Dual Coppersmith: Finding Small Roots Without Multivariate Polynomial Systems

Taehun Kang[1] Jebin Kim[1] Changmin Lee[1,*] JeongHwan Lee[1]

[1]Korea University, Seoul, Republic of Korea

{kth9999,wpqls1025,changminlee,hwani0814}@korea.ac.kr

[*]Corresponding author

Abstract

The multivariate Coppersmith algorithm is a central tool in public-key cryptanalysis for finding a small hidden root of modular polynomial equations. It consists of two stages. First, a lattice-reduction (LR) stage constructs auxiliary equations satisfied by the hidden root over $\mathbb{Z}$. Second, a root-recovery (RR) stage solves these equations. The RR stage can be a bottleneck: auxiliary polynomials may not isolate the root, and solving them can take exponential time in the worst case, even when LR is feasible.

We introduce *Dual Coppersmith*, which replaces algebraic root recovery with geometric recovery from a modular orthogonal lattice. Our construction embeds the hidden root as a short lattice vector, and we prove that LLL recovers its direction whenever a sufficient gap separates it from all independent lattice vectors. The root can then be recovered directly from coordinate ratios. To analyze this gap, we propose a heuristic model for the second successive minimum by examining both the full lattice and a structured sublattice. Under this model, Dual Coppersmith achieves the standard Coppersmith product-size bound without solving an auxiliary polynomial system. Experiments on the approximate common divisor (ACD) and Herrmann–May (HM) problems support the model's predictions and demonstrate root recovery in higher-dimensional instances where the Gröbner-based baseline reaches its timeout.

Keywords: Coppersmith algorithm, Multivariate poynomial systems, Modular orthogonal lattice, Lattice reduction.

1 Introduction

Coppersmith's method and its multivariate extensions are central tools in cryptanalysis. Applications include recovering partially known messages in low-public-exponent RSA and factoring with partial knowledge of a prime factor [Cop96, Cop97], attacks on RSA with small private exponents [BD99], and approximate common divisor problems underlying integer-based fully homomorphic encryption [CH13]. Multivariate extensions have also been applied to the cryptanalysis of elliptic-curve pseudorandom generators with truncated outputs [MV20].

The standard multivariate Coppersmith method, which we call the **primal** algorithm, turns modular polynomial equations with a small hidden root into a lattice problem [HG97, JM06, HM10]. Its lattice-reduction (LR) stage constructs a lattice from polynomial coefficient vectors and applies a reduction algorithm such as LLL, which runs in polynomial time in the lattice input size. Sufficiently short vectors yield auxiliary polynomials that vanish at the hidden root over the integers. Its root-recovery (RR) stage then solves the resulting polynomial system, for example by Gröbner-basis computation or resultant elimination.

Much of the theoretical progress has focused on the LR stage. Determinant estimates provide criteria for obtaining sufficiently short polynomial coefficient vectors. Better shift selection, sharper small-root bounds, and automated lattice constructions have made the production of useful auxiliary polynomials increasingly systematic [JM06, HM10, FNP24, FLC+25, Rya25].

The RR stage poses a separate problem that these LR guarantees do not resolve. Producing short auxiliary polynomials does not by itself guarantee that they isolate the target root or form a system that can be solved efficiently. Polynomial-time guarantees for LR therefore do not automatically extend to the complete algorithm [Rya25]. The cost of solving the auxiliary system must therefore be assessed separately from the cost of producing its equations, and can remain a bottleneck even when lattice reduction is feasible.

This motivates the question behind this work:

Can multivariate Coppersmith find the small root without running an RR stage at all?

We answer this question with the **Dual Coppersmith algorithm**. Rather than improving the small-root exponent, it replaces algebraic RR with recovery from a sufficiently isolated root direction in a modular orthogonal lattice. This makes the lattice's second successive minimum central to the recovery analysis.

Contributions

Our contribution in this work is two-fold, consisting of the novel algorithmic design itself and a theoretical analysis of its success conditions.

- **The Dual Coppersmith algorithm.** We reinterpret the generated modular relations as linear congruence constraints on the root's monomial-evaluation vector. Their modular orthogonal lattice contains a scaled root vector, and we prove that the lattice points on its real line are precisely its integer multiples. A nonzero vector on this line therefore reveals the root through ratios of its constant and degree-one coordinates, which cancel the unknown multiple; verification against the original congruences and bounds then makes the algorithm unconditionally sound. For a known prime modulus p, our main recovery theorem (Theorem 3.4) guarantees recovery whenever $\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp}) > \delta^{2(n-1)}\rho\sqrt{n}$. Here n is the lattice dimension, $\rho\sqrt{n}$ bounds the root-vector length, and $\delta^{2(n-1)}$ is a valid approximation factor for the lattice reduction used.

This guarantee is deterministic and does not require the second-minimum model. Dual Coppersmith thereby replaces the algebraic RR stage with linear algebra. Figure 1 contrasts the two algorithms.

- **An analysis of the second minimum.** To predict when this gap occurs, we analyze the second successive minimum λ_2 of the orthogonal lattice. An estimate based only on the full projected covolume may miss short vectors arising from the algebraic structure. We identify the first-jet sublattice in the span of the root vector and its first derivatives, characterize membership of integer linear combinations of the derivative vectors through Jacobian congruences, and compute its projected covolume. Gaussian-heuristic estimates for the full projection and the projected first-jet lattice provide two candidate length scales, λ_{GH} and λ_{jet}. Under additional geometric assumptions on short vectors inside and outside the first-jet sublattice, Heuristic 4.1 predicts $\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp}) \approx \min\{\lambda_{\mathrm{jet}}, \lambda_{\mathrm{GH}}\}$. Under the stated heuristics, Corollary 4.2 predicts recovery when $\prod_{i=1}^{\ell} X_i < p^{1/d-\varepsilon}$, for inputs of degree at most d whose Jacobian has positive rank at the target root modulo p. Here $0 < \varepsilon < 1/d$ is fixed, the shift parameters are chosen suitably, and p is sufficiently large. This matches the product-bound exponent of primal total-degree Coppersmith while avoiding algebraic RR.

Remark 1.1. Although our predictive model relies on the Gaussian heuristic, this should not be viewed as a localized limitation. In contrast, the previous primal algorithm operates on two separate empirical heuristics. First, it assumes that the auxiliary equations form a solvable system. Second, to obtain the ℓ independent equations required for root recovery, it assumes the first ℓ LLL-reduced vectors have roughly the same norm. The dual product-bound analysis instead uses Heuristic 4.1, which combines Gaussian-heuristic estimates for the full projection and the projected first-jet lattice. This second-minimum model is an additional assumption and does not follow from the Gaussian heuristic alone.

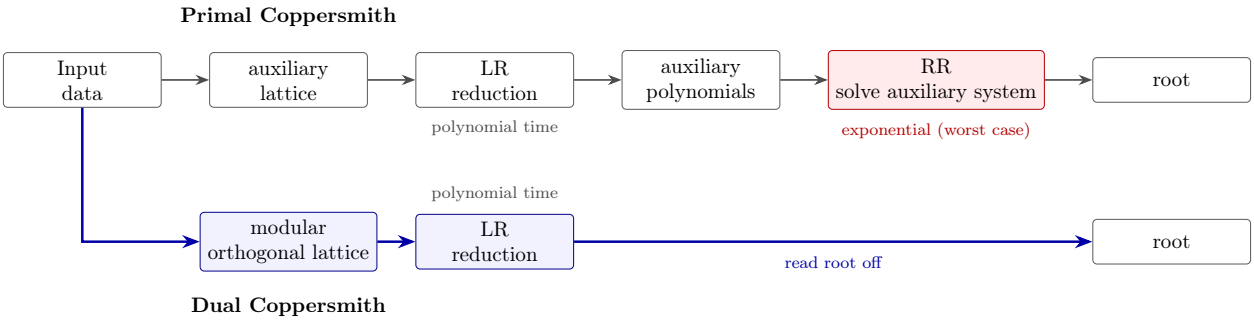

Figure 1: Primal and Dual Coppersmith start from the same input data but use the lattice output differently. The primal algorithm uses lattice reduction to produce auxiliary equations, and then runs a separate algebraic root-recovery stage. Dual Coppersmith builds a dual lattice in which the hidden root is represented by a short direction; after lattice reduction, the root is obtained directly by coordinate ratios.

We test this on two problems that require small-root finding. On the approximate common divisor (ACD) and Herrmann–May (HM), Dual Coppersmith recovers the root in seconds while the Gröbner-based baseline crosses a fixed timeout as the number of variables grows. Also, the test shows our prediction $\lambda_2^{\mathrm{pred}}$ provides a tight approximation to λ_2. The full data, including its scaling behavior with respect to ℓ, is available in Table 1 and 2.

2 Preliminaries

This section fixes notation and records the facts used later: basic lattice and reduction background, and the primal Coppersmith pipeline against which we compare.

2.1 Notation

We write $\mathbb{Z}, \mathbb{R}$ for the integers and reals, respectively, and $\mathbb{Z}_{\geq a}$ for the integers at least a. For any positive integer k, we write $[k] = \{1, \ldots, k\}$. For any finite set S, we let $|S|$ denote its cardinality.

Throughout, ℓ is the number of variables. For a vector $\mathbf{x} = (x_1, \ldots, x_\ell)$, e_i is the i-th standard basis vector. For $\mathbf{r} \in \mathbb{R}^\ell$, we write $\|\mathbf{r}\| = (\sum_i r_i^2)^{1/2}$. For a nonzero vector $u \in \mathbb{R}^d$, let $\pi_u : \mathbb{R}^d \to (\mathbb{R}u)^\perp$ denote the orthogonal projection onto the hyperplane perpendicular to u. When the vector u is clear from context, we simply write π.

A multi-index $\alpha \in \mathbb{Z}_{\geq 0}^\ell$ has $|\alpha| = \sum_i \alpha_i$, and, abusing notation, for a vector $\mathbf{z} = (z_1, \ldots, z_\ell)$ we write $\mathbf{z}^\alpha = z_1^{\alpha_1} \cdots z_\ell^{\alpha_\ell}$. The polynomials in our constructions have integer coefficients; for a polynomial $0 \neq f = \sum_\alpha c_\alpha \mathbf{x}^\alpha \in \mathbb{Z}[\mathbf{x}]$, we set $\deg(f) = \max\{|\alpha| : c_\alpha \neq 0\}$. Fix a finite ordered multi-index set $\mathcal{T} \subset \mathbb{Z}_{\geq 0}^\ell$. We denote by $(\mathbf{z}^\alpha)_{\alpha \in \mathcal{T}}$ the vector of monomials indexed by $\mathcal{T}$, and for a vector $\mathbf{z}$ with coordinates indexed by $\mathcal{T}$ we write $\mathbf{z}_\alpha$ for its α-coordinate. We identify $\mathbb{R}^\mathcal{T}$ with coefficient vectors in this monomial basis and write $\mathrm{coeff}_\mathcal{T}(h)$ for the coefficient vector of a polynomial h (missing coefficients padded with zeros); as our polynomials are integer-coefficient, $\mathrm{coeff}_\mathcal{T}(h) \in \mathbb{Z}^\mathcal{T}$. If every monomial occurring in h lies in $\mathcal{T}$, then for any $\mathbf{z} \in \mathbb{R}^\ell$, $h(\mathbf{z}) = \langle \mathrm{coeff}_\mathcal{T}(h), (\mathbf{z}^\alpha)_{\alpha \in \mathcal{T}} \rangle$.

2.2 Lattices and lattice reduction

An integral lattice $\mathcal{L} \subseteq \mathbb{Z}^d$ of full rank is a discrete additive subgroup generated by a linearly independent basis $b_1, \ldots, b_d \in \mathbb{Z}^d$; that is, $\mathcal{L} = \{\sum_{i=1}^d z_i b_i : z_i \in \mathbb{Z}\}$. Let $B \in \mathbb{Z}^{d \times d}$ be the basis matrix whose rows are $b_1, \ldots, b_d$. Since $\mathcal{L}$ has full rank, its determinant satisfies $\det(\mathcal{L}) = |\det B| = [\mathbb{Z}^d : \mathcal{L}]$.

Definition 2.1 (Successive minima)**.** Let $\mathcal{L} \subset \mathbb{R}^d$ be a rank-d lattice. For $i \in [d]$, the i-th successive minimum $\lambda_i(\mathcal{L})$ is the smallest $R > 0$ such that $\mathcal{L}$ contains i linearly independent vectors of norm at most R; in particular $\lambda_1(\mathcal{L}) = \min\{\|v\| : v \in \mathcal{L} \setminus \{0\}\}$.

The Coppersmith algorithm uses lattice reduction (LR) as a core subroutine. In practice, we use the LLL algorithm [LLL82], which runs in time polynomial in the lattice rank d and the bit-length of the input basis vectors. We measure the output quality by the root-Hermite factor δ, which describes the first-vector norm relative to $\det(\mathcal{L})^{1/d}$ on a per-dimension scale.

Lemma 2.2 (LLL first-vector bound [LLL82])**.** *Let $b_1, \ldots, b_d$ be the output of LLL on a rank-d lattice $\mathcal{L}$, with root-Hermite factor δ. Then*

$$\|b_1\| \leq \delta^{2(d-1)} \lambda_1(\mathcal{L}).$$

The LLL bound is a worst-case guarantee on the vector returned by reduction. For the size of the shortest vector itself, and later for the second minimum of our dual algorithm, we use the standard Gaussian heuristic.

Heuristic 2.3 (Gaussian heuristic [GN08])**.** *Let $\mathcal{L}$ be a rank-d lattice. The Gaussian heuristic predicts that*

$$\lambda_1(\mathcal{L}) \approx \sqrt{\frac{d}{2\pi e}} \det(\mathcal{L})^{1/d}.$$

2.3 Bounded modular roots and total degree shifts

We first define the *bounded root problem* and the *shift polynomials* used by both algorithms. We then establish an index estimate for the shift coefficient matrix. This estimate will be used to analyze the two lattice constructions.

Definition 2.4 (Bounded modular root problem)**.** Let p be a prime, $f_1, \dots, f_s \in \mathbb{Z}[x_1, \dots, x_\ell]$, and let $\mathbf{X} = (X_1, \dots, X_\ell) \in \mathbb{Z}_{>0}^{\ell}$. The bounded modular root problem asks to find an $\mathbf{r} \in \mathbb{Z}^\ell$ satisfying

$$|r_i| < X_i \quad (i \in [\ell]), \qquad f_a(\mathbf{r}) \equiv 0 \pmod{p} \quad (a \in [s]).$$

The known-moduli formulation of [Rya25] allows congruences with distinct known moduli p_a. Setting $P = \operatorname{lcm}(p_1, \dots, p_s)$ and $h_a = (P/p_a) f_a$ gives the equivalent relations

$$f_a(\mathbf{r}) \equiv 0 \pmod{p_a} \iff h_a(\mathbf{r}) \equiv 0 \pmod{P}.$$

Since P may be composite, this equivalence does not reduce the known-moduli problem to Definition 2.4. The results below concern the common prime-modulus setting of that definition.

The Coppersmith algorithm starts from shift polynomials that vanish at every target root modulo p^m. We construct them by multiplying powers of the input polynomials by monomials and suitable powers of p, subject to a total degree bound.

Definition 2.5 (Total-degree shifts)**.** Let p be prime, let $m, t \in \mathbb{Z}_{\geq 1}$, and let $f_1, \dots, f_s \in \mathbb{Z}[x_1, \dots, x_\ell]$ with $t \geq \max_a \deg(f_a)$. Put $\mathcal{T}_t = \{\alpha \in \mathbb{Z}_{\geq 0}^{\ell} : |\alpha| \leq t\}$, and use standard multi-index notation for $\mathbf{f}^{\mathbf{i}}$. The total-degree shift index set is

$$\mathcal{S}_{m,t} = \left\{ (\mathbf{i}, \beta) \in \mathbb{Z}_{\geq 0}^{s} \times \mathbb{Z}_{\geq 0}^{\ell} : 1 \leq |\mathbf{i}| \leq m,\ \deg(\mathbf{x}^{\beta} \mathbf{f}^{\mathbf{i}}) \leq t \right\}.$$

For $(\mathbf{i}, \beta) \in \mathcal{S}_{m,t}$, the corresponding shift polynomial is

$$g_{\mathbf{i},\beta} = p^{m-|\mathbf{i}|} \mathbf{x}^{\beta} \mathbf{f}^{\mathbf{i}}.$$

Indeed, if $\mathbf{r}$ is a bounded root modulo p, then $\mathbf{f}^{\mathbf{i}}(\mathbf{r})$ is divisible by $p^{|\mathbf{i}|}$, which implies $g_{\mathbf{i},\beta}(\mathbf{r}) \equiv 0 \pmod{p^m}$. We refer to each $g_{\mathbf{i},\beta}$ as a *shift polynomial*.

Constructing these shift polynomials modulo p^m is only the first step. To successfully extract the target root $\mathbf{r}$, we must convert this modular relation into an equation over the integers. We use the following multivariate form of the lemma of Howgrave-Graham [HG97], as stated by Jochemsz and May [JM06, Lemma 1]. It guarantees that a small modular root is also a root over $\mathbb{Z}$ when the scaled coefficient norm is sufficiently small.

Lemma 2.6 (Howgrave-Graham)**.** *Let $h \in \mathbb{Z}[x_1, \dots, x_\ell]$ have at most ω nonzero coefficients. Let $M \in \mathbb{Z}_{>0}$, and let $\mathbf{r} \in \mathbb{Z}^\ell$ satisfy $h(\mathbf{r}) \equiv 0 \pmod{M}$ and $|r_i| < X_i$ for all i. If the Euclidean norm of the coefficient vector of $h(X_1 x_1, \dots, X_\ell x_\ell)$ is less than $M/\sqrt{\omega}$, then $h(\mathbf{r}) = 0$ over $\mathbb{Z}$.*

Proof. Write $h(\mathbf{x}) = \sum_\alpha c_\alpha \mathbf{x}^\alpha$. Since $|r_i| < X_i$ for all i and at most ω coefficients are nonzero, the Cauchy-Schwarz inequality gives

$$|h(\mathbf{r})| \leq \sum_{\alpha} |c_\alpha \mathbf{r}^\alpha| \leq \sqrt{\omega}\, \|h(X_1 x_1, \dots, X_\ell x_\ell)\| < M.$$

Since $h(\mathbf{r})$ is an integer divisible by M, we conclude that $h(\mathbf{r}) = 0$ over $\mathbb{Z}$. □

To analyze the determinants of the lattices used by Coppersmith algorithms, we collect the coefficient vectors of the shift polynomials in an integer matrix A. Its $(\mathbf{i}, \beta)$-row is $\mathrm{coeff}_{\mathcal{T}_t}(g_{\mathbf{i},\beta})$. Write

$$n = |\mathcal{T}_t| = \binom{\ell + t}{\ell}, \qquad B_t = \sum_{\alpha \in \mathcal{T}_t} \alpha_i = \binom{\ell + t}{\ell + 1}.$$

The last sum is independent of i by symmetry. Thus coefficient scaling by X^α contributes the determinant factor $\prod_{\alpha \in \mathcal{T}_t} X^\alpha = \left(\prod_{i=1}^{\ell} X_i^{\alpha_i}\right)^{B_t}$. We also define the row module

$$\mathcal{R}_{m,t} = \left\{ \mathbf{c}^T A \bmod p^m \mid \mathbf{c} \in \mathbb{Z}^{|\mathcal{S}_{m,t}|} \right\}.$$

Its cardinality measures the number of independent modular relations supplied by the shifts. The following estimate is independent of the number of input polynomials.

Lemma 2.7 (Modular row module bound)**.** *Suppose that* $\max_a \deg(f_a) \le d$ *for an integer* $d \ge 1$*, and that* $f_a \bmod p \neq 0$ *for some* $a \in [s]$*. For* $t = dm$*, put*

$$S_m = \sum_{j=1}^{m} \binom{\ell + t - jd}{\ell}.$$

Then

$$|\mathcal{R}_{m,t}| \ge p^{S_m}.$$

Moreover, for fixed ℓ, d*,*

$$\frac{S_m}{B_t} \longrightarrow \frac{1}{d} \qquad (m \longrightarrow \infty). \tag{1}$$

Proof. Choose $f = f_{a_0}$ with $\bar{f} = f \bmod p \neq 0$. For $1 \le j \le m$, the shift polynomials include

$$p^{m-j} \mathbf{x}^\beta f^j, \qquad |\beta| \le t - jd.$$

Consider the spaces $V_j = \bar{f}^j \mathbb{F}_p[\mathbf{x}]_{\le t - jd}, V_{m+1} = \{0\}$. They form a descending flag because $\deg(\bar{f}) \le d$. Multiplication by $\bar{f}^j$ is injective in $\mathbb{F}_p[\mathbf{x}]$, so

$$\dim_{\mathbb{F}_p} V_j = \binom{\ell + t - jd}{\ell}.$$

Extend bases successively from V_{m+1} to V_1, adding $c_j = \dim V_j - \dim V_{j+1}$ vectors in V_j. Lift these vectors using integer combinations of the coefficient vectors of $\mathbf{x}^\beta f^j$. Their multiples by p^{m-j} belong to $\mathcal{R}_{m,t}$ modulo p^m. Since the lifts reduce to a basis of V_1, they extend to a square matrix invertible modulo p^m. In the resulting coordinates, the multiplied vectors occupy distinct coordinates, with c_j coordinates taking p^j values each. Consequently,

$$|\mathcal{R}_{m,t}| \ge \prod_{j=1}^{m} p^{j c_j} = p^{\sum_{j=1}^{m} \dim V_j} = p^{S_m}.$$

Finally,

$$S_m = \frac{d^\ell}{(\ell+1)!} m^{\ell+1} + O(m^\ell), \quad B_t = \frac{d^{\ell+1}}{(\ell+1)!} m^{\ell+1} + O(m^\ell),$$

which proves (1). □

The estimate uses only the shifts of one polynomial. Additional relations can enlarge $\mathcal{R}_{m,t}$, but cannot decrease its cardinality. Their contribution depends on the algebraic relations among all shift polynomials modulo p^m.

2.4 The primal Coppersmith algorithm

The primal algorithm applies lattice reduction to the scaled coefficient vectors of the shift polynomials and the trivial polynomials $p^m\mathbf{x}^\alpha$. Sufficiently short vectors yield auxiliary polynomials that vanish at the target root over $\mathbb{Z}$, and the RR stage solves the resulting system.

Let $\mathcal{L}_0 \subseteq \mathbb{Z}^n$ be the lattice generated by their coefficient vectors before scaling, and define

$$\mathcal{L} = \{(X^\alpha c_\alpha)_{\alpha\in\mathcal{T}_t} \mid \mathbf{c} \in \mathcal{L}_0\}.$$

The coefficient vectors of the trivial polynomials generate $p^m\mathbb{Z}^n$, ensuring that $\mathcal{L}_0$ has full rank. Reduction modulo p^m identifies $\mathcal{L}_0/p^m\mathbb{Z}^n$ with $\mathcal{R}_{m,t}$. Hence

$$\det(\mathcal{L}) = \frac{p^{mn}}{|\mathcal{R}_{m,t}|}\left(\prod_{i=1}^{\ell} X_i\right)^{B_t}. \tag{2}$$

Therefore, Lemma 2.7 gives an upper bound for the primal lattice determinant.

Algorithm 1 describes the primal Coppersmith procedure for solving the bounded modular root problem. The LR stage constructs short auxiliary polynomials, and the RR stage enumerates their common bounded integer roots and verifies the candidates against the original congruences.

Algorithm 1 Primal Coppersmith algorithm

Require: a prime p, integers $m, t \in \mathbb{Z}_{\geq 1}$ with $t \geq \max_a \deg(f_a)$, relations $\mathcal{F} = (f_1, \ldots, f_s)$, bounds $\mathbf{X}$, and an RR algorithm

Ensure: bounded root vectors $\mathbf{r}$, possibly none

1: Form $\mathcal{S}_{m,t}$ and $g_{\mathbf{i},\beta} = p^{m-|\mathbf{i}|}\mathbf{x}^\beta\mathbf{f}^\mathbf{i}$

▷ **LR stage**

2: Scale the shifts: $\widetilde{g}_{\mathbf{i},\beta}(\mathbf{x}) = g_{\mathbf{i},\beta}(X_1x_1, \ldots, X_\ell x_\ell)$

3: Build the full LR lattice $\mathcal{L}$ generated by $\mathrm{coeff}_{\mathcal{T}_t}(\widetilde{g}_{\mathbf{i},\beta})$ for $(\mathbf{i},\beta) \in \mathcal{S}_{m,t}$, and by the trivial rows $\mathrm{coeff}_{\mathcal{T}_t}(p^m X^\alpha \mathbf{x}^\alpha)$ for $\alpha \in \mathcal{T}_t$

4: Apply LLL to the basis of $\mathcal{L}$

5: Select reduced vectors passing the Howgrave-Graham bound and interpret them as shift combinations $g^{(1)}, \ldots, g^{(k)} \in \langle g_{\mathbf{i},\beta},\ p^m\mathbf{x}^\alpha : (\mathbf{i},\beta) \in \mathcal{S}_{m,t},\ \alpha \in \mathcal{T}_t\rangle_\mathbb{Z}$

▷ **RR stage**

6: Solve $g^{(1)}(\mathbf{x}) = \cdots = g^{(k)}(\mathbf{x}) = 0$ using the RR algorithm

7: Output the candidates satisfying $|r_i| < X_i$ and $f_a(\mathbf{r}) \equiv 0 \pmod p$ for all a

LR stage. Every integer combination g of the shift polynomials and the trivial polynomials $p^m\mathbf{x}^\alpha$ satisfies $g(\mathbf{r}) \equiv 0 \pmod{p^m}$. If the coefficient vector of $\widetilde{g}(\mathbf{x}) = g(X_1x_1, \ldots, X_\ell x_\ell)$ satisfies the norm bound of Lemma 2.6, then $g(\mathbf{r}) = 0$ over $\mathbb{Z}$. The determinant of the scaled shift lattice predicts when LLL successfully produces such short auxiliary polynomials. Ignoring polynomial factors, this is achieved once the determinant falls below the required modulus threshold [HG97, JM06, HM10, GGM16, Rya25]. The trivial rows are included to ensure that the lattice has full rank. Once the LR stage successfully outputs a collection of such short auxiliary polynomials, they jointly define a system of integer equations. Solving this multivariate system, however, requires the following algebraic assumption.

Heuristic 2.8 (Algebraic root recovery [FLC+25, Assumption 1])**.** *The ideal generated by the auxiliary polynomials produced by the LR stage is zero-dimensional [BD99, HM10, FNP24], ensuring that the RR subroutine can enumerate its bounded integer zeros.*

RR stage. Assuming Heuristic 2.8 holds, the RR stage enumerates the bounded integer roots of the auxiliary system, typically using a lexicographic Gröbner basis. Even for ideals of dimension zero, known algorithms for Gröbner basis computation have arithmetic complexity upper bounds that are exponential in the number of variables in the worst case [HHPS20]. The LR determinant criterion does not control this algebraic cost, which can become a computational bottleneck as the number of variables grows. The RR cost must therefore be assessed separately from the LR runtime. Appendix A reviews the relevant complexity bound.

We now combine the determinant estimate from Lemma 2.7 with the two assumptions above to obtain the primal product bound.

Lemma 2.9. *Let $(p, \mathcal{F}, \mathbf{X})$ be a bounded modular root instance, and let $\mathbf{r} \in \mathbb{Z}^\ell$ be a target bounded root. Write $\mathcal{F} = (f_1, \ldots, f_s) \subset \mathbb{Z}[x_1, \ldots, x_\ell]$. Suppose that $\max_a \deg(f_a) \le d$ for an integer $d \ge 1$ and that $f_a \bmod p \neq 0$ for some $a \in [s]$. Assume Heuristic 2.8 and the length heuristic discussed in Remark 1.1. For any $0 < \varepsilon < 1/d$, choose m sufficiently large, depending only on ℓ, d, ε, and set $t = dm$. If p is sufficiently large and*

$$\prod_{i=1}^{\ell} X_i < p^{1/d-\varepsilon},$$

then the primal Coppersmith determinant estimate predicts that the LR stage produces auxiliary polynomials satisfying the Howgrave-Graham bound of Lemma 2.6. Consequently, Algorithm 1 outputs $\mathbf{r}$.

Proof. The proof of this lemma is provided in Appendix C. □

The preceding product bound is conservative and independent of s, since Lemma 2.7 uses only the shift polynomials generated by a single input polynomial whose reduction modulo p is nonzero. Additional relations may enlarge $\mathcal{R}_{m,t}$, decreasing the primal lattice determinant and potentially improving the product bound. The dual algorithm in the next section constructs a modular orthogonal lattice from the same shift polynomials. A geometric gap condition then enables direct root recovery from a single reduced vector, replacing the algebraic RR stage.

3 Dual Coppersmith

In this section, we present Dual Coppersmith, an algorithm for solving the bounded modular root problem without an algebraic RR stage. Our approach uses the same shift polynomials as the primal Coppersmith algorithm to construct a modular orthogonal lattice containing a scaled monomial evaluation vector of the target root. A sufficient geometric gap allows the root to be recovered directly from a single reduced vector through coordinate ratios. We first establish the lattice construction and the properties of the root line. We then present the algorithm and prove its correctness and *polynomial time complexity* for fixed m, t.

3.1 The modular orthogonal lattice

This subsection constructs the modular orthogonal lattice for Dual Coppersmith and establishes its root line properties and determinant formula.

Fix a bounded modular root instance $(p, \mathcal{F}, \mathbf{X})$. We use the shift polynomials and their coefficient matrix A from Section 2.3. The construction allows any $m, t \ge 1$ with $t \ge \max_a \deg(f_a)$.

Definition 3.1 (Modular orthogonal lattice)**.** Let A be the integer matrix from Section 2.3 whose $(\mathbf{i}, \beta)$-row is $\mathrm{coeff}_{\mathcal{T}_t}(g_{\mathbf{i},\beta})$. Write $n = |\mathcal{T}_t|$ and define the modular kernel lattice by

$$K = \{\mathbf{z} \in \mathbb{Z}^n \mid A\mathbf{z} \equiv 0 \pmod{p^m}\}.$$

Let $\rho = \mathrm{lcm}\{\mathbf{X}^\alpha \mid \alpha \in \mathcal{T}_t\}$. The scaled orthogonal lattice is

$$\mathcal{L}_{\mathbf{X}}^{\perp} = \left\{ \left(\frac{\rho}{X^\alpha} z_\alpha \right)_{\alpha \in \mathcal{T}_t} \,\middle|\, \mathbf{z} \in K \right\}.$$

The choice of ρ makes every coordinate scaling factor ρ/X^α an integer. Since $p^m\mathbb{Z}^n \subseteq K$, both K and $\mathcal{L}_{\mathbf{X}}^{\perp}$ have full rank n. The trivial polynomials $p^m\mathbf{x}^\alpha$ impose no additional constraints modulo p^m, so their coefficient vectors need not be included in A.

For a target bounded root $\mathbf{r}$, define its monomial evaluation vector and its scaled form by

$$\mathbf{z}_{\mathbf{r}} = (\mathbf{r}^\alpha)_{\alpha \in \mathcal{T}_t}, \qquad \mathbf{v}_{\mathbf{r}} = \left(\frac{\rho}{X^\alpha} \mathbf{r}^\alpha \right)_{\alpha \in \mathcal{T}_t}.$$

We call $\mathbb{R}\mathbf{v}_{\mathbf{r}}$ the *root line* associated with $\mathbf{r}$.

Lemma 3.2 (Root line)**.** *For every bounded target root* $\mathbf{r}$*,*

$$\mathbf{v}_{\mathbf{r}} \in \mathcal{L}_{\mathbf{X}}^{\perp}, \qquad \mathcal{L}_{\mathbf{X}}^{\perp} \cap \mathbb{R}\mathbf{v}_{\mathbf{r}} = \mathbb{Z}\mathbf{v}_{\mathbf{r}}, \qquad \|\mathbf{v}_{\mathbf{r}}\| \le \rho\sqrt{n}.$$

Proof. Every shift polynomial vanishes at $\mathbf{r}$ modulo p^m, so $A\mathbf{z}_{\mathbf{r}} \equiv 0 \pmod{p^m}$. Thus $\mathbf{z}_{\mathbf{r}} \in K$ and $\mathbf{v}_{\mathbf{r}} \in \mathcal{L}_{\mathbf{X}}^{\perp}$. Suppose that $c\mathbf{v}_{\mathbf{r}} \in \mathcal{L}_{\mathbf{X}}^{\perp}$ for some $c \in \mathbb{R}$. By the definition of the scaled lattice, there is a vector $\mathbf{z} \in K$ whose constant coordinate satisfies $c\rho = \rho z_0$. Hence $c = z_0 \in \mathbb{Z}$. Together with $\mathbf{v}_{\mathbf{r}} \in \mathcal{L}_{\mathbf{X}}^{\perp}$, this proves the identity for the lattice points on the root line. Finally, the bounds $|r_i| < X_i$ give

$$\|\mathbf{v}_{\mathbf{r}}\|^2 = \rho^2 \sum_{\alpha \in \mathcal{T}_t} \left| \frac{\mathbf{r}^\alpha}{X^\alpha} \right|^2 \le \rho^2 n.$$

□

The determinant of the modular orthogonal lattice can be expressed using the modular row module $\mathcal{R}_{m,t}$ defined in Section 2.3.

Lemma 3.3 (Determinant)**.** *The modular kernel lattice satisfies* $\det(K) = |\mathcal{R}_{m,t}|$*. Consequently,*

$$\det(\mathcal{L}_{\mathbf{X}}^{\perp}) = |\mathcal{R}_{m,t}| \frac{\rho^n}{\left(\prod_{i=1}^{\ell} X_i\right)^{B_t}}. \tag{3}$$

Proof. Consider the homomorphism $\phi(\mathbf{z}) = A\mathbf{z} \bmod p^m$ from $\mathbb{Z}^n$ to $(\mathbb{Z}/p^m\mathbb{Z})^{|\mathcal{S}_{m,t}|}$. Its kernel is K, so $\det(K) = [\mathbb{Z}^n : K] = |\operatorname{im}\phi|$. Let $d_1, \ldots, d_k$ be the nonzero entries in the Smith normal form of A. The corresponding unimodular transformations remain invertible modulo p^m. Thus the image of ϕ and the modular row module $\mathcal{R}_{m,t}$ both have cardinality

$$|\operatorname{im}\phi| = \prod_{j=1}^{k} \frac{p^m}{\gcd(p^m, d_j)} = |\mathcal{R}_{m,t}|$$

Algorithm 2 Dual Coppersmith algorithm

Require: a prime p, integers $m, t \in \mathbb{Z}_{\geq 1}$ with $t \geq \max_a \deg(f_a)$, relations $\mathcal{F} = (f_1, \cdots, f_s)$, and bounds $\mathbf{X}$

Ensure: a bounded root vector $\mathbf{r}$, possibly none

Use the total-degree shifts of Definition 2.5: $\mathcal{S}_{m,t}$ and $g_{\mathbf{i},\beta} = p^{m-|\mathbf{i}|}\mathbf{x}^{\beta}\mathbf{f}^{\mathbf{i}}$
Build $A = (\mathrm{coeff}_{\mathcal{T}_t}(g_{\mathbf{i},\beta}))_{(\mathbf{i},\beta)\in\mathcal{S}_{m,t}}$ and $\mathcal{L}_{\mathbf{X}}^{\perp}$ of Definition 3.1
Apply LLL to the basis of $\mathcal{L}_{\mathbf{X}}^{\perp}$ and let w be the first reduced vector
if $w_0 \neq 0$ **then**
    Set $r_i = X_i w_{e_i}/w_0$ for $i \in [\ell]$
    if all r_i are integers **then**
        Set $\mathbf{r} = (r_1, \ldots, r_\ell)$
        if $|r_i| < X_i$ for all i, and $f_a(\mathbf{r}) \equiv 0 \pmod{p}$ for all a **then**
            Output $\mathbf{r}$
        end if
    end if
end if

The coordinate scaling contributes the factor

$$\prod_{\alpha\in\mathcal{T}_t} \frac{\rho}{X^{\alpha}} = \frac{\rho^n}{\left(\prod_{i=1}^{\ell} X_i\right)^{B_t}},$$

which proves Equation (3). □

The same module cardinality therefore controls both lattice determinants. In Equation (2) it appears in the denominator, whereas in Equation (3) it appears in the numerator.

3.2 Algorithm

We now present the Dual Coppersmith algorithm for solving the bounded modular root problem of Definition 2.4.

We now state the correctness and time complexity of Algorithm 2.

Theorem 3.4 (Root recovery)**.** *Let $(p, \mathcal{F}, \mathbf{X})$ be a bounded modular root instance, and let $\mathbf{r} \in \mathbb{Z}^{\ell}$ be a target bounded root. Fix integers $m, t \geq 1$ with $t \geq \max_a \deg(f_a)$. Write $n = |\mathcal{T}_t|$ and $\rho = \mathrm{lcm}\{X^{\alpha} : \alpha \in \mathcal{T}_t\}$. For input $(p, m, t, \mathcal{F}, \mathbf{X})$, Algorithm 2 outputs $\mathbf{r}$ if*

$$\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp}) > \delta^{2(n-1)}\rho \cdot \sqrt{n} \ ,$$

where $\delta \geq 1$ is a root-Hermite factor of an LLL algorithm used.

Proof. By Lemma 3.2, we have $\mathbf{v_r} \in \mathcal{L}_{\mathbf{X}}^{\perp}$ and $\|\mathbf{v_r}\| \leq \rho\sqrt{n}$. Let $\mathbf{w}$ be the first LLL-reduced basis vector. The LLL first-vector bound gives

$$\|\mathbf{w}\| \leq \delta^{2(n-1)}\lambda_1(\mathcal{L}_{\mathbf{X}}^{\perp}) \leq \delta^{2(n-1)}\rho\sqrt{n} < \lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp}).$$

Since $\|\mathbf{w}\| \geq \lambda_1(\mathcal{L}_{\mathbf{X}}^{\perp})$, the same bound implies $\delta^{2(n-1)} \geq 1$. Hence $\mathbf{v_r}$ also has norm below $\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp})$. If $\mathbf{w}$ and $\mathbf{v_r}$ were linearly independent, they would contradict the definition of the second successive minimum. Thus $\mathbf{w} \in \mathbb{R}\mathbf{v_r}$. The root line identity gives $\mathbf{w} = c\mathbf{v_r}$ for some $c \in \mathbb{Z} \setminus \{0\}$.

The constant coordinate of $\mathbf{v_r}$ is ρ, so $w_0 = c\rho \neq 0$. For each $i \in [\ell]$, we obtain

$$X_i \frac{w_{e_i}}{w_0} = X_i \frac{c(\rho/X_i)r_i}{c\rho} = r_i.$$

Thus, Algorithm 2 recovers $\mathbf{r}$, which passes the integrality, bound, and modular verification checks. □

Theorem 3.5 (Time complexity of Algorithm 2)**.** *Let $(p, \mathcal{F}, \mathbf{X})$ be a bounded modular root instance. Assume that every coefficient of $\mathcal{F}$ has absolute value at most p. Fix integers $m, t \geq 1$ with $t \geq \max_a \deg(f_a)$. Then Algorithm 2 terminates in* $\mathrm{poly}(\log p,\ \ell,\ \max_{i\in[\ell]} \log X_i, |\mathcal{F}|)$.

Proof. Set $s = |\mathcal{F}|$ and $n = |\mathcal{T}_t| = \binom{\ell+t}{\ell} = O(\ell^t)$. The definition of $\mathcal{S}_{m,t}$ gives at most $\binom{s+m}{m}$ choices of $\mathbf{i}$ and at most n choices of β for each $\mathbf{i}$. Hence $|\mathcal{S}_{m,t}| \leq \binom{s+m}{m} n = O(s^m \ell^t)$, so A has polynomially many rows and columns for fixed m, t.

Each input polynomial has at most n monomials and coefficients bounded in absolute value by p. The coefficients of each shift polynomial have bit lengths $O(m(\log p + \log n))$. Thus A can be constructed in polynomial time. Since $\log(p^m) = O(\log p)$, a basis of the modular kernel K can be computed in polynomial time using Smith normal form, with basis entries of polynomial bit length.

Every X^α with $\alpha \in \mathcal{T}_t$ divides $(X_1 \cdots X_\ell)^t$. Consequently,

$$\log \rho \leq t \sum_{i=1}^{\ell} \log X_i \leq t\ell \max_{i\in[\ell]} \log X_i.$$

The least common multiple ρ and all scaling factors ρ/X^α can therefore be computed in polynomial time and polynomial bit length. Hence the scaled lattice basis also has polynomial bit length. LLL reduction, coordinate ratio extraction, and the integrality, bound, and modular verification checks all take polynomial time in the stated parameters. □

Space complexity For fixed m, t, Algorithm 2 also uses polynomial space in the parameters of Theorem 3.5. The shift matrix and the lattice basis have polynomially many entries of polynomial bit-length. Using polynomial-bit-time implementations of modular kernel computation and LLL reduction therefore requires only polynomial space. Coefficient scaling, coordinate-ratio decoding, and verification also use polynomial space.

4 Analysis of the second minimum

This section derives a product bound for the dual Coppersmith algorithm by estimating the second successive minimum of $\mathcal{L}_{\mathbf{X}}^{\perp}$. The key idea is to isolate the root line through a sufficiently large length gap between the root vector and every lattice vector independent of it. Theorem 3.4 guarantees root recovery whenever

$$\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp}) > \delta^{2(n-1)} \rho \sqrt{n}, \tag{4}$$

where $\delta \geq 1$ is the root-Hermite factor of the LLL algorithm. We use the construction from Sections 2 and 3 with $t \geq m \geq 1$ and $t \geq \max_a \deg(f_a)$, retaining the notation $n = |\mathcal{T}_t|$ and $B_t = \binom{\ell+t}{\ell+1}$. Throughout this section, $\mathbf{r}$ is a bounded target root, and π denotes orthogonal projection onto $(\mathbb{R}\mathbf{v_r})^{\perp}$.

For affine relations with $t = m$, the complete shift construction retains the multiplicative structure of the degree-one modular orthogonal lattice. The complete-power comparison in Proposition D.1 shows that first-jet directions determine the second-minimum scale when the shortest nonzero projection of the degree-one lattice is asymptotically longer than its primitive root vector. Applying the Gaussian heuristic to this smaller projected lattice then gives the estimate λ_{jet}.

For $t > m$, additional monomial shifts change the lattice construction and can alter the relative sizes of the structured and generic length scales. The generic estimate λ_{GH} is obtained by applying the Gaussian heuristic to the full projected lattice $\pi(\mathcal{L}_{\mathbf{X}}^{\perp})$. These considerations motivate a model that accounts for both scales through their minimum.

Heuristic 4.1 (Second-minimum model)**.** *For the modular orthogonal lattice $\mathcal{L}_{\mathbf{X}}^{\perp}$ associated with a target root $\mathbf{r}$, we predict*

$$\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp}) \approx \lambda_2^{\mathrm{pred}} := \min\{\lambda_{\mathrm{jet}}, \lambda_{\mathrm{GH}}\}.$$

The model assumes that the shortest nonzero vectors of $\pi(\mathcal{L}_{\mathrm{jet}})$ occur at the scale λ_{jet}, and that projections of vectors outside $\mathcal{L}_{\mathrm{jet}}$ are not substantially shorter than $\lambda_2^{\mathrm{pred}}$. Here $\mathcal{L}_{\mathrm{jet}}$ is defined in Section 4.2. For fixed ℓ, m, t, we apply this prediction in the range $\lambda_2^{\mathrm{pred}}/\rho \to \infty$ as $p \to \infty$.

The experimental results in Section 5 are consistent with this two-scale prediction. Sections 4.1 and 4.2 derive the two estimates. The product-bound calculation below shows that both predicted lengths grow faster than the recovery threshold in (4). Under Heuristic 4.1, their minimum therefore gives the recovery prediction stated in the following corollary.

Corollary 4.2 (Product bound for Dual Coppersmith)**.** *Let $(p, \mathcal{F}, \mathbf{X})$ be a bounded modular root instance with a bounded target root $\mathbf{r} \in \mathbb{Z}^{\ell}$. Write $\mathcal{F} = (f_1, f_2, \cdots, f_s) \subset \mathbb{Z}[x_1, \ldots, x_{\ell}]$, and fix an integer $d \geq 1$ with $\max_a \deg(f_a) \leq d$. Suppose that the Jacobian matrix $(\partial_i f_a(\mathbf{r}))_{a\in[s],\, i\in[\ell]}$ has positive rank over $\mathbb{F}_p$. Assume Gaussian Heuristic 2.3 and Heuristic 4.1. For any $0 < \varepsilon < 1/d$, choose m sufficiently large, depending only on ℓ, d, ε, and set $t = dm$. If p is sufficiently large and*

$$\prod_{i=1}^{\ell} X_i < p^{1/d-\varepsilon},$$

then the model predicts the gap condition (4) for the lattice $\mathcal{L}_{\mathbf{X}}^{\perp}$ associated to the input $(p, m, t, \mathcal{F}, \mathbf{X})$. Consequently, whenever the predicted gap holds for the actual lattice, Algorithm 2 outputs $\mathbf{r}$.

Under the stated heuristics, the dual construction matches the $1/d$ product-bound exponent of the primal total-degree Coppersmith algorithm. Once the actual gap condition (4) holds, the root is recovered directly from coordinate ratios, without an algebraic root-recovery step over $\mathbb{Z}$.

4.1 The full projected estimate

To model $\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp})$, we first estimate the generic length scale of lattice vectors independent of the root line. Then, we obtain the full projected estimate by removing the root line and applying the Gaussian heuristic to the resulting lattice of rank $n-1$. Every lattice vector outside the root line has a nonzero projection, so this projected lattice describes directions relevant to the second minimum.

Recall the module $\mathcal{R}_{m,t}$ generated modulo p^m by the coefficient vectors of the shift polynomials from Section 2.3, and the scaled root vector $\mathbf{v}_{\mathbf{r}}$ from Section 3.1. By Lemma 3.2, the vector $\mathbf{v}_{\mathbf{r}}$ generates all lattice points on its line and is therefore primitive in $\mathcal{L}_{\mathbf{X}}^{\perp}$. Extending it to a lattice

basis and projecting the remaining basis vectors gives the volume identity

$$\det(\pi(\mathcal{L}_{\mathbf{X}}^{\perp})) = \frac{\det(\mathcal{L}_{\mathbf{X}}^{\perp})}{\|\mathbf{v_r}\|} = \frac{|\mathcal{R}_{m,t}|\rho^n}{\|\mathbf{v_r}\| \left(\prod_{i=1}^{\ell} X_i\right)^{B_t}},$$

where the last equality follows from Lemma 3.3. Applying Gaussian Heuristic 2.3 defines

$$\lambda_{\mathrm{GH}} := \sqrt{\frac{n-1}{2\pi e}} \left(\frac{\det(\mathcal{L}_{\mathbf{X}}^{\perp})}{\|\mathbf{v_r}\|}\right)^{1/(n-1)}. \tag{5}$$

This estimate uses the volume of the full projected lattice. The next subsection accounts for structured directions whose lengths may differ from this generic scale.

4.2 The first-jet estimate

The full projected estimate depends only on the lattice covolume and may miss short vectors arising from its algebraic structure. In our construction, multiplying each first derivative of the scaled monomial evaluation vector by p gives a lattice vector independent of the root line: differentiating a shift polynomial at the root can remove at most one factor of p, which this multiplication restores. These vectors may be shorter than the full projected estimate. We therefore study the lattice in the span of the root vector and its first derivatives, called the first-jet lattice, and estimate its shortest nonzero projection to account for these competing directions.

We restrict the local derivative-based analysis to first-order directions. Appendix B explains why, in the parameter regime of Corollary 4.2, pure higher-order derivative generators do not improve upon the first-jet scale. This comparison does not exclude cancellations between different derivative orders or other structured sublattices.

Recall the evaluation vectors $\mathbf{z_r}$ and $\mathbf{v_r}$ from Section 3.1. For $i \in [\ell]$, write $\partial_i = \partial/\partial x_i$. Define

$$(\partial_i \mathbf{z_r})_\alpha = \begin{cases} \alpha_i\, \mathbf{r}^{\alpha - e_i}, & \alpha_i > 0, \\ 0, & \alpha_i = 0, \end{cases} \qquad (\partial_i \mathbf{v_r})_\alpha = \frac{\rho}{X^\alpha}(\partial_i \mathbf{z_r})_\alpha.$$

The first-jet lattice is

$$\mathcal{L}_{\mathrm{jet}} = \mathcal{L}_{\mathbf{X}}^{\perp} \cap \mathrm{span}_{\mathbb{R}}\{\mathbf{v_r}, \partial_1 \mathbf{v_r}, \ldots, \partial_\ell \mathbf{v_r}\}.$$

To determine which derivative combinations belong to this lattice, define the Jacobian and its rank over $\mathbb{F}_p$ as

$$\mathrm{Jac}_{\mathcal{F}}(\mathbf{y}) = (\partial_i f_a(\mathbf{y}))_{a\in[s],\, i\in[\ell]}, \quad \mathrm{rk}_{\mathcal{F}}(\mathbf{y}) = \mathrm{rank}_{\mathbb{F}_p} \mathrm{Jac}_{\mathcal{F}}(\mathbf{y}).$$

We now characterize exactly which integral combinations of the derivative directions lie in $\mathcal{L}_{\mathrm{jet}}$. The p-scaled derivative directions and the rank of $\mathcal{L}_{\mathrm{jet}}$ will follow as immediate consequences.

Lemma 4.3 (Derivative congruence). *For* $\mathbf{b} = (b_1, \ldots, b_\ell)^\top \in \mathbb{Z}^\ell$,

$$\sum_{i=1}^{\ell} b_i \partial_i v_{\mathbf{r}} \in \mathcal{L}_{\mathbf{X}}^{\perp} \quad \Longleftrightarrow \quad \sum_{i=1}^{\ell} b_i \partial_i f_a(\mathbf{r}) \equiv 0 \pmod{p} \quad (a \in [s]).$$

Proof. The congruences on the right imply that the directional derivative of $\mathbf{x}^\beta \mathbf{f}^{\mathbf{i}}$ in direction $\mathbf{b}$ is divisible by $p^{|\mathbf{i}|}$ when evaluated at $\mathbf{r}$. Indeed, when the derivative acts on $\mathbf{x}^\beta$, the product retains all $|\mathbf{i}|$ factors that vanish modulo p. When it acts on a factor f_a, the corresponding Jacobian congruence

supplies the lost factor of p. Therefore, multiplying by $p^{m-|\mathbf{i}|}$ makes the pairing of every coefficient row with $\sum_{k=1}^{\ell} b_k \partial_k \mathbf{z_r}$ vanish modulo p^m.

Conversely, the row $p^{m-1} f_a$ is present for every $a \in [s]$, since $(e_a, 0) \in \mathcal{S}_{m,t}$. Applying the modular kernel condition to this row yields $p^{m-1} \sum_{k=1}^{\ell} b_k \partial_k f_a(\mathbf{r}) \equiv 0 \pmod{p^m}$, which implies the required Jacobian congruence modulo p, for every $a \in [s]$. □

Lemma 4.3 determines which integer coefficient vectors produce derivative combinations in $\mathcal{L}_{\mathbf{X}}^{\perp}$. These coefficient vectors form the lattice

$$K_{\text{jet}} = \left\{ \mathbf{b} \in \mathbb{Z}^{\ell} : \text{Jac}_{\mathcal{F}}(\mathbf{r})\mathbf{b} \equiv 0 \pmod{p} \right\}.$$

Equivalently, K_{jet} is the kernel of the reduction modulo p of the Jacobian map $\mathbf{b} \mapsto \text{Jac}_{\mathcal{F}}(\mathbf{r})\mathbf{b}$. Hence $[\mathbb{Z}^{\ell} : K_{\text{jet}}] = p^{\text{rk}_{\mathcal{F}}(\mathbf{r})}$.

The defining congruences also imply $p\mathbb{Z}^{\ell} \subseteq K_{\text{jet}}$. Hence Lemma 4.3 shows that $p\partial_i \mathbf{v_r} \in \mathcal{L}_{\text{jet}}$ for every $i \in [\ell]$. Together with the root vector, we obtain

$$\mathbf{v_r}, p\partial_1 \mathbf{v_r}, \ldots, p\partial_{\ell} \mathbf{v_r}.$$

On the coordinates $0, e_1, \ldots, e_{\ell}$, the matrix formed by these vectors as columns is triangular with nonzero diagonal entries $\rho, p\rho/X_1, \ldots, p\rho/X_{\ell}$. Thus the vectors are linearly independent, and $\mathcal{L}_{\text{jet}}$ has rank $\ell + 1$. Projecting away from the root line leaves a lattice of rank ℓ.

To apply Gaussian Heuristic 2.3 to this projected lattice, we next compute its determinant. The coefficient index above accounts for the Jacobian constraints. To compute the Euclidean norm of the projected derivative vectors, let

$$\mathbf{U}_i = \pi(\partial_i \mathbf{v_r}) \qquad (i \in [\ell]).$$

We express these vectors in normalized coordinates to separate their scaling factors from the position of the target root. Put $\mathbf{u} = (r_1/X_1, \ldots, r_{\ell}/X_{\ell}) \in \mathbb{R}^{\ell}$, and let $J_{t,\mathbf{r}} \in \mathbb{R}^{n \times \ell}$ be the matrix whose (α, i)-entry is $\partial \mathbf{u}^{\alpha} / \partial u_i$. Define a matrix

$$\Pi_{t,\mathbf{r}} = I_n - \frac{(\mathbf{u}^{\alpha})_{\alpha \in \mathcal{T}_t} (\mathbf{u}^{\alpha})_{\alpha \in \mathcal{T}_t}^T}{\sum_{\alpha \in \mathcal{T}_t} (\mathbf{u}^{\alpha})^2} = I_n - \frac{\mathbf{v_r} \mathbf{v_r}^{\top}}{\|\mathbf{v_r}\|^2}.$$

Here $\Pi_{t,\mathbf{r}}$ represents the orthogonal projection π. The coordinate definitions imply

$$\mathbf{v_r} = \rho(\mathbf{u}^{\alpha})_{\alpha \in \mathcal{T}_t}, \qquad \mathbf{U}_i = \frac{\rho}{X_i} \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}} e_i.$$

The following proposition combines the Gram determinant of these vectors with the coefficient index.

Proposition 4.4 (Volume of the jet lattice)**.** *Assume* $\text{rk}_{\mathcal{F}}(\mathbf{r}) \geq 1$. *Then*

$$\det(\mathcal{L}_{\text{jet}}) = \|v_{\mathbf{r}}\| \; p^{\text{rk}_{\mathcal{F}}(\mathbf{r})} \frac{\rho^{\ell}}{\prod_{i=1}^{\ell} X_i} \sqrt{\det\left(J_{t,\mathbf{r}}^T \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}}\right)}.$$

Proof. We first describe the projected lattice in terms of K_{jet}. Take $\mathbf{w} \in \mathcal{L}_{\text{jet}}$ and write

$$\mathbf{w} = c\mathbf{v_r} + \sum_{i=1}^{\ell} y_i \partial_i \mathbf{v_r}, \qquad c, y_i \in \mathbb{R}.$$

Since $\mathbf{w} \in \mathcal{L}_{\mathbf{X}}^{\perp}$, its coordinates have the form $w_\alpha = (\rho/X^\alpha)z_\alpha$ with integers z_α. Examining the 0-coordinate yields $c \in \mathbb{Z}$. Subsequently, examining the e_i-coordinate yields $cr_i + y_i \in \mathbb{Z}$, which forces $y_i \in \mathbb{Z}$ since both c and r_i are integers.

Subtracting $c\mathbf{v_r}$ and applying Lemma 4.3 shows that $\sum_{i=1}^{\ell} y_i \partial_i \mathbf{v_r} \in \mathcal{L}_{\mathbf{X}}^{\perp}$. Conversely, every combination with $c \in \mathbb{Z}$ and $\mathbf{b} \in K_{\text{jet}}$ lies in $\mathcal{L}_{\text{jet}}$. Therefore,

$$\pi(\mathcal{L}_{\text{jet}}) = \left\{ \sum_{i=1}^{\ell} b_i \mathbf{U}_i \;\middle|\; \mathbf{b} \in K_{\text{jet}} \right\}.$$

The projected lattice volume is this index times the covolume of the lattice generated by $\mathbf{U}_1, \dots, \mathbf{U}_\ell$ and the independence established above implies that $\mathbf{U}_1, \dots, \mathbf{U}_\ell$ are linearly independent. The index of $\pi(\mathcal{L}_{\text{jet}})$ in the lattice generated by these vectors equals $[\mathbb{Z}^\ell : K_{\text{jet}}] = p^{\text{rk}_{\mathcal{F}}(\mathbf{r})}$. Therefore, it remains to compute their covolume, which is given by the square root of the determinant of their Gram matrix, namely $\sqrt{\det((\langle U_i, U_j \rangle)_{i,j})}$.

Using $\mathbf{U}_i = (\rho/X_i)\Pi_{t,\mathbf{r}} J_{t,\mathbf{r}} e_i$ and $\Pi_{t,\mathbf{r}}^\top \Pi_{t,\mathbf{r}} = \Pi_{t,\mathbf{r}}$, we obtain

$$\langle \mathbf{U}_i, \mathbf{U}_j \rangle = \frac{\rho^2}{X_i X_j} \left(J_{t,\mathbf{r}}^\top \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}} \right)_{i,j}.$$

Factoring ρ/X_i from row i and column i of the Gram matrix contributes $\left(\rho^\ell / \prod_i X_i\right)^2$ to its determinant. Taking the square root and multiplying by the index computed above, we obtain

$$\begin{aligned} \det(\pi(\mathcal{L}_{\text{jet}})) &= p^{\text{rk}_{\mathcal{F}}(\mathbf{r})} \sqrt{\det\left((\langle \mathbf{U}_i, \mathbf{U}_j \rangle)_{i,j=1}^{\ell} \right)} \\ &= p^{\text{rk}_{\mathcal{F}}(\mathbf{r})} \frac{\rho^\ell}{\prod_{i=1}^{\ell} X_i} \sqrt{\det\left(J_{t,\mathbf{r}}^\top \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}} \right)} \end{aligned}$$

Finally, we include the root-line direction to compute the volume of $\mathcal{L}_{\text{jet}}$. The primitive root-line property from Section 3.1 implies $\mathcal{L}_{\text{jet}} \cap \mathbb{R}\mathbf{v_r} = \mathbb{Z}\mathbf{v_r}$. Thus $\mathbf{v_r}$ can be extended to a basis of $\mathcal{L}_{\text{jet}}$, and the projection determinant identity gives $\det(\mathcal{L}_{\text{jet}}) = \|\mathbf{v_r}\| \det(\pi(\mathcal{L}_{\text{jet}}))$. □

Applying Heuristic 2.3 to this rank-ℓ projected jet lattice gives the local structured estimate

$$\lambda_{\text{jet}} := \sqrt{\frac{\ell}{2\pi e}} \left(\frac{\det(\mathcal{L}_{\text{jet}})}{\|v_{\mathbf{r}}\|} \right)^{1/\ell} = \rho \sqrt{\frac{\ell}{2\pi e}} \left(\frac{p^{\text{rk}_{\mathcal{F}}(\mathbf{r})}}{\prod_{i=1}^{\ell} X_i} \sqrt{\det\left(J_{t,\mathbf{r}}^T \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}} \right)} \right)^{1/\ell}. \tag{6}$$

In the proof of Corollary 4.2, we need a lower bound for λ_{jet} that depends only on the product $\prod_i X_i$ and the normalized root $\mathbf{u} = \mathbf{r}/\mathbf{X}$. The following lemma provides precisely this auxiliary estimate.

Lemma 4.5 (Projected derivative lower bound)**.** *With the notation above,*

$$\sqrt{\det(J_{t,\mathbf{r}}^T \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}})} \geq \frac{1}{\sqrt{1 + \|\mathbf{u}\|^2}}.$$

Proof. For $y \in \mathbb{R}^\ell$,

$$y^T J_{t,\mathbf{r}}^T \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}} y = \min_{a \in \mathbb{R}} \|J_{t,\mathbf{r}} y - a(\mathbf{u}^\alpha)_{\alpha \in \mathcal{T}_t}\|^2.$$

Restricting to the coordinates $0, e_1, \ldots, e_\ell$ can only decrease the minimum. On these coordinates, we have

$$(\mathbf{u}^\alpha)_{\alpha \in \mathcal{T}_t} = (1, u_1, \ldots, u_\ell)^T \qquad \text{and} \qquad J_{t,\mathbf{r}} = \begin{pmatrix} 0 \\ I_\ell \end{pmatrix}.$$

Thus the corresponding restricted projected Gram matrix is $I_\ell - \frac{\mathbf{u}\mathbf{u}^T}{1+\|\mathbf{u}\|^2}$, where here $\mathbf{u} = (u_1, \ldots, u_\ell)^T$. By the matrix determinant lemma,

$$\det\left(I_\ell - \frac{\mathbf{u}\mathbf{u}^T}{1+\|\mathbf{u}\|^2}\right) = 1 - \frac{\mathbf{u}^T\mathbf{u}}{1+\|\mathbf{u}\|^2} = \frac{1}{1+\|\mathbf{u}\|^2}.$$

Consequently, the restricted block contributes a factor of $(1+\|\mathbf{u}\|^2)^{-1}$, and the determinant of the full projected Gram matrix is bounded below by the same quantity. □

When $\mathrm{rk}_{\mathcal{F}}(\mathbf{r}) \geq 1$, substituting this uniform bound into (6) yields

$$\lambda_{\text{jet}} \geq \rho\sqrt{\frac{\ell}{2\pi e}}\left(\frac{p}{\sqrt{\ell+1}\prod_{i=1}^{\ell} X_i}\right)^{1/\ell}.$$

This lower bound depends on the root bounds only through $\prod_i X_i$ and is the first-jet estimate used to derive Corollary 4.2.

4.3 Product bound for Dual Coppersmith

We now prove Corollary 4.2 by showing that its product condition places both λ_{GH} and λ_{jet} above the recovery threshold in (4). The index bound of Lemma 2.7 controls the full projected estimate, while positive Jacobian rank and Lemma 4.5 control the first-jet estimate.

Proof of Corollary 4.2. Positive Jacobian rank implies that $f_a \bmod p \neq 0$ for at least one a. Thus, Lemma 2.7 gives

$$|\mathcal{R}_{m,t}| \geq p^{S_m}, \qquad S_m = \sum_{j=1}^{m} \binom{\ell + t - jd}{\ell}$$

and shows that $S_m/B_t \to 1/d$ as $m \to \infty$ with $t = dm$, where $B_t = \binom{\ell+t}{\ell+1}$. By Lemma 3.2, we have $\|\mathbf{v}_\mathbf{r}\| \leq \rho\sqrt{n}$. Combining Lemma 3.3, the index bound, and (5) yields

$$\lambda_{\text{GH}} \geq \rho\sqrt{\frac{n-1}{2\pi e}}\left(\frac{p^{S_m}}{\sqrt{n}\left(\prod_{i=1}^{\ell} X_i\right)^{B_t}}\right)^{1/(n-1)},$$

where $n = |\mathcal{T}_t|$. Thus $\lambda_{\text{GH}} > \delta^{2(n-1)}\rho\sqrt{n}$ whenever

$$\prod_{i=1}^{\ell} X_i < \left[\frac{1}{\sqrt{n}}\left(\frac{\sqrt{(n-1)/(2\pi e)}}{\delta^{2(n-1)}\sqrt{n}}\right)^{n-1}\right]^{1/B_t} p^{S_m/B_t}.$$

For the first-jet estimate, positive Jacobian rank and Lemma 4.5, applied to (6), imply

$$\lambda_{\text{jet}} \geq \rho\sqrt{\frac{\ell}{2\pi e}}\left(\frac{p}{\sqrt{\ell+1}\prod_{i=1}^{\ell} X_i}\right)^{1/\ell}.$$

Hence $\lambda_{\text{jet}} > \delta^{2(n-1)}\rho\sqrt{n}$ whenever

$$\prod_{i=1}^{\ell} X_i < \frac{1}{\sqrt{\ell+1}} \left(\frac{\sqrt{\ell/(2\pi e)}}{\delta^{2(n-1)}\sqrt{n}} \right)^{\ell} p.$$

These two product thresholds have exponent S_m/B_t and 1 respectively. Choose m, depending only on ℓ, d, ε, so that

$$\frac{S_m}{B_t} \geq \frac{1}{d} - \frac{\varepsilon}{2}.$$

Both exponents then strictly exceed $1/d - \varepsilon$, since $d \geq 1$ and $\varepsilon > 0$. For this choice of m, the prefactors in both thresholds are positive and independent of p and $\mathbf{X}$. Therefore, for sufficiently large p, the product condition

$$\prod_{i=1}^{\ell} X_i < p^{1/d-\varepsilon}$$

satisfies both thresholds. Consequently,

$$\lambda_2^{\text{pred}} = \min\{\lambda_{\text{GH}}, \lambda_{\text{jet}}\} > \delta^{2(n-1)}\rho\sqrt{n}.$$

Therefore, Heuristic 4.1 predicts the gap in (4). Whenever this gap holds for the actual lattice, Theorem 3.4 guarantees that Algorithm 2 outputs $\mathbf{r}$. □

4.4 Extension to a composite modulus

We extend the recovery guarantee and the product-bound analysis to a public modulus whose relevant divisor is unknown, as required by the experiments below. We retain n, B_t, ρ, $\mathbf{z_r}$, and $\mathbf{v_r}$ from the preceding sections.

Theorem 4.6 (Recovery over a public modulus)**.** *Let $N \geq 2$, $0 < \beta \leq 1$, and $\max_a \deg(f_a) \leq d$ for an integer $d \geq 1$. Fix integers $m, t \geq 1$ with $t \geq dm$, and construct $\mathcal{L}_{\mathbf{X}}^{\perp}$ modulo N^m from the total-degree shifts $g_{\mathbf{i},\boldsymbol{\beta}} = N^{m-|\mathbf{i}|}\mathbf{x}^{\boldsymbol{\beta}}\mathbf{f}^{\mathbf{i}}$. For a bounded target $\mathbf{r}$, put $p = \gcd(N, f_1(\mathbf{r}), \ldots, f_s(\mathbf{r})) \geq N^{\beta}$ and $q_0 = N/p$. Then*

$$\mathcal{L}_{\mathbf{X}}^{\perp} \cap \mathbb{R}\mathbf{v_r} = \mathbb{Z}q_0^m\mathbf{v_r}.$$

Suppose reduction returns a nonzero vector $\mathbf{w}$ satisfying $\|\mathbf{w}\| \leq \delta^{2(n-1)}\lambda_1(\mathcal{L}_{\mathbf{X}}^{\perp})$ for some $\delta \geq 1$. If

$$\lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp}) > \delta^{2(n-1)}q_0^m\|\mathbf{v_r}\|,$$

then $w_0 \neq 0$ and $r_i = X_i w_{e_i}/w_0$ for every $i \in [\ell]$. No primality assumption on p is required.

Proof. Let A be the public shift coefficient matrix. The construction uses

$$K = \{\mathbf{z} \in \mathbb{Z}^n : A\mathbf{z} \equiv 0 \pmod{N^m}\}, \qquad \mathcal{L}_{\mathbf{X}}^{\perp} = \{((\rho/X^{\alpha})z_{\alpha})_{\alpha \in T_t} : \mathbf{z} \in K\}.$$

Every shift is divisible by p^m at $\mathbf{r}$, so $N^m \mid q_0^m g_{\mathbf{i},\boldsymbol{\beta}}(\mathbf{r})$ and $q_0^m\mathbf{v_r} \in \mathcal{L}_{\mathbf{X}}^{\perp}$.

To prove primitivity on the target line, suppose $c\mathbf{v_r} \in \mathcal{L}_{\mathbf{X}}^{\perp}$. Its unscaled constant coordinate gives $c \in \mathbb{Z}$. Since $t \geq dm$, every f_a^m is a shift, so $N^m \mid cf_a(\mathbf{r})^m$ for all a. The identity $\gcd(N^m, f_1(\mathbf{r})^m, \ldots, f_s(\mathbf{r})^m) = p^m$ then implies $q_0^m \mid c$, proving the asserted intersection.

For recovery, the reduction bound and the gap give $\|\mathbf{w}\| \leq \delta^{2(n-1)}q_0^m\|\mathbf{v_r}\| < \lambda_2(\mathcal{L}_{\mathbf{X}}^{\perp})$. Since $\delta \geq 1$, the vector $q_0^m\mathbf{v_r}$ is also shorter than λ_2. The two vectors must therefore be collinear. The constant coordinate of $\mathbf{v_r}$ is $\rho \neq 0$, and its e_i-coordinate is $(\rho/X_i)r_i$. Thus, their ratios give the result. □

We next state the product bound for the Gaussian heuristic of the full and first-jet projections of the public lattice.

Corollary 4.7 (Product bound for the public lattice)**.** *Fix $\ell, d \geq 1$, $\ell/(\ell+1) < \beta \leq 1$, and $\delta \geq 1$ independent of N and $\mathbf{X}$. In the setting of Theorem 4.6, suppose $\mathrm{Jac}_{\mathcal{F}}(\mathbf{r})$ has positive rank modulo every prime dividing p. For every $0 < \varepsilon < ((\ell+1)\beta - \ell)/d$, choose m sufficiently large depending only on ℓ, d, ε, and set $t = dm$. For sufficiently large N, the condition*

$$\prod_{i=1}^{\ell} X_i < N^{((\ell+1)\beta-\ell)/d-\varepsilon}$$

implies

$$\min\{\lambda_{\mathrm{GH}}, \lambda_{\mathrm{jet}}\} > \delta^{2(n-1)} q_0^m \rho\sqrt{n},$$

where both estimates are computed from the public lattice.

Proof. The full projected covolume follows from the primitive-vector identity in Theorem 4.6. Write $\mathcal{R}_{m,t}$ for the row module of A modulo N^m. The Smith-normal-form argument of Lemma 3.3 remains valid modulo N^m, and therefore

$$\det\bigl(\pi(\mathcal{L}_{\mathbf{X}}^{\perp})\bigr) = \frac{|\mathcal{R}_{m,t}|\rho^n}{q_0^m \|\mathbf{v_r}\| (\prod_i X_i)^{B_t}}.$$

The first-jet projection requires mixed coefficients because $\mathbf{v_r}$ itself need not lie in the public lattice. Its coefficient lattice is

$$K_{\mathrm{jet}} = \left\{ \mathbf{b} \in \mathbb{Z}^{\ell} : \exists c \in \mathbb{Z}, \quad A\left(c\mathbf{z_r} + \sum_i b_i \partial_i \mathbf{z_r} \right) \equiv 0 \pmod{N^m} \right\}.$$

Here c is common to all rows of A. The Gram-matrix calculation of Section 4.2 gives

$$\det\bigl(\pi(\mathcal{L}_{\mathrm{jet}})\bigr) = [\mathbb{Z}^{\ell} : K_{\mathrm{jet}}] \frac{\rho^{\ell}}{\prod_i X_i} \sqrt{\det(J_{t,\mathbf{r}}^{\mathsf{T}} \Pi_{t,\mathbf{r}} J_{t,\mathbf{r}})}.$$

Thus $[\mathbb{Z}^{\ell} : K_{\mathrm{jet}}]$ replaces the prime-field factor $p^{\mathrm{rk}_{\mathcal{F}}(\mathbf{r})}$. The two Gaussian estimates use these covolumes in ranks $n-1$ and ℓ.

The row-module bound is $|\mathcal{R}_{m,t}| \geq N^{S_m}$. Indeed, for each prime dividing N, some input polynomial is nonzero modulo that prime: this follows from the Jacobian assumption for primes dividing p, and from the definition of p otherwise. Applying the proof of Lemma 2.7 at each prime power and combining the indices proves the bound.

The derivative congruences give $[\mathbb{Z}^{\ell} : K_{\mathrm{jet}}] \geq q_0^{(m-1)\ell} p$. For $m \geq 2$, compare the rows $N f^{m-1}$, $N x_i f^{m-1}$, and $f^{m-1} f_a$ at each prime divisor of N, choosing f with minimum valuation at $\mathbf{r}$ when that valuation is smaller than the exponent in N. At prime powers fully dividing p, the rows $N^{m-1} f_a$ suffice; the case $m = 1$ follows from the rows f_a. These comparisons imply $K_{\mathrm{jet}} \subseteq q_0^{m-1}\{\mathbf{b} \in \mathbb{Z}^{\ell} : \mathrm{Jac}_{\mathcal{F}}(\mathbf{r})\mathbf{b} \equiv 0 \pmod{p}\}$. The local Jacobian assumption makes the index of the latter lattice at least $q_0^{(m-1)\ell} p$.

Combining these index bounds with Lemma 4.5 and $\|\mathbf{v_r}\| \leq \rho\sqrt{n}$ gives the product-exponent comparisons of Section 4.3, now with the factor q_0^m in the recovery threshold. The full-projection comparison has exponent $(S_m - (1-\beta)mn)/B_t$, whereas the jet comparison has exponent $(\ell+1)\beta - \ell$. Choose m so that $S_m/B_t > 1/d - \varepsilon/2$ and set $t = dm$. Since $mn/B_t = (\ell+1)/d$, both exponents exceed $((\ell+1)\beta - \ell)/d - \varepsilon$. All remaining factors are positive constants independent of N and $\mathbf{X}$, so sufficiently large N gives the claimed inequalities. □

Corollary 4.7 compares the predicted lengths with the recovery threshold. For the recovery prediction, retain both geometric conditions of Heuristic 4.1 on the public lattice, with constants independent of N and $\mathbf{X}$ and separation regime $\lambda_2^{\mathrm{pred}}/(q_0^m\rho) \to \infty$. The product condition alone does not certify the actual gap required by Theorem 4.6. At $p = N$ and $\beta = 1$, the exponent reduces to $1/d$.

5 Experiments

The cost of the primal Coppersmith method is dominated by its RR stage, whose Gröbner computation is exponential in the number of variables. However, Algorithm 2 removes the RR stage and reads the root off the reduced basis by coordinate ratios. This section demonstrates the resulting gap empirically. The two problems below are only vehicles: the primal method inherits the Gröbner blow-up as the number of variables grows, while our method reduces recovery to polynomial-time linear algebra.

Because our method rests on the γ-uSVP assumption, we report two applications—the approximate common divisor problem (Section 5.1) and the Herrmann-May problem (Section 5.2). In each, we test the assumption on the reduced modular orthogonal lattice for λ_2. All experiments were run single-threaded on a machine with an Intel Core i9-14900KF CPU and 32 GB of RAM, running Ubuntu 24.04 LTS (Linux kernel 6.17) and SageMath 10.8, with a wall-clock timeout of 600 seconds per instance. The RR stage of the primal method uses SageMath's Gröbner computation.

In our experiments, primal runtimes are split into lattice reduction (LR) and Gröbner-based root recovery (RR), with 't/o' denoting a timeout (600s); Dual reports our full runtime. All vector lengths are reported on a $\log_2$ scale and refer to the actual public-modulus lattice constructed modulo N^m. We report exact second minima when the certifying search completes, and certified lower and upper bounds otherwise. These values are compared with the predictor $\lambda_2^{\mathrm{pred}} = \min\{\lambda_{\mathrm{GH}}, \lambda_{\mathrm{jet}}\}$.

We use $\delta_{\mathrm{emp}} = 1.02$ as a literature-based empirical reference for lattice reduction. Following the heuristic approximation model discussed in [GGM16], we compare the second minimum with

$$T_{\mathrm{emp}}(L) = \delta_{\mathrm{emp}}^{2n}\lambda_1(L).$$

This quantity serves as an empirical gap benchmark, rather than a worst-case approximation guarantee for the reduction algorithm. In every reported instance, the exact second minimum or its certified lower bound exceeds this benchmark. Successful root recovery is verified independently by coordinate-ratio decoding and application-specific verification.

The implementation of Algorithm 2, together with all scripts needed to reproduce the experiments in this section, is available at

https://github.com/kimjebin/Dual_Coppersmith

Remark 5.1 (`flatter`)**.** We utilize the `flatter` algorithm [RH23], which approximates LLL reduction to achieve significant speedups on large bases. While `flatter` outputs the exact same b_1 as LLL, the subsequent tail vectors $b_2, b_3, \ldots$ differ because it reduces more shallowly. Since our dual method only requires b_1 to satisfy the recovery condition, we can fully leverage `flatter`. Conversely, the primal algorithm requires $\ell + 1$ short vectors that fall below a strict Howgrave-Graham bound, which the looser tail of `flatter` fails to supply. Therefore, the primal baseline must employ LLL, while we use `flatter`. Although the main algorithmic difference between the two methods is the presence or absence of the RR stage, this distinction ultimately allows us to secure a substantial speed advantage during the LR stage as well.

Second-minimum validation. Using the planted root for experimental validation, we project the public-modulus lattice orthogonally away from its root line. Lower bounds are obtained from certified block-SVP bounds on bases of this projected lattice. We improve these bounds by considering unimodular changes of basis, and certify the resulting bounds using the original exact data.

Upper bounds are obtained from vectors of the original lattice that are linearly independent of the root line. When enumeration is incomplete, the shortest available witness supplies an upper bound rather than an exact second minimum. We report an exact value only when the search has certified that no shorter independent vector exists.

5.1 The approximate common divisor problem

As a first application we take the approximate common divisor (ACD) problem. We use the standard multivariate-polynomial formulation of ACD as a bounded modular root problem, following [HG01, CH13, GGM16].

Definition 5.2 (Approximate common divisor problem [HG01, GGM16])**.** Let p be a hidden η-bit integer and let $R = 2^{\omega}$. Given a modulus $N = pq_0$ and samples

$$a_i = pq_i + r_i \qquad (i \in [\ell]),$$

where $q_0, q_1, \ldots, q_\ell$ are unknown and $|r_i| < R$, the task is to recover p.

Lemma 5.3 (ACD as bounded modular roots [HG01, CH13, GGM16])**.** *For an ACD instance as in Definition 5.2, the hidden error vector $\mathbf{r} = (r_1, \ldots, r_\ell)$ is the planted bounded root of the relations*

$$f_i(\mathbf{x}) = x_i - a_i \equiv 0 \pmod{p} \qquad (i \in [\ell]), \qquad |x_i| < R,$$

an instance of the bounded modular root problem (Definition 2.4). Conversely, recovering such a bounded root $\mathbf{r}$ gives a nontrivial divisor

$$\gcd(N, a_1 - r_1, \ldots, a_\ell - r_\ell)$$

of N, which equals p up to the usual cofactor on the planted root.

For the Coppersmith step, we use the standard ACD shift polynomials

$$g_{\mathbf{i}} = N^{\max\{m-|\mathbf{i}|,0\}} \prod_{j=1}^{\ell} (x_j - a_j)^{i_j}, \qquad |\mathbf{i}| \le t,$$

as in the multivariate-polynomial approach to ACD [CH13, GGM16]. These shifts vanish at the planted error vector modulo p^m. We form A from their coefficient vectors and construct $\mathcal{L}^{\perp}_{\mathbf{X}}$ over N^m as in Section 4.4. Under its gap condition, coordinate-ratio decoding recovers $\mathbf{r}$, from which the ACD divisor is obtained as in Lemma 5.3.

We refer to [CH13, GGM16] for the shift construction and its solvability analysis. In the experiments we vary only the degree t, the multiplicity m (with $t - m \in \{0, 1\}$, the useful range identified in [GGM16]), and the number of samples ℓ.

Remark 5.4 (Monomial multipliers)**.** In the coordinates $y_i = x_i - a_i$ the ACD relations are simply $f_i = y_i$, and the shifts $g_{\mathbf{i}} = N^{\max\{m-|\mathbf{i}|,0\}} \prod_j y_j^{i_j}$ are pure powers of the y_i, without the monomial multipliers $\mathbf{x}^{\boldsymbol{\beta}}$ of the usual total-degree presentation $N^{m-|\mathbf{u}|}\mathbf{x}^{\boldsymbol{\beta}} \prod_j y_j^{u_j}$ $(1 \le |\mathbf{u}| \le m,\ |\boldsymbol{\beta}| + |\mathbf{u}| \le t)$. Here this costs nothing: since $y_i = x_i - a_i$ is an affine change of variables, $\mathbf{x}^{\boldsymbol{\beta}} = (\mathbf{y} + \mathbf{a})^{\boldsymbol{\beta}}$ expands

into y-powers, and the pure powers $\{y^{\mathbf{i}} : |\mathbf{i}| \le t\}$ already span every monomial of degree at most t; the two presentations therefore define the *same* modular orthogonal lattice $\mathcal{L}^{\perp}_{\mathbf{X}}$. We checked this directly on the settings of Table 1 small enough to enumerate (identical Hermite normal form and determinant of $\mathcal{L}^{\perp}_{\mathbf{X}}$), so the recovered root and the minima λ_1, λ_2 are unchanged. The total-degree presentation only adds redundant rows: they leave $\mathcal{L}^{\perp}_{\mathbf{X}}$ the same but give a worse-conditioned basis, so it never reduces faster and is markedly slower at higher degree. The equivalence is special to ACD, where $f_i = x_i - a_i$ is itself a coordinate system.

Setup. We fix $\eta = 100$ and $\omega = 10$ (so $|r_i| < 2^{\omega}$) and keep $t \ge 2$, so that the recovery is genuinely nonlinear: at $t = 1$ the shifts are linear and no Gröbner basis is needed. For each (t, m, ℓ) we size the modulus at 90% of the largest value [GGM16] predicts solvable, so that each instance sits in its own solvable regime, and we increase ℓ until the primal algorithm reaches the timeout.

Results. Table 1 reports the outcome for the four parameter settings. Once $t \ge 2$, the baseline's Gröbner stage grows super-polynomially in the number of variables ℓ and crosses the timeout at a threshold. Our method recovers the root in every case, and its running time grows only polynomially with the dimension—so on the timed-out instances it is the only method of the two that solves them at all.

Table 1: ACD recovery time and second-minimum validation ($\eta = 100, \omega = 10$). For λ_2, a single entry spanning the Lower and Upper columns reports an exact computation; otherwise, the entries give certified bounds, rounded outward to one decimal place. In every row, the exact value or certified lower bound exceeds the empirical recovery threshold $\delta^{2n}\lambda_1$ with $\delta = 1.02$.

			time (s)			vector norm ($\log_2$)					
			Primal					λ_2			
(t, m)	ℓ	dim	LR	RR	Dual	λ_1	$\delta^{2n}\lambda_1$	Lower	Upper	λ_{GH}	λ_{jet}
$(2, 1)$	10	66	1.6	35.4	2.2	4309.5	4313.3	4324.0		4324.0	4398.2
$(2, 1)$	11	78	3.3	95.0	3.5	5056.7	5061.2	5071.1	5072.1	5071.6	5145.3
$(2, 1)$	12	91	6.8	280.0	5.8	5853.3	5858.5	5868.1	5869.8	5868.9	5942.1
$(2, 1)$	13	105	13.8	**t/o**	**8.9**	6698.5	6704.5	6713.3	6715.7	6714.5	6787.2
$(2, 2)$	10	66	1.3	111.2	2.5	1541.8	1545.6	1554.7		1566.3	1554.4
$(2, 2)$	11	78	2.5	305.7	4.3	1690.2	1694.7	1703.4		1715.2	1703.0
$(2, 2)$	12	91	4.8	**t/o**	**7.4**	1836.2	1841.4	1849.7		1862.0	1849.3
$(3, 2)$	5	56	1.1	19.6	2.0	2242.8	2246.0	2266.5		2266.5	2331.6
$(3, 2)$	6	84	10.6	209.2	7.1	2955.3	2960.1	2979.8	2981.3	2980.5	3044.2
$(3, 2)$	7	120	42.5	**t/o**	**27.4**	3751.8	3758.7	3776.3	3779.5	3777.9	3840.7
$(3, 3)$	5	56	0.8	28.6	1.6	1172.5	1175.7	1185.6		1206.7	1185.2
$(3, 3)$	6	84	4.3	315.5	5.6	1401.5	1406.3	1414.5		1436.3	1414.1
$(3, 3)$	7	120	17.8	**t/o**	**21.6**	1625.7	1632.6	1639.1		1662.2	1638.7

5.2 The Herrmann-May problem

To show that the advantage is not tied to ACD, our second application is the Herrmann–May problem [HM08], which asks for small solutions to a linear equation modulo an unknown divisor of a known composite integer.

Definition 5.5 (Herrmann–May problem [HM08])**.** Let N have an unknown divisor $p \geq N^{\beta}$, and let

$$f(\mathbf{x}) = a_0 + a_1 x_1 + \cdots + a_\ell x_\ell \in \mathbb{Z}[x_1, \ldots, x_\ell]$$

be linear. Given N, β, f, and a bound X, the task is to find a small root $\mathbf{r} = (r_1, \ldots, r_\ell)$ with

$$f(\mathbf{r}) \equiv 0 \pmod{p}, \qquad |r_i| < X.$$

Lemma 5.6 (HM as bounded modular roots [HM08])**.** *An HM instance is an instance of the bounded modular root problem (Definition 2.4) for the single relation*

$$f(\mathbf{x}) \equiv 0 \pmod{p}, \qquad |x_i| < X.$$

Recovering a bounded root $\mathbf{r}$ *gives*

$$p \mid \gcd(f(\mathbf{r}), N),$$

and therefore exposes a nontrivial divisor of N *whenever this gcd is neither* 1 *nor* N*.*

For the Coppersmith step, we use the standard Herrmann–May shifts

$$g_{\mathbf{i},k} = \mathbf{x}^{\mathbf{i}} f(\mathbf{x})^k N^{m-k}, \qquad 0 \leq k \leq m,$$

with the usual degree restriction on $\mathbf{i}$. These shift polynomials vanish at the planted root modulo p^m. We form A from their coefficient vectors and construct $\mathcal{L}_{\mathbf{X}}^{\perp}$ over the public modulus N^m as in Section 4.4. Under the reduction and gap assumptions of that Theorem 4.6, coordinate-ratio decoding recovers $\mathbf{r}$. We then check the coordinate bounds and compute $\gcd(N, f(\mathbf{r}))$, accepting a nontrivial divisor as in Lemma 5.6. We refer to [HM08] for the underlying shift-polynomial approach and its solvability analysis.

Setup. We fix the multiplicity and degree $m = t = 3$ for the HM small-root instances, utilizing a 400-bit RSA modulus N with $\beta = 0.95$, and root bounds $X = 2^6$ (so that $|x_i| < 2^6$). In this setting where $t = m$, the projected search terminates exactly, which allows us to obtain exact values for the second minimum λ_2. To evaluate scalability, we increase the number of variables ℓ until the primal algorithm reaches the timeout.

Results. Table 2 shows the same qualitative behaviour as for ACD, though it is less extreme because a single linear equation constrains the ℓ unknowns more weakly than the ℓ separate ACD relations. The primal Gröbner time grows super-polynomially in ℓ. Our method recovers the root throughout, finishing in 60 seconds even at $\ell = 9$, and it is the only one of the two that solves these largest instances.

6 Discussion

6.1 Significance

Dual Coppersmith changes the RR stage of multivariate Coppersmith. In the primal algorithm, lattice reduction first produces auxiliary equations, and a separate RR stage then solves them algebraically, typically by a Gröbner basis. The RR stage has been the main obstacle to scaling the algorithm to many variables. As a result, many high-dimensional Coppersmith constructions were not explored to completion: even when the LR stage looked feasible, the subsequent RR stage could stop the attack.

Table 2: Herrmann-May recovery time and second-minimum validation (400-bit N, $\beta = 0.95$, $m = t = 3$, $X = 2^6$). The column $\delta^{2n}\lambda_1$ ($\delta = 1.02$) is the empirical reduction threshold; the measured λ_2 exceeds it on every row, so recovery holds. Every row has $t = m$, so the projected search terminates and λ_2 is exact. At $\ell = 8$, the primal RR stage times out, and by $\ell = 9$, the LR stage itself fails to complete. Conversely, our Gröbner-free and `flatter`-based dual algorithm completely bypasses these scaling issues, successfully recovering the roots in 24.6 and 59.9 seconds for $\ell = 8$ and $\ell = 9$, respectively.

		time (s)			vector norm ($\log_2$)				
ℓ	dim	Primal (LR)	Primal (RR)	Dual	λ_1	$\delta^{2n}\lambda_1$	λ_2	λ_{GH}	λ_{jet}
2	10	0.0	0.1	0.2	76.8	77.4	241.2	441.0	239.7
3	20	0.0	0.1	0.3	76.8	77.9	177.7	315.7	176.7
4	35	0.3	0.4	0.5	77.2	79.2	146.1	248.3	145.6
5	56	2.1	4.0	1.1	77.4	80.6	127.4	205.6	127.0
6	84	13.5	27.8	3.2	77.4	82.2	115.2	175.8	114.4
7	120	79.0	171.8	8.9	77.5	84.4	106.2	153.8	105.6
8	165	442.8	**t/o**	**24.6**	77.9	87.3	99.5	136.9	99.2
9	220	**t/o**	–	**59.9**	78.1	90.7	94.7	123.5	94.1

The dual algorithm removes this obstruction for isolated planted-root instances. It does not ask lattice reduction to produce several equations to be solved later. Instead, it builds a modular orthogonal lattice in which the planted root itself appears as a short direction. Once this direction is found, the root coordinates are obtained by coordinate ratios and then verified directly. Thus the final task is no longer algebraic elimination, but a geometric isolation problem in an explicit lattice.

This opens a broader search space for multivariate Coppersmith attacks. Parameters that were previously unattractive because of the Gröbner bottleneck can now be tested through the $\mathcal{L}_{\mathbf{X}}^{\perp}$. The relevant question becomes whether the planted root line is separated from the rest of the lattice, as measured by the λ_2 gap. This gives an algorithmic advantage.

6.2 Limitations

The main limitation of Dual Coppersmith is its dependence on an isolated short direction. The algorithm is sound for the bounded modular root problem, because every vector decoded by coordinate ratios is checked against the original bounds and congruences. However, its completeness theorem applies to the isolated-root-line regime.

Concretely, a bounded root $\mathbf{r}$ defines the scaled vector $\mathbf{v_r} \in \mathcal{L}_{\mathbf{X}}^{\perp}$. If the line $\mathbb{R}v_{\mathbf{r}}$ is separated from every independent lattice direction by the gap required in Theorem 3.4, then lattice reduction finds a vector on this line, and the ratio decoder outputs $\mathbf{r}$. This is a geometric condition on the constructed lattice. It is not implied by the formal statement of the bounded modular root problem alone.

The algorithm is therefore best suited to planted-root instances in which the target root is expected to be the unique short direction. This is the domain covered by our analysis and experiments. If several bounded roots are present, then each root may contribute its own short line. The reduced basis may then contain a combination of several such lines rather than a vector on one of them. In that case, the ratio decoder has no single line from which to read the coordinates.

This failure mode is safe. If a reduced vector does not decode to an integral candidate satisfying $|r_i| < X_i$ and $f_a(\mathbf{r}) \equiv 0 \pmod{p}$, Algorithm 2 rejects it. Thus Dual Coppersmith may miss a root when the isolation condition fails, but it does not certify an incorrect one.

6.3 Open Problems

The first open problem is to go beyond the isolated-root-line regime. When several bounded roots coexist, the corresponding vectors $v_{\mathbf{r}}, v_{\mathbf{r}'}, \ldots$ may form a low-dimensional cluster of short directions. A natural question is whether one can identify this cluster and then separate the individual root lines. Such an algorithm would extend the dual approach from single planted-root instances toward instances with several bounded solutions, while still avoiding a full Gröbner-based RR stage.

The second open problem is to construct better orthogonal lattices by exploiting the structure of the input problem. In this paper we use a general modular orthogonal lattice built from the standard total-degree construction. For specific problems, such as ACD or Herrmann–May, the equations may contain additional algebraic structure that could be used to choose rows, weights, or coordinates more carefully. A problem-adapted orthogonal lattice might increase the λ_2 gap, reduce the dimension, or remove structured short vectors that compete with the planted root line.

More broadly, these questions suggest a new direction for multivariate Coppersmith theory. Instead of studying only when auxiliary equations with small coefficients exist, one can study when the associated orthogonal lattice isolates the planted root. Understanding this geometry may lead to sharper predictors, stronger rigorous gap results, and new high-variable Coppersmith attacks that were previously blocked by the RR stage.

Appendix

A Cost of Root Recovery

We recall the algebraic background for the RR step discussed in Section 2.4. In the standard approach, root recovery is reduced to variable elimination in the ideal generated by the auxiliary polynomials. For the Gröbner-basis formulation of elimination, we fix the lexicographic order satisfying $x_\ell \succ \cdots \succ x_1$. For a nonzero polynomial f, let $\mathrm{LT}(f)$ denote its leading term with respect to this order.

Definition A.1 (Gröbner basis)**.** Let $I \subset \mathbb{Q}[x_1, \ldots, x_\ell]$ be an ideal. The leading-term ideal of I is defined as $\mathrm{LT}(I) = \langle \mathrm{LT}(f) : f \in I \setminus \{0\} \rangle$. A finite set $G = \{g_1, \ldots, g_s\} \subset I$ is called a Gröbner basis of I if $\mathrm{LT}(I) = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) \rangle$.

We recall the algebraic tool underlying this form of root recovery. Set

$$\mathcal{R} = \mathbb{Q}[x_1, \ldots, x_\ell],$$

and let $I \subset \mathcal{R}$ be an ideal. Eliminating the variables $x_{k+1}, \ldots, x_\ell$ means finding all polynomial consequences of I that involve only $x_1, \ldots, x_k$. This corresponds precisely to computing the k-th elimination ideal $I_k = I \cap \mathbb{Q}[x_1, \ldots, x_k]$.

Theorem A.2 (Elimination theorem [CLO98])**.** *Let $I \subset \mathbb{Q}[x_1, \ldots, x_\ell]$ be an ideal, and let G be a Gröbner basis of I with respect to the lexicographic order fixed above. Then, for every $k \in [\ell]$, the set*

$$G_k = G \cap \mathbb{Q}[x_1, \ldots, x_k]$$

is a Gröbner basis of the elimination ideal $I_k = I \cap \mathbb{Q}[x_1, \ldots, x_k]$.

While Theorem A.2 provides a rigorous methodology for systematic variable elimination, computing such a lexicographic Gröbner basis can require $2^{2^{\Omega(\ell)}}$ time in the worst case [MM82, Dub90].

This worst-case statement is used only to separate the two stages. It does not claim that every root-recovery instance arising from Coppersmith shifts attains the doubly exponential bound. Rather, it explains why the polynomial-time guarantee for the LR stage does not extend automatically to the full primal pipeline once algebraic RR is included.

B Higher-order jets

This appendix compares individual higher-order derivative vectors with the first-order weighted derivatives used in Section 4.2. We prove that the specified weighted vectors belong to the modular orthogonal lattice and that, under an explicit size condition, every higher-order vector in this family is longer than a first-order one. We work in the prime-modulus setting and retain the notation of Sections 2.3, 3.1 and 4.2.

For a multi-index $\kappa \in \mathbb{Z}_{\geq 0}^{\ell}$, write $\partial^{\kappa} = \partial_{x_1}^{\kappa_1} \cdots \partial_{x_\ell}^{\kappa_\ell}$ and define $\partial^{\kappa} z_{\mathbf{r}} = (\partial^{\kappa} x^{\alpha}|_{x=\mathbf{r}})_{\alpha \in T_t}$. Its scaled form is $\partial^{\kappa} v_{\mathbf{r}} = ((\rho/\mathbf{X}^{\alpha})(\partial^{\kappa} z_{\mathbf{r}})_{\alpha})_{\alpha \in T_t}$. The case $|\kappa| = 1$ gives the derivative vectors of Section 4.2.

Lemma B.1 (Sufficient weighting for derivatives)**.** *For every κ with $|\kappa| \leq m$,*

$$p^{|\kappa|} \partial^{\kappa} z_{\mathbf{r}} \in K, \qquad p^{|\kappa|} \partial^{\kappa} v_{\mathbf{r}} \in \mathcal{L}_{\mathbf{X}}^{\perp}.$$

Proof. Put $k = |\kappa|$. Consider a shift $g_{\mathbf{i},\beta} = p^{m-|\mathbf{i}|} x^{\beta} \mathbf{f}^{\mathbf{i}}$ from Definition 2.5. Each term in its kth-order derivative differentiates at most k of the $|\mathbf{i}|$ factors from $\mathbf{f}^{\mathbf{i}}$. At $\mathbf{r}$, at least $\max\{|\mathbf{i}| - k, 0\}$ undifferentiated factors therefore remain divisible by p. Together with $p^{m-|\mathbf{i}|}$, these factors make every term divisible by p^{m-k}. Hence

$$\left\langle \mathrm{coeff}_{T_t}(g_{\mathbf{i},\beta}), p^k \partial^{\kappa} z_{\mathbf{r}} \right\rangle = p^k \partial^{\kappa} g_{\mathbf{i},\beta}(\mathbf{r}) \equiv 0 \pmod{p^m}.$$

This holds for every row of A, so $p^k \partial^{\kappa} z_{\mathbf{r}} \in K$. Coordinate scaling gives the second inclusion. □

The multiplier $p^{|\kappa|}$ is sufficient for membership; smaller multipliers may suffice for particular derivative directions. We now compare the individual vectors obtained with this specified multiplier.

Proposition B.2 (Comparison of individual weighted derivatives)**.** *Put $X = \max_i X_i$ and $n = |T_t|$, and choose i with $X_i = X$. If $p > t\sqrt{n}\, X$, then, for every $\kappa \in T_t$ with $2 \leq |\kappa| \leq m$,*

$$\left\| p^{|\kappa|} \partial^{\kappa} v_{\mathbf{r}} \right\| > \left\| p \partial_i v_{\mathbf{r}} \right\|.$$

Proof. Fix $\kappa \in T_t$ with $k = |\kappa| \geq 2$. The $\alpha = \kappa$ coordinate gives

$$\left\| p^k \partial^{\kappa} v_{\mathbf{r}} \right\| \geq \frac{\rho p^k}{\mathbf{X}^{\kappa}} \prod_{j=1}^{\ell} \kappa_j! \geq \rho \left(\frac{p}{X} \right)^k.$$

For the first-order vector, $|r_j| < X_j$ and $\alpha_i \leq t$ imply $|(\partial_i z_{\mathbf{r}})_{\alpha}| \leq t\mathbf{X}^{\alpha}/X_i$ whenever $\alpha_i > 0$; the other coordinates are zero. Since $X_i = X$,

$$\left\| p \partial_i v_{\mathbf{r}} \right\| \leq \frac{\rho t \sqrt{n}\, p}{X}.$$

Because $p/X > t\sqrt{n} \geq 1$ and $k \geq 2$, the first lower bound is strictly larger than this upper bound. □

The size condition holds for sufficiently large p in Corollary 4.2. Indeed, $X_i \geq 1$ gives $X \leq \prod_i X_i$, and the product bound implies $p/X > p^{1-1/d+\varepsilon} \longrightarrow \infty$. The parameters ℓ, m, t, and therefore $t\sqrt{n}$, are fixed as p grows in that corollary.

Remark B.3 (Scope of the comparison)**.** Proposition B.2 compares only the specified individual vectors before projection. It does not bound shorter admissible multiples or linear combinations of derivative directions. The quantity λ_{jet} remains the projected covolume-based estimate of Section 4.2. The second-minimum prediction still relies on Heuristic 4.1, and deterministic recovery remains governed by the gap condition in Theorem 3.4.

C Proof of Lemma 2.9

Proof. Let $\mathcal{L}$ be the scaled coefficient lattice defined above, with $n = |\mathcal{T}_t|$, and let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be the LLL-reduced basis returned in the LR stage. Each $\mathbf{b}_j$ is the coefficient vector of the corresponding scaled auxiliary polynomial. By Lemma 2.7, we have $|\mathcal{R}_{m,t}| \geq p^{S_m}$ and $S_m/B_t \to 1/d$ as $m \to \infty$. Choose m sufficiently large that

$$\frac{S_m}{B_t} \geq \frac{1}{d} - \frac{\epsilon}{2},$$

and fix this m, with $t = dm$. Then, Equation (2) and the product condition give

$$\det(\mathcal{L})^{1/n} \leq p^{m-S_m/n} \left(\prod_{i=1}^{\ell} X_i \right)^{B_t/n} < p^{m-S_m/n+(1/d-\varepsilon)B_t/n} \leq p^{m-\varepsilon B_t/(2n)}.$$

With the reduction parameters fixed, the LLL determinant bound and the length heuristic discussed in Remark 1.1 yield, for some constant $C \geq 1$ independent of p,

$$\max_{1 \leq j \leq \ell} \|\mathbf{b}_j\| \leq C\|\mathbf{b}_1\| \leq C\delta^{n-1} \det(\mathcal{L})^{1/n} < C\delta^{n-1} p^{m-\epsilon B_t/(2n)}.$$

Since $\epsilon B_t/(2n) > 0$, for sufficiently large p we have

$$\max_{1 \leq j \leq \ell} \|\mathbf{b}_j\| < \frac{p^m}{\sqrt{n}},$$

so the first ℓ reduced vectors pass the LR norm test.

Every polynomial retained by the LR stage satisfies $g(\mathbf{r}) \equiv 0 \pmod{p^m}$ and Howgrave-Graham bound. Therefore, Lemma 2.6 gives $g(\mathbf{r}) = 0$ over $\mathbb{Z}$ for all retained polynomials. By Heuristic 2.8, the RR stage enumerates their bounded integer zeros. The bounded target $\mathbf{r}$ is among these zeros and satisfies the original congruences and bounds, so the final verification accepts and outputs it. □

D Complete-power comparison for affine relations

This appendix proves that a sufficiently large gap in the degree-one lattice forces the second minimum of the complete affine construction to occur in its first-jet lattice. We use the public-modulus setting of Section 4.4, set $t = m$, and assume that every f_a is a nonconstant affine polynomial. The construction includes all total-degree shifts from Definition 2.5, with N replacing p.

Write $f_a(\mathbf{x}) = a_{a0} + \sum_{i=1}^{\ell} a_{ai}x_i$. The degree-one lattice, before multiplication by a common integer scaling factor, is

$$\Gamma = \left\{ (z_0, z_1/X_1, \ldots, z_\ell/X_\ell) : \mathbf{z} \in \mathbb{Z}^{\ell+1}, \quad a_{a0}z_0 + \sum_{i=1}^{\ell} a_{ai}z_i \equiv 0 \pmod{N} \text{ for every } a \right\}.$$

As in Theorem 4.6, put $p = \gcd(N, f_1(\mathbf{r}), \ldots, f_s(\mathbf{r}))$ and $q_0 = N/p$. The primitive vector on the target root line in Γ is $\mathbf{u} = q_0(1, r_1/X_1, \ldots, r_\ell/X_\ell)$. Let $Q = \pi_{\mathbf{u}}(\Gamma)$, where $\pi_{\mathbf{u}}$ is orthogonal projection onto $\mathbf{u}^\perp$, and put $\nu = \|\mathbf{u}\|$ and $\mu = \lambda_1(Q)$. Both $\mathcal{L}_{\mathbf{X}}^\perp$ and its first-jet lattice $\mathcal{L}_{\text{jet}}$ are taken over the public modulus N^m.

Proposition D.1 (Complete-power comparison)**.** *Fix $\ell, m \geq 1$. Along any family of the above instances with $\mu/\nu \longrightarrow \infty$, for all sufficiently large μ/ν,*

$$\lambda_2(\mathcal{L}_{\mathbf{X}}^\perp) = \lambda_2(\mathcal{L}_{\text{jet}}) \asymp_{\ell,m} \rho\nu^{m-1}\mu.$$

If, in addition, $\mu \asymp_\ell \det(Q)^{1/\ell}$, then

$$\lambda_2(\mathcal{L}_{\mathbf{X}}^\perp) \asymp_{\ell,m} \lambda_{\text{jet}} \asymp_{\ell,m} \min\{\lambda_{\text{GH}}, \lambda_{\text{jet}}\}.$$

Here the comparison constants depend only on the indicated fixed dimensions, and the two estimates use the actual projected covolumes of the public lattice: $\lambda_{\text{jet}} = \sqrt{\ell/(2\pi e)}\, \det(\pi(\mathcal{L}_{\text{jet}}))^{1/\ell}$ and $\lambda_{\text{GH}} = \sqrt{(n-1)/(2\pi e)}\, \det(\pi(\mathcal{L}_{\mathbf{X}}^\perp))^{1/(n-1)}$.

Proof. All norms below are the Euclidean monomial-coordinate norms unless an auxiliary norm is specified.

The exact lattice identity. Homogenization identifies the coordinates indexed by $\mathcal{T}_m$ with the degree-m monomials in $\ell+1$ variables: $\alpha \in \mathcal{T}_m$ corresponds to $(m-|\alpha|, \alpha)$. For a square matrix A and an integer $j \geq 1$, define $M_j(A)$ by $(A\mathbf{y})^\alpha = \sum_{|\beta|=j} M_j(A)_{\alpha,\beta}\mathbf{y}^\beta$ for $|\alpha| = j$. Thus $M_j(AB) = M_j(A)M_j(B)$, and $M_j(U)$ is unimodular whenever U is an integer unimodular matrix. This definition fixes all multinomial coefficients in the lift.

Let H have as rows an integer basis of the coefficient lattice generated by the homogenized forms f_a^h and $Ny_0, \ldots, Ny_\ell$. The unscaled degree-one modular kernel is $NH^{-1}\mathbb{Z}^{\ell+1}$. Every homogenized total-degree shift is a product of m of these linear generators, and conversely all such products occur in their integer span. The products with no f_a^h factor give the trivial rows $N^m\mathbf{y}^\alpha$, which impose no additional modular constraints. Because every f_a has degree one and $t = m$, the allowed monomial multipliers give precisely all the remaining products. Consequently, the homogeneous degree-m coefficient lattice has row basis $M_m(H)$, and its modular kernel is $N^m M_m(H)^{-1}\mathbb{Z}^n = M_m(NH^{-1})\mathbb{Z}^n$.

Put $D = \operatorname{diag}(1, X_1^{-1}, \ldots, X_\ell^{-1})$ and $G = DNH^{-1}$, so the columns of G form a basis of Γ. The original coordinate scaling is $\rho M_m(D)$, giving the exact identity

$$\mathcal{L}_{\mathbf{X}}^\perp = \rho M_m(G)\mathbb{Z}^n.$$

It holds for any basis matrix G of Γ, by the unimodularity noted above. No primality or coprimality assumption is needed.

Coordinates adapted to the root line. On the degree-one target line, a lattice point has the form $c(1, r_1/X_1, \ldots, r_\ell/X_\ell)$ with $c \in \mathbb{Z}$ and $N \mid cf_a(\mathbf{r})$ for every a. These conditions are equivalent to $q_0 \mid c$, proving that $\mathbf{u}$ is primitive. Choose a basis $\mathbf{q}_1, \ldots, \mathbf{q}_\ell$ of Q and lifts $\mathbf{b}_i \in \Gamma$ whose components along $\mathbf{u}/\nu$ have absolute value at most $\nu/2$. Then $G = (\mathbf{u}, \mathbf{b}_1, \ldots, \mathbf{b}_\ell)$ is a basis matrix of Γ.

Choose the basis of Q recursively by taking a shortest vector in each successive orthogonal projection and size-reducing along the preceding vectors. If d_i are its Gram–Schmidt lengths, then $d_1 = \mu$ and $d_{i+1} \geq (\sqrt{3}/2)d_i$: a shortest vector in the next projection lifts to a vector of length at most $\sqrt{d_{i+1}^2 + d_i^2/4}$ in the preceding projected lattice, whose minimum is d_i. Thus every d_i is at least $(\sqrt{3}/2)^{\ell-1}\mu$.

To rotate coordinates without introducing input-dependent norm distortion, use the auxiliary norm $\|z\|_{(j)}^2 = \sum_{|\alpha|=j} \binom{j}{\alpha} z_\alpha^2$. The multinomial theorem, applied to $(\mathbf{x} \cdot \mathbf{y})^j$, shows that $M_j(O)$ preserves this norm for every orthogonal matrix O. Also $\|z\| \leq \|z\|_{(j)} \leq \sqrt{j!}\,\|z\|$. Rotate the degree-one coordinates so that $\mathbf{u} = \nu e_0$ and the projected basis matrix is upper triangular, with diagonal $d_1, \ldots, d_\ell$. Write this projected matrix as R.

Bounds for all transverse blocks. Write a lattice vector as $\mathbf{w} = \rho M_m(G)c$ with $c \in \mathbb{Z}^n$, and group the coordinates of c by their transverse degree $k = \alpha_1 + \cdots + \alpha_\ell$. Suppose $k \geq 1$ is the largest degree with a nonzero block c_k. The transverse-degree-k output block is exactly $\rho\nu^{m-k}M_k(R)c_k$. Blocks of smaller degree cannot contribute to it, and blocks of greater degree are zero by the choice of k.

In decreasing lexicographic monomial order, $M_k(R)$ is upper triangular with diagonal entries $\prod_i d_i^{\alpha_i}$, $|\alpha| = k$. At the last nonzero integer coordinate of c_k, the corresponding output coordinate has absolute value at least $((\sqrt{3}/2)^{\ell-1}\mu)^k$. Subtracting any real multiple of the root vector changes only the transverse-degree-zero block. The norm comparisons therefore give

$$\|\pi(\mathbf{w})\| \geq c_{\ell,m}\rho\nu^{m-k}\mu^k \qquad (c_{\ell,m} > 0).$$

The constant is independent of N, the coefficients, the bounds, and the target root. This bound applies to every integral combination in the block.

Let $v_m(\mathbf{x}) = (\mathbf{x}^\alpha)_{|\alpha|=m}$. The degree-zero column is $v_m(\mathbf{u})$, while the degree-one columns are $\mathrm{d}v_m(\mathbf{u})[\mathbf{b}_i]$. Their span is the tangent space to the monomial-evaluation map at the root, namely the span of $\mathbf{v_r}$ and its first derivatives. Since the remaining columns are linearly independent, intersection with this span consists exactly of the lattice vectors with $c_k = 0$ for every $k \geq 2$. These are precisely $\mathcal{L}_{\mathrm{jet}}$.

For an upper bound, use $\mathbf{q}_1$ of length μ and its chosen lift $\mathbf{b}_1$. The lattice vector $\rho\mathrm{d}v_m(\mathbf{u})[\mathbf{b}_1]$ is independent of the root line and has norm at most $C_m\rho\nu^{m-1}(\mu + \nu)$ for a constant C_m. Indeed, in the rotated weighted norm its squared length is $\rho^2\nu^{2m-2}(m\mu^2 + m^2a^2)$, where $|a| \leq \nu/2$ is the component of $\mathbf{b}_1$ along $\mathbf{u}/\nu$.

The primitive lifted root vector is $\rho v_m(\mathbf{u}) = q_0^m\mathbf{v_r}$, of norm comparable to $\rho\nu^m$. When $\mu/\nu \to \infty$, the block lower bound puts every independent vector above this root vector and at scale at least $\rho\nu^{m-1}\mu$. For $m \geq 2$, every vector outside $\mathcal{L}_{\mathrm{jet}}$ has scale at least $\rho\nu^{m-2}\mu^2$, which eventually exceeds the first-jet upper bound. Hence the second minimum is attained in the first-jet lattice and has the scale asserted in the proposition. For $m = 1$, the full lattice is already its first-jet lattice, giving the same conclusion.

From the actual minimum to the covolume estimates. Put $\Delta = \det(Q)^{1/\ell}$. After removing the root component, the derivative map $\mathbf{q} \mapsto \pi(\mathrm{d}v_m(\mathbf{u})[\mathbf{q}])$ has all its singular values comparable to ν^{m-1}. In the weighted norm, its transverse length is exactly $\sqrt{m}\,\nu^{m-1}\|\mathbf{q}\|$; the norm equivalence gives the claim for the original Euclidean quotient norm. Its image of Q, scaled by ρ, is exactly the projected first-jet lattice. Therefore $\det(\pi(\mathcal{L}_{\mathrm{jet}}))^{1/\ell} \asymp_{\ell,m} \rho\nu^{m-1}\Delta$.

For the full lattice, triangularization gives $|\det M_m(G)| = |\det G|^{B_m}$, where $B_m = \binom{\ell+m}{\ell+1} = mn/(\ell+1)$. Since $\det\Gamma = \nu \det Q$ and the primitive root length is comparable to $\rho\nu^m$, the two covolume estimates satisfy

$$\lambda_{\mathrm{jet}} \asymp_{\ell,m} \rho\nu^{m-1}\Delta, \qquad \lambda_{\mathrm{GH}} \asymp_{\ell,m} \rho\nu^m(\Delta/\nu)^{\ell B_m/(n-1)}.$$

For $m > 1$, the exponent $\ell B_m/(n-1)$ is greater than one. Under $\mu \asymp_\ell \Delta$ and $\mu/\nu \to \infty$, it follows that $\lambda_{\mathrm{GH}}/\lambda_{\mathrm{jet}} \to \infty$. For $m = 1$, the two projected lattices coincide and their estimates are equal. Combining these facts with the deterministic comparison proves the second assertion. □

The first comparison is deterministic; predicting μ from $\det(Q)$ is the additional assumption in the second comparison.

The ACD and Herrmann–May constructions in Section 5 satisfy the complete-shift condition when $t = m$. The cases $t > m$ and nonlinear input relations remain outside this proposition.

AI methodology

We used AI for generating experiments code.

References

[BD99] Dan Boneh and Glenn Durfee. Cryptanalysis of rsa with private key d less than n 0.292. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–11. Springer, 1999.

[CH13] Henry Cohn and Nadia Heninger. Approximate common divisors via lattices. *The Open Book Series*, 1(1):271–293, 2013.

[CLO98] David A Cox, John Little, and Donal O'shea. *Using algebraic geometry*. Springer, 1998.

[Cop96] Don Coppersmith. Finding a small root of a univariate modular equation. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 155–165. Springer, 1996.

[Cop97] Don Coppersmith. Small solutions to polynomial equations, and low exponent rsa vulnerabilities. *Journal of cryptology*, 10(4):233–260, 1997.

[Dub90] Thomas W Dubé. The structure of polynomial ideals and gröbner bases. *SIAM Journal on Computing*, 19(4):750–773, 1990.

[FLC+25] Yansong Feng, Hengyi Luo, Qiyuan Chen, Abderrahmane Nitaj, and Yanbin Pan. Computing asymptotic bounds for small roots in coppersmith's method via sumset theory. In *Annual International Cryptology Conference*, pages 3–32. Springer, 2025.

[FNP24] Yansong Feng, Abderrahmane Nitaj, and Yanbin Pan. Partial prime factor exposure attacks on some rsa variants. *Theoretical Computer Science*, 999:114549, 2024.

[GGM16] Steven D Galbraith, Shishay W Gebregiyorgis, and Sean Murphy. Algorithms for the approximate common divisor problem. *LMS Journal of Computation and Mathematics*, 19(A):58–72, 2016.

[GN08] Nicolas Gama and Phong Q Nguyen. Predicting lattice reduction. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 31–51. Springer, 2008.

[HG97] Nicholas Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *IMA International Conference on Cryptography and Coding*, pages 131–142. Springer, 1997.

[HG01] Nick Howgrave-Graham. Approximate integer common divisors. In *International cryptography and lattices conference*, pages 51–66. Springer, 2001.

[HHPS20] Amir Hashemi, Joos Heintz, Luis M Pardo, and Pablo Solernó. Intrinsic complexity for constructing zero-dimensional gröbner bases. In *International Workshop on Computer Algebra in Scientific Computing*, pages 245–265. Springer, 2020.

[HM08] Mathias Herrmann and Alexander May. Solving linear equations modulo divisors: On factoring given any bits. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 406–424. Springer, 2008.

[HM10] Mathias Herrmann and Alexander May. Maximizing small root bounds by linearization and applications to small secret exponent rsa. In *International workshop on public key cryptography*, pages 53–69. Springer, 2010.

[JM06] Ellen Jochemsz and Alexander May. A strategy for finding roots of multivariate polynomials with new applications in attacking rsa variants. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 267–282. Springer, 2006.

[LLL82] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[MM82] Ernst W Mayr and Albert R Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in mathematics*, 46(3):305–329, 1982.

[MV20] Thierry Mefenza and Damien Vergnaud. Inferring sequences produced by elliptic curve generators using coppersmith's methods. *Theoretical Computer Science*, 830-831:20–42, 2020.

[RH23] Keegan Ryan and Nadia Heninger. Fast practical lattice reduction through iterated compression. In *Annual International Cryptology Conference*, pages 3–36. Springer, 2023.

[Rya25] Keegan Ryan. Solving multivariate coppersmith problems with known moduli. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 355–384. Springer, 2025.