Worst-case Analysis of Lattice Enumeration Algorithm over Modules ⋆

Jiseung Kim[1], Changmin Lee[2], and Yongha Son[3]

[1] Jeonbuk National University
jiseungkim@jbnu.ac.kr
[2] Korea University
changminlee@korea.ac.kr
[3] Sungshin Women's University
yongha.son@sungshin.ac.kr

Abstract. We study lattice enumeration in rank-n modules over number fields. Building on the refined Euclidean analysis of Hanrot and Stehlé (CRYPTO 2007), we develop a module analogue of Kannan–Hanrot–Stehlé enumeration. Our approach works directly with pseudo-bases and introduces module counterparts of HKZ reduction and quasi-HKZ reduction. We also extend the notion of a lattice *profile* to the module setting and prove tail-product bounds for the $\mathbb{Z}$-basis obtained from a module-HKZ-reduced pseudo-basis. These bounds yield a strictly improved asymptotic enumeration complexity compared to the naive reduction that views a rank-n module as an (nd)-dimensional Euclidean lattice.

As a representative consequence, for $K = \mathbb{Q}[x]/\langle x^d + 1\rangle$ with d a power of two and a rank-n module $M \subset K^n$ with $n \geq 3$, we show that module-SVP can be solved in time $\Delta_K \cdot n^{\frac{nd}{2e}+o(nd)}$, where Δ_K is the discriminant of K. This improves over the Euclidean enumeration bound $(nd)^{\frac{nd}{2e}+o(nd)}$. As an application, for $R = \mathbb{Z}[x]/\langle x^t + 1\rangle$ with t a power of two and an ideal $I \subset R$, we obtain an ideal-SVP algorithm running in time $\exp\big(\frac{t}{2e}\ln\ln t + O(t)\big)$, improving upon the worst-case enumeration bound $\exp\big(\frac{t}{2e}\ln t + o(t)\big)$.

Keywords: Lattice Enumeration, Module Lattice, Module-SVP

1 Introduction

The Learning with Errors (LWE) problem [21] has become a cornerstone of modern cryptography, especially in the post-quantum setting. Its importance stems from both its versatility in cryptographic constructions [3,8] and its worst-case-to-average-case connection to lattice problems such as the Shortest Vector Problem (SVP) [21]. In particular, the hardness of LWE is supported by reductions from (approximate) SVP, a problem known to be NP-hard (under randomized

⋆ Author list in alphabetical order; see https://www.ams.org/profession/leaders/culture/JointResearchandItsPublicationfinal.pdf.

reductions) for certain approximation factors [10,14,19]. These connections provide a robust theoretical foundation for designing primitives believed to resist quantum adversaries.

Beyond standard Euclidean lattices, algebraic variants such as Ring-LWE [18] and Module-LWE [15] have emerged as leading candidates for practical post-quantum cryptography. They exploit algebraic structure to improve efficiency while retaining security guarantees comparable to LWE. Notably, the Module-LWE-based primitives *Kyber* [2] (KEM) and *Dilithium* [6] (signatures) were selected in the NIST Post-Quantum Cryptography standardization effort, underscoring the practical relevance of module lattices.

Despite this success, a fundamental algorithmic gap remains: while algebraic structure is crucial for efficiency in cryptographic implementations, it is still unclear to what extent it can be exploited to *accelerate* the core lattice algorithms underlying security analyses and concrete attacks. Several module-lattice algorithms have been proposed [4, 5, 13, 16, 20], yet they often do not outperform their Euclidean counterparts in a way that matches the gains seen at the level of cryptographic constructions. For instance, module-LLL [16] extends LLL [17] to the module setting, but the practical improvements over treating the module as an nd-dimensional Euclidean lattice are often modest. A similar phenomenon appears for enumeration-based methods, whose theoretical potential over modules has not been fully realized.

This leads to the following guiding question:

Can algorithms for module lattices be accelerated by leveraging their intrinsic module structure?

1.1 This work

We provide a partial answer by revisiting Kannan–Hanrot–Stehlé lattice enumeration [9, 12] in the setting of rank-n module lattices over a number field K of degree d. Our main contribution is a *module-aware* geometric analysis of enumeration that avoids the naive reduction to an nd-dimensional Euclidean lattice and instead exploits the block structure induced by the module. As a result, the enumeration node bound improves from the classical worst-case form $(nd)^{\frac{nd}{2e}+o(nd)}$ to $\Delta_K \cdot n^{\frac{nd}{2e}+o(nd)}$ (up to polynomial factors), once an appropriate module-HKZ notion is available.

Theorem 1.1 (Informal)**.** *Let K be a number field of degree d and discriminant Δ_K, and let $M \subset K^n$ be a rank-n module with $n \geq 3$, given by a pseudo-basis $((I_i, \mathbf{b}_i))_{1 \leq i \leq n}$ together with $\mathbb{Z}$-bases of the ideals I_i. Writing the total bit-length of input representation by B, one can find a shortest nonzero vector in M in time*

$$\mathsf{poly}(n, d, B, \log \Delta_K) \cdot \left(\Delta_K \cdot n^{\frac{nd}{2e}+o(nd)} \; + \; T_{\mathsf{HKZ}}(2d) \right)$$

where $T_{\mathsf{HKZ}}(2d)$ is a time complexity of HKZ on $2d$-dimension. In particular, using the bound $T_{\mathsf{HKZ}}(2d) = d^{\frac{d}{e}+o(d)}$ from [9], this becomes

$$\mathsf{poly}(n, d, B, \log \Delta_K) \cdot \left(\Delta_K \cdot n^{\frac{nd}{2e}+o(nd)} + d^{\frac{d}{e}+o(d)} \right).$$

A concrete application is obtained by combining the module analysis with the standard reduction from ideal-SVP to module-SVP in cyclotomic fields.

Application 1.2. For a power-of-two integer t, one can find a shortest nonzero element of an ideal $I \subset \mathbb{Z}[x]/\langle x^t + 1\rangle$ in time

$$\exp\left(\frac{t}{2e}\ln\ln t + O(t)\right)$$

up to polynomial factors, improving on the worst-case bound $\exp\left(\frac{t}{2e}\ln t + o(t)\right)$ from [9].

Technical overview. For a full-rank Euclidean lattice $\Lambda \subset \mathbb{R}^m$ with an HKZ-reduced basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_m)$, Hanrot–Stehlé [9, Thm. 3] show that for all $1 \le i \le m$,

$$\frac{\|\mathbf{b}_1\|^i}{\prod_{j=m-i+1}^{m}\|\mathbf{b}_j^*\|} \le \sqrt{m}^{\,i\left(1+\ln\frac{m}{i}\right)}. \tag{1}$$

Plugging (1) into the standard node-counting bound for enumeration [9,12] yields a worst-case running time $m^{\frac{m}{2e}+o(m)}$.

A naive way to handle a rank-n module $M \subset K^n$ (with $[K:\mathbb{Q}] = d$) is to embed it as an $m = nd$ dimensional Euclidean lattice, which leads to the bound $\sqrt{nd}^{\,\frac{nd}{e}+o(nd)}$.

Our approach introduces a notion of *module-HKZ-reduced pseudo-basis* and proves a sharper tail-product control for the $\mathbb{Z}$-basis $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_{nd})$ obtained from a module-HKZ-reduced pseudo-basis. Informally, for indices of the form $i = dk$ (block boundaries), we show a bound of the form

$$\frac{\|\mathbf{w}_1\|^i}{\prod_{j=nd-i+1}^{nd}\|\mathbf{w}_j^*\|} \;\lesssim\; \sqrt{nd}^{\,i}\cdot\sqrt{n}^{\,i\left(1+\ln\frac{nd}{i}\right)}. \tag{2}$$

This replaces the dependence on nd in (1) by a dependence on n at the block level and yields the improved enumeration bound $\sqrt{n}^{\,\frac{nd}{e}+o(nd)}$ (up to polynomial factors).

Application to Module-CVP. The same framework extends to module-CVP as in the Euclidean case [9], but it does not yield a comparable speedup. Let $\mathbf{D} = (\mathbf{d}_1, \ldots, \mathbf{d}_{nd})$ be the $\mathbb{Z}$-basis of M obtained from a (module-)reduced pseudo-basis and let $\mathbf{t} \in \mathrm{span}(M)$ be a target. Babai's nearest-plane algorithm [1] outputs $\mathbf{d} \in M$ such that

$$\|\mathbf{t} - \mathbf{d}\| \le \sqrt{nd}\cdot\max_{1\le i\le nd}\|\mathbf{d}_i^*\| \;=\; \sqrt{nd}\cdot\|\mathbf{d}_1^*\|. \tag{3}$$

Therefore, to guarantee finding a closest vector by enumeration, one may set the search radius $A = nd\cdot\|\mathbf{d}_1^*\|^2$. The corresponding node bound satisfies (cf. [9])

$$N_i^{\mathrm{CVP}} \;\lesssim\; \frac{2^{O(nd)}\cdot A^{i/2}}{\sqrt{nd}^{\,i}\cdot\prod_{j=nd-i+1}^{nd}\|\mathbf{d}_j^*\|} \;=\; \frac{2^{O(nd)}\cdot(\sqrt{nd}\cdot\|\mathbf{d}_1^*\|)^i}{\sqrt{nd}^{\,i}\cdot\prod_{j=nd-i+1}^{nd}\|\mathbf{d}_j^*\|}. \tag{4}$$

Since A already carries a $\sqrt{nd}$ factor and the worst case is attained for $i = nd$, the denominator becomes $\det(\Lambda(\mathbf{D}))$ and the bound essentially reduces to the full-dimensional Euclidean behavior. Consequently, unlike SVP, the module block structure does not lead to a meaningful improvement for CVP within this enumeration-based approach.

Replacing enumeration by sieving yields no gain. Our speedup is inherently tied to *enumeration*: it comes from a refined node-counting argument that exploits the module–HKZ Gram–Schmidt profile of the converted $\mathbb{Z}$-basis. In contrast, in the standard heuristic sieving model the running time depends essentially only on the dimension, namely $T_{\mathsf{sieve}}(m) = 2^{cm+o(m)}$ for some constant $c > 0$.

As a result, replacing the enumeration calls in our algorithms by sieving makes the overall cost dominated by a sieve in dimension nd, i.e., $T_{\mathsf{sieve}}(nd) = 2^{cnd+o(nd)}$, which matches applying sieving directly to the ambient lattice $\sigma(M)$. Hence, sieving does not preserve the enumeration-based advantage of the module structure. A detailed discussion of this point is deferred to the appendix Appendix C.

2 Preliminaries

Notations. Vectors and matrices are denoted in bold letters. For $n \in \mathbb{Z}_{\geq 1}$ and $R > 0$, the (closed) Euclidean ball of radius R in $\mathbb{R}^n$ is denoted by $\mathcal{B}_n(R)$, and

$$\mathrm{vol}(\mathcal{B}_n(R)) = R^n \cdot \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)}.$$

For any finite set U, the number of elements in U is denoted by $\#U$. For any measurable set $S \subset \mathbb{R}^n$, its volume is denoted by $\mathrm{vol}(S)$. The Euclidean norm of a vector $\mathbf{v} \in \mathbb{R}^n$ is denoted by $\|\mathbf{v}\|$. The natural logarithm of a positive real number x is denoted by $\ln x$, and the base-2 logarithm is denoted by $\log_2 x$ or simply $\log x$ when the base is clear from the context.

2.1 Lattice Backgrounds

A lattice $\Lambda \subset \mathbb{R}^m$ of rank n is the set of all integer combinations of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^m$:

$$\Lambda = \Lambda(\mathbf{B}) = \mathbb{Z}\mathbf{b}_1 + \cdots + \mathbb{Z}\mathbf{b}_n, \qquad \mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^{m \times n}.$$

The parameter m is called the (ambient) dimension and n the rank. If $m = n$, the lattice is said to be full-rank.

Gram–Schmidt Orthogonalization. Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of Λ. For each $i \in \{1, \ldots, n\}$, define $\mathbf{b}_i^*$ as the orthogonal projection of $\mathbf{b}_i$ onto $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^{\perp}$. The Gram–Schmidt orthogonalized family is $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*\}$

and we write $\mathbf{B}^* = (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$. For simplicity, let $\mathcal{P}_i$ denote the orthogonal projection onto $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^\perp$ (equivalently, onto $\mathrm{span}(\mathbf{b}_1^*, \ldots, \mathbf{b}_{i-1}^*)^\perp$).

For a full-rank lattice $\Lambda(\mathbf{B})$, its volume is $\det(\Lambda) = \mathrm{vol}(\Lambda) = \prod_{i=1}^n \|\mathbf{b}_i^*\|$, which is independent of the choice of the basis $\mathbf{B}$.

Definition 2.1 (Shortest Vector Problem (SVP))**.** *Given a basis of a lattice Λ, find a shortest nonzero vector $\mathbf{x} \in \Lambda \setminus \{\mathbf{0}\}$ such that $\|\mathbf{x}\| = \lambda_1(\Lambda)$, where $\lambda_1(\Lambda)$ denotes the minimum length of a nonzero lattice vector.*

Lemma 2.2 (Minkowski's theorem)**.** *For any rank-n lattice $\Lambda \subset \mathbb{R}^m$ with $n \geq 2$, there exists a nonzero lattice vector $\mathbf{x} \in \Lambda$ such that $\|\mathbf{x}\| \leq \sqrt{n} \cdot \det(\Lambda)^{1/n}$.*

Definition 2.3 (HKZ-reduction)**.** *A basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ is Hermite–Korkine–Zolotarev (HKZ) reduced if*

1. *$\|\mathbf{b}_1\| = \lambda_1(\Lambda(\mathbf{B}))$, and*
2. *for each $i \geq 2$, the projected basis $(\mathcal{P}_i(\mathbf{b}_i), \ldots, \mathcal{P}_i(\mathbf{b}_n))$ is HKZ-reduced in $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})^\perp$.*

As shown in [9, Thm. 3], an HKZ-reduced basis $\{\mathbf{b}_i\}_{i=1}^n$ satisfies

$$\frac{\|\mathbf{b}_1\|^i}{\prod_{j \geq n-i+1} \|\mathbf{b}_j^*\|} \leq \sqrt{n}^{\,i + i \ln \frac{n}{i}} \leq \sqrt{n}^{\,i + \frac{n}{e}}.$$

The proof uses only multiplicative relations among tail products of Gram–Schmidt norms. Consequently, one may replace the sequence $\{\|\mathbf{b}_j^*\|\}_{j=1}^n$ by any positive sequence obeying the same HKZ-type recursion, yielding the following abstraction. We defer the proof of Lemma 2.4 to Appendix A.

Lemma 2.4. *Let $\{v_i\}_{1 \leq i \leq n} \subset \mathbb{R}_{>0}$ satisfy*

$$v_i \leq \sqrt{n-i+1} \cdot \Big(\prod_{j \geq i} v_j\Big)^{\frac{1}{n-i+1}} \qquad (1 \leq i \leq n).$$

Then for every $1 \leq i \leq n$,

$$\frac{v_1^i}{\prod_{j \geq n-i+1} v_j} \leq \sqrt{n}^{\,i + i \ln \frac{n}{i}} \leq \sqrt{n}^{\,i + \frac{n}{e} + O(1)}.$$

2.2 Algebraic Number Theory Background

This section introduces basic algebraic objects. For more details, see [16]. Let K be a number field of degree d, and let $K_\mathbb{R} := K \otimes_\mathbb{Q} \mathbb{R}$. Let r_1 be the number of real embeddings and $2r_2$ the number of complex embeddings, so that $r_1 + 2r_2 = d$. Let $\sigma_1, \ldots, \sigma_{r_1+r_2}$ be the distinct embeddings into $\mathbb{R}$ (for $1 \leq i \leq r_1$) and into $\mathbb{C}$ (for $r_1 < i \leq r_1 + r_2$). We identify $K_\mathbb{R} \cong \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ via the canonical embedding

$\sigma : K \to \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ where $\sigma(x) = (\sigma_1(x), \ldots, \sigma_{r_1+r_2}(x))$. For $x \in K$, define the field norm and trace by

$$\mathcal{N}_{K/\mathbb{Q}}(x) := \prod_{i=1}^{r_1} \sigma_i(x) \cdot \prod_{j=1}^{r_2} \sigma_{r_1+j}(x)\overline{\sigma_{r_1+j}(x)},$$

$$\mathrm{Tr}_{K/\mathbb{Q}}(x) := \sum_{i=1}^{r_1} \sigma_i(x) \ + \ \sum_{j=1}^{r_2} \big(\sigma_{r_1+j}(x) + \overline{\sigma_{r_1+j}(x)}\big).$$

We also use the Euclidean norms

$$\|x\| := \sqrt{\sum_{i=1}^{r_1} |\sigma_i(x)|^2 + \sum_{i=r_1+1}^{r_1+r_2} 2|\sigma_i(x)|^2}, \qquad \|x\|_\infty := \max_{1 \le i \le r_1+r_2} |\sigma_i(x)|.$$

For $y \in K_\mathbb{R} \cong \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$, define $\overline{y}$ by complex conjugation on the $\mathbb{C}^{r_2}$-coordinates. Let $K_\mathbb{R}^+ := \{(y_1, \ldots, y_{r_1+r_2}) \in \mathbb{R}^{r_1+r_2} : y_i \ge 0\}$ viewed as the subset of $K_\mathbb{R}$ whose coordinates are nonnegative real numbers. Define $\sqrt{\cdot} : K_\mathbb{R}^+ \to K_\mathbb{R}^+$ as the coordinate-wise square root.

Let $R = \mathcal{O}_K$ be the ring of integers of K. A *fractional ideal* I of K is an additive subgroup of K closed under multiplication by R, such that there exists $0 \ne x \in R$ with $xI \subset R$ an integral ideal. The norm of a fractional ideal I is defined by $\mathcal{N}(I) = \frac{\mathcal{N}(xI)}{|\mathcal{N}(x)|}$ for any $0 \ne x \in R$ such that xI is an integral ideal; this is well-defined and multiplicative.

Any nonzero fractional ideal $I \subset K$ can be viewed as a rank-d lattice via σ: there exist $x_1, \ldots, x_d \in K$ such that $I = \mathbb{Z}x_1 + \cdots + \mathbb{Z}x_d$ and $(x_1, \ldots, x_d)$ is called a $\mathbb{Z}$-basis of I. The discriminant of K is defined by

$$\Delta_K = \big|\det([\sigma_i(x_j)]_{i,j})\big|^2$$

for any $\mathbb{Z}$-basis $(x_1, \ldots, x_d)$ of R.

Computing an explicit $\mathbb{Z}$-basis of a given fractional ideal generally requires standard computational algebraic number theory routines. Throughout this paper, we assume that every input fractional ideal is provided together with a $\mathbb{Z}$-basis.

2.3 Module Background

We call an (R-)module $M \subset K_\mathbb{R}^m$ any set of the form $M = I_1\mathbf{b}_1 + \cdots + I_n\mathbf{b}_n$, where each I_i is a nonzero fractional ideal of K and $\mathbf{b}_i \in K_\mathbb{R}^m$ are $K_\mathbb{R}$-linearly independent. The tuple $((I_1, \mathbf{b}_1), \ldots, (I_n, \mathbf{b}_n))$ is called a *pseudo-basis* of M, and n is the *rank* of M.

Lemma 2.5. *Let $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ be a pseudo-basis of a module M. Fix an index i, and assume that $\mathbf{b}_i \in M$ (equivalently, $\mathcal{O}_K \cdot \mathbf{b}_i \subseteq M$). Then $\mathcal{O}_K \subseteq I_i$, and in particular*

$$\mathcal{N}(I_i) \le 1.$$

Proof. Since $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ is a pseudo-basis of M, we have $M = \sum_{j=1}^{n} I_j \mathbf{b}_j$, and hence $I_i \mathbf{b}_i \subseteq M$. The assumption $\mathbf{b}_i \in M$ implies $\mathcal{O}_K \mathbf{b}_i \subseteq M$ as well. Thus the rank-1 submodule $\mathcal{O}_K \mathbf{b}_i$ is contained in $I_i \mathbf{b}_i$, which is possible only if $\mathcal{O}_K \subseteq I_i$. Taking norms gives $\mathcal{N}(I_i) \le \mathcal{N}(\mathcal{O}_K) = 1$. □

$\mathbb{Z}$-generators induced by a pseudo-basis. Let $M = \sum_{i=1}^{n} I_i \mathbf{b}_i \subset K_{\mathbb{R}}^m$ be given with a pseudo-basis $((I_i, \mathbf{b}_i))_{i \le n}$. Assume that each fractional ideal I_i is provided with a $\mathbb{Z}$-basis

$$I_i = \mathbb{Z} x_{i,1} + \cdots + \mathbb{Z} x_{i,d}.$$

Then M is generated as a $\mathbb{Z}$-module by the nd vectors $\{x_{i,j} \mathbf{b}_i\}_{i,j}$:

$$M = \sum_{i=1}^{n} \sum_{j=1}^{d} \mathbb{Z}(x_{i,j} \mathbf{b}_i).$$

In particular, these generators yield a concrete $\mathbb{Z}$-basis of the embedded lattice $\sigma(M)$ after applying any standard basis extraction procedure (e.g., the pseudo-basis to $\mathbb{Z}$-basis conversion in [7, 16]).

Extended canonical embedding. For $\mathbf{v} = (v_1, \ldots, v_m) \in K_{\mathbb{R}}^m$, define

$$\sigma(\mathbf{v}) := (\sigma(v_1), \ldots, \sigma(v_m)) \in (\mathbb{R}^{r_1} \times \mathbb{C}^{r_2})^m \cong \mathbb{R}^{md}.$$

Thus a rank-n module M yields an (embedded) $\mathbb{Z}$-lattice of rank nd. As a common abuse of notation, we use M to denote both the module and its embedded lattice $\sigma(M)$.

Inner products and norms. For $\mathbf{a}, \mathbf{b} \in K_{\mathbb{R}}^m$, define

$$\langle \mathbf{a}, \mathbf{b} \rangle_{K_{\mathbb{R}}} := \sum_{i=1}^{m} a_i \overline{b_i} \ \in K_{\mathbb{R}}, \qquad \langle \mathbf{a}, \mathbf{b} \rangle := \mathrm{Tr}\left(\langle \mathbf{a}, \mathbf{b} \rangle_{K_{\mathbb{R}}} \right) \ \in \mathbb{R}.$$

For $\mathbf{v} \in K_{\mathbb{R}}^m$, define

$$\|\mathbf{v}\|_{K_{\mathbb{R}}} := \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle_{K_{\mathbb{R}}}} \in K_{\mathbb{R}}^{+}, \qquad \|\mathbf{v}\| := \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle} \in \mathbb{R}_{\ge 0}.$$

Then $\|\mathbf{v}\|$ coincides with the Euclidean norm of the embedded vector $\sigma(\mathbf{v}) \in \mathbb{R}^{md}$. Let $\lambda_1(M)$ denote the shortest nonzero length in M with respect to $\|\cdot\|$.

Definition 2.6 (Module-SVP)**.** *Given a module $M \subset K_{\mathbb{R}}^m$, the module-SVP is to find a nonzero vector $\mathbf{x} \in M$ minimizing $\|\mathbf{x}\|$.*

Determinant and algebraic norm. For a pseudo-basis $((I_i, \mathbf{b}_i))_{i \le n}$, let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ and define $\det_{K_{\mathbb{R}}}(M)$ as in [16]. The lattice determinant satisfies $\det(M) = \Delta_K^{n/2} \cdot \mathcal{N}(\det_{K_{\mathbb{R}}}(M))$. Extending the ideal norm to vectors by $\mathcal{N}(\mathbf{v}) := \mathcal{N}(\|\mathbf{v}\|_{K_{\mathbb{R}}})$, define $\lambda_1^{\mathcal{N}}(M) := \inf\{\mathcal{N}(\mathbf{v}) : \mathbf{v} \in M \setminus \{\mathbf{0}\}\}$.

Lemma 2.7 ([16, Lem. 2.2]). *For any rank-n module M, we have*

$$d^{-d/2}\lambda_1(M)^d \Delta_K^{-1/2} \le \lambda_1^{\mathcal{N}}(M) \le d^{-d/2}\lambda_1(M)^d \le n^{d/2}\Delta_K^{1/2}\mathcal{N}(\det_{K_\mathbb{R}} M)^{1/n}.$$

Gram–Schmidt orthogonalization for modules. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n \in K_\mathbb{R}^m$ be $K_\mathbb{R}$-linearly independent. Define $\mathbf{b}_1^* = \mathbf{b}_1$ and for $i > 1$,

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j<i} \mu_{i,j}\mathbf{b}_j^*, \qquad \mu_{i,j} = \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle_{K_\mathbb{R}}}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle_{K_\mathbb{R}}} \in K_\mathbb{R}.$$

Each embedding of $\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle_{K_\mathbb{R}}$ is a positive real number, hence the inverse is well-defined in $K_\mathbb{R}$. Let $r_{i,i} = \|\mathbf{b}_i^*\|_{K_\mathbb{R}}$ and obtain a QR factorization $\mathbf{B} = \mathbf{QR}$ over $K_\mathbb{R}$, where $\bar{\mathbf{Q}}^T\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} = (r_{i,j})$ is upper-triangular. We denote by $\tau_i(\mathbf{v})$ the orthogonal projection of $\mathbf{v}$ onto $\mathrm{span}(\mathbf{b}_1, \ldots, \mathbf{b}_i)^\perp$.

Lemma 2.8 ([16, Lem. 2.6]). *Let $((I_i, \mathbf{b}_i))_{i\le n}$ be a pseudo-basis of $M \subset K_\mathbb{R}^m$. Then*

$$\det_{K_\mathbb{R}}(M) = \prod_{i=1}^n r_{i,i}I_i, \qquad \det(M) = \Delta_K^{n/2}\prod_{i=1}^n \mathcal{N}(r_{i,i}I_i).$$

Moreover, applying Minkowski's bound to the embedded $\mathbb{Z}$-lattice of rank nd yields

$$\lambda_1(M) \le \sqrt{nd}\cdot \det(M)^{1/(nd)} = \sqrt{nd}\cdot \Delta_K^{1/(2d)} \cdot \left(\prod_{i=1}^n \mathcal{N}(r_{i,i}I_i)\right)^{1/(nd)}. \tag{5}$$

Lemma 2.9 (Thm. 4, [7]). *Suppose a pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ of $M \subset K_\mathbb{R}^m$ and a full-rank set of vectors $(\mathbf{s}_i)$ of M are given. Then there exists an algorithm returning a pseudo-basis $\{(J_i, \mathbf{c}_i)\}_{i\le n}$ such that $\mathbf{c}_i \in M$ and $\mathbf{c}_i^* = \mathbf{s}_i^*$ for all i. Furthermore, if $M \subset K^m$, the running time is polynomial in $\log \Delta_K$ and the input bit-length.*

Handling bit-lengths. Lee et al. [16] proposed algorithms for handling the bit-length of ideals and pseudo-bases. All algorithms preserve the underlying module, while producing efficient representations of ideals and the R-factor of a pseudo-basis. For example, a scaled pseudo-basis implies $\mathcal{N}(I_i) \le 1$ for all i. All such procedures terminate in time polynomial in $\log \Delta_K$ and the input bit-length.

We now recall the notion of a *scaled* (and size-reduced) pseudo-basis.

Definition 2.10 ([16, Def. 3.5]). *A pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ is* scaled *if for all $i \le n$,*

$$R \subset I_i, \qquad \mathcal{N}(I_i) > 2^{-d^2}\Delta_K^{-1/2}, \qquad \|r_{i,i}\| \le 2^d \Delta_K^{1/(2d)}\mathcal{N}(r_{i,i}I_i)^{1/d}.$$

If, in addition, $\|r_{i,j}/r_{i,i}\| \le (4d)^d\Delta_K^{1/2}$ for all $i < j \le n$, it is called size-reduced.

Algorithm 1: Scaling the ideals

Input: A pseudo-basis $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ of a module M
Output: A scaled pseudo-basis of M

for $i \leftarrow 1$ **to** n **do**
  Use LLL to find $s_i \in r_{i,i} I_i \setminus \{0\}$ such that
  $$\|s_i\| \le 2^d \Delta_K^{1/(2d)} \mathcal{N}(r_{i,i} I_i)^{1/d}.$$
  Find $x_i \in I_i$ such that $s_i = r_{i,i} x_i$.
  Update $I_i \leftarrow I_i \cdot \langle x_i \rangle^{-1}$ and $\mathbf{b}_i \leftarrow x_i \mathbf{b}_i$.
end for
return $((I_i, \mathbf{b}_i))_{i \le n}$.

Lemma 2.11 ([16, Lem. 3.6]). *Given a pseudo-basis $((I_i, \mathbf{b}_i))_{i \le n}$ of a module M, Algo 1 returns a scaled pseudo-basis of M and preserves $\mathcal{N}(r_{i,i} I_i)$ for all i. The algorithm terminates in time polynomial in $\log \Delta_K$ and the input bit-length.*

Throughout this paper, we assume that every pseudo-basis is scaled (Definition 2.10) by applying scaling and size-reduction procedures to control bit-lengths.

Size-reduction on modules. We further introduce the notion of a size-reduced pseudo-basis. Fix a $\mathbb{Z}$-basis $(x_1, \ldots, x_d)$ of R. For $y = \sum_{t=1}^{d} y_t x_t \in K_{\mathbb{R}}$ with $y_t \in \mathbb{R}$, define

$$\lfloor y \rceil_R := \sum_{t=1}^{d} \lfloor y_t \rceil x_t \in R,$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer (with any fixed tie-breaking rule).

Algorithm 2: Size-reduction

Input: A pseudo-basis $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ of a module M
Output: A size-reduced pseudo-basis of M

for $j \leftarrow 1$ **to** n **do**
  for $i \leftarrow j - 1$ **downto** 1 **do**
    Compute $c_{i,j} \leftarrow \lfloor r_{i,j} / r_{i,i} \rceil_R$.
    Update $\mathbf{b}_j \leftarrow \mathbf{b}_j - c_{i,j} \mathbf{b}_i$.
  end for
end for
return $((I_i, \mathbf{b}_i))_{i \le n}$.

Lemma 2.12 ([16, Lem. 3.7]). *Given a pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ of a module M, Algo 2 returns a scaled, size-reduced pseudo-basis of M and preserves $\mathcal{N}(r_{i,i}I_i)$ for all i. The algorithm terminates in time polynomial in $\log \Delta_K$ and the input bit-length.*

3 Enumeration Algorithm over Euclidean Lattices

This section briefly recalls the analysis of Hanrot–Stehlé [9] for Kannan's lattice enumeration and the induced HKZ/SVP solver. Throughout the paper, this Euclidean enumeration procedure will be used as a subroutine within the module-HKZ routines.

Kannan's Enumeration. Let $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_d)$ be a $\mathbb{Z}$-basis of a rank-d lattice $\Lambda = \Lambda(\mathbf{B}) \subset \mathbb{R}^d$, and let $A > 0$. The enumeration algorithm outputs all $\mathbf{x} = \sum_{i=1}^{d} x_i \mathbf{b}_i \in \Lambda$ such that $\|\mathbf{x}\|^2 \le A$.

Algorithm 3: Lattice enumeration (depth-first search)

Input: A basis $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_d)$ and a bound $A > 0$
Output: All $\mathbf{x} = \sum_{i=1}^{d} x_i \mathbf{b}_i \in \Lambda(\mathbf{B})$ such that $\|\mathbf{x}\|^2 \le A$

Compute Gram–Schmidt data $(\mu_{i,j})_{1\le j<i\le d}$ and $(\|\mathbf{b}_i^*\|)_{1\le i\le d}$ for $\mathbf{B}$.
Initialize $x_1, \dots, x_d \leftarrow 0$, $l_1, \dots, l_d \leftarrow 0$, $S \leftarrow \emptyset$, and $i \leftarrow d$.
while $1 \le i \le d$ **do**
  Set $l_i \leftarrow \left(x_i + \sum_{j=i+1}^{d} x_j \mu_{j,i}\right)^2 \cdot \|\mathbf{b}_i^*\|^2$.
  if $\sum_{t=i}^{d} l_t \le A$ **then**
    if $i = 1$ **then**
      $S \leftarrow S \cup \{(x_1, \dots, x_d)\}$
      $x_1 \leftarrow x_1 + 1$
    else
      $i \leftarrow i - 1$
      Set $x_i \leftarrow \left\lceil -\sum_{j=i+1}^{d} x_j \mu_{j,i} - \sqrt{\frac{A - \sum_{t=i+1}^{d} l_t}{\|\mathbf{b}_i^*\|^2}} \right\rceil$.
    end if
  else
    $i \leftarrow i + 1$
    if $i \le d$ **then**
      $x_i \leftarrow x_i + 1$
    end if
  end if
end while
return S

Counting nodes. To measure the time complexity of the enumeration algorithm, we first count the number of integer tuples, called nodes, in the hyper-

ellipsoids depending on A. More precisely, for $k \in \{1, \dots, d\}$, let N_k denote the number of integer tuples $(y_{d-k+1}, \dots, y_d) \in \mathbb{Z}^k$ such that

$$\sum_{j=d-k+1}^{d} y_j^2 \cdot \|\mathbf{b}_j^*\|^2 \;\le\; A.$$

Then the running time of Algo 3 is bounded by $O\left(\sum_{k=1}^{d} N_k\right)$ up to a polynomial factor. A standard volume argument (see, e.g., [9, Sec. 4.1]) gives, for every $1 \le k \le d$,

$$N_k \le \mathsf{poly}(d) \cdot \frac{\mathrm{vol}(\mathcal{B}_k(\sqrt{A}))}{\prod_{j=d-k+1}^{d} \|\mathbf{b}_j^*\|} \le \mathsf{poly}(d) \cdot 2^{O(d)} \frac{A^{k/2}}{\sqrt{d}^k \cdot \prod_{j=d-k+1}^{d} \|\mathbf{b}_j^*\|}. \tag{6}$$

Here, $\mathrm{vol}(\mathcal{B}_k(\sqrt{A}))$ is the volume of the Euclidean ball of radius $\sqrt{A}$ in $\mathbb{R}^k$. Throughout, we keep explicit the dependence on tail Gram–Schmidt products, and absorb lower-order terms into $\mathsf{poly}(d)$.

Kannan's SVP solver / HKZ reduction. One can build an HKZ-reduction algorithm from enumeration, following Kannan [12] and Hanrot–Stehlé's analysis [9]. The procedure makes two recursive calls in dimension $d-1$; see Algo 4.

Algorithm 4: Kannan's SVP solver / HKZ reduction

Input: An LLL-reduced $\mathbb{Z}$-basis $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_d)$ of a lattice L
Output: An HKZ-reduced basis of the same lattice L

Compute the orthogonal projections $(\mathbf{b}'_2, \dots, \mathbf{b}'_d)$ of $(\mathbf{b}_2, \dots, \mathbf{b}_d)$ onto $\mathbf{b}_1^{\perp}$.
HKZ-reduce $(\mathbf{b}'_2, \dots, \mathbf{b}'_d)$ in dimension $d-1$ (recursive call).
Lift back to a basis of L by adding suitable integer multiples of $\mathbf{b}_1$ so that $|\mu_{i,1}| \le \frac{1}{2}$ for all $i \ge 2$.
if $\|\mathbf{b}_2^*\| < \|\mathbf{b}_1^*\|/2$ *i.e., B is not quasi-HKZ reduced* **then**
  Swap $\mathbf{b}_1$ and $\mathbf{b}_2$ and restart from the projection step.
end if
Call Algo 3 on $\mathbf{B}$ with $A = \|\mathbf{b}_1^*\|^2$ to find a shortest vector $\mathbf{b}_0 \in L$.
Replace the basis by an LLL-reduced basis of $\mathrm{span}_{\mathbb{Z}}\{\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_d\}$.
Compute the projections $(\mathbf{b}'_2, \dots, \mathbf{b}'_d)$ of $(\mathbf{b}_2, \dots, \mathbf{b}_d)$ onto $\mathbf{b}_1^{\perp}$.
HKZ-reduce $(\mathbf{b}'_2, \dots, \mathbf{b}'_d)$ in dimension $d-1$ (recursive call).
Lift back as above so that $|\mu_{i,1}| \le \frac{1}{2}$ for all $i \ge 2$.
return $(\mathbf{b}_1, \dots, \mathbf{b}_d)$

Definition 3.1 (Quasi-HKZ reduction [9, Thm. 3])**.** *A basis $(\mathbf{b}_1, \dots, \mathbf{b}_d)$ is* quasi-HKZ-reduced *if*

1. $\|\mathbf{b}_2^*\| \;\ge\; \|\mathbf{b}_1^*\|/2$, *and*
2. *the projected basis $(\mathcal{P}_2(\mathbf{b}_2), \dots, \mathcal{P}_2(\mathbf{b}_d))$ is HKZ-reduced in $\mathbf{b}_1^{\perp}$.*

Cost of the enumeration step on a quasi-HKZ basis. Assume $\mathbf{B}$ is quasi-HKZ-reduced and run enumeration with $A = \|\mathbf{b}_1^*\|^2$. Using $\|\mathbf{b}_1^*\| \le 2\|\mathbf{b}_2^*\|$ and (6), we obtain (up to polynomial factors)

$$N_k \le \frac{2^{O(d)} \cdot \|\mathbf{b}_1^*\|^k}{\sqrt{d}^k \cdot \prod_{j=d-k+1}^{d} \|\mathbf{b}_j^*\|} \le \frac{2^{O(d)} \cdot \|\mathbf{b}_2^*\|^k}{\sqrt{d}^k \cdot \prod_{j=d-k+1}^{d} \|\mathbf{b}_j^*\|}.$$

Since $(\mathcal{P}_2(\mathbf{b}_2), \ldots, \mathcal{P}_2(\mathbf{b}_d))$ is HKZ-reduced in dimension $d-1$, we may apply Lemma 2.4 (with $n = d-1$) to the tail Gram–Schmidt sequence, yielding

$$\frac{\|\mathbf{b}_2^*\|^k}{\prod_{j=d-k+1}^{d} \|\mathbf{b}_j^*\|} \le d^{\frac{k}{2} + \frac{d}{2e} + o(d)} \qquad (1 \le k \le d).$$

Therefore the enumeration step costs at most

$$O\left(\sum_{k=1}^{d} N_k\right) = d^{\frac{d}{2e} + o(d)}.$$

Recursion depth. It is known [11] that the loop of "reduce in dimension $d-1$ + possible swap" reaches a quasi-HKZ basis within $\log d + O(1)$ iterations. Let $C(d)$ denote the cost of HKZ reduction in dimension d. Then

$$C(d) \le (\log d + O(1))C(d-1) + d^{\frac{d}{2e} + o(d)},$$

which implies $C(d) \le d^{\frac{d}{2e} + o(d)}$ (see [9] for details).

4 Module HKZ-Reduction and $\mathbb{Z}$-Basis Conversion

This section provides the geometric toolkit needed for our module-SVP algorithm based on module-HKZ reduction in Section 5. In particular, the module-SVP algorithm operates on pseudo-bases of R-modules, while the final enumeration complexity is most conveniently analyzed on an explicit $\mathbb{Z}$-basis in the Euclidean space obtained via the canonical embedding. Accordingly, our goal here is twofold: (i) to formalize an HKZ-type notion for R-modules that supports a recursive, projection-based structure, and (ii) to convert a module-HKZ-reduced pseudo-basis into a carefully structured $\mathbb{Z}$-basis that is suitable for enumeration and amenable to norm estimates.

The main quantitative outcome is a *tail-product bound* for the Gram–Schmidt lengths of a $\mathbb{Z}$-basis $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{nd}]$ obtained from a module-HKZ-reduced pseudo-basis by a blockwise conversion procedure. Concretely, we will establish that for every $1 \le i \le nd$,

$$\frac{\|\mathbf{w}_1\|^i}{\prod_{j=nd-i+1}^{nd} \|\mathbf{w}_j^*\|} \le \sqrt{nd}^{\lceil i/d \rceil \cdot d} \cdot \sqrt{n}^{\,i \ln \frac{dn}{i} + o(nd)}. \tag{7}$$

This estimate is the key input for the running-time analysis of module enumeration in subsequent sections.

For comparison, a naive embedding of $M \subset K^n$ as an $m = nd$-dimensional Euclidean lattice would feed (8) into the standard enumeration analysis. Concretely, for an HKZ-reduced basis $(\mathbf{b}_1, \ldots, \mathbf{b}_{nd})$ of the embedded rank-nd lattice and all $1 \le i \le nd$, Hanrot–Stehlé [9, Thm. 3] prove

$$\frac{\|\mathbf{b}_1\|^i}{\prod_{j=nd-i+1}^{nd} \|\mathbf{b}_j^*\|} \;\le\; \sqrt{nd}^{\,i\left(1+\ln \frac{nd}{i}\right)}. \tag{8}$$

In our bound (7), the factor $\sqrt{nd}^{\lceil i/d \rceil \cdot d}$ is within a $\sqrt{nd}^{O(d)}$ multiplicative loss of $\sqrt{nd}^{\,i}$, while the main term improves the Euclidean exponent from $\sqrt{nd}^{\,i \ln(nd/i)}$ to $\sqrt{n}^{\,id \ln(n/i)}$ by exploiting the d-block structure. In this sense, (7) can be viewed as a module analogue of the Hanrot–Stehlé profile bound, and it is the key input for the running-time analysis of module enumeration in subsequent sections.

Throughout this section, we work in $K_{\mathbb{R}}^n$ equipped with the canonical inner product $\langle \cdot, \cdot \rangle_{K_{\mathbb{R}}}$, and we use σ to denote the (extended) canonical embedding into $\mathbb{R}^{nd}$.

We next introduce module-HKZ reduction and describe a blockwise $\mathbb{Z}$-basis conversion procedure whose d-blocks are HKZ-reduced as Euclidean lattice bases under σ. We then combine Minkowski-type bounds for projected rank-1 submodules with this blockwise HKZ structure to prove (7).

Definition 4.1 (Module-HKZ-reduction)**.** *Let K be a number field of degree d, and let $M \subset K^n$ be a rank-n module. A pseudo-basis $\big((I_i, \mathbf{b}_i)\big)_{1 \le i \le n}$ of M is* module-HKZ-reduced *if*

1. *$\|\mathbf{b}_1\| = \lambda_1(M)$, and*
2. *the orthogonal projections of $\{(I_i, \mathbf{b}_i)\}_{i \ge 2}$ onto $\mathbf{b}_1^{\perp}$ form a module-HKZ-reduced pseudo-basis of the module they generate.*

4.1 Block-wise HKZ $\mathbb{Z}$-basis conversion

Module-HKZ reduction is formulated at the level of pseudo-bases and projected rank-1 submodules $(I_i, \mathbf{b}_i^*)$. In particular, it provides strong control on their *volumes* (equivalently, on $\mathcal{N}(I_i \cdot \mathbf{b}_i^*)\sqrt{\Delta_K}$), but it does not directly control the lengths of vectors in a concrete $\mathbb{Z}$-basis of each module $I_i \cdot \mathbf{b}_i^*$. Since the enumeration algorithm analyzed in Section 5 runs on an explicit $\mathbb{Z}$-lattice in Euclidean space (via the embedding σ), we need a conversion from pseudo-bases to $\mathbb{Z}$-bases that turns these volume constraints into usable Gram–Schmidt bounds.

To this end, we convert a pseudo-basis into a $\mathbb{Z}$-basis $\mathbf{W}$ with a block structure adapted to the field degree: each d-block corresponds to the embedded ideal lattice $\sigma(I_i \cdot \mathbf{b}_i^*) \subset K_{\mathbb{R}}$, and we additionally enforce that every such block is HKZ-reduced as an ordinary Euclidean lattice basis under σ. This blockwise HKZ property is exactly what allows us to control the Gram–Schmidt profile of $\mathbf{W}$ in terms of the module-HKZ data. We use the following conversion procedure.

Algorithm 5: Block-wise HKZ $\mathbb{Z}$-basis conversion

Input: A module-HKZ-reduced pseudo-basis $\big((I_i, \mathbf{b}_i)\big)_{1\le i\le n}$ of a module M, with an LLL-reduced $\mathbb{Z}$-basis $\mathbf{c}_i$ of each ideal I_i

Output: A $\mathbb{Z}$-basis $\mathbf{W}$ of M

1 Obtain $\mathbf{Q}$ and $\big((I_i, \mathbf{r}_i)\big)_{1\le i\le n}$ by applying Gram–Schmidt orthogonalization / QR-factorization to $\big((I_i, \mathbf{b}_i)\big)_{1\le i\le n}$.

2 Form a block-upper-triangular $\mathbb{Z}$-basis

$$\mathbf{B}_{\mathbb{Z}} \;:=\; [\mathbf{c}_j \cdot r_{i,j}]_{1\le i,j\le n} \in K_{\mathbb{R}}^{n\times nd},$$

where $\mathbf{c}_j \cdot r_{i,j} = (c_{j,1} r_{i,j}, \dots, c_{j,d} r_{i,j}) \in K_{\mathbb{R}}^{d}$.

3 **for** $i = 1$ **to** n **do** `// HKZ in Euclidean dimension` d

4 | Compute $\mathbf{U}_i \in \mathbb{Z}^{d\times d}$ unimodular such that $\sigma(\mathbf{c}_i \cdot r_{i,i}) \cdot \mathbf{U}_i$ is HKZ-reduced.

5 **end for**

6 **return** $\mathbf{W} = \mathbf{Q} \cdot \mathbf{B}_{\mathbb{Z}} \cdot \mathsf{diag}(\mathbf{U}_1, \dots, \mathbf{U}_n)$.

Theorem 4.2. *Let $M \subset K^n$ be a rank-n module, and let $\big((I_i, \mathbf{b}_i)\big)_{1\le i\le n}$ be its module-HKZ-reduced pseudo-basis. Assume that an LLL-reduced $\mathbb{Z}$-basis of each ideal I_i, say $\mathbf{c}_i \in K_{\mathbb{R}}^d$, is given. Then Algo 5 outputs a $\mathbb{Z}$-basis $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_{nd})$ of M such that for every $i \in \{1, \dots, n\}$,*

$$\prod_{j=1}^{d} \|\mathbf{w}^*_{(i-1)d+j}\| = \mathcal{N}(I_i \cdot \mathbf{b}_i^*) \cdot \sqrt{\Delta_K}, \tag{9}$$

$$\|\mathbf{w}^*_{(i-1)d+1}\| = \|\mathbf{b}_i^*\|, \tag{10}$$

$$\|\mathbf{w}^*_{(i-1)d+k}\| \le \sqrt{d-k+1} \cdot \left(\prod_{j=(i-1)d+k}^{id} \|\mathbf{w}_j^*\| \right)^{1/(d-k+1)} \quad \text{for } 1 \le k < d. \tag{11}$$

Furthermore, the running time of the conversion is $\mathsf{poly}(B, \log \Delta_K) \cdot n \cdot 2^{O(d)}$, where B is the bit-size of inputs.

Proof. We first prove that $\mathbf{W}$ is a $\mathbb{Z}$-basis of M. The matrix $\mathbf{B}_{\mathbb{Z}}$ is a $\mathbb{Z}$-basis of $\mathbf{Q}^T \cdot M$ by construction, and multiplying on the right by $\mathsf{diag}(\mathbf{U}_1, \dots, \mathbf{U}_n)$ preserves the $\mathbb{Z}$-span since each $\mathbf{U}_i$ is unimodular. Finally, multiplying by $\mathbf{Q}$ maps back to M and preserves norms since $\mathbf{Q}$ is orthogonal.

For (9), note that each block $\{\mathbf{w}_{(i-1)d+1}, \dots, \mathbf{w}_{id}\}$ spans (over $\mathbb{Z}$) the embedded ideal lattice $\sigma(I_i \cdot \mathbf{b}_i^*)$ in $K_{\mathbb{R}}$. The standard determinant identity for ideal lattices gives $\det(\sigma(I)) = \mathcal{N}(I)\sqrt{\Delta_K}$ (see, e.g., [7, 16]), hence the product of Gram–Schmidt lengths of that block equals $\mathcal{N}(I_i \cdot \mathbf{b}_i^*)\sqrt{\Delta_K}$, proving (9). Equality (10) holds because the first Gram–Schmidt length of each HKZ-reduced d-block equals the length of the shortest vector in that block-lattice, which is $\|\mathbf{b}_i^*\|$ by its module-HKZ-reduced construction. Inequality (11) is precisely the usual HKZ tail inequality in Euclidean dimension d, applied blockwise.

For the complexity, the dominant step is computing HKZ in Euclidean dimension d for each block, which costs $2^{O(d)} \cdot \mathsf{poly}(B, \log \Delta_K)$ per block; multiplying by n yields the stated bound. □

4.2 Geometry of module-HKZ-reduced bases

Let $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{nd}]$ be the $\mathbb{Z}$-basis obtained by applying Algo 5 to a module-HKZ-reduced pseudo-basis $\big((I_i, \mathbf{b}_i)\big)_{1 \le i \le n}$. The conversion ensures that each d-block of $\mathbf{W}$ is HKZ-reduced (under σ), while module-HKZ reduction provides volume control on the projected rank-1 submodules $(I_i, \mathbf{b}_i^*)$. Combining these two ingredients, we now prove the tail-product inequality (7).

Theorem 4.3. *Let K be a number field of degree d, and let $M \subset K^n$ be a rank-n module. Suppose $\big((I_i, \mathbf{b}_i)\big)_{1 \le i \le n}$ is module-HKZ-reduced, and let $\mathbf{W} = (\mathbf{w}_1, \cdots, \mathbf{w}_{nd})$ be the $\mathbb{Z}$-basis output by Algo 5. Then for every $1 \le i \le nd$,*

$$\frac{\|\mathbf{w}_1\|^i}{\prod_{j=nd-i+1}^{nd} \|\mathbf{w}_j^*\|} \le \sqrt{nd}^{\lceil i/d \rceil d} \cdot \sqrt{n}^{\,i \ln \frac{nd}{i} + o(nd)}.$$

Proof. Since the input pseudo-basis is module-HKZ-reduced, applying Minkowski's inequality to successive projected rank-1 submodules yields, for every $1 \le i \le n$,

$$\|\mathbf{b}_i^*\| \le \sqrt{(n-i+1)d} \cdot \Delta_K^{1/(2d)} \cdot \left(\prod_{j=i}^{n} \mathcal{N}(I_j \cdot \mathbf{b}_j^*) \right)^{1/(n-i+1)d}. \tag{12}$$

Using the blockwise volume identity for the $\mathbb{Z}$-basis $\mathbf{W}$ output by Algo 5 (cf. Equation (9)), we have

$$\prod_{j=(i-1)d+1}^{nd} \|\mathbf{w}_j^*\| = \Delta_K^{(n-i+1)/2} \cdot \prod_{j=i}^{n} \mathcal{N}(I_j \cdot \mathbf{b}_j^*),$$

and thus (12) becomes

$$\|\mathbf{w}_{(i-1)d+1}^*\| \le \sqrt{(n-i+1)d} \cdot \left(\prod_{j=(i-1)d+1}^{nd} \|\mathbf{w}_j^*\| \right)^{1/(n-i+1)d}. \tag{13}$$

Moreover, inside each block, applying Minkowski to successive rank-1 submodules in dimension d yields, for $1 \le k < d$,

$$\|\mathbf{w}_{(i-1)d+k}^*\| \le \sqrt{d-k+1} \cdot \left(\prod_{j=(i-1)d+k}^{id} \|\mathbf{w}_j^*\| \right)^{1/(d-k+1)}. \tag{14}$$

Combining (13), (14) with the blockwise volume identity (again from Equation (9)) gives the following constraints on $\|\mathbf{w}_j^*\|$:

(a) For every block i,

$$\prod_{j=(i-1)d+1}^{id} \|\mathbf{w}_j^*\| = \mathcal{N}(I_i \cdot \mathbf{b}_i^*) \cdot \sqrt{\Delta_K}.$$

(b) For every $1 \le i \le n$,

$$\|\mathbf{w}_{(i-1)d+1}^*\| \le \sqrt{(n-i+1)d} \cdot \left(\prod_{j=(i-1)d+1}^{nd} \|\mathbf{w}_j^*\| \right)^{1/(n-i+1)d}.$$

This is an global constraint: it controls the first Gram–Schmidt length of block i in terms of the entire *tail* from block i to block n, coming from Minkowski applied to projected rank-1 submodules across successive blocks.

(c) For every $1 \le i \le n$ and $1 \le k < d$,

$$\|\mathbf{w}_{(i-1)d+k}^*\| \le \sqrt{d-k+1} \cdot \left(\prod_{j=(i-1)d+k}^{id} \|\mathbf{w}_j^*\| \right)^{1/(d-k+1)}.$$

This is an local constraint: it only relates Gram–Schmidt lengths *within the same d-block*, coming from Minkowski applied to successive rank-1 submodules inside the d-dimensional Euclidean lattice $\sigma(I_i \mathbf{b}_i)$.

Motivated by these, define $v_1, \ldots, v_{nd} \in \mathbb{R}_{>0}$ as the (unique) positive solution to the *blockwise* equalities obtained by turning (a)–(c) into equalities:

(a*) For every block i,

$$\prod_{j=(i-1)d+1}^{id} v_j = \mathcal{N}(I_i \cdot \mathbf{b}_i^*) \cdot \sqrt{\Delta_K}.$$

(b*) For every $1 \le i \le n$,

$$v_{(i-1)d+1} = \sqrt{(n-i+1)d} \cdot \left(\prod_{j=(i-1)d+1}^{nd} v_j \right)^{1/(n-i+1)d}.$$

(c*) For every $1 \le i \le n$ and $1 \le k < d$,

$$v_{(i-1)d+k} = \sqrt{d-k+1} \cdot \left(\prod_{j=(i-1)d+k}^{id} v_j \right)^{1/(d-k+1)}.$$

We first prove the following lemmas. These proofs are deferred to Appendix B.

Lemma 4.4 (Existence and uniqueness of the extremal profile)**.** *There exists a* unique *vector* $V = (v_1, \dots, v_{nd}) \in \mathbb{R}^{nd}_{>0}$ *satisfying the equalities* (b*) *and* (c*) *together with* (a*)*.*

Lemma 4.5. *Let* $m = nd$*, and let* $\alpha_j := \log \|\mathbf{w}_j^*\|$*,* $\beta_j := \log v_j$*. Then for every* $1 \le i \le m$*,*

$$\frac{\|\mathbf{w}_1^*\|^i}{\prod_{j=m-i+1}^{m} \|\mathbf{w}_j^*\|} \le \frac{v_1^i}{\prod_{j=m-i+1}^{m} v_j}. \tag{15}$$

We now complete the proof of Theorem 4.3. By Lemma 4.5, it suffices to show that for every $1 \le i \le nd$,

$$\frac{v_1^i}{\prod_{j=nd-i+1}^{nd} v_j} \le (\sqrt{nd})^{\lceil i/d \rceil d} \cdot \sqrt{n}^{\,i \ln \frac{nd}{i} + o(nd)}. \tag{16}$$

Write $i = kd + r$ with integers $k \ge 0$ and $0 \le r < d$, and set $s := \lceil i/d \rceil$ (so $s = k$ if $r = 0$, and $s = k + 1$ if $r > 0$). Split the ratio as

$$\frac{v_1^i}{\prod_{j=nd-i+1}^{nd} v_j} = \underbrace{\frac{v_1^r}{\prod_{j=nd-i+1}^{nd-kd} v_j}}_{=:R} \cdot \underbrace{\frac{v_1^{kd}}{\prod_{j=nd-kd+1}^{nd} v_j}}_{=:Q}. \tag{17}$$

Step 1 (bounding the partial-block factor R**)** If $r = 0$ then $R = 1$ and there is nothing to prove. Assume $r > 0$ (so $s = k + 1$). Since the sequence $(v_j)_{1 \le j \le nd}$ is monotone decreasing, we have $\prod_{j=nd-i+1}^{nd-kd} v_j \ge v_{nd}^r$ and thus

$$R = \frac{v_1^r}{\prod_{j=nd-i+1}^{nd-kd} v_j} \le \left(\frac{v_1}{v_{nd}}\right)^r \le \left(\frac{v_1}{v_{nd}}\right)^d. \tag{18}$$

To eliminate the factor v_1/v_{nd}, we use the Hermite/Minkowski constraints inside blocks: by (c*) and the block product constraints (a*), the last block $\{nd - d + 1, \dots, nd\}$ satisfies the Euclidean HKZ tail inequality, hence

$$\frac{v_{nd-d+1}^d}{\prod_{j=nd-d+1}^{nd} v_j} \le (\sqrt{d})^d. \tag{19}$$

Since $v_1 \ge v_{nd-d+1} \ge v_{nd}$ and $\prod_{j=nd-d+1}^{nd} v_j$ is fixed, (19) implies $v_1/v_{nd} \le (\sqrt{d})^{O(1)} = \sqrt{nd}^{\,o(1)}$. Plugging this into (18) gives

$$R \le (\sqrt{d})^{O(d)} \le (\sqrt{nd})^d \cdot \sqrt{n}^{\,o(nd)}. \tag{20}$$

Step 2 (block geometric means) Define the block geometric means

$$V_t := \left(\prod_{j=(t-1)d+1}^{td} v_j\right)^{1/d} \qquad (1 \le t \le n).$$

The constraint (b*) yields

$$V_t = \sqrt{n-t+1} \cdot \left(\prod_{u=t}^{n} V_u \right)^{1/(n-t+1)}.$$

Hence $(V_1, \ldots, V_n)$ satisfies the hypotheses of Lemma 2.4.

Step 3 (bounding the full-block factor Q) If $k = 0$ then $Q = 1$. Otherwise, the tail product $\prod_{j=nd-kd+1}^{nd} v_j$ contains exactly the last k blocks, and

$$Q = \frac{v_1^{kd}}{\prod_{j=nd-kd+1}^{nd} v_j} = \left(\frac{v_1^k}{\prod_{u=n-k+1}^{n} V_u} \right)^d.$$

Moreover, $v_1 = \sqrt{d} V_1$ by (b*) at $t = 1$, so

$$Q = (\sqrt{d})^{kd} \cdot \left(\frac{V_1^k}{\prod_{u=n-k+1}^{n} V_u} \right)^d.$$

Applying Lemma 2.4 to $(V_1, \ldots, V_n)$ gives

$$\frac{V_1^k}{\prod_{u=n-k+1}^{n} V_u} \le \sqrt{n}^{k(1+\ln(n/k))+O(1)}.$$

Therefore,

$$Q \le (\sqrt{nd})^{kd} \cdot \sqrt{n}^{kd \ln(n/k)+O(d)}. \tag{21}$$

Step 4 (assembling and rewriting in terms of i) If $r = 0$ then $s = k$ and $i = kd$, so (21) already gives

$$Q \le (\sqrt{nd})^{sd} \cdot \sqrt{n}^{id \ln(n/i)+o(nd)}.$$

Assume $r > 0$ (so $s = k + 1$). Combining (20) and (21) yields

$$RQ \le (\sqrt{nd})^{d} \cdot (\sqrt{nd})^{kd} \cdot \sqrt{n}^{kd \ln(n/k)+o(nd)} = (\sqrt{nd})^{sd} \cdot \sqrt{n}^{kd \ln(n/k)+o(nd)}.$$

Finally, since $i = kd + r$ with $0 < r < d$, we have $k = \lfloor i/d \rfloor$ and thus $k = \Theta(i/d)$, and

$$kd \ln \frac{n}{k} \le (kd + d) \ln \frac{n}{kd + d} \;=\; id \ln \frac{n}{i} + O(d \ln n).$$

Absorbing the additive $O(d \ln n)$ term into the $\sqrt{n}^{o(nd)}$ slack completes the proof of (16). □

5 An Algorithm for Module HKZ Reduction

In the Euclidean setting, Hanrot–Stehlé [9] analyze Kannan's enumeration through two ingredients: (i) a tail-product bound for (quasi-)HKZ bases, which translates into a node-counting bound for enumeration, and (ii) a short loop of the form "HKZ in dimension $d-1$ + possible swap", which reaches quasi–HKZ after at most $\log d + O(1)$ iterations.

Our module analogue follows the same blueprint as Hanrot–Stehlé. We first introduce an appropriate relaxation of module–HKZ reduction, namely *quasi–module–HKZ reduction* (Section 5.2). We then design and analyze an iterative reduction procedure that reaches a quasi–module–HKZ-reduced pseudo-basis in only a logarithmic number of iterations, mirroring the Hanrot–Stehlé loop in the Euclidean setting.

Why a head-recursive balancing is ineffective. A natural (but ultimately unhelpful) way to define a quasi–module–HKZ analogue would be to enforce a *front* balance, e.g.,

$$\mathcal{N}(I_2\mathbf{b}_2^*) \geq \mathcal{N}(I_1\mathbf{b}_1^*)/c$$

for some constant $c > 1$, by mimicking the swap step. However, unlike in the Euclidean case, performing such a swap—even if followed by module-LLL [16]—does not come with a provable monotone progress measure for the algebraic sizes $\mathcal{N}(I_i\mathbf{b}_i^*)^{1/d}$.

One may attempt to replace module-LLL by a stronger rank-2 module–HKZ step on the first two blocks. Yet this introduces an intrinsic *field-dependent* loss. Indeed, module–HKZ in rank 2 can only control the two successive blocks up to a discriminant factor: in the simplest form, it yields a balance of the type

$$\mathcal{N}(I_2\mathbf{b}_2^*)^{1/d} \gtrsim \Delta_K^{-1/d} \cdot \mathcal{N}(I_1\mathbf{b}_1^*)^{1/d}, \tag{22}$$

(or equivalently $\lambda_1(I_1\mathbf{b}_1) \lesssim \Delta_K^{1/d}\lambda_1(I_2\mathbf{b}_2^*)$ at the Euclidean level), reflecting the unavoidable discriminant term in the blockwise volume identity.

The problem is that such a $\Delta_K^{1/d}$-loss propagates *multiplicatively* in the node-counting bound for enumeration. To see the effect, consider a pseudo-basis whose last $n-1$ blocks already satisfy the desired module–HKZ conditions with a $\mathbb{Z}$-basis $\{\mathbf{w}_i\}_{i\leq nd}$, and suppose we enumerate with target radius comparable to $\lambda_1(I_1\mathbf{b}_1)$. At depth i (Euclidean dimension), the standard bound has the shape

$$N_i \lesssim \frac{\lambda_1(I_1\mathbf{b}_1)^i}{(\sqrt{nd})^i \cdot \prod_{j=m-i+1}^{m} \|\mathbf{w}_j^*\|}, \qquad m := nd.$$

Replacing $\lambda_1(I_1\mathbf{b}_1)$ using (22) therefore incurs a factor $\Delta_K^{i/d}$, yielding

$$N_i \lesssim \Delta_K^{i/d} \cdot \frac{\lambda_1(I_2\mathbf{b}_2^*)^i}{(\sqrt{nd})^i \cdot \prod_{j=m-i+1}^{m} \|\mathbf{w}_j^*\|}.$$

By Theorem 4.3, we obtain an estimate of the form

$$N_i \lesssim \Delta_K^{i/d} \cdot \sqrt{n}^{id\ln(n/i)}.$$

Since the dominating depths satisfy $i = \Theta(m)$ (near the maximizer of the profile bound), this translates into an overhead of size $\Delta_K^{\Theta(m/d)} = \Delta_K^{\Theta(n)}$ in the final complexity. For instance, if K is a power-of-two cyclotomic field of degree d, then $\Delta_K = d^d$ and thus $\Delta_K^{\Theta(n)} = d^{\Theta(nd)}$. Consequently, enforcing strong reduction *at the head* would erase the asymptotic benefit of module enumeration, essentially reducing to the naive nd-dimensional Euclidean behavior up to field-dependent factors.

This motivates our tail-recursive strategy. Rather than trying to force progress at the head—which would repeatedly incur field-dependent losses—we keep the first $n-1$ blocks reduced and enforce only a mild two-block balance at the tail. In particular, a naive swap of the last two pseudo-basis elements $(I_{n-1}, \mathbf{b}_{n-1})$ and $(I_n, \mathbf{b}_n)$ does not come with a provable progress measure (i.e., a guaranteed decrease of the relevant algebraic size quantities), so we instead use a rank-2 module–HKZ subroutine on the projected tail.

Road map. We conclude this introductory discussion with a brief road map of the section. In Section 5.1, we present the rank-2 module–HKZ subroutine that serves as the analogue of the Euclidean "swap" step, and explain how it acts on the projected tail. In Section 5.2, we introduce the relaxed notion of *quasi–module–HKZ reduction*, which captures the invariant maintained by the iteration while imposing only a mild balance condition at the tail. Then, in Section 5.3, we describe the full reduction algorithm that combines quasi–module–HKZ reduction with the rank-2 module–HKZ subroutine to compute a genuinely module–HKZ-reduced pseudo-basis. The correctness and complexity analyses are given in Sections 5.4 and 5.5, respectively, and we conclude with applications in Section 5.6.

5.1 A rank-2 module–HKZ subroutine

We first isolate the rank-2 subroutine that serves as the module analogue of the Euclidean "swap" step, namely line 5 of Algo 4.

As a simple and fully explicit instantiation of this rank-2 module-SVP oracle, we solve it by embedding the module into a $2d$-dimensional Euclidean lattice and running standard enumeration.

To calibrate the cost of this approach, let K be a number field of degree d, and let $M \subset K_{\mathbb{R}}^2$ be a rank-2 R-module given by a pseudo-basis $((I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2))$. Using the (extended) canonical embedding $\sigma : K_{\mathbb{R}}^2 \to \mathbb{R}^{2d}$, we obtain the Euclidean lattice

$$\Lambda := \sigma(M) \ \subset \ \mathbb{R}^{2d},$$

of rank $2d$. Concretely, if each ideal I_i is given as $I_i = \sum_{j=1}^{d} \mathbb{Z} \cdot x_{i,j}$, then Λ is $\mathbb{Z}$-generated by the $2d$ vectors $\{\sigma(x_{i,j}\mathbf{b}_i)\}_{i\in\{1,2\},j\in\{1,\dots,d\}}$.

Once a $\mathbb{Z}$-basis of Λ is obtained and LLL-reduced, one can apply Kannan's enumeration-based SVP solver (HKZ reduction) from Section 3 directly in Euclidean dimension $2d$. By the Hanrot–Stehlé analysis [9] recalled above, this

yields the bound

$$\mathsf{SVP}\text{ on }\Lambda\subset\mathbb{R}^{2d}\quad\text{can be solved in time}\quad (2d)^{\frac{2d}{2e}+o(2d)}=(2d)^{\frac{d}{e}+o(d)},\qquad(23)$$

up to factors polynomial in $\log\Delta_K$ and the input bit-length.

Algorithm 6: Rank-2 module–HKZ reduction (subroutine)

Input: A rank-2 pseudo-basis $((I_1,\mathbf{b}_1),(I_2,\mathbf{b}_2))$ of a submodule $M_2\subset K^n$, together with $\mathbb{Z}$-bases of I_1, I_2

Output: A module–HKZ-reduced rank-2 pseudo-basis $((J_1,\mathbf{c}_1),(J_2,\mathbf{c}_2))$ of the same submodule M_2

1 Convert $((I_1,\mathbf{b}_1),(I_2,\mathbf{b}_2))$ into a $\mathbb{Z}$-basis $\mathbf{V}=(\mathbf{v}_1,\dots,\mathbf{v}_{2d})$ using Algo 5.

2 Run enumeration (Algo 3) on $\mathbf{V}$ to obtain a shortest nonzero vector $\mathbf{c}_1\in M_2$.

3 Project $\mathbf{V}$ onto $\mathbf{c}_1^{\perp}$ and run enumeration again to obtain a second independent vector $\mathbf{c}_2'\in M_2\cap\mathbf{c}_1^{\perp}$.

4 Lift $\mathbf{c}_2'$ to $\mathbf{c}_2\in M_2$ by adding a suitable R-multiple of $\mathbf{c}_1$ so that $\lfloor\mu_{2,1}\rceil_R=0$ (module size-reduction).

5 Call the pseudo-basis update routine (Lemma 2.9) with $((I_1,\mathbf{b}_1),(I_2,\mathbf{b}_2))$ and $(\mathbf{c}_1,\mathbf{c}_2)$ to obtain ideals (J_1,J_2) such that $((J_1,\mathbf{c}_1),(J_2,\mathbf{c}_2))$ is a pseudo-basis of M_2. Return $((J_1,\mathbf{c}_1),(J_2,\mathbf{c}_2))$.

Remark 1. The projection step computes SVP in the projected $\mathbb{Z}$-lattice $\pi_{\mathbf{c}_1^{\perp}}(\sigma(M_2))$, hence returns a vector $\mathbf{c}_2'\in M_2$ that is independent from $\mathbf{c}_1$. The lifting by an R-multiple of $\mathbf{c}_1$ enforces the module size-reduction condition $\lfloor\mu_{2,1}\rceil_R=0$.

5.2 Quasi–module–HKZ reduction

We now define an intermediate notion that serves as a tail-recursive analogue of quasi–HKZ reduction in the Euclidean setting.

Definition 5.1 (Quasi–module–HKZ reduction (tail-recursive))**.** *Let K be a number field of degree d, and let $M\subset K^n$ be a rank-n module. A pseudo-basis $((I_i,\mathbf{b}_i))_{1\le i\le n}$ of M is* quasi–module–HKZ-reduced *(in the tail-recursive sense) if*

1. ***(Two-block balance on the last two blocks)***

$$\mathcal{N}(I_n\cdot\mathbf{b}_n^*)^{1/d}\ge\frac{1}{3\Delta_K^{1/d}}\mathcal{N}(I_{n-1}\cdot\mathbf{b}_{n-1}^*)^{1/d}.$$

2. ***(Module–HKZ on the first $n-1$ blocks)*** *the truncated pseudo-basis $\big((I_i,\mathbf{b}_i)\big)_{1\le i\le n-1}$ is module–HKZ-reduced for the rank-$(n-1)$ submodule*

$$M_{<n}\;:=\;\sum_{i=1}^{n-1}I_i\mathbf{b}_i\;\subseteq\;M.$$

5.3 Computing module–HKZ reduction

We now give Algo 7, our main procedure for computing a module–HKZ-reduced pseudo-basis.

5.4 Correctness

We prove that Algo 7 outputs a module–HKZ-reduced pseudo-basis of M. The argument follows Kannan's HKZ template, with a tail-recursive quasi-reduction phase: Parts (I)–(II) maintain module–HKZ on the first $n-1$ blocks while enforcing the two-block balance on $(n-1, n)$, Part (III) installs a true shortest vector, and Part (IV) performs the final recursive reduction in $\mathbf{d}_1^{\perp}$.

Proposition 5.2. *If the loop formed by Parts (I)–(II) of Algo 7 terminates, then the current pseudo-basis is quasi–module–HKZ-reduced in the sense of Definition 5.1.*

Proof. Upon termination, the loop guard is false, so Definition 5.1(1) holds. In the last iteration, Part (II) recursively applies Algo 7 to the truncated pseudo-basis of $M_{<n} := \sum_{i=1}^{n-1} I_i \mathbf{b}_i$, and then lifts back with size-reduction. By induction on the rank, the truncated pseudo-basis is module–HKZ-reduced, proving Definition 5.1(2). □

Lemma 5.3. *In Part (III) of Algo 7, enumeration returns $\mathbf{c} \in M \setminus \{0\}$ with $\|\mathbf{c}\| = \lambda_1(M)$, and after the update we have $\|\mathbf{d}_1\| = \lambda_1(M)$.*

Proof. The converted $\mathbb{Z}$-basis spans M, so Algo 3 returns a shortest nonzero element of M. The update sets $\mathbf{d}_1 := \mathbf{c}$, and size-reduction does not change $\mathbf{d}_1$. □

Theorem 5.4. *Algo 7 outputs a module–HKZ-reduced pseudo-basis of M (in the sense of Definition 4.1).*

Proof. By Theorem 5.2 we obtain a quasi–module–HKZ pseudo-basis after Parts (I)–(II). By Lemma 5.3, Part (III) installs $\mathbf{d}_1$ with $\|\mathbf{d}_1\| = \lambda_1(M)$. Part (IV) then recursively module–HKZ-reduces in $\langle \mathbf{d}_1 \rangle_R^{\perp}$ and lifts back with size-reduction, matching Definition 4.1. □

5.5 Complexity

We now bound the running time of Algo 7. The algorithm has two conceptual phases: first it enforces the quasi–module–HKZ conditions (Parts (I)–(II)), namely it keeps the first $n-1$ blocks module–HKZ-reduced while enforcing the two-block balance on the tail blocks $(n-1, n)$; and then it installs a true shortest vector by Euclidean enumeration in dimension nd (Part (III)), followed by one more recursive reduction in $\mathbf{d}_1^{\perp}$ (Part (IV)).

For the complexity accounting, the quasi–module–HKZ phase contributes an outer-loop overhead consisting of multiple calls to the rank-2 routine Algo 6, say

Algorithm 7: Module–HKZ reduction via enumeration

Input: A size-reduced pseudo-basis $((I_i, \mathbf{b}_i))_{1\le i\le n}$ of a rank-n module $M \subset K^n$ with $n \ge 3$, together with $\mathbb{Z}$-bases of the ideals I_i

Output: A module–HKZ-reduced pseudo-basis of M

(0) Recursive module–HKZ on the first $n-1$ blocks

Let $M_{<n} := \sum_{i=1}^{n-1} I_i \mathbf{b}_i \subseteq M$.

Recursively apply Algo 7 to the truncated pseudo-basis $\big((I_i, \mathbf{b}_i)\big)_{1\le i\le n-1}$ (viewed as a pseudo-basis of $M_{<n}$).

Lift the output back (keeping $(I_n, \mathbf{b}_n)$ unchanged), size-reduce the whole pseudo-basis, and recompute the module Gram–Schmidt data.

while $\mathcal{N}(I_n \cdot \mathbf{b}_n^*)^{1/d} < \frac{1}{3\Delta_K^{1/d}} \mathcal{N}(I_{n-1} \cdot \mathbf{b}_{n-1}^*)^{1/d}$ **do**

  (I) Rank-2 tail-correction on blocks $(n-1, n)$

  Let $M_{<n-2} := \sum_{i=1}^{n-2} I_i \mathbf{b}_i$ and let π denote the (module) orthogonal projection onto $\langle M_{<n-2} \rangle_R^{\perp}$.

  Apply Algo 6 to the projected rank-2 submodule generated by $((I_{n-1}, \pi(\mathbf{b}_{n-1})), (I_n, \pi(\mathbf{b}_n)))$, obtaining $((J_{n-1}, \mathbf{c}'_{n-1}), (J_n, \mathbf{c}'_n))$.

  Lift the projected vectors back to M by setting $\mathbf{c}_{n-1}$ and $\mathbf{c}_n$ so that $\pi(\mathbf{c}_{n-1}) = \mathbf{c}'_{n-1}$ and $\pi(\mathbf{c}_n) = \mathbf{c}'_n$.

  Replace $(I_{n-1}, \mathbf{b}_{n-1}), (I_n, \mathbf{b}_n)$ by $(J_{n-1}, \mathbf{c}_{n-1}), (J_n, \mathbf{c}_n)$.

  Size-reduce the whole pseudo-basis (in particular enforce $\lfloor \mu_{n,n-1} \rceil_R = 0$) and recompute the module Gram–Schmidt data.

  (II) Re-run module–HKZ on the first $n-1$ blocks.

  Recursively apply Algo 7 to the truncated pseudo-basis $\big((I_i, \mathbf{b}_i)\big)_{1\le i\le n-1}$ (viewed as a pseudo-basis of $M_{<n}$).

  Lift the output back (keeping $(I_n, \mathbf{b}_n)$ unchanged), size-reduce the whole pseudo-basis, and recompute the module Gram–Schmidt data.

end while

(III) Enumeration step to install a shortest vector of M

Convert the current pseudo-basis into a $\mathbb{Z}$-basis $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_{nd})$ using Algo 5.

Run Euclidean enumeration (Algo 3) on $\mathbf{W}$ with squared bound $A = \|\mathbf{b}_1\|^2$ to obtain a shortest nonzero vector $\mathbf{c} \in M$.

Set $\mathbf{d}_1 := \mathbf{c}$, complete to a pseudo-basis using $n-1$ independent vectors among the current $\{\mathbf{b}_i\}_{i\ge 2}$, and apply the pseudo-basis update routine (e.g., Lemma 2.9) to obtain a size-reduced pseudo-basis $((I'_i, \mathbf{d}_i))_{1\le i\le n}$ with $\|\mathbf{d}_1\| = \|\mathbf{c}\|$.

(IV) Recursive HKZ in $\mathbf{d}_1^{\perp}$

For each $i \ge 2$, set $\mathbf{d}'_i := \mathbf{d}_i - \mu_{i,1}\mathbf{d}_1$ so that $\mathbf{d}'_i \perp \mathbf{d}_1$.

Recursively module–HKZ-reduce $((I'_i, \mathbf{d}'_i))_{2\le i\le n}$ in rank $n-1$, lift back, and size-reduce as usual.

return $((I'_i, \mathbf{d}'_i))_{1\le i\le n}$

N_{iter} calls, whose dominant cost is Euclidean enumeration/SVP in dimension $2d$. The recursive call in Part (II) (re-installing module–HKZ on the first $n-1$ blocks) is absorbed into the overall recursion. Thus it suffices to bound N_{iter} and the enumeration cost in dimension nd, which we do next. Note that all remaining operations –basis conversion, module Gram–Schmidt, pseudo-basis updates, and size-reduction– contribute only polynomial overhead in the input size.

Iteration bound for the rank-2 loop. We bound the number of iterations N_{iter} of the loop in Part (I) by combining (i) a uniform *progress* guarantee whenever the tail balance test fails and (ii) a positive *lower bound* on the algebraic size of the $(n-1)$-st block. More precisely, if the balance test fails, a call to the rank-2 module–HKZ subroutine on the submodule generated by $(I_{n-1}, \mathbf{b}_{n-1})$ and $(I_n, \mathbf{b}_n)$ produces an updated $(n-1)$-st block whose algebraic size $\mathcal{N}(I_{n-1} \cdot \mathbf{b}_{n-1}^*)^{1/d}$ decreases by a constant factor, which will be shown in Lemma 5.5. On the other hand, this quantity cannot decrease indefinitely: Lemma 5.6 shows that it always bounded below by the "algebraic shortest" invariant $\lambda_1^I(M)$, defined by

$$\lambda_1^I(M) \;:=\; \inf\Big\{\mathcal{N}(I \cdot \mathbf{v}) : \mathbf{v} \in K^n,\; I \subseteq \mathcal{O}_K \text{ an ideal},\; I\mathbf{v} \subseteq M,\; I\mathbf{v} \neq 0\Big\}.$$

The next lemma relates $\lambda_1^I(M)$ to the Euclidean shortest vector length $\lambda_1(M)$, which yields an explicit logarithmic bound on the number of possible strict decreases, and hence on the number of loop iterations.

Lemma 5.5. *Let $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ be a size-reduced pseudo-basis of M. Let a pseudo basis $((J_{n-1}, \mathbf{c}'_{n-1}), (J_n, \mathbf{c}'_n))$ be the output of Algo 6 on the rank-2 submodule generated by $((I_{n-1}, \pi(\mathbf{b}_{n-1})), (I_n, \pi(\mathbf{b}_n)))$. If*

$$\mathcal{N}(I_n \cdot \mathbf{b}_n^*)^{1/d} \;<\; \frac{1}{3\Delta_K^{1/d}} \mathcal{N}(I_{n-1} \cdot \mathbf{b}_{n-1}^*)^{1/d},$$

then

$$\mathcal{N}(J_{n-1} \cdot \mathbf{c}_{n-1}^*)^{1/d} \le \sqrt{\frac{2}{3}} \mathcal{N}(I_{n-1} \cdot \mathbf{b}_{n-1}^*)^{1/d}.$$

Proof. Let M_2 be the rank-2 submodule generated by $((I_{n-1}, \pi(\mathbf{b}_{n-1})), (I_n, \pi(\mathbf{b}_n)))$. By Algo 6, $\mathbf{c}'_{n-1}$ is a shortest nonzero vector of the Euclidean lattice $\sigma(M_2)$, hence $\|\mathbf{c}'_{n-1}\| = \lambda_1(\sigma(M_2))$. Minkowski's theorem in dimension $2d$ gives

$$\|\mathbf{c}'_{n-1}\| \le \sqrt{2d}\, \det(\sigma(M_2))^{1/(2d)} = \sqrt{2d}\Delta_K^{1/(4d)} \Big(\mathcal{N}(I_{n-1} \cdot \mathbf{b}_{n-1}^*) \mathcal{N}(I_n \cdot \mathbf{b}_n^*)\Big)^{1/(2d)}.$$

On the other hand, by AM–GM over the d embeddings we have

$$\|\mathbf{c}'_{n-1}\|^2 = \sum_{\ell=1}^{d} \|\sigma_\ell(\mathbf{c}'_{n-1})\|^2 \ge d\Big(\prod_{\ell=1}^{d} \|\sigma_\ell(\mathbf{c}'_{n-1})\|^2\Big)^{1/d},$$

hence

$$\|\mathbf{c}'_{n-1}\| \ge \sqrt{d}\mathcal{N}(\mathcal{O}_K \cdot \mathbf{c}'_{n-1})^{1/d} \ge \sqrt{d}\mathcal{N}(J_{n-1} \cdot \mathbf{c}_{n-1}^*)^{1/d},$$

where the last inequality uses $\mathcal{N}(J_{n-1}) \le 1$ by the Lemma 2.5, and also $\mathbf{c}^*_{n-1} = \mathbf{c}'_{n-1}$ by definition.

Combining,

$$\mathcal{N}(J_{n-1} \cdot \mathbf{c}^*_{n-1})^{1/d} \le \sqrt{2}\Delta_K^{1/(2d)} \left(\mathcal{N}(I_{n-1} \cdot \mathbf{b}^*_{n-1})\mathcal{N}(I_n \cdot \mathbf{b}^*_n)\right)^{1/(2d)}.$$

Using the imbalance hypothesis $\mathcal{N}(I_n \cdot \mathbf{b}^*_n)^{1/d} < \frac{1}{3\Delta_K^{1/d}}\mathcal{N}(I_{n-1} \cdot \mathbf{b}^*_{n-1})^{1/d}$ in the right-hand side gives $\mathcal{N}(J_{n-1} \cdot \mathbf{c}^*_{n-1})^{1/d} \le \sqrt{2/3}\mathcal{N}(I_{n-1} \cdot \mathbf{b}^*_{n-1})^{1/d}$. □

Lemma 5.6. *For any rank-n module M,*

$$d^{-d/2}\lambda_1(M)^d \Delta_K^{-1/2} \le \lambda_1^I(M) \le \inf_{0 \ne \mathbf{v} \in M} \mathcal{N}(\mathcal{O}_K \cdot \mathbf{v}).$$

Proof. The upper bound is immediate by restricting to $I = \mathcal{O}_K$ and $\mathbf{v} \in M \setminus \{0\}$. For the lower bound, fix any admissible pair $(I, \mathbf{v})$ with $I \cdot \mathbf{v} \subseteq M$ and $I \cdot \mathbf{v} \ne 0$. Consider the Euclidean lattice $\sigma(I \cdot \mathbf{v}) \subset \mathbb{R}^d$. By Minkowski,

$$\lambda_1(\sigma(I\mathbf{v})) \le \sqrt{d}\Delta_K^{1/(2d)}\mathcal{N}(I \cdot \mathbf{v})^{1/d}.$$

Since $I \cdot \mathbf{v} \subseteq M$, we have $\lambda_1(M) \le \lambda_1(\sigma(I \cdot \mathbf{v}))$, hence

$$\mathcal{N}(I \cdot \mathbf{v}) \ge d^{-d/2}\lambda_1(M)^d \Delta_K^{-1/2}.$$

Taking the infimum over all admissible $(I, \mathbf{v})$ yields the claim. □

Lemma 5.7 (Iteration bound)**.** *Let $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ be the input to Algo 7. The number of iterations of the rank-2 loop in Part (I) is bounded by*

$$N_{\text{iter}} \le \frac{1}{\log \sqrt{3/2}} O\left(\log\left(\frac{\left(\sqrt{(n-1)d}\right)^{(n-1)d} \cdot \Delta_K^{(n-1)/2} \cdot \prod_{i=1}^{n-1} \mathcal{N}(I_i \cdot \mathbf{b}^*_i)}{\lambda_1(M)^{(n-1)d}}\right)\right)$$

and in particular is polynomial in $\log \Delta_K$ and in the input bit-length.

Proof. Each time the balance test in Part (I) fails, Lemma 5.5 yields a strict decrease

$$\mathcal{N}(I_{n-1} \cdot \mathbf{b}^*_{n-1})^{1/d} \;\mapsto\; \mathcal{N}(J_{n-1} \cdot \mathbf{c}^*_{n-1})^{1/d}$$

by a factor of at least $\sqrt{2/3}$. Since the underlying rank-2 submodule is unchanged, the corresponding blockwise volume is invariant (up to the fixed field factor). Hence the update also implies a strict increase

$$\mathcal{N}(I_n \cdot \mathbf{b}^*_n)^{1/d} \;\mapsto\; \mathcal{N}(J_n \cdot \mathbf{c}^*_n)^{1/d}$$

by a factor of at least $\sqrt{3/2}$.

Let $M_{<n} := \sum_{i=1}^{n-1} I_i \mathbf{b}_i$ be the submodule generated by the first $n-1$ blocks, and write $\det(M_{<n})$ for the determinant of its embedded $\mathbb{Z}$-lattice. By the blockwise volume identity for the last block, we have

$$\det(M_{<n}) = \frac{\det(M)}{\sqrt{\Delta_K}\mathcal{N}(I_n \mathbf{b}^*_n)}.$$

Since $\det(M)$ and Δ_K are fixed, the above increase of $\mathcal{N}(I_n\mathbf{b}_n^*)^{1/d}$ implies that $\det(M_{<n})$ decreases by a factor of at least $\sqrt{2/3}$ in each iteration.

On the other hand, Minkowski's theorem in dimension $(n-1)d$ gives

$$\lambda_1(M_{<n})^{(n-1)d} \le \left(\sqrt{(n-1)d}\right)^{(n-1)d} \cdot \det(M_{<n}).$$

Moreover, since $M_{<n} \subseteq M$, we have $\lambda_1(M_{<n}) \ge \lambda_1(M)$, and therefore

$$\left(\sqrt{(n-1)d}\right)^{(n-1)d} \cdot \det(M_{<n}) \ge \lambda_1(M)^{(n-1)d}.$$

Thus the quantity $\left(\sqrt{(n-1)d}\right)^{(n-1)d} \cdot \det(M_{<n})$ admits a fixed positive lower bound, yet decreases by a factor of at least $\sqrt{2/3}$ at each iteration.

Finally, using the blockwise volume identity for the first $n-1$ blocks, the initial value satisfies

$$\det(M_{<n}) \le \Delta_K^{(n-1)/2} \cdot \prod_{i=1}^{n-1} \mathcal{N}(I_i \cdot \mathbf{b}_i^*).$$

Consequently, only

$$O\left(\log\left(\frac{\left(\sqrt{(n-1)d}\right)^{(n-1)d} \cdot \Delta_K^{(n-1)/2} \cdot \prod_{i=1}^{n-1} \mathcal{N}(I_i \cdot \mathbf{b}_i^*)}{\lambda_1(M)^{(n-1)d}}\right)\right)$$

such strict decreases are possible, which yields the stated bound on N_{iter}.

□

Running-time recurrence and its solution. Up to polynomial overhead, Algo 7 performs: (i) one initial recursive call in rank $n-1$ (Part (0)); (ii) an outer loop (Parts (I)–(II)) executed N_{iter} times, where each iteration consists of one rank-2 reduction (cost $T_{\mathsf{HKZ}}(2d)$) and one recursive call in rank $n-1$; (iii) one enumeration call in dimension nd (Part (III)); and (iv) one final recursive call in rank $n-1$ (Part (IV)). Thus, writing $T_{\mathsf{MHKZ}}(n)$ by the running time of Algo 7 on rank-n modules,

$$\begin{aligned} T_{\mathsf{MHKZ}}(n) \le & \left(1 + N_{\mathrm{iter}}\right) \cdot T_{\mathsf{MHKZ}}(n-1) + N_{\mathrm{iter}} \cdot T_{\mathsf{HKZ}}(2d) \\ & + \mathsf{poly}(n, d, B, \log \Delta_K) \cdot \max_{1 \le i \le nd} N_i, \end{aligned} \tag{24}$$

where N_i is the standard enumeration node bound for the call in Part (III).

Theorem 5.8. *Let $n \ge 3$. Given an input pseudo-basis $\left((I_i, \mathbf{b}_i)\right)_{1 \le i \le n}$ of a rank-n module M, Algo 7 outputs a module–HKZ-reduced pseudo-basis and runs in time*

$$\mathsf{poly}(n, d, B, \log \Delta_K) \cdot \left(T_{\mathsf{HKZ}}(2d) + \max_{1 \le i \le nd} N_i\right).$$

In particular, the enumeration step in Part (III) satisfies

$$\max_{1 \le i \le nd} N_i \le \mathsf{poly}(n, d, B, \log \Delta_K) \cdot \Delta_K \cdot n^{\frac{nd}{2e} + o(nd)}.$$

Proof. Correctness is proved in Section 5.4. The recurrence (24) follows from the above cost accounting. By Lemma 5.7, $N_{\text{iter}} = \mathsf{poly}(n, d, B, \log \Delta_K)$, so unrolling (24) over recursion depth at most n yields

$$T_{\mathsf{MHKZ}}(n) \le \mathsf{poly}(n, d, B, \log \Delta_K) \cdot \Big(T_{\mathsf{HKZ}}(2d) + \max_{1 \le i \le nd} N_i \Big).$$

It remains to bound $\max_i N_i$ for the enumeration in Part (III). Let $m := nd$ and let $W = (\mathbf{w}_1, \dots, \mathbf{w}_m)$ be the converted $\mathbb{Z}$-basis with Gram–Schmidt vectors $(\mathbf{w}_1^*, \dots, \mathbf{w}_m^*)$. We use the squared radius $A = \|\mathbf{b}_1\|^2$ in Algo 3; in particular, $A = \|\mathbf{b}_1\|^2 = \|\mathbf{w}_1\|^2$. From (6) we obtain, up to polynomial factors,

$$N_i \le \mathsf{poly}(n, d, B, \log \Delta_K) \cdot 2^{O(m)} \cdot \frac{\|\mathbf{w}_1\|^i}{(\sqrt{m})^i \cdot \prod_{j=m-i+1}^{m} \|\mathbf{w}_j^*\|}. \tag{25}$$

Set

$$R_i \;:=\; \frac{\|\mathbf{w}_1\|^i}{\prod_{j=m-i+1}^{m} \|\mathbf{w}_j^*\|}.$$

By (25), it suffices to upper bound R_i.

Step 1: split off the last block. For $i > d$, split the tail product at the last block:

$$R_i = \underbrace{\frac{\|\mathbf{w}_1\|^{i-d}}{\prod_{j=m-i+1}^{m-d} \|\mathbf{w}_j^*\|}}_{=:R^{(1)}} \cdot \underbrace{\frac{\|\mathbf{w}_1\|^{d}}{\prod_{j=m-d+1}^{m} \|\mathbf{w}_j^*\|}}_{=:R^{(2)}}. \tag{26}$$

(If $i \le d$, then $R_i = R^{(2)}$ and the factor $R^{(1)}$ is vacuous.)

Step 2: bound $R^{(1)}$ using module–HKZ on the first $n-1$ blocks. After Parts (I)–(II), the truncated pseudo-basis $\big((I_t, \mathbf{b}_t)\big)_{1 \le t \le n-1}$ is module–HKZ-reduced (by Definition 5.1(2)), hence the corresponding block profile of Gram–Schmidt lengths in the first $n-1$ blocks satisfies the HKZ tail-product bounds of Theorem 4.3. Therefore,

$$R^{(1)} \le (\sqrt{m})^{(k-1)d} \cdot \sqrt{n}^{\,i \ln(nd/i) + o(m)} \tag{27}$$

$$\le (\sqrt{m})^{(k-1)d} \cdot \sqrt{n}^{\,\frac{nd}{e} + o(m)}, \tag{28}$$

where the second inequality uses that the function $i \ln(nd/i)$ is maximized at $i = nd/e$, with maximum value nd/e.

Step 3: bound $R^{(2)}$ directly from quasi balance and reduce to the $(n-1)$-st block. By the blockwise volume identity (9) for the last block,

$$\prod_{j=m-d+1}^{m} \|\mathbf{w}_j^*\| = \mathcal{N}(I_n \cdot \mathbf{b}_n^*) \sqrt{\Delta_K}.$$

Hence

$$R^{(2)} = \frac{\|\mathbf{w}_1\|^d}{\mathcal{N}(I_n \cdot \mathbf{b}_n^*)\sqrt{\Delta_K}}.$$

Using the quasi–module–HKZ tail balance Definition 5.1(1),

$$\mathcal{N}(I_n \cdot \mathbf{b}_n^*)^{1/d} \geq \frac{1}{3\Delta_K^{1/d}} \mathcal{N}(I_{n-1} \cdot \mathbf{b}_{n-1}^*)^{1/d},$$

we obtain

$$\frac{1}{\mathcal{N}(I_n \cdot \mathbf{b}_n^*)\sqrt{\Delta_K}} \leq \frac{3^d \Delta_K}{\mathcal{N}(I_{n-1} \cdot \mathbf{b}_{n-1}^*)\sqrt{\Delta_K}} = \frac{3^d \Delta_K}{\prod_{j=m-2d+1}^{m-d} \|\mathbf{w}_j^*\|},$$

where the last equality uses (9) for the $(n-1)$-st block. Therefore

$$R^{(2)} \leq 3^d \Delta_K \cdot \frac{\|\mathbf{w}_1\|^d}{\prod_{j=m-2d+1}^{m-d} \|\mathbf{w}_j^*\|}. \tag{29}$$

The denominator in (29) lies entirely in the first $n-1$ blocks that is module-HKZ-reduced, hence we could again apply Theorem 4.3 (restricted to the first $n-1$ blocks) to get

$$\begin{aligned} \frac{\|\mathbf{w}_1\|^d}{\prod_{j=m-2d+1}^{m-d} \|\mathbf{w}_j^*\|} &\leq (\sqrt{m})^d \cdot \sqrt{n}^{d \ln n + o(m)} \\ &\leq (\sqrt{m})^d \cdot \sqrt{n}^{o(m)}. \end{aligned} \tag{30}$$

Combining (29) and (30) yields a bound on $R^{(2)}$ of the form

$$R^{(2)} \leq \Delta_K \cdot (\sqrt{m})^d \cdot \sqrt{n}^{o(m)},$$

the factor 3^d is absorbed into the term $\sqrt{n}^{o(m)}$.

Step 4: assemble and conclude. Multiplying the bounds for $R^{(1)}$ and $R^{(2)}$ from (27) gives

$$R_i \leq \Delta_K \cdot (\sqrt{m})^{kd} \cdot \sqrt{n}^{\frac{m}{e} + o(m)}.$$

Plugging this into (25) yields

$$N_i \leq \mathsf{poly}(n, d, B, \log \Delta_K) \cdot \Delta_K \cdot \sqrt{n}^{\frac{m}{e} + o(m)}.$$

Hence

$$N_i \leq \mathsf{poly}(n, d, B, \log \Delta_K) \cdot \Delta_K \cdot n^{\frac{m}{2e} + o(m)} = \mathsf{poly}(n, d, B, \log \Delta_K) \cdot \Delta_K \cdot n^{\frac{nd}{2e} + o(nd)}.$$

Taking the maximum over $1 \leq i \leq m$ completes the proof. □

5.6 Applications

Proposition 5.9. *Let $n \geq 3$ and let $K = \mathbb{Q}[X]/\langle X^d + 1\rangle$ with d a power of two, and let $M \subset K^n$ be a rank-n module. Assume Euclidean HKZ/SVP in dimension $2d$ runs in time $T_{\mathsf{HKZ}}(2d) = d^{\frac{d}{e}+o(d)}$. Then one can find a shortest nonzero vector in M in time*

$$\mathsf{poly}(n, d, B, \log \Delta_K) \cdot \left(d^d \cdot n^{\frac{nd}{2e}+o(nd)} + d^{\frac{d}{e}+o(d)}\right).$$

Proof. For $K = \mathbb{Q}[X]/\langle X^d + 1\rangle$ with d a power of two, $\Delta_K = d^d$. The claim follows directly from Theorem 5.8 together with the assumption $T_{\mathsf{HKZ}}(2d) = d^{\frac{d}{e}+o(d)}$. □

Corollary 5.10. *Let I be an ideal in $\mathbb{Z}[X]/\langle X^t + 1\rangle$ where t is a power of two. Then ideal-SVP on I can be solved in time*

$$\exp\left(\frac{t}{2e}\ln\ln t + O(t)\right)$$

up to polynomial factors.

Proof. The ideal-SVP $\mathbb{Z}[X]/\langle X^t + 1\rangle$, where t is a power of two, can be reduced to a module-SVP in $\left(\mathbb{Z}[X]/\langle X^d + 1\rangle\right)^{t/d}$ for any power-of-two $d \leq t$. Up to polynomial factors, the updated bound on $\max_i N_i$ yields

$$T(d) = d^d \cdot (t/d)^{\frac{t}{2e}} + d^{\frac{d}{e}}$$

due to the result of Proposition 5.9. For $d \leq t$ we have $d^{d/e} \leq d^d(t/d)^{t/(2e)}$, hence it suffices to minimize

$$A(d) := d^d \cdot (t/d)^{\frac{t}{2e}}.$$

Write

$$f(d) := \ln A(d) = d\ln d + \frac{t}{2e}\big(\ln t - \ln d\big).$$

Treat d as a positive real and differentiate:

$$f'(d) = \ln d + 1 - \frac{t}{2e}\cdot\frac{1}{d}.$$

The minimizer satisfies $f'(d) = 0$, i.e.,

$$d(\ln d + 1) = \frac{t}{2e}.$$

Let $y := \ln d + 1$, so $d = e^{y-1}$. The condition becomes

$$ye^y = \frac{t}{2},$$

Equivalently,

$$y = W\left(\frac{t}{2}\right),$$

where $W(\cdot)$ denotes the (principal branch of the) *Lambert W-function*, defined as the inverse of the map $y \mapsto ye^y$ on $\mathbb{R}_{\geq 0}$. Thus,

$$d_\star = \exp\big(W(t/2) - 1\big) = \frac{t}{2eW(t/2)}.$$

Choosing the nearest power-of-two d changes $f(d)$ by $O(t)$, hence changes $A(d)$ by at most a factor $\exp(O(t))$.

Using $\ln d_\star = W(t/2) - 1$ and $d_\star = t/(2eW(t/2))$, we get

$$\begin{aligned} f(d_\star) &= d_\star \ln d_\star + \frac{t}{2e}\big(\ln t - \ln d_\star\big) \\ &= \frac{t}{2e}\Big(1 - \frac{1}{W(t/2)}\Big) + \frac{t}{2e}\Big(\ln t - W(t/2) + 1\Big) \\ &= \frac{t}{2e}\Big(\ln t - W(t/2) + 2 - \frac{1}{W(t/2)}\Big). \end{aligned}$$

Since $W(t/2) = \ln t - \ln\ln t + O(1)$, we have $\ln t - W(t/2) = \ln\ln t + O(1)$, and therefore

$$\min_d \ln T(d) = \min_d \ln A(d) = f(d_\star) = \frac{t}{2e}\ln\ln t + O(t).$$

Exponentiating gives

$$\min_d T(d) = \exp\left(\frac{t}{2e}\ln\ln t + O(t)\right)$$

up to polynomial factors. □

For comparison, recall that Kannan's Euclidean SVP solver runs in time $t^{\frac{t}{2e}+o(t)}$ in dimension t [9], whereas Corollary 5.10 yields $\exp\big(\frac{t}{2e}\ln\ln t + O(t)\big)$ (up to polynomial factors).

Replacing enumeration by sieving yields no gain. Our speedup is inherently tied to *enumeration*: it comes from a refined node-counting argument that exploits the module–HKZ Gram–Schmidt profile of the converted $\mathbb{Z}$-basis. In contrast, in the standard heuristic sieving model the running time depends essentially only on the dimension, namely $T_{\mathsf{sieve}}(m) = 2^{cm+o(m)}$ for some constant $c > 0$. As a result, replacing the enumeration calls in our algorithms by sieving makes the overall cost dominated by a sieve in dimension nd, i.e., $T_{\mathsf{sieve}}(nd) = 2^{cnd+o(nd)}$, which matches applying sieving directly to the ambient lattice $\sigma(M)$. Hence, sieving does not preserve the enumeration-based advantage of the module structure. A detailed discussion of this point is deferred to the Appendix C.

References

1. László Babai. On lovász'lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
2. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.
3. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.
4. Gabrielle De Micheli, Daniele Micciancio, Alice Pellet-Mary, and Nam Tran. Reductions from module lattices to free module lattices, and application to dequantizing module-lll. In *Annual International Cryptology Conference*, pages 836–865. Springer, 2023.
5. Léo Ducas, Lynn Engelberts, and Paola de Perthuis. Predicting module-lattice reduction. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 133–166. Springer, 2025.
6. Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.
7. Claus Fieker and Damien Stehlé. Short bases of lattices over number fields. In *International Algorithmic Number Theory Symposium*, pages 157–173. Springer, 2010.
8. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206, 2008.
9. Guillaume Hanrot and Damien Stehlé. Improved analysis of kannan's shortest lattice vector algorithm. In *Annual international cryptology conference*, pages 170–186. Springer, 2007.
10. Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. *Theory of Computing*, 8(23):513–531, 2012.
11. Bettina Helfrich. Algorithms to construct minkowski reduced and hermite reduced lattice bases. *Theoretical Computer Science*, 41:125–139, 1985.
12. Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 193–206, 1983.
13. Alexander Karenin and Elena Kirshanova. Finding dense submodules with algebraic lattice reduction. In *International Conference on Cryptology in Africa*, pages 403–427. Springer, 2024.
14. Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM (JACM)*, 52(5):789–808, 2005.
15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.
16. Changmin Lee, Alice Pellet-Mary, Damien Stehlé, and Alexandre Wallet. An lll algorithm for module lattices. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 59–90. Springer, 2019.
17. Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

18. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 1–23. Springer, 2010.
19. Daniele Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM journal on Computing*, 30(6):2008–2035, 2001.
20. Tamalika Mukherjee and Noah Stephens-Davidowitz. Lattice reduction for modules, or how to reduce modulesvp to modulesvp. In *Annual International Cryptology Conference*, pages 213–242. Springer, 2020.
21. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

A Missing Proofs in Section 2

This section presents the proofs deferred from Section 2. We begin with the proof of Lemma 2.4 and then turn to Lemma 2.7.

Proof of Lemma 2.4. Let $P_i := \prod_{j \geq i} v_j$ and $G_i := P_i^{1/(n-i+1)}$ for $1 \leq i \leq n$ (so $P_{n+1} = 1$). Then, we have $v_i \leq \sqrt{n-i+1} G_i$ and $P_{i+1} = \frac{P_i}{v_i}$, which directly implies that

$$P_{i+1} = \frac{P_i}{v_i} \geq \frac{G_i^{n-i+1}}{\sqrt{n-i+1} G_i} = \frac{G_i^{n-i}}{\sqrt{n-i+1}}.$$

Based on the relation, we also get $G_{i+1} \geq G_i (n-i+1)^{-\frac{1}{2(n-i)}}$ from $G_{i+1}^{n-i} = P_{i+1}$. Fix i and put $k := n-i+1$. Since $P_k = G_k^i$ and $v_1 \leq \sqrt{n} G_1$, we conclude

$$\frac{v_1^i}{\prod_{j \geq n-i+1} v_j} = \frac{v_1^i}{P_k} \leq \left(\sqrt{n} \frac{G_1}{G_k} \right)^i.$$

Iterating the previous inequality for G from 1 to $k-1$ gives a relation

$$\frac{G_1}{G_k} \leq \prod_{t=1}^{k-1} (n-t+1)^{\frac{1}{2(n-t)}} = \prod_{m=i}^{n-1} (m+1)^{1/(2m)}.$$

Therefore,

$$\frac{1}{i} \ln \left(\frac{v_1^i}{\prod_{j \geq n-i+1} v_j} \right) \leq \frac{1}{2} \ln n + \frac{1}{2} \sum_{m=i}^{n-1} \frac{\ln(m+1)}{m}.$$

For $i \geq 2$, we use $\ln(m+1) \leq \ln n$ and the harmonic-sum bound $\sum_{m=i}^{n-1} \frac{1}{m} \leq \int_{i-1}^{n-1} \frac{dx}{x} = \ln \frac{n-1}{i-1}$ to obtain

$$\frac{1}{i} \ln \left(\frac{v_1^i}{\prod_{j \geq n-i+1} v_j} \right) \leq \frac{1}{2} \ln n + \frac{1}{2} \ln n \cdot \ln \frac{n-1}{i-1}.$$

Exponentiating yields

$$\frac{v_1^i}{\prod_{j \geq n-i+1} v_j} \leq \sqrt{n}^{\, i + i \ln \frac{n-1}{i-1}} \qquad (i \geq 2).$$

Finally, the coarse bound $i \ln \frac{n-1}{i-1} \le \frac{n}{e} + O(1)$ gives the second displayed inequality. For $i = 1$, use $\sum_{m=1}^{n-1} \frac{1}{m} \le 1 + \ln(n-1)$ to argue similarly. □

We now prove Lemma 2.7.

Proof of Lemma 2.7. Let $s \in M \setminus \{0\}$ be an element of minimal Euclidean norm. By the arithmetic-geometric mean inequality, we have $N(s) \le d^{-d/2}\|s\|^d$. Applying Minkowski's theorem to the canonical embedded lattice of M, it follows that

$$\|s\| \le \sqrt{nd} \cdot (\det M)^{1/(nd)}.$$

Using the relation $\det M = \Delta_K^{n/2} N(\det_{K_\mathbb{R}} M)$, we derive the last two inequalities.

Next, for any $s \in M \setminus \{0\}$, applying Minkowski's theorem to the lattice $R \cdot s$ gives

$$\lambda_1(M) \le \lambda_1(R \cdot s) \le \sqrt{d} \cdot \Delta_K^{1/2d} \cdot \mathcal{N}(s)^{1/d}.$$

Then it guarantees the first inequality by taking the minimum for $s \in M$. □

B Missing Proofs in Section 4

Lemma B.1 (Existence and uniqueness of the extremal profile)**.** *There exists a* unique *vector* $V = (v_1, \dots, v_{nd}) \in \mathbb{R}_{>0}^{nd}$ *satisfying the equalities* (b*) *and* (c*) *together with* (a*).

Proof. We proceed by backward reconstruction over the blocks. Let $m = nd$ and $t_i := (i-1)d + 1$ for $1 \le i \le n$ for a simple description.

Step 1 (one-block reconstruction from the last entry). Fix a block index $i \in \{1, \dots, n\}$ and denote $x := v_{id}$. The relations (c*) determine uniquely the entries $v_{id-1}, \dots, v_{t_i+1}$ as positive functions of x. For $\ell = 1, \dots, d-1$, applying (c*) with $k = d - \ell$ gives

$$v_{id-\ell} = \sqrt{\ell+1} \cdot \left(\prod_{j=id-\ell}^{id} v_j \right)^{1/(\ell+1)}.$$

Let $P_\ell := \prod_{j=id-\ell}^{id} v_j$ (with $P_0 := v_{id} = x$). Then $P_\ell = v_{id-\ell} P_{\ell-1}$, hence

$$v_{id-\ell}^{\ell} = (\ell+1)^{(\ell+1)/2} \cdot P_{\ell-1}.$$

Thus $v_{id-\ell}$ is uniquely determined from $P_{\ell-1}$, and inductively from x. Consequently, there exists an explicit constant $C'_d > 0$ depending only on d such that

$$\prod_{j=t_i+1}^{id} v_j = C'_d \cdot v_{id}^{d-1}. \tag{31}$$

Note that the relations (c*) are identical in every block, hence C'_d depends only on d (and not on i).

Step 2 (determine one block from the tail). Assume blocks $i+1, \dots, n$ have already been determined, and set

$$P_{i+1} := \prod_{j=t_{i+1}}^{m} v_j, \qquad \text{with } P_{n+1} := 1.$$

We determine block i.

(i) Fix v_{t_i} from (b*) *and* (a*). Using $\prod_{j=t_i}^{m} v_j = \left(\prod_{j=t_i}^{id} v_j\right) \cdot P_{i+1}$, the equality (b*) yields

$$v_{t_i} = \sqrt{(n-i+1)d} \cdot \left(\left(\mathcal{N}(I_i \cdot \mathbf{b}_i^*)\sqrt{\Delta_K}\right) \cdot P_{i+1} \right)^{1/((n-i+1)d)}. \tag{32}$$

Hence v_{t_i} is uniquely determined and positive.

(ii) Fix v_{id} from (a*) *and* (31). By (31),

$$\prod_{j=t_i}^{id} v_j = v_{t_i} \cdot \prod_{j=t_i+1}^{id} v_j = v_{t_i} \cdot C'_d \cdot v_{id}^{d-1}.$$

By (a*) we have

$$\prod_{j=t_i}^{id} v_j = \mathcal{N}(I_i \cdot \mathbf{b}_i^*)\sqrt{\Delta_K}.$$

Combined with (31) and the already fixed value of v_{t_i} from (32), this uniquely determines $v_{id} > 0$. Then Step 2 uniquely determines $v_{t_i+1}, \dots, v_{id-1}$.

Step 3 (backward induction). Starting from $i = n$ and iterating backward for $i = n-1, \dots, 1$ yields a unique $V = (v_1, \dots, v_m) \in \mathbb{R}^m_{>0}$ satisfying (a*), (b*), and (c*). □

Lemma B.2. *Let $m = nd$, and let $\alpha_j := \log \|\mathbf{w}_j^*\|$, $\beta_j := \log v_j$. Then for every $1 \le i \le m$,*

$$\frac{\|\mathbf{w}_1^*\|^i}{\prod_{j=m-i+1}^{m} \|\mathbf{w}_j^*\|} \le \frac{v_1^i}{\prod_{j=m-i+1}^{m} v_j}. \tag{33}$$

Proof. Write $t_i := (i-1)d + 1$, $L_i := (n-i+1)d$, and $L'_k := d - k + 1$. Taking logs in (b) and (b*) gives, for each block start t_i,

$$\alpha_{t_i} \le c_i + \frac{1}{L_i} \sum_{j=t_i}^{m} \alpha_j, \qquad \beta_{t_i} = c_i + \frac{1}{L_i} \sum_{j=t_i}^{m} \beta_j, \qquad c_i := \tfrac{1}{2} \log L_i. \tag{34}$$

Similarly, taking logs in (c) and (c*) gives, for $t = t_i + k - 1$ with $2 \le k \le d$,

$$\alpha_t \le c'_k + \frac{1}{L'_k} \sum_{j=t}^{id} \alpha_j, \qquad \beta_t = c'_k + \frac{1}{L'_k} \sum_{j=t}^{id} \beta_j, \qquad c'_k := \tfrac{1}{2} \log L'_k. \tag{35}$$

We claim that for every $s \in \{1, \dots, m\}$,

$$\sum_{j=1}^{s} \beta_j \geq \sum_{j=1}^{s} \alpha_j. \tag{36}$$

Indeed, define prefix-sum gaps within block i by

$$\Delta_{i,k} := \sum_{r=1}^{t_i+k-1} (\beta_r - \alpha_r), \qquad i = 1, \dots, n,\ k = 1, \dots, d,$$

and set $\Delta_{1,0} := 0$ and $\Delta_{i,0} := \Delta_{i-1,d}$ for $i \geq 2$. We prove by double induction on (i, k) that

$$\Delta_{i,k} \geq 0 \qquad \text{for all } i, k. \tag{37}$$

This implies (36) since every s equals $t_i + k - 1$ for a unique (i, k).

Auxiliary identity (blockwise tail vs. prefix) By (a) and (a*), for each block i we have the *blockwise equality of sums*

$$\sum_{j=t_i}^{id} \alpha_j = \sum_{j=t_i}^{id} \beta_j,$$

hence for any $t = t_i + k - 1 \in [t_i, id]$,

$$\sum_{j=t}^{id} (\beta_j - \alpha_j) = -\sum_{j=t_i}^{t-1} (\beta_j - \alpha_j) = -(\Delta_{i,k-1} - \Delta_{i,0}). \tag{38}$$

Outer base $i = 1$, inner base $k = 1$ At $t_1 = 1$, subtracting the two relations in (34) gives

$$\beta_1 - \alpha_1 \geq \frac{1}{L_1} \Big(\sum_{j=1}^{m} \beta_j - \sum_{j=1}^{m} \alpha_j \Big) = 0,$$

where the last equality follows by summing the blockwise equalities $\sum_{j=t_i}^{id} \alpha_j = \sum_{j=t_i}^{id} \beta_j$ over $i = 1, \dots, n$. Thus $\Delta_{1,1} \geq 0$.

Outer base $i = 1$, inner step $k \to k + 1$ Assume $\Delta_{1,k-1} \geq 0$ for some $2 \leq k \leq d$ and set $t = t_1 + k - 1$. Subtracting (35) for α from that for β yields

$$\beta_t - \alpha_t \geq \frac{1}{L'_k} \Big(\sum_{j=t}^{d} \beta_j - \sum_{j=t}^{d} \alpha_j \Big) = -\frac{1}{L'_k} \Delta_{1,k-1},$$

using (38) with $\Delta_{1,0} = 0$. Hence

$$\Delta_{1,k} = \Delta_{1,k-1} + (\beta_t - \alpha_t) \geq \Big(1 - \frac{1}{L'_k} \Big) \Delta_{1,k-1} \geq 0.$$

Outer step $i \to i+1$, inner base $k = 1$ Assume $\Delta_{i,k} \geq 0$ for all k (so $\Delta_{i,d} \geq 0$). At t_{i+1}, subtracting the two relations in (34) gives

$$\beta_{t_{i+1}} - \alpha_{t_{i+1}} \geq \frac{1}{L_{i+1}} \Big(\sum_{j=t_{i+1}}^{m} \beta_j - \sum_{j=t_{i+1}}^{m} \alpha_j \Big) = 0,$$

where the last equality holds by summing the blockwise equalities over blocks $i+1, \ldots, n$. Therefore $\Delta_{i+1,1} = \Delta_{i,d} + (\beta_{t_{i+1}} - \alpha_{t_{i+1}}) \geq 0$.

Outer step $i \to i+1$, inner step $k \to k+1$ Assume $\Delta_{i+1,k-1} \geq 0$ and set $t = t_{i+1} + k - 1$. Subtracting (35) gives

$$\beta_t - \alpha_t \geq \frac{1}{L'_k} \Big(\sum_{j=t}^{(i+1)d} \beta_j - \sum_{j=t}^{(i+1)d} \alpha_j \Big) = -\frac{1}{L'_k} (\Delta_{i+1,k-1} - \Delta_{i+1,0}),$$

using (38). Since $\Delta_{i+1,0} = \Delta_{i,d} \geq 0$, we obtain

$$\Delta_{i+1,k} = \Delta_{i+1,k-1} + (\beta_t - \alpha_t) \geq \Big(1 - \frac{1}{L'_k}\Big) \Delta_{i+1,k-1} + \frac{1}{L'_k} \Delta_{i,d} \geq 0.$$

This completes the double induction and proves (37), hence (36).

Finally, (33) is equivalent to $i(\beta_1 - \alpha_1) + \sum_{j=1}^{m-i} (\beta_j - \alpha_j) \geq 0$, which follows immediately from (36) together with the base case $\beta_1 - \alpha_1 \geq 0$ proved above. □

C Replacing Enumeration by Sieving

We discuss how the cost bound in Section 5.5 changes if we replace the enumeration (Algo 3) by a heuristic lattice sieving algorithm.

Heuristic sieving model. Assume that there exists an SVP sieve which, given an m-dimensional Euclidean lattice basis, outputs a shortest nonzero vector in expected time

$$T_{\mathsf{sieve}}(m) = 2^{c \cdot m + o(m)},$$

for some constant $c > 0$ (depending on the chosen sieve variant). As before, we keep polynomial factors in $(n, d, B, \log \Delta_K)$ implicit.

Algorithmic replacement. We modify Algo 6 and Algo 7 by replacing every call to Euclidean enumeration (Algo 3) by the above sieve:

- In Algo 6, the two SVP calls in dimension $2d$ (for $\mathbf{c}_1$ and for $\mathbf{c}'_2$) become sieving calls, costing $T_{\mathsf{sieve}}(2d)$ each (up to polynomial overhead).
- In Part (III) of Algo 7, the SVP call on the converted $\mathbb{Z}$-basis $\mathbf{W}$ in dimension nd becomes a sieving call, costing $T_{\mathsf{sieve}}(nd)$.

Resulting running time. Substituting sieving into the complexity bound of Theorem 5.8 yields the following estimate. The contribution of the rank-2 subroutine (called within the outer loop) is obtained by replacing the Euclidean

HKZ/SVP cost $T_{\mathsf{HKZ}}(2d)$ by the sieving cost $T_{\mathsf{sieve}}(2d)$, and the contribution of the SVP call in Part (III) is obtained by replacing the enumeration node bound $\max_i N_i$ by $T_{\mathsf{sieve}}(nd)$. Consequently,

$$T^{\mathsf{sieve}}_{\mathsf{MHKZ}}(n) \le \mathsf{poly}(n, d, B, \log \Delta_K) \cdot \Big(T_{\mathsf{sieve}}(2d) + T_{\mathsf{sieve}}(nd)\Big). \tag{39}$$

In particular, for $n \ge 3$ we have $nd \ge 2d$, and hence

$$T^{\mathsf{sieve}}_{\mathsf{MHKZ}}(n) \;=\; \mathsf{poly}(n, d, B, \log \Delta_K) \cdot 2^{c \cdot nd + o(nd)}.$$

Remark 2. The speedup $n^{\frac{nd}{2e}+o(nd)}$ in Theorem 5.8 is specific to enumeration: it comes from a refined node-counting argument that exploits the module–HKZ Gram–Schmidt profile of the converted $\mathbb{Z}$-basis. In contrast, in the standard sieving model the running time $T_{\mathsf{sieve}}(m)$ is essentially a function of the dimension m and is largely insensitive to such profile information. Hence, after replacing enumeration by sieving, the leading term is governed by a sieve in dimension nd, giving

$$T^{\mathsf{sieve}}_{\mathsf{MHKZ}}(n) = 2^{c \cdot nd + o(nd)}.$$

In particular, this matches the cost of applying sieving directly to the ambient $\mathbb{Z}$-lattice $\sigma(M)$ in dimension nd; i.e., using sieving does not preserve the enumeration-based advantage of the module–HKZ structure.