Faster Matrix Multiplication Leads to Faster Lattice Sieving

Yiming Gao[1] Honggang Hu[1,2]

[1]School of Cyber Science and Technology,
University of Science and Technology of China
qw1234567@mail.ustc.edu.cn

[2]Hefei National Laboratory, Hefei, China
hghu2005@ustc.edu.cn

Abstract

Let ω denote the matrix multiplication exponent. Very recently, OpenAI showed that $\omega \leq 9/4$. We demonstrate that, for any fixed $\eta > 0$, if $\omega \leq 9/4 - \eta$, then there is a constant $\delta = \delta(\eta) > 0$ such that heuristic lattice sieving can be improved to $(3/2 - \delta)^{n/2+o(n)}$ time under the spherical heuristic used in the BDGL analysis. With known matrix multiplication algorithms, our method already yields subquadratic search within BDGL buckets.

1 Introduction

For more than a decade, the Becker–Ducas–Gama–Laarhoven (BDGL) algorithm has provided the best known classical heuristic time complexity for lattice sieving, namely $(3/2)^{n/2+o(n)}$ [BDGL16]. The BDGL exponent also serves as a central asymptotic benchmark for the sieving cost models used in concrete security estimates of lattice-based cryptography [APS15]. Kirshanova and Laarhoven further proved that this exponent is optimal within a broad framework of hash-based nearest neighbor searching on uniformly random spherical data [KL21, Theorem 6]. It has remained unclear where a further improvement in the leading exponent might come from.

We connect this filtering framework to fast matrix multiplication by changing the search inside its buckets. Recent progress has established $\omega \leq 9/4$ [OAI26], and we show that any fixed improvement below this threshold would also improve the BDGL sieving exponent under its spherical heuristic.

The geometric starting point is that bucket membership creates positive correlation even between independently sampled points, because they share a component along the center f. We remove this common direction by projection onto $f^{\perp}$ and normalization. For points near the cap boundary of threshold τ, an original correlation t becomes

$$t_{\mathrm{proj}} \approx \frac{t - \tau^2}{1 - \tau^2}.$$

At the original threshold $\tau = 1/2$, the background correlation drops from about $1/4$ to zero, while pairs with original correlation just above $1/2$ give shorter differences and retain residual correlation close to $1/3$.

This gap allows us to search for useful pairs in groups. Gaussian tensor features amplify the separation, while random signs allow a target pair to remain detectable after aggregation. Fast matrix multiplication constructs the compressed group representations and computes scores for all

group pairs. We expand only those group pairs whose scores repeatedly exceed a threshold and check their point pairs exactly.

For F filters with bucket size parameter M, the bucket search term becomes

$$FM^2 \quad \longrightarrow \quad FM^C \operatorname{poly}(n), \qquad \frac{2\omega}{3} < C < 2.$$

Faster search makes larger buckets viable, and their greater pair coverage reduces the number of filters. A bucket exponent $C < 3/2$ makes this tradeoff improve the leading sieve exponent, giving the threshold $\omega < 9/4$.

1.1 The results

Throughout, we work in a randomized real-RAM model with unit cost arithmetic and comparisons, together with standard Gaussian and spherical sampling.

Fix constants $0 < a < b$ and write

$$\mathcal{I}_n = \left[\frac{1}{2} + \frac{a}{n}, \frac{1}{2} + \frac{b}{n}\right]. \tag{1.1}$$

Our search reports all unordered pairs whose inner product lies in $\mathcal{I}_n$. For unit vectors, these pairs give differences of norm $1 - \Theta(1/n)$. For $N = 2^{O(n)}$ independent uniform points in S^{n-1}, fix $0 < \tau \le 1/2$ and let $\mathcal{C}_n(\tau)$ be the spherical cap measure. Use the BDGL filters of threshold τ with total count F, including padding and repetitions, and put

$$M = \max\{1, N\mathcal{C}_n(\tau)\}.$$

Theorem 1.1 (Spherical pair search)**.** *For every fixed C satisfying*

$$\frac{2\omega}{3} < C < 2,$$

there is a constant $c > 0$ such that, whenever $M \le 2^{cn}$, a randomized algorithm reports every unordered pair whose inner product lies in $\mathcal{I}_n$ with probability $1-2^{-\Omega(n)}$. After the bucket memberships have been constructed, its expected additional running time satisfies

$$\mathbb{E}[T_{\text{search}}] \le FM^C \operatorname{poly}(n). \tag{1.2}$$

Let T_{filt} include filter generation, membership enumeration, and bucket construction. The total expected time is therefore bounded by

$$\mathbb{E}[T_{\text{total}}] \le \mathbb{E}[T_{\text{filt}}] + FM^C \operatorname{poly}(n), \qquad \mathbb{E}[T_{\text{filt}}] \le (N + FM)2^{o(n)}. \tag{1.3}$$

With known matrix multiplication algorithms, our method already yields subquadratic bucket search for suitable BDGL filter parameters. For $M = n^{\omega(1)}$ satisfying the size condition in Theorem 1.1, the expected search cost is $FM^C \operatorname{poly}(n)$ with a fixed $C < 2$. This is asymptotically smaller than the FM^2 pair enumeration term for the same filters.

Optimizing the filters gives the following consequence.

Corollary 1.2. *Under the spherical heuristic used in the BDGL analysis [BDGL16], if $\omega \le 9/4 - \eta$ for a fixed $\eta > 0$, then lattice sieving solves SVP with high probability in time*

$$(3/2 - \delta)^{n/2+o(n)}$$

for a constant $\delta = \delta(\eta) > 0$.

Our correlation search follows Gregory Valiant's framework of tensor amplification and signed aggregation [Val15], with compressed aggregation adapted from Karppa, Kaski, and Kohonen [KKK18, Section 4.2]. We apply these ideas to Gaussian features on projected spherical data and use the resulting search bound to optimize the BDGL filters.

Remark. For rational lattice inputs, Gaussian sampling with polynomial precision and truncation, conservative filter membership tests, and fixed score margins give a word-RAM implementation with the same leading exponent, provided the matrix algorithm admits an exact implementation with polynomial bit overhead. Final pair tests use the original rational vectors.

AI disclosure. We used large language models through Codex (GPT-6.1-sol and GPT-6-astra) to explore possible improvements to the BDGL sieve under the spherical heuristic used in its analysis. The models proposed the main approach developed in this manuscript. The authors subsequently recognized its connection to OpenAI's recent breakthrough establishing $\omega \le 9/4$, proposed simplifications to the algorithm, and verified the proofs. The authors then worked with the models to organize and draft the manuscript. The entire paper was thoroughly checked and revised by the authors, who remain responsible for all mathematical claims, proofs, and references.

2 Spherical filtering and residual correlation

2.1 Revisiting BDGL

A spherical sieve searches a list of nearly equal length vectors for pairs whose differences are shorter. After normalization, the identity $\|x-y\|^2 = 2 - 2\langle x, y\rangle$ turns this task into finding pairs with inner product close to $1/2$.

BDGL organizes the search using overlapping spherical caps. A center $f \in S^{n-1}$ defines the bucket

$$\mathcal{C}_f = \{x : \langle x, f\rangle \ge \tau\},$$

and a point may enter several buckets. The original method enumerates point pairs within each bucket. At its symmetric parameters, $\tau = 1/2$, the list size and filter count are $N = 2^{\ell n + o(n)}$ and $F = 2^{\kappa_{\mathrm{B}} n + o(n)}$, where

$$\ell = \frac{1}{2}\log_2 \frac{4}{3}, \qquad \kappa_{\mathrm{B}} = \frac{1}{2}\log_2 \frac{3}{2} = 0.2924812503\ldots. \tag{2.1}$$

The bucket size parameter is then $M = 2^{o(n)}$. We retain the same spherical filtering framework and change the search performed inside its buckets.

2.2 Removing the common direction

Membership in a bucket already creates positive correlation: even two independently sampled points share a component along f. To see its contribution, first consider points at the same latitude $\langle x, f\rangle = \langle y, f\rangle = \tau$ and write

$$x = \tau f + \sqrt{1-\tau^2}\, u, \qquad y = \tau f + \sqrt{1-\tau^2}\, v,$$

where u, v are unit vectors in $f^{\perp}$. Their correlation splits as

$$\langle x, y\rangle = \tau^2 + (1-\tau^2)\langle u, v\rangle. \tag{2.2}$$

For independent background points, u, v are independent uniform directions with $\mathbb{E}[\langle u, v\rangle] = 0$ and $\mathbb{E}[\langle u, v\rangle^2] = 1/(n-1)$. Thus background fluctuations have scale $n^{-1/2}$ around the correlation τ^2 induced by the common direction.

Useful pairs have original correlation close to $1/2$, so their residual correlation is close to $(1/2 - \tau^2)/(1 - \tau^2)$. At $\tau = 1/2$, removing the common direction takes the background correlation from about $1/4$ to 0 and the target correlation from about $1/2$ to $1/3$. This gives a constant positive signal against a nearly orthogonal background.

We make the equal latitude calculation uniform by retaining a thin band inside each cap:

$$B_f = \{x \in S^{n-1} : \tau \le \langle x, f\rangle \le \tau + \Delta_n\}, \qquad \Delta_n \longrightarrow 0. \tag{2.3}$$

Project each retained point onto $f^\perp$ and normalize:

$$\pi_f(x) = \frac{x - \langle x, f\rangle f}{\sqrt{1 - \langle x, f\rangle^2}}. \tag{2.4}$$

For $\alpha = \langle x, f\rangle$ and $\beta = \langle y, f\rangle$, the exact residual correlation is

$$\langle \pi_f(x), \pi_f(y)\rangle = \frac{\langle x, y\rangle - \alpha\beta}{\sqrt{1-\alpha^2}\sqrt{1-\beta^2}}.$$

Since $\alpha, \beta = \tau + o(1)$ in the band, every target pair has

$$\langle \pi_f(x), \pi_f(y)\rangle = \frac{1/2 - \tau^2}{1 - \tau^2} + o(1) \ge \frac{1}{3} - o(1), \qquad 0 < \tau \le 1/2. \tag{2.5}$$

We use the fixed lower threshold $\rho = 1/4$ in the local algorithm. The remaining question is whether enough common filters survive the band restriction.

2.3 Coverage and background estimates

Write $\mathcal{W}_n(\tau, \tau, t)$ for the common cap measure of two points with inner product t, and put $P(s) = -\frac{1}{2}\log_2(1 - s^2)$. The standard cap and common cap estimates [BDGL16, Sections 2 and 7] are $\mathcal{C}_n(\tau) = 2^{-P(\tau)n + o(n)}$ and

$$\mathcal{W}_n(\tau, \tau, t) = 2^{-K_\tau(t)n + o(n)}, \qquad K_\tau(t) = -\frac{1}{2}\log_2\left(1 - \frac{2\tau^2}{1+t}\right). \tag{2.6}$$

The filters come from the random product codes of [BDGL16, Lemma 5.1 and Theorem 5.1], drawn independently of the input. Let

$$F_0 = \mathcal{W}_n(\tau, \tau, 1/2)^{-1}, \qquad F = F_0\, 2^{o(n)}, \qquad M = \max\{1, N\mathcal{C}_n(\tau)\}. \tag{2.7}$$

Here F counts all centers across $R = \lceil c_0 n\rceil$ independent codes of the same fixed padded size, for sufficiently large c_0. Each code has F/R centers and covers any fixed target pair with constant probability. Each fixed code address has a uniform marginal center, and the decoder enumerates all cap memberships with expected cost $(N + FM)2^{o(n)}$ across all codes, including filter generation.

The common cap mass concentrates near both latitude boundaries, which lets us retain that coverage inside the band. For thresholds r, s and point correlation t, the squared projection norm determining the common cap is

$$D(r, s, t) = \frac{r^2 + s^2 - 2trs}{1 - t^2}, \qquad \left.\frac{\partial D}{\partial r}\right|_{r=s=\tau} = \frac{2\tau}{1+t} > 0.$$

For $t \in \mathcal{I}_n$, raising either threshold by Δ_n therefore increases the common cap exponent by $\Omega(\Delta_n)$ uniformly. Choose $n\Delta_n$ to dominate the logarithm of the product code padding and the errors in the volume estimates. In one code, the expected number of common centers at which either endpoint lies above $\tau + \Delta_n$ is then $2^{o(n)-\Omega(n\Delta_n)} = o(1)$. Subtracting this event from the cap coverage event gives constant probability of a common center in the band. The R independent codes give simultaneous coverage of all target pairs with probability $1 - 2^{-\Omega(n)}$.

For background points, projection has an exact distributional description. The latitude of a uniform point and its direction in $f^{\perp}$ are independent, so conditioning independent points on membership in B_f leaves their projected directions independent and uniform. Consequently, for every fixed $s \in (0, 1)$,

$$\Pr[|\langle \pi_f(x), \pi_f(y) \rangle| > s \,|\, x, y \in B_f] = 2^{-P(s)n+o(n)}. \tag{2.8}$$

The bucket search is now a correlation detection problem: most pairs have residual inner product near zero, while the useful pairs retain correlation at least $1/3 - o(1)$. The next section constructs an algebraic search for this gap.

3 Compressed correlation search

The projection in Section 2 separates targets of constant positive correlation from a nearly orthogonal background. Gaussian tensor features amplify this gap. For independent standard Gaussian vectors $G_1, \ldots, G_d$ and an even degree d, the feature

$$\varphi(x) = \prod_{j=1}^{d} \langle G_j, x \rangle \qquad \text{satisfies} \qquad \mathbb{E}[\varphi(x)\varphi(y)] = \langle x, y \rangle^d.$$

A target with correlation at least $\rho = 1/4$ therefore has mean feature product at least ρ^d, whereas a pair with absolute correlation at most $\sigma < \rho$ has mean at most σ^d.

To search in groups, write $\Phi(x)$ for a vector of such features. For groups I, J on opposite sides, independent random signs give

$$z_I = \sum_{x \in I} a_x \Phi(x), \qquad z_J = \sum_{y \in J} b_y \Phi(y), \qquad S_{I,J} = \langle z_I, z_J \rangle.$$

All group scores are entries of the matrix product $\mathsf{S} = \mathcal{Z}_0 \mathcal{Z}_1^{\mathsf{T}}$, where the rows are the group representations. The random signs preserve the contribution of a target even when other pairs are present. Indeed, conditional on the Gaussian features $\mathcal{G}$, the kernel $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$ satisfies

$$\mathbb{E}_{a,b}\Big[S_{I,J}^2 \mid \mathcal{G}\Big] = \sum_{x \in I, y \in J} K(x, y)^2.$$

A target contributes a squared term to this sum, and the moment bounds below turn that contribution into a constant detection probability. Repeating the test identifies group pairs to expand for exact inner product checks.

Following [KKK18, Section 4.2], we also form the group representations by matrix multiplication: splitting each tensor feature into two halves lets us construct the signed aggregates without computing every full feature of every point. Groups of about $m^{1/3}$ points give $O(m^{2/3})$ groups. With suitable feature dimensions, both matrix computations have base exponent arbitrarily close to $2\omega/3$. The remaining exact work is controlled by the number of exceptional pairs whose absolute correlation exceeds σ.

3.1 The search algorithm

Lemma 3.1 (Compressed search). *Fix $0 < \xi < 2 - 2\omega/3$ and put $C = 2\omega/3 + \xi$. There is a constant $0 < \sigma = \sigma(\xi) < 1/4$ with the following property. Given two lists in S^{n-1} with total size m, let Q be the number of cross pairs with absolute inner product greater than σ. For any $0 < \zeta < 1/2$, a randomized algorithm reports all cross pairs with inner product at least $1/4$ with probability at least $1 - \zeta$, in time*

$$(m^C + Qm^{2/3}) \operatorname{poly}(n, \log m, \log(1/\zeta)). \tag{3.1}$$

The parameters below balance the matrix costs and separate targets from the background. Set

$$\rho = \frac{1}{4}, \qquad \gamma = \frac{\xi}{16}, \qquad \lambda = \left\lceil 2 + \frac{1}{\gamma} \right\rceil, \qquad r_0 = \left\lceil m^{1/3+\xi/4} \right\rceil, \qquad r = r_0^2, \tag{3.2}$$

and put $g = \lceil m^{1/3} \rceil$ and $d = 2h = 2\lfloor (\gamma/2) \log_{1/\rho} m \rfloor$. Choose $\sigma = \rho^\lambda$. Each trial uses independent standard Gaussian vectors to define

$$A_s(x) = \prod_{j=1}^{h} \langle G^A_{s,j}, x \rangle, \qquad B_t(x) = \prod_{j=1}^{h} \langle G^B_{t,j}, x \rangle, \qquad 1 \le s, t \le r_0. \tag{3.3}$$

The same two independent Gaussian families serve both lists. For a group I, let U_I, V_I contain its half features as rows. Independent Rademacher signs a_x give the aggregate

$$Z_I = U_I^{\mathsf{T}} \operatorname{diag}(a_x : x \in I) V_I, \qquad Z_I(s,t) = \sum_{x \in I} a_x A_s(x) B_t(x). \tag{3.4}$$

This forms the signed sum of the full features $\phi_{s,t}(x) = A_s(x) B_t(x)$ directly from r_0 coordinates per half family. If $\mathcal{Z}_0, \mathcal{Z}_1$ have the flattened aggregates as rows, the score matrix is

$$\mathsf{S} = \mathcal{Z}_0 \mathcal{Z}_1^{\mathsf{T}}, \qquad S_{I,J} = \sum_{s,t} Z_I(s,t) Z_J(s,t). \tag{3.5}$$

Algorithm 1 specifies the search. The fixed enumeration threshold m_0 and repetition constant c_1 are chosen sufficiently large as functions of ξ.

Algorithm 1. Compressed correlation search

Input: Two indexed lists L_0, L_1 of unit vectors with total size m, slack ξ, and failure probability ζ.
Output: All cross pairs with inner product at least $\rho = 1/4$, or failure.

if $m \le m_0$ **then return** *all target pairs by direct enumeration*
Partition each list into groups of at most g points, keeping these groups fixed
Set $R_{\mathrm{loc}} \leftarrow \lceil c_1(\log m + \log(1/\zeta)) \rceil$, all counts $c_{I,J} \leftarrow 0$, and $\mathcal{P} \leftarrow \varnothing$
for $k = 1, \dots, R_{\mathrm{loc}}$ **do**
    Draw fresh Gaussian families and independent signs for every point on both sides, and compute the half features (3.3)
    Form every Z_I by the matrix products in (3.4)
    Compute S by (3.5), splitting the inner dimension into blocks of length $O(m^{2/3})$
    Increment $c_{I,J}$ for every group pair with $|S_{I,J}| \ge r\rho^d/8$
foreach *group pair* (I, J) *with* $c_{I,J} \ge R_{\mathrm{loc}}/64$ **do**
    Compute all cross inner products in $I \times J$
    if *none has absolute value greater than* σ **then return** *failure*
    Insert the pairs with inner product at least ρ into $\mathcal{P}$
return $\mathcal{P}$

3.2 Analysis

Proof of Lemma 3.1. The shared half features determine both the target signal and the background noise. For a point pair of correlation c, put $U_s = A_s(x)A_s(y)$ and $V_t = B_t(x)B_t(y)$. The Gaussian moments give

$$\mathbb{E}[U_s] = c^h, \qquad \mathbb{E}\Big[U_s^2\Big] = (1+2c^2)^h.$$

The pair kernel factors as

$$K(x,y) = \sum_{s,t} \phi_{s,t}(x)\phi_{s,t}(y) = \left(\sum_s U_s\right)\left(\sum_t V_t\right). \tag{3.6}$$

Independence of the two families therefore gives

$$\mathbb{E}[K(x,y)] = rc^d, \tag{3.7}$$

$$\begin{aligned}\mathbb{E}\Big[K(x,y)^2\Big] &= \Big[r_0(1+2c^2)^h + r_0(r_0-1)c^{2h}\Big]^2 \\ &\le r(1+2c^2)^d + 2r^{3/2}(1+2c^2)^{d/2}c^d + r^2c^{2d}.\end{aligned} \tag{3.8}$$

The middle term comes from feature coordinates that share one half feature. This is the additional noise introduced by compression, and choosing σ sufficiently below ρ will control it together with the diagonal term and the squared mean.

For a target pair with $c \ge \rho$, each half sum concentrates around r_0c^h, since its relative variance is at most $r_0^{-1}(\rho^{-2}+2)^h = O(m^{-1/3-5\xi/32})$. Hence, for sufficiently large m, with probability at least $1/2$,

$$K(x,y) \ge \frac{1}{4}r\rho^d. \tag{3.9}$$

Writing the signs on the two sides as a_x and b_y, the group score in (3.5) is

$$S_{I,J} = \sum_{x\in I, y\in J} a_x b_y K(x,y).$$

Conditional on the complete Gaussian feature collection $\mathcal{G}$, the independent signs satisfy

$$\mathbb{E}_{a,b}\Big[S_{I,J}^2 \mid \mathcal{G}\Big] = \sum_{x,y} K(x,y)^2, \qquad \mathbb{E}_{a,b}\Big[S_{I,J}^4 \mid \mathcal{G}\Big] \le 9\left(\sum_{x,y} K(x,y)^2\right)^2. \tag{3.10}$$

The second moment sums squared pair contributions, so a single target supplies variance even in the presence of other pairs. Together with the twofold Khintchine bound on the fourth moment, this preserves a constant probability of detecting that target. Paley–Zygmund and (3.9) give marking probability at least $1/32$ for every group pair containing a target.

For a group pair in which every cross pair has $|c| \le \sigma$, (3.8) and the first identity in (3.10) bound the marking probability by a constant multiple of

$$g^2\left(\frac{(1+2\sigma^2)^d}{r\rho^{2d}} + \frac{2(1+2\sigma^2)^{d/2}\sigma^d}{\sqrt{r}\,\rho^{2d}} + \left(\frac{\sigma}{\rho}\right)^{2d}\right). \tag{3.11}$$

The choices $\gamma = \xi/16$ and $\lambda \ge 2 + 1/\gamma$ make (3.11) $O(m^{-5\xi/16})$, including the mixed term. Thus, for $m \ge m_0$, a pure background group pair is marked with probability at most $1/256$.

Since the marking probabilities differ by a fixed constant, Chernoff bounds for the repeated trials give simultaneous expansion of every target group pair and no pure background group pair with probability at least $1-\zeta$.

For the running time, there are $H = O(m^{2/3})$ groups per side. A fixed matrix algorithm with cost $O(q^{\omega+\theta})$ gives score computation cost

$$m^{2(\omega+\theta)/3+\xi/2}\,\mathrm{poly}(n, \log m).$$

Generating half features and forming the padded products in (3.4) have respective exponents $4/3+\xi/4$ and $(\omega+\theta+2)/3+(\omega+\theta)\xi/4$. Since $\omega \geq 2$ and $\xi < 2/3$, the score computation dominates both. Choosing a fixed $\theta > 0$ with $2\theta/3 < \xi/2$ bounds the base work per trial by $m^C\,\mathrm{poly}(n, \log m)$.

Each expanded group pair containing an exceptional pair is charged to a distinct such pair. Halting at the first expanded group pair containing no exceptional pair ensures that at most $Q+1$ group pairs are expanded on every execution, even when the randomized detection fails. Their exact tests cost $O((Q+1)g^2 n)$, with the extra g^2 term absorbed by the base bound. □

4 Subquadratic search in spherical buckets

The local bound has two contributions to sum over the filters: compressed score computation and exact recovery. Bucket occupancy moments control the first, while the projected spherical tail controls the second. Fix $2\omega/3 < C < 2$, put $\xi = C - 2\omega/3$, and take σ from Lemma 3.1. Algorithm 2 pairs each of the $R = \lceil c_0 n \rceil$ independent product codes with a fresh random bipartition, so a fixed pair is separated with probability $1/2$. It retains input indices through projection for the final interval checks and removal of repeated reports.

Algorithm 2. Spherical pair search

Input: An indexed list $L = (x_1, \ldots, x_N) \subset S^{n-1}$, $0 < \tau \leq 1/2$, and $2\omega/3 < C < 2$.
Output: All unordered pairs $\{i, j\}$ with $\langle x_i, x_j \rangle \in \mathcal{I}_n$, or failure.

Set $\xi \leftarrow C - 2\omega/3$ and take σ from Lemma 3.1
Choose R, F, M and Δ_n as in Section 2.3
Set $G \leftarrow \lceil M \rceil$, $\zeta \leftarrow 2^{-An}$ for a sufficiently large constant A, and $\mathcal{P} \leftarrow \varnothing$
for $t = 1, \ldots, R$ **do**
    Draw an independent BDGL product code with F/R centers and a uniform bipartition $L = L_0 \sqcup L_1$
    Enumerate cap memberships and form the band buckets $B_f^a = \{x \in L_a : \tau \leq \langle x, f \rangle \leq \tau + \Delta_n\}$
    foreach *f with $B_f^0 \neq \varnothing$ and $B_f^1 \neq \varnothing$* **do**
        Project both sides by π_f in (2.4), retaining input indices, and split each side into blocks of at most G points
        foreach *cross block pair (U, V)* **do**
            $\mathcal{R} \leftarrow \mathsf{CompressedSearch}_\xi(U, V, \zeta)$ from Algorithm 1
            if *the local call fails* **then return** *failure*
            Insert every $\{i, j\} \in \mathcal{R}$ with $\langle x_i, x_j \rangle \in \mathcal{I}_n$ into $\mathcal{P}$
return $\mathcal{P}$

4.1 The bucket moments

Index all F band buckets by i, including the code index, and let m_i be the total number of points in bucket i. Set

$$S_1 = \sum_i m_i, \qquad S_2 = \sum_i m_i^2.$$

For each center, the independent input points lie in its band with probability at most $\mathcal{C}_n(\tau)$, so the binomial moments give

$$\mathbb{E}[S_1] \le FM, \qquad \mathbb{E}[S_2] \le F(M + M^2) \le 2FM^2. \tag{4.1}$$

For independent uniform $u, v \in S^{n-2}$, write

$$p_n(s) = \Pr[|\langle u, v\rangle| > s]\,.$$

For each fixed $s \in (0, 1)$, the spherical tail bound gives

$$p_n(s) \le \mathrm{poly}(n)\, 2^{-P(s)n}. \tag{4.2}$$

Let Q_{all} count exceptional cross pairs over all buckets, with multiplicity. Conditional on membership in a band, the two projected directions are independent and uniform, so

$$\mathbb{E}[Q_{\mathrm{all}}] \le FM^2 p_n(\sigma). \tag{4.3}$$

4.2 Search cost and recovery

Splitting each bucket side into blocks of at most $G = \lceil M \rceil$ points caps the recovery factor at $O(M^{2/3})$, while the second occupancy moment controls the cost of processing large buckets. Apply Lemma 3.1 to every cross block pair. For small buckets,

$$\sum_{m_i \le G} m_i^C \le S_1^{2-C} S_2^{C-1}.$$

Hölder's inequality and (4.1) give expected base cost $O(FM^C)$. For $m_i > G$, the two sides split into $O(m_i^2/G^2)$ cross block pairs, with total base cost $O(m_i^2 G^{C-2})$. Summing over these buckets gives the same expected bound.

Each original cross pair belongs to one block pair in its bucket, so (4.3) bounds the expected recovery work by $O(FM^{8/3} p_n(\sigma))$. With local failure probability $\zeta = 2^{-An}$, the logarithmic factors in Lemma 3.1 are polynomial in n, giving

$$\mathbb{E}[T_{\mathrm{search}}] \le F\big(M^C + M^{8/3} p_n(\sigma)\big)\,\mathrm{poly}(n). \tag{4.4}$$

Adding the filtering cost and using (4.2) gives

$$\mathbb{E}[T_{\mathrm{total}}] \le (N + FM)2^{o(n)} + F\big(M^C + M^{8/3} 2^{-P(\sigma)n}\big)\,\mathrm{poly}(n). \tag{4.5}$$

Proof of Theorem 1.1. Choose

$$c = \frac{P(\sigma)}{2(8/3 - C)} > 0.$$

When $M \le 2^{cn}$, (4.2) gives

$$M^{8/3} p_n(\sigma) \le M^C\,\mathrm{poly}(n)\, 2^{-P(\sigma)n/2},$$

so (4.4) proves the claimed search cost.

Product code coverage from Section 2.3 and the independent bipartitions place every target pair on opposite sides of a common band bucket with probability $1 - 2^{-\Omega(n)}$ simultaneously. Its projected correlation is at least $1/3 - o(1)$ by (2.5), so Lemma 3.1 recovers it. Choosing A sufficiently large gives simultaneous success of the at most $FN^2 = 2^{O(n)}$ local calls with the same probability, and the final exact checks retain precisely the target interval. □

5 Optimizing the filters

A larger cap makes each bucket more expensive to search, but also increases the probability that a useful pair shares a filter. Theorem 1.1 lets us compare these two effects through the single bucket exponent C.

5.1 The filtering tradeoff

Choose a small fixed $\varepsilon > 0$ and set

$$P(\tau) = \ell - \varepsilon, \qquad 1 - \tau^2 = \frac{3}{4}\, 2^{2\varepsilon}. \tag{5.1}$$

At list size $N = 2^{\ell n + o(n)}$, this gives $M = 2^{\varepsilon n + o(n)}$. The common cap estimate (2.6), together with (2.7), gives the actual filter count

$$F = 2^{k(\varepsilon) n + o(n)}, \qquad k(\varepsilon) = -\frac{1}{2} \log_2 \left(2^{2\varepsilon} - \frac{1}{3} \right). \tag{5.2}$$

For $\varepsilon < c$, the size condition in Theorem 1.1 holds for all sufficiently large n. The full expected cost in (1.3) becomes

$$\mathbb{E}[T_{\text{total}}] \le 2^{\ell n + o(n)} + 2^{[k(\varepsilon) + \varepsilon] n + o(n)} + 2^{[k(\varepsilon) + C\varepsilon] n + o(n)}. \tag{5.3}$$

Since $C > 1$, the search term dominates the membership term.

Expanding the filter rate gives

$$k(\varepsilon) = \kappa_{\text{B}} - \frac{3}{2}\varepsilon + \frac{3 \ln 2}{4} \varepsilon^2 + O(\varepsilon^3). \tag{5.4}$$

The relevant search exponent is therefore

$$k(\varepsilon) + C\varepsilon = \kappa_{\text{B}} + \left(C - \frac{3}{2} \right) \varepsilon + \frac{3 \ln 2}{4} \varepsilon^2 + O(\varepsilon^3).$$

To first order, increasing the bucket exponent by ε saves $3\varepsilon/2$ in the filter exponent and costs $C\varepsilon$ in the search exponent. Thus larger buckets improve the total exponent when $C < 3/2$. At $C = 3/2$, the linear saving is exhausted and the quadratic term is positive. Together with $C > 2\omega/3$, this explains the strict threshold $\omega < 9/4$.

Constructive formulations of the current bound $\omega \le 9/4$ appear in [Pass26]. However, Amanov, van den Berg, and Zuiddam show that the degeneration ingredients underlying this bound cannot lower it within their framework [ABZ26, Theorem 5.4], suggesting that improving on $9/4$ will require new ideas.

5.2 A smaller leading exponent

Suppose $\omega \le 9/4 - \eta$ for a fixed $\eta > 0$ and choose

$$C = \frac{3}{2} - \frac{\eta}{3} > \frac{2\omega}{3}. \tag{5.5}$$

The slack $\xi = C - 2\omega/3 \ge \eta/3$ lets us take the bucket size constant c in Theorem 1.1 to depend only on η. Choose a sufficiently small $\varepsilon = \varepsilon(\eta) > 0$ with $\varepsilon < c$ and $2\varepsilon < \ell$. Since $\ell < \kappa_{\text{B}}$, the expansion above gives a constant $\beta = \beta(\eta) > 0$ such that

$$\max\{\ell, k(\varepsilon) + C\varepsilon\} \le \kappa_{\text{B}} - \beta. \tag{5.6}$$

At $N = \lceil n^2 2^{\ell n} \rceil$, the bucket size condition holds, so (5.3) gives expected total cost

$$\mathbb{E}[T_{\text{total}}] \le 2^{(\kappa_B - \beta)n + o(n)}. \tag{5.7}$$

The next section uses this fixed saving to obtain the full lattice sieving bound in Corollary 1.2.

6 Consequences for lattice sieving

Section 5 gives a fixed exponent saving for one round of spherical pair search. To carry this saving through the sieve, each round must supply enough distinct shorter vectors, and the selected list must satisfy the same estimates in the next round.

6.1 Output supply and the spherical heuristic

For independent spherical points, the probability that a fixed pair belongs to $\mathcal{I}_n$ is

$$q_n = \Theta\left(n^{-1/2} 2^{-\ell n}\right).$$

At $N = \lceil n^2 2^{\ell n} \rceil$, the expected number of such pairs exceeds N by a polynomial factor. Rotational invariance makes distinct pair indicators pairwise independent even when they share an endpoint, so the second moment bound gives at least N such pairs with probability $1 - 2^{-\Omega(n)}$. Their differences have norm between $\sqrt{1 - 2b/n}$ and $\sqrt{1 - 2a/n}$, and distinct unordered pairs almost surely give distinct differences after fixing an orientation.

For lattice sieving, we assume the following form of the spherical heuristic underlying the BDGL analysis. Each round has $N = \lceil n^2 2^{\ell n} \rceil$ distinct vectors in a radial shell of relative width $O(1/n)$. Their normalized directions satisfy the membership and exceptional pair moment bounds (4.1) and (4.3) up to $2^{o(n)}$ factors, and pairs in a window of the form (1.1) supply a new list of N distinct nonzero vectors with a radius smaller by $1 - \Theta(1/n)$. We assume that these properties continue to hold after selecting the new list in each round.

Initialization and termination follow Nguyen and Vidick [NV08], as used in the BDGL sieve analysis [BDGL16, Section 7]. There are polynomially many rounds, and the $2^{o(n)}$ factors are absorbed by the fixed saving from Section 5.

6.2 The full sieve

Proof of Corollary 1.2. Use the parameters from Section 5.2. Coverage and the local search guarantee hold for fixed input lists, while the heuristic in Section 6.1 gives expected cost $2^{(\kappa_B - \beta)n + o(n)}$ per round by (5.7).

Run Algorithm 2 with operation limit $\lceil 2^{(\kappa_B - \beta/2)n} \rceil$ per round. Markov's inequality bounds the probability of reaching this limit by $2^{-\beta n/2 + o(n)}$, so a union bound over the polynomially many rounds, together with the search guarantee and heuristic output supply, gives high probability of success. For $\delta = \frac{3}{2}(1 - 2^{-\beta}) > 0$, every execution takes time at most

$$2^{(\kappa_B - \beta/2)n + o(n)} = (3/2 - \delta)^{n/2 + o(n)}. \qquad \square$$

Remark. For independent spherical inputs, process product codes in batches of $2^{(\ell - \varepsilon)n + o(n)}$ centers and local block pairs sequentially. Bucket concentration bounds the memberships stored in a batch by $2^{\ell n + o(n)}$, while the local workspace is at most $G^2 2^{o(n)}$. Since $2\varepsilon < \ell$ and the output dictionary has expected size $2^{\ell n + o(n)}$, the expected peak space is $2^{\ell n + o(n)}$, with the same time bound. For lattice sieving, retain only the required new list. The corresponding space bound also uses bucket occupancy concentration as part of the spherical heuristic.

References

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015. https://eprint.iacr.org/2015/046.

[ABZ26] Alimzhan Amanov, Maxim van den Berg, and Jeroen Zuiddam. Multiplying matrices via polynomials: exposition and barriers. arXiv:2610.30054, 2026. https://arxiv.org/abs/2610.30054.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 10–24, 2016. doi:10.1137/1.9781611974331.ch2. https://eprint.iacr.org/2015/1128.

[KKK18] Matti Karppa, Petteri Kaski, and Jukka Kohonen. A faster subquadratic algorithm for finding outlier correlations. *ACM Transactions on Algorithms*, 14(3), Article 31, pages 31:1–31:26, 2018. doi:10.1145/3174804. https://arxiv.org/abs/1510.03895.

[KL21] Elena Kirshanova and Thijs Laarhoven. Lower bounds on lattice sieving and information set decoding. In *Advances in Cryptology – CRYPTO 2021*, pages 791–820, 2021. doi:10.1007/978-3-030-84245-1_27. https://eprint.iacr.org/2021/785.

[NV08] Phong Q. Nguyen and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. *Journal of Mathematical Cryptology*, 2(2):181–207, 2008. doi:10.1515/jmc.2008.009.

[OAI26] OpenAI. An upper bound of 9/4 for the matrix multiplication exponent. OpenAI Math Release preprint, October 2, 2026. https://github.com/openai/math/blob/main/preprints/Matrix-Multiplication-Nine-Fourths-October-2-2026/paper.pdf.

[Pass26] Rafael Pass. Matrix multiplication from polynomial packings: A constructive and elementary proof of $\omega \leq 9/4$. arXiv:2610.28627, 2026. https://arxiv.org/abs/2610.28627.

[Val15] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and the closest pair problem. *Journal of the ACM*, 62(2), Article 13, 2015.