, r(x,y))$ is attained at $z = \max\{\xi(y), g^{-1}(\min\{r(x,y), 3/2\})\}$, with the root $\xi$ of Appendix F.3 and the inverse of $g$ on $[0, 1/4]$. Rational root and inverse-entropy brackets provide a lower enclosure. The elementary bound $\psi(y,d) \geq y$, from (66), handles boundary cases as well.

The repetition-cost bound $G(\alpha, w)$ is nonincreasing in $\alpha$. For fixed $\alpha$, it is concave in $w$ before its zero region. Its lower bound on an interval is thus obtained from endpoints, or from zero if that interval meets the zero region. Combining these bounds with the lower endpoint of $\delta_*$ and the upper endpoint of $\rho$ bounds the normalized cost.

**A local convex lower bound.** Direct interval bounds converge slowly near the minimum. We instead use a convex lower function on a small neighborhood, allowing every box inside it to be excluded at once. Write

$$(\bar{x}, \bar{y}, \bar{z}) = (0.09796686677105, 0.06606292369075, 0.03479412735314).$$

Consider $\rho \in [0.995, 1.005]$ and $|x - \bar{x}|, |y - \bar{y}| \leq 0.0004$. The root and inverse-entropy sign tests above put a minimizing $z$ inside $|z - \bar{z}| \leq 0.001$. Put

$$\begin{aligned}\varphi(\rho, x, y, z) = {} & 0.417633157509395\,\{2g(y) - r(c_*\rho, x) - r(x,y)\} \\ & + 0.066847676684586\,\{g(z) - r(y,z)\} \\ & + 0.515519165806019\,\{2g(z) - r(x,y) - r(y,z)\}.\end{aligned}$$

The three coefficients are positive and sum to one, and each brace is bounded by one of the second and third combination costs. Therefore $\varphi \leq T_{\mathrm{in}}$ for a minimizing $z$. Interval arithmetic also proves $\varphi > 0$ throughout the larger box $|x - \bar{x}|, |y - \bar{y}|, |z - \bar{z}| \leq 0.001$ and $\rho \in [0.995, 1.005]$; hence $\delta_*\varphi \geq \delta_-\varphi$ for a lower enclosure $\delta_-$ of $\delta_*$.

Convexity in the first argument of $G$ supplies the supporting bound

$$\rho\, G\left(\frac{\delta_*}{\rho} - 2c_*\delta_*, w_*\right) \geq -(1 - w_* + 2c_*\delta_*)\log(1 - w_*)\,[\rho - \rho_*]_+, \qquad \rho_* = \frac{\delta_*}{1 - w_* + 2c_*\delta_*}.$$

Indeed $G = 0$ at $\alpha = 1 - w_*$, and its derivative on the nonzero side is $\log(1 - w_*)$. Let $s_-$ be a rational lower enclosure of the slope $-(1 - w_* + 2c_*\delta_*)\log(1 - w_*)$, $\rho_+$ a rational upper enclosure of $\rho_*$, and $e_-$ a lower enclosure of $e_*$. The derivative with respect to $\rho$ of

$$\delta_-\varphi(\rho, x, y, z) + \rho(e_- - \vartheta)$$

is strictly negative throughout the box, and becomes strictly positive after $s_-$ is added. Thus the sum with the supporting term $s_-[\rho - \rho_+]_+$ is minimized at $\rho = \rho_+$.

At $\rho = \rho_+$, the Hessian of $\varphi$ in $(x, y, z)$ is bounded below by a rational positive-definite matrix. To give an explicit enclosure, define the column vectors

$$\boldsymbol{u}_2 = (-1/2, 1, 0)^\top, \quad \boldsymbol{v}_2 = (-1, -2, 0)^\top, \quad \boldsymbol{u}_3 = (0, -1/2, 1)^\top, \quad \boldsymbol{v}_3 = (0, -1, -2)^\top.$$

Differentiating $\varphi$ gives three diagonal terms and four rank-one terms. Rounding their coefficients downward gives

$$\begin{aligned}(\ln 2)\nabla^2\varphi \succeq\ & \operatorname{diag}(17.196, -32.3432, -69.9851) \\ & + 100.4497\boldsymbol{u}_2\boldsymbol{u}_2^\top + 1.2073\boldsymbol{v}_2\boldsymbol{v}_2^\top \\ & + 356.9883\boldsymbol{u}_3\boldsymbol{u}_3^\top + 0.6714\boldsymbol{v}_3\boldsymbol{v}_3^\top = \mathbf{M}, \\ \mathbf{M} = & \begin{pmatrix} 43.515725 & -47.81025 & 0 \\ -47.81025 & 162.854175 & -177.15135 \\ 0 & -177.15135 & 289.6888 \end{pmatrix}.\end{aligned}$$

The notation $\succeq$ means that the difference is positive semidefinite. The three pivots in symmetric elimination exceed 43.5157, 110.3255, and 5.2343, respectively, so $\mathbf{M}$ is positive definite. At the center, the value is greater than 0.26524289723923955, and the absolute values of the three partial derivatives are less than $(1.600 \cdot 10^{-9}, 8.380 \cdot 10^{-6}, 2.176 \cdot 10^{-15})$. The supporting plane at $(\bar{x}, \bar{y}, \bar{z})$ then gives a uniform lower bound by subtracting 0.001 times the sum of the absolute gradient bounds. After multiplication by $\delta_-$ and addition of $\rho_+(e_- - \vartheta)$, its residual is greater than $6.18 \cdot 10^{-8} > 0$. This excludes the whole local region.

**Exhaustive verification of the parameter domain.** Start from the two- and three-dimensional rectangles specified above, one for each depth case. A box is discarded only if it violates a coordinate inequality, has a direct certified lower bound above $\vartheta$, or is contained in the certified local region. Otherwise split one coordinate at its exact rational midpoint and keep both children. Exhaustion of all three queues proves the lower bound for every fixed depth. The exact node and leaf counts give an additional check that no generated child was omitted. The following algorithm makes the subdivision and its stopping rule explicit. A box is a Cartesian product of closed rational intervals; overlap at a subdivision boundary does not affect coverage.

**Algorithm 15** Certifying the global SVP lower bound

**Input:** The threshold $\vartheta = 0.2740094$ and the three depth cases $k = 1$, $k = 2$, and $k \geq 3$.

**Output:** A partition of the initial boxes into boxes certified as infeasible, directly bounded above $\vartheta$, or contained in the local region.

Initialize the pending boxes to $[0.4, 2] \times [0, 1/2]$ for $k = 1$ and $[0.4, 2] \times [0, 1/2]^2$ for each other case.
**while** a pending box remains **do**
    Remove one pending box, retaining its depth case.
    **if** the case is $k \geq 3$ and the box lies in the certified local region **then**
        Discard the box.
    **else**
        Check the necessary parent–child inequalities by intersecting their allowed coordinate intervals.
        **if** the necessary intervals are empty **then**
            Discard the box.
        **else**
            Compute its rational lower bound from (78) and the repetition-cost bound, as appropriate to its depth case.
            **if** the lower bound is greater than $\vartheta$ **then**
                Discard the box.
            **else**
                Divide the widths in $(\rho, x, y)$ by $(0.5, 0.2, 0.1)$, omitting $y$ for $k = 1$.
                Bisect a coordinate of largest normalized width at its exact rational midpoint; add both resulting boxes.
            **end if**
        **end if**
    **end if**
**end while**
**return** the certificate that every initial box has been covered.

The completed calculation visits 454,329 boxes and leaves no box pending. Its 227,166 leaves consist of 17 direct exclusions for depth one, 262 for depth two, 225,852 for all larger depths, 1,005 local exclusions, and 30 infeasible boxes. The identity $2 \cdot 227{,}166 = 454{,}329 + 3$ accounts for all three initial rectangles and both children of every subdivision. The smallest direct lower bound exceeds 0.27400940032267.

**Numerical enclosures.** The interval calculations use the logarithm enclosure of Appendix F.6 with $N = 40$ terms and outward rounding to multiples of $10^{-35}$. The initial fixed Gaussian quantities $\delta_*, w_*, e_*$ in the lower certificate are enclosed with $N = 24$ terms on a $10^{-30}$ grid and then converted outward to the finer grid; their initial uncertainty is retained throughout. For $x = e^{-a} < 1/4$, the discrete Gaussian sums retain $|j| \leq 5$. The omitted zeroth, second, and fourth moments satisfy

$$2 \sum_{j \geq 6} j^{2i} x^{j^2} \leq \frac{2 \cdot 6^{2i} x^{36}}{1 - (7/6)^{2i} x^{13}} \qquad (i \in \{0, 1, 2\}),$$

since consecutive terms have ratio at most $(7/6)^{2i}x^{13}$; for $i = 0$ this is the theta bound of Appendix F.6. To evaluate the small entropy $\mathcal{H}(r_a)$, put

$$P = \frac{2\sum_{j\geq 2} x^{j^2}}{\Theta(a)}, \qquad U = \frac{2\sum_{j\geq 2} j^2 x^{j^2}}{\Theta(a)}, \qquad \mathcal{H}(r_a) = \frac{-(1-P)\ln(1-P) + P\ln\Theta(a) - U\ln x}{\ln 2}. \tag{79}$$

The identity follows from $p_a(j) = x^{j^2}/\Theta(a)$ and avoids subtracting two entropies of comparable size. All displayed lower and upper bounds include truncation and rounding errors. Floating-point optimization supplies candidate parameters and may prioritize subdivisions; it is never used to accept an inequality.

**Theorem G.2.** *Consider the objective in* (45) *of the SVP construction on* $0 \leq \delta \leq 1$ *with every fixed decomposition depth* $k \geq 1$*, every exact-search cutoff* $\delta_0 \in (1/2, 1)$*, and parameters that may depend on the input density, subject to* (3)*,* (5)*,* (8)*, and the padding condition* (72)*. The parameters are selected deterministically from the input dimensions and the counted radius, independently of the random input matrix. All strict constraints hold with uniform positive margins throughout* $\delta_0 \leq \mathcal{H}(p_a) \leq 1$*. Its infimum lies in* $[0.2740094, 0.27400949)$*. With the constants in Eq.* (46)*, set* $n_0 = \lceil n/(1-\kappa)\rceil$ *and choose the proportions as in Eq.* (34)*. This parameter rule gives time* $2^{0.27400949n+o(n)}$ *and space* $2^{0.261392024n+o(n)}$*.*

*Proof.* The upper certificate gives a feasible rule with complete time exponent strictly below 0.27400949. At the single density used in the lower certificate, every fixed depth and every permitted choice has complete exponent at least 0.2740094; allowing different choices at other densities cannot lower this bound. Thus the infimum lies in the stated interval. □

## G.5 Certifying the SVP Density Functions above One

The functions in Section 5.3 describe the cost at every density in a feasible profile's domain. The coordinate analysis in Lemma 5.5 permits every fixed bounded interval of positive densities. We now verify the remaining parameter conditions and uniform time and space upper bounds through density 1.5.

Keep $c_0 = 1$ and the constants $k, \kappa, \lambda, \eta, \varepsilon, \nu$ from (74), and choose the remaining ratios from Table 3. All displayed decimals are exact rational numbers. For the actual algorithm, set $t_i = c_i(\widehat{p}+\lambda)$ with $0 \leq \widehat{p} - p_a(1) \leq \min\{n^{-2}, 10^{-12}\}$, as before. In particular, the last row gives a single fixed profile whose functions $T_c(\delta), S_c(\delta)$ apply throughout $[1, 1.5]$. No separate parameter fitting is needed to evaluate an intermediate density. The other rows improve the bounds on shorter intervals.

| Upper density $D$ | $c_1$ | $c_2$ | $c_3$ |
|---|---|---|---|
| 1.05 | 0.8939052281 | 0.6177516046 | 0.3386702762 |
| 1.10 | 0.8678977552 | 0.5915218016 | 0.3209603588 |
| 1.20 | 0.8656984966 | 0.5982906774 | 0.3350929465 |
| 1.30 | 0.8501069734 | 0.5873979844 | 0.3331341199 |
| 1.40 | 0.8191452039 | 0.5562355082 | 0.3124046192 |
| 1.50 | 0.8022369518 | 0.5405699407 | 0.3037290770 |

Table 3: Rational parameter profiles. Each row is certified on the entire interval $1 \leq \delta \leq D$.

**Lemma G.3.** *Let* $(D, c_1, c_2, c_3)$ *be a row of Table 3,* $c_0 = 1$*,* $k = 3$*,* $\kappa = 10^{-8}$*,* $\lambda = 10^{-9}$*, and* $\eta = \varepsilon = 10^{-10}$*. Let* $a > 0$ *satisfy* $1 \leq \mathcal{H}(p_a) \leq D$*, put* $\rho = (1-\kappa)\mathcal{H}(p_a)$*, and, for any* $0 \leq \theta \leq 10^{-12}$*,*

$t_i = c_i(p_a(1) + \lambda + \theta)$ *for* $0 \le i \le 3$. *These parameters satisfy the coordinate, nested-equation, and even-modulus conditions* (3), (5), *and* (8), *and the two padding conditions* (39) *with* $\lambda$ *replaced by* $\lambda + \theta$, *all with uniform positive margins. The exponents* (76) *are strictly below the time and space bounds of the corresponding row of Table 4 throughout this interval.*

| $D$ | Time upper bound | Space upper bound | Accepted intervals |
|---|---|---|---|
| 1.05 | 0.285998580 | 0.266611111 | 335 |
| 1.10 | 0.297998391 | 0.274778021 | 342 |
| 1.20 | 0.325997508 | 0.284999867 | 351 |
| 1.30 | 0.360999854 | 0.306587328 | 342 |
| 1.40 | 0.407979218 | 0.317854425 | 231 |
| 1.50 | 0.467999492 | 0.328474928 | 223 |

Table 4: Uniform interval certificates for the complete completion cost, with decimal upper bounds rounded upward. Exact search and the original profile below density one have smaller worst-case costs.

*Proof.* Set $x = e^{-a}$. Strict monotonicity of $a \mapsto \mathcal{H}(p_a)$ (Lemma A.2(4)) allows rational bisection in $x$ to enclose the endpoints with entropy 1 and $D$. Starting from $[1/10, 7/10]$, perform 75 bisections; each decision is accepted only when an outward interval for $\mathcal{H}(p_a)$ lies strictly on the relevant side of the target. The left endpoint of the entropy-one bracket and the right endpoint of the entropy-$D$ bracket therefore enclose the entire required range. The former is approximately 0.1444784036; the latter is at most 0.3450178247.

We use a theta enclosure that also applies when $x > 1/4$. For $i \in \{0, 1\}$, the terms with $2 \le j \le 16$ are summed explicitly, and their remaining two-sided tail satisfies

$$2 \sum_{j \ge 17} j^{2i} x^{j^2} \le \frac{2 \cdot 17^{2i} x^{289}}{1 - (18/17)^{2i} x^{35}} \qquad (0 < x \le 3/4).$$

Indeed, the ratio of consecutive terms is at most $(18/17)^{2i} x^{35} < 1$. Adding the contributions from $0, \pm 1$ gives outward intervals for $\Theta(a)$, $\mu(a)$, $p_a(1)$, and $\mathcal{H}(p_a)$, and (79) gives $\mathcal{H}(r_a)$ without subtracting two nearly equal entropies.

All arithmetic uses integer endpoints with scale $10^{35}$ and outward rounding, with the logarithm enclosure of Appendix F.6 and $N = 40$ terms. This is the same arithmetic used in Appendix G.4; the enlarged theta domain is checked separately here.

Bisect each complete interval of $x$ at rational midpoints. On each subinterval, evaluate every constraint and both costs, allowing the full interval $0 \le \theta \le 10^{-12}$. Accept only if all required margins are positive and both costs are below the corresponding rounded targets in Table 2; otherwise bisect again. Sorting the accepted intervals and checking exact equality of adjacent endpoints verifies that they cover the initial interval with neither gaps nor overlaps of positive length. The resulting bounds and leaf counts are in Table 4.

For the padding bounds, $\mu(a)$ increases in $x$ because its derivative with respect to $\ln x$ is the variance of $j^2$ under $p_a$. The upper endpoint of every interval has $\mu(a) < 1$, as verified by the same arithmetic; hence $p_a(1) = x/\Theta(a)$, whose derivative in $x$ has the sign of $1 - \mu(a)$, increases throughout the interval. Using its upper endpoint value $p_{\max}$, the two margins are at least

$$\frac{\lambda}{1 - \kappa} - \varepsilon > 9 \cdot 10^{-10}, \qquad \frac{\kappa(1 - 2p_{\max}) - 2(\lambda + 10^{-12})}{1 - \kappa} - 2\varepsilon > 3.78 \cdot 10^{-9}.$$

The equation margin is $\eta = 10^{-10}$, all other feasibility margins are positive, and $\nu(1-\kappa)^{-2/3} > 1$. These are uniform margins on compact intervals, so the bound on the probability of omitting a fixed vector in Lemma 3.2 applies uniformly by Section 3.1.1. Finally, continuity extends the strict feasibility and target-cost bounds to neighborhoods of the closed intervals. This absorbs $\mathcal{H}(p_a) = \delta + o(1)$, including at $\delta = 1$ and $\delta = D$, and proves the asserted asymptotic bounds for actual inputs. □

**Certified bounds up to density 3/2.** All six interval certificates record the exact rational endpoints, upper bounds, and smallest margins. Equation (42) can be evaluated at any rational density in $[0, 1.5]$. The evaluation uses exact search below the cutoff 0.54, the original profile through density one, and above one selects the smallest certified time upper endpoint among the applicable profiles. The reported space bound belongs to that same profile. This is a certified evaluation of feasible bounds, not a solution of the optimization problem (43). Floating-point samples for plots are explicitly separate from the interval certificates. Any jumps at a profile boundary reflect that selection policy, not an intrinsic discontinuity in SVP complexity.

**Limitation of the large-coordinate bound at density two.** The same inverse-entropy and theta enclosures, now at $\mathcal{H}(p_a) = 2$, certify

$$0.62508 < \mathcal{H}(r_{a(2)}) < 0.62509.$$

Thus the large-coordinate term in the proved completion bound (42) alone exceeds 0.62508 there. Changing only the inner representation parameters in this expression cannot prove a total exponent below 1/2. This is a limitation of that upper-bound expression, not a runtime lower bound for the enumeration and not a lower bound for other algorithms. A fixed bounded density range remains essential to the uniform estimates; we make no claim for $\delta \to \infty$.

## References


[Aba26] Navid Abapour. Sharper and closed-form attacks on SIS when the modulus is small. *IACR Communications in Cryptology*, 3(2), 2026. doi:10.62056/abe0zogxq.

[ADSD15] Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the closest vector problem in $2^n$ time—the discrete Gaussian strikes again! In *56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 563–582, 2015. URL: https://arxiv.org/abs/1504.01995, doi:10.1109/FOCS.2015.41.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 99–108, 1996. doi:10.1145/237814.237838.

[ASD18] Divesh Aggarwal and Noah Stephens-Davidowitz. Just take the average! An embarrassingly simple $2^n$-time algorithm for SVP (and CVP). In *1st Symposium on Simplicity in Algorithms (SOSA 2018)*, volume 61 of *OASIcs*, pages 12:1–12:19, 2018. doi:10.4230/OASIcs.SOSA.2018.12.

[BCJ11] Anja Becker, Jean-Sébastien Coron, and Antoine Joux. Improved generic algorithms for hard knapsacks. In *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 364–385. Springer, 2011. doi:10.1007/978-3-642-20465-4_21.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 10–24, 2016. doi:10.1137/1.9781611974331.ch2.

[BGJ14] Anja Becker, Nicolas Gama, and Antoine Joux. A sieve algorithm based on overlattices. *LMS Journal of Computation and Mathematics*, 17(A):49–70, 2014. doi:10.1112/S1461157014000229.

[BGLS19] Shi Bai, Steven D. Galbraith, Liangze Li, and Daniel Sheffield. Improved combinatorial algorithms for the inhomogeneous short integer solution problem. *Journal of Cryptology*, 32(1):35–83, 2019. doi:10.1007/s00145-018-9304-1.

[BPS26] Maiara F. Bollauf, Amaury Pouly, and Yixin Shen. Solving SIS in any norm via Gaussian sampling. Cryptology ePrint Archive, Paper 2026/225, 2026. URL: https://eprint.iacr.org/2026/225.

[CDAE21] André Chailloux, Thomas Debris-Alazard, and Simona Etinski. Classical and quantum algorithms for generic syndrome decoding problems and applications to the Lee metric. In *Post-Quantum Cryptography – PQCrypto 2021*, volume 12841 of *Lecture Notes in Computer Science*, pages 44–62. Springer, 2021. doi:10.1007/978-3-030-81293-5_3.

[DEL25] Léo Ducas, Lynn Engelberts, and Johanna Loyer. Wagner's algorithm provably runs in subexponential time for $\mathrm{SIS}^\infty$. In *Advances in Cryptology – CRYPTO 2025, Part I*, volume 16000 of *Lecture Notes in Computer Science*, pages 353–384. Springer, 2025. doi:10.1007/978-3-032-01855-7_12.

[DEP23] Léo Ducas, Thomas Espitau, and Eamonn W. Postlethwaite. Finding short integer solutions when the modulus is small. In *Advances in Cryptology – CRYPTO 2023, Part III*, volume 14083 of *Lecture Notes in Computer Science*, pages 150–176. Springer, 2023. doi:10.1007/978-3-031-38548-3_6.

[DL26] Emmanouil Doulgerakis and Thijs Laarhoven. Smooth sailing through spherical shells: Provable random-lattice sieving in time $2^{0.292n}$. arXiv:2609.37482, 2026. URL: https://arxiv.org/abs/2609.37482.

[DLMF] NIST Digital Library of Mathematical Functions. https://dlmf.nist.gov/, Release 1.2.8 of 2026-09-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[DLvW20] Léo Ducas, Thijs Laarhoven, and Wessel P. J. van Woerden. The randomized slicer for CVPP: Sharper, faster, smaller, batchier. In *Public-Key Cryptography – PKC 2020, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 3–36. Springer, 2020. doi:10.1007/978-3-030-45388-6_1.

[DSW18] Martin Dietzfelbinger, Philipp Schlag, and Stefan Walzer. A subquadratic algorithm for 3XOR. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *LIPIcs*, pages 59:1–59:15, 2018. doi:10.4230/LIPIcs.MFCS.2018.59.

[ER65] Paul Erdős and Alfréd Rényi. Probabilistic methods in group theory. *Journal d'Analyse Mathématique*, 14:127–138, 1965. doi:10.1007/BF02806383.

[EV22] Friedrich Eisenbrand and Moritz Venzin. Approximate $\mathrm{CVP}_p$ in time $2^{0.802n}$. *Journal of Computer and System Sciences*, 124:129–139, 2022. doi:10.1016/j.jcss.2021.09.006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 197–206, 2008. Full version dated August 25, 2008. URL: https://web.eecs.umich.edu/~cpeikert/pubs/trap_lattice.pdf, doi:10.1145/1374376.1374407.

[HGJ10] Nick Howgrave-Graham and Antoine Joux. New generic algorithms for hard knapsacks. In *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 235–256. Springer, 2010. doi:10.1007/978-3-642-13190-5_12.

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

[Laa17] Thijs Laarhoven. Sieving for closest lattice vectors (with preprocessing). In *Selected Areas in Cryptography – SAC 2016*, volume 10532 of *Lecture Notes in Computer Science*, pages 523–542. Springer, 2017. doi:10.1007/978-3-319-69453-5_28.

[McD89] Colin McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics, 1989*, pages 148–188. Cambridge University Press, 1989. doi:10.1017/CBO9781107359949.008.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007. doi:10.1137/S0097539705447360.

[Oze16] Ilya Ozerov. *Combinatorial Algorithms for Subset Sum Problems*. PhD thesis, Ruhr-Universität Bochum, 2016. URL: https://d-nb.info/1089007094/34.

[RV22] Thomas Rothvoss and Moritz Venzin. Approximate CVP in time $2^{0.802n}$ – now in any norm! In *Integer Programming and Combinatorial Optimization (IPCO 2022)*, volume 13265 of *Lecture Notes in Computer Science*, pages 440–453. Springer, 2022. doi:10.1007/978-3-031-06901-7_33.

[Sha12] Andrew Shallue. Division algorithms for the fixed weight subset sum problem. arXiv:1201.2739, 2012. URL: https://arxiv.org/abs/1201.2739.

[SP15] Aria Ghasemian Sahebi and S. Sandeep Pradhan. Abelian group codes for channel coding and source coding. *IEEE Transactions on Information Theory*, 61(5):2399–2414, 2015. doi:10.1109/TIT.2015.2407874.

[SS81] Richard Schroeppel and Adi Shamir. A $T = O(2^{n/2})$, $S = O(2^{n/4})$ algorithm for certain NP-complete problems. *SIAM Journal on Computing*, 10(3):456–464, 1981. doi:10.1137/0210033.

[ZWW26] Yang Zhou, Zhedong Wang, and Mingsheng Wang. Exact SVP on Haar-random lattices at the heuristic sieving exponents. Cryptology ePrint Archive, Paper 2026/2271, 2026. URL: https://eprint.iacr.org/2026/2271.

## AI methodology

We used Codex and ChatGPT to assist with proof development, literature analysis, manuscript revision, implementation, and computational verification of finite cases. These tools materially influenced the CVP reduction, representation and streaming arguments, exact search, and modulus extensions. In particular, Codex was used to develop the SVP arguments and to construct the counted-radius, matrix-input, and basis-input reference implementations. The authors independently verified the correctness and originality of all AI-assisted results, including mathematical arguments, implementations, and references.