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. This paper presents a systematic study of module lattices. We extend the lattice enumeration algorithm from Euclidean lattices to module lattices, providing a generalized framework. To incorporate the refined analysis by Hanrot and Stehlè (CRYPTO'07), we adapt key definitions from Euclidean lattices, such as HKZ-reduced bases and quasi-HKZ-reduced bases, adapting them to the pseudo-basis of modules.
Furthermore, we revisit the lattice profile, a crucial aspect of enumeration algorithm analysis, and extend its analysis to module lattices. As a result, we improve the asymptotic performance of the module lattice enumeration algorithm and module-SVP.
For instance, let $K = \mathbb{Q}[x]/\langle x^d+1\rangle$ be a number field with a power-of-two integer d, and suppose that $n \ln n = o(\ln d)$. Then, the nonzero shortest vector in $M \subset K^n$ can be found in time $d^{\frac{d}{2e}+o(d)}$, improving upon the previous lattice enumeration bound of $(nd)^{\frac{nd}{2e}+o(nd)}$.
Our algorithm naturally extends to solving ideal-SVP. Given an ideal $I \subset R$, where $R = \mathbb{Z}[x]/\langle x^t + 1\rangle$ with a power-of-two integer $t = nd$, we can find the nonzero shortest element of I in time $\exp(O(\frac{t}{2e} \ln \ln t))$, improving upon the previous enumeration bound of $\exp(O(\frac{t}{2e} \ln t))$.
Keywords: Lattice Enumeration, Module Lattice, Module-SVP

1 Introduction

The Learning with Errors (LWE) problem [20] has emerged as one of the foundational problems in cryptography, particularly in the context of post-quantum cryptography. Its significance stems from its post-quantum hardness and its versatility in cryptographic applications [3, 9]. The hardness of LWE is closely tied to the Shortest Vector Problem (SVP) in lattices, a problem that is well-known to be NP-hard (under randomized reductions) with some approximate factors [11,13,18]. This connection provides a robust theoretical foundation for designing cryptographic schemes resistant to quantum attacks.

Recent advancements have focused on algebraic lattice problems, such as Ring-LWE [17] and Module-LWE [14], which are widely regarded as strong candidates for post-quantum cryptographic primitives. These variants leverage algebraic structures to achieve improved efficiency while maintaining their security

guarantees. Notably, cryptographic primitives like Kyber [2] and Dilithium [6], which are based on the hardness of the Module-LWE problem, have gained global recognition. Kyber (a key encapsulation mechanism) and Dilithium (a digital signature scheme), selected as part of the winners in the NIST Post-Quantum Cryptography standardization process, underscore their importance as practical and efficient solutions for securing digital systems against quantum adversaries.

Despite the increasing interest in module lattices, a notable gap remains in the development of efficient algorithms that fully exploit their algebraic structure. While some algorithms [5, 15, 19] have been proposed, they often fail to outperform their counterparts designed for traditional Euclidean lattices.

For instance, the classical Lenstra–Lenstra–Lovász (LLL) [16] lattice basis reduction algorithm has been extended to module lattices, leading to algorithms such as module-LLL [15]. Although module-LLL incorporates additional algebraic constraints, its practical performance improvements are often limited compared to its application in Euclidean lattices. This underscores the challenges of effectively using module structures in algorithmic design. Similarly, enumeration techniques tailored for module lattices have yet to achieve their theoretical potential, particularly in comparison to their success in Euclidean lattices.

This leads to a fundamental question:

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

Addressing this question is essential for advancing the practical applicability of Module-LWE based cryptography.

1.1 This work

We provide a partial answer to this question. Specifically, we present an improved analysis of the lattice enumeration algorithm [10, 12] for module lattices by leveraging module structures. This analysis leads to the following theorem.

Theorem 1.1 (Informal)**.** *Let K be a number field of extension degree d with its discriminant Δ_K. Given a pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ of a rank-n module $M \subset K^m$, one can find the nonzero shortest vector in M in time*

$$n^{\frac{nd}{2e}} \cdot d^{\frac{d}{2e}+o(nd)}$$

up to a polynomial factor.

As direct applications, we introduce two results. The detailed discussion of applications is given in Section 4.1.

Application 1.2. Let $n \ln n = o(\ln d)$ and $R = \mathbb{Z}[x]/\langle x^d + 1\rangle$, where d is a power of two and $M \subset K^n$, where $K = \mathbb{Q}[x]/\langle x^d + 1\rangle$. Then, one can find the nonzero shortest vector $\mathbf{v} \in M$ in time $d^{\frac{d}{2e}+o(d)}$ up to a polynomial factor.

Previously, it terminates in time $(nd)^{\frac{nd}{2e}+o(nd)}$ up to a polynomial factor.

Application 1.3. For a power-of-two integer t, we can find the nonzero shortest element of an ideal $I \subset \mathbb{Z}[x]/\langle x^t + 1\rangle$ in asymptotic time $e^{\frac{t}{2e}\ln\ln t}$, improving upon the previous worst-case bound of $e^{\frac{t}{2e}\ln t}$ given in [10].

Technical Overview. Our improvement leverages algebraic structures to enhance the performance of lattice enumeration algorithms [8,10,12] in the context of modules.

To achieve this, we first introduce *module analogs* of definitions from Euclidean lattices. Specifically, we define a module-HKZ-reduced pseudo-basis, which serves as a counterpart to the HKZ-reduced basis in the Euclidean lattice. Furthermore, we introduce the notion of a quasi-module-HKZ-reduced pseudo-basis, which extends the concept of a quasi-HKZ-reduced basis from [10] to module lattices. Using these definitions, we propose a module lattice enumeration algorithm.

On the other hand, a rank-n module $M \subset K^m$ can be regarded as an nd-dimensional Euclidean lattice generated by a HKZ-reduced basis $\{\vec{b}_1, \ldots, \vec{b}_{nd}\}$. According to the analysis of Hanrot and Stehlé [10], it satisfies

$$\|\vec{b}_1^*\| \leq \|\vec{b}_{(i-1)d+1}^*\| \cdot \sqrt{nd}^{\ln\left(\frac{n}{n-i+1}\right)}.$$

This result directly affects the complexity of the enumeration algorithm. The dominant cost in lattice enumeration is determined by the number of integer points in a given search space, which can be approximated as:

$$O\left(\sum_{i=1}^{nd} \frac{(2\pi e)^i \cdot \|\vec{b}_1^*\|^i}{\sqrt{nd}^i \cdot \prod_{j\geq n-i+1} \|\vec{b}_j^*\|}\right) = \max_i O\left(\frac{(2\pi e)^i \cdot \|\vec{b}_1^*\|^i}{\sqrt{nd}^i \cdot \prod_{j\geq n-i+1} \|\vec{b}_j^*\|}\right).$$

In order to adapt the analysis, we prove that given a module-HKZ-reduced pseudo-basis $((I_i, \mathbf{b}_i))_{1\leq i\leq n}$ of a module M, the following holds:

$$\|\mathbf{b}_1^*\| \leq \|\mathbf{b}_i^*\| \cdot \sqrt{n}^{\ln\left(\frac{n}{n-i+1}\right)}.$$

Compared to the profile of an HKZ-reduced basis of the nd-dimensional Euclidean lattice, the profile of a module-HKZ-reduced pseudo-basis gives a different relation, which is a motivation of the new analysis of the lattice enumeration algorithm over modules.

Observation 1.4. *Let $((I_i, \mathbf{b}_i))_{i\leq n}$ be a module-HKZ-reduced pseudo-basis of a rank-n module with extension degree d, and let $\{\vec{b}_1, \ldots, \vec{b}_{nd}\}$ be an HKZ-reduced basis of an arbitrary Euclidean lattice. Then, the following bounds hold:*

$$(\textit{Module lattice})\ \|\mathbf{b}_1^*\| \leq \|\mathbf{b}_i^*\| \cdot \sqrt{n}^{\ln\left(\frac{n}{n-i+1}\right)}$$

$$(\textit{Euclidean lattice})\ \|\vec{b}_1^*\| \leq \|\vec{b}_{(i-1)d+1}^*\| \cdot \sqrt{nd}^{\ln\left(\frac{n}{n-i+1}\right)}$$

From Observation 1.4, we see that for each index i, the ratio $\|\mathbf{b}_1^*\|/\|\mathbf{b}_i^*\|$ is smaller than $\|\vec{b}_1^*|/\|\vec{b}_{(i-1)\cdot d+1}^*\|$. We thus expect that the complexity of the module enumeration algorithm is estimated as follows:

$$O\left(\sum_{i=1}^{n} \frac{(2\pi e)^i \cdot \|\mathbf{b}_1^*\|^i}{\sqrt{nd}^i \cdot \prod_{j\geq n-i+1} \|\mathbf{b}_j^*\|}\right) = \max_i O\left(\frac{(2\pi e)^i \cdot \|\mathbf{b}_1^*\|^i}{\sqrt{nd}^i \cdot \prod_{j\geq n-i+1} \|\mathbf{b}_j^*\|}\right).$$

We emphasize that a smaller ratio of $\|\mathbf{b}_1^*\|/\|\mathbf{b}_i^*\|$ increases the denominator, which effectively reduces the overall time complexity of the enumeration algorithm.

Based on the module lattice enumeration algorithm, we additionally provide an improved analysis of Module-SVP algorithm.

Application to Module-CVP. This paragraph begins with a natural question: how can the algorithm be applied to solve module-CVP, and does it improve the algorithm's performance?

As shown in [10], our analysis can be extended to the module-CVP. The module-CVP asks us to find the closest lattice vector given a pseudo-basis $\mathbf{B} = ((I_i, \mathbf{b}_i))_{1\leq i\leq n}$ of a module M and a target vector $\mathbf{t} \in \mathrm{span}(M)$. The high-level strategy for solving module-CVP is essentially the same as for solving the module-SVP. However, unlike the module-SVP case, the module enumeration algorithm does not improve the performance of solving module-CVP.

Let $\mathbf{D} = (\mathbf{d}_i)_{1\leq i\leq nd}$ be the corresponding $\mathbb{Z}$-module basis of M. The first step in solving module-CVP is to compute the module-HKZ reduced pseudo-basis $(I_i, \mathbf{b}_i)$.

On the other hand, with a $\mathbb{Z}$-basis for the module M, Babai's nearest plane algorithm [1] can be employed in the same way as for the lattice enumeration algorithm. By Babai's nearest plane algorithm, we know that there exists a lattice vector $\mathbf{d}$ at a distance of at most $\sqrt{nd} \cdot \max_i \|\mathbf{d}_i^*\|$ from the target vector $\mathbf{t}$, where $\mathbf{d}_i^*$ are the Gram-Schmidt orthogonalized vectors corresponding to the $\mathbb{Z}$-basis $\mathbf{D}$.

Additionally, due to the definition of the profile of $\mathbf{d}_i^*$, it follows that $\max_i \|\mathbf{d}_i^*\| = \|\mathbf{d}_1^*\|$. Consequently, if we set $A = nd \cdot \|\mathbf{d}_1\|^2$ in the module lattice enumeration procedure, we can find all solutions, and it terminates in time $\max_{i\leq nd} N_i^{\mathrm{CVP}}$ where

$$N_i^{\mathrm{CVP}} \leq \frac{\sqrt{2\pi e}^i \cdot \sqrt{nd}^i \cdot \|\mathbf{d}_1^*\|^i}{\sqrt{nd}^i \prod_{j\geq nd-i} \|\mathbf{d}_j^*\|}.$$

Since $\|\mathbf{d}_j^*\|$ for all j is less than $\|\mathbf{d}_1^*\|$, the term is maximized when $i = nd$.

At this point, the denominator becomes equal to the volume of the lattice $\Lambda(\mathbf{D})$. This result implies that the ring structure cannot be exploited in the module-CVP scenario. Therefore, we conclude that under these circumstances, the algebraic structure provides no performance improvement.

2 Preliminaries

Notations. Vectors and matrices are denoted in bold letters. The n-dimensional unit ball with radius R is denoted by $\mathcal{B}_n(R)$, and its volume is given by $R^n \cdot \frac{\pi^{n/2}}{\Gamma(n/2+1)}$. 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 nonzero $x \in \mathbb{R}$ is denoted by $\ln x$, while the logarithm with base 2 is denoted by either $\log x$ or $\log_2 x$.

2.1 Algebraic Number Theory Backgrounds

This section introduces the definitions of key algebraic objects. For more details, please refer to [15].

Let K be a number field of extension degree d. The field K has r_1 real embeddings and $2r_2$ complex embeddings, denoted by σ_i, where $r_1 + 2r_2 = d$. Using these embeddings, we define a function $\sigma : K \to \mathbb{R}^{r_1} \times \mathbb{C}^{2r_2}$ as $\sigma(x) = (\sigma_1(x), \ldots, \sigma_d(x))$, which is called the *canonical embedding*. For convenience, we set $\sigma_{r_1+i}(x) = \overline{\sigma_{r_1+r_2+i}(x)}$, where $\bar{\cdot}$ denotes complex conjugation.

The embeddings are used to define the *field norm* and the *field trace* for any $x \in K$ as follows: $\mathcal{N}(x) = \prod_{i=1}^{d} \sigma_i(x)$ and $\mathrm{Tr}(x) = \sum_{i=1}^{d} \sigma_i(x)$. Additionally, we define the following norms for every $x \in K$: $\|x\| = \sqrt{\sum_{i=1}^{d} |\sigma_i(x)|^2}$ and $\|x\|_\infty = \max_{1 \le i \le d} |\sigma_i(x)|$. Given a number field K of extension degree d, we denote its real tensor product by $K_\mathbb{R} = K \otimes_\mathbb{Q} \mathbb{R}$.

Let R be the ring of integers of K. The ring R has a $\mathbb{Z}$-basis $(x_1, \ldots, x_d)$ and can also be regarded as a lattice under the canonical embedding. The discriminant of K is defined as $\Delta_K = |\det(\sigma_i(x_j))|^2$.

A *fractional ideal* I of K is an additive subgroup of K that is closed under multiplication by any element of the ring of integers R. Additionally, there exists a nonzero integer $x \in \mathbb{Z} \setminus \{0\}$ such that $xI \subseteq R$.

A nonzero ideal can be viewed as a lattice in $K_\mathbb{R}$ using the canonical embedding, and we refer to this lattice as the *ideal lattice*. In general, finding a $\mathbb{Z}$-basis for an arbitrary ideal I is computationally difficult. However, throughout this paper, we assume that a $\mathbb{Z}$-basis for I is given. In other words, we focus only on ideals for which a $\mathbb{Z}$-basis is known.

The *norm* of an ideal I, denoted by $\mathcal{N}(I)$, is defined as $\mathcal{N}(I) = \frac{\mathcal{N}(xI)}{|\mathcal{N}(x)|}$, for any $x \in R \setminus \{0\}$ such that $xI \subseteq R$. The norm satisfies the multiplicative property: $\mathcal{N}(I \cdot J) = \mathcal{N}(I) \cdot \mathcal{N}(J)$, for any fractional ideals I and J.

2.2 Lattice Backgrounds

A lattice $\Lambda \subset \mathbb{R}^m$ is the set of vectors generated by all integer combinations of n linearly independent vectors $\vec{b}_1, \ldots, \vec{b}_n \in \mathbb{R}^m$. The parameters m and n are called the dimension and the rank of the lattice Λ, respectively. The vectors $\vec{b}_1, \ldots, \vec{b}_n$

are called a basis of the lattice and are denoted by $\mathbf{B} = (\vec{b}_1, \ldots, \vec{b}_n) \in \mathbb{R}^{m \times n}$. In other words, the lattice Λ can be expressed as $\Lambda = \{\sum_{i=1}^{n} x_i \vec{b}_i : x_i \in \mathbb{Z}\}$. Here, n is called the rank of Λ, and Λ is called a full-rank lattice if $m = n$. The set $P(\mathbf{B}) = \{\sum_{i=1}^{n} x_i \vec{b}_i : x_i \in [0, 1)\}$ is called the *parallelepiped* of the basis $\mathbf{B}$.

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

Gram-Schmidt Orthogonalization. For a basis $\mathbf{B} = (\vec{b}_1, \ldots, \vec{b}_n)$ of a lattice Λ and $i \in \{1, \ldots, n\}$, we define $\vec{b}_i^*$ as the projection of $\vec{b}_i$ orthogonally onto $\mathrm{span}(\vec{b}_1, \ldots, \vec{b}_{i-1})$. The Gram-Schmidt orthogonalization of $\mathbf{B}$ is the set of vectors $\{\vec{b}_1^*, \ldots, \vec{b}_n^*\}$. For simplicity, we define $\mathcal{P}_i$ as the projection operator orthogonal to $\mathrm{span}(\vec{b}_1^*, \ldots, \vec{b}_{i-1}^*)$.

The Gram-Schmidt orthogonalized basis can be represented in matrix form as $\mathbf{B}^* = (\vec{b}_1^*, \ldots, \vec{b}_n^*)$. The volume of a full-rank lattice Λ with basis $\mathbf{B}$ is defined as $\prod_{i=1}^{n} \|\vec{b}_i^*\|$ and is denoted by $\det(\mathbf{B})$ or $\mathrm{vol}(\Lambda)$. We note that the lattice volume is well defined: it does not depend on the choice of basis $\mathbf{B}$.

Definition 2.2 (HKZ-reduction)**.** *We say that $\mathbf{B} = (\vec{b}_1, \ldots, \vec{b}_n)$ is an Hermite-Korkine-Zolotarev (HKZ) reduced basis if $\|\vec{b}_1\| = \lambda_1(\Lambda(\mathbf{B}))$, and the remaining basis vectors are HKZ-reduced when projected onto the subspace orthogonal to $\vec{b}_1$.*

Lemma 2.3 ([10, Lem. 1.])**.** *If $(\vec{b}_1, \ldots, \vec{b}_n)$ is HKZ-reduced, then for any $i \leq n$, we have*

$$\|\vec{b}_i^*\| \leq \sqrt{\frac{n-i+5}{4}} \cdot \left(\prod_{j \geq i} \|\vec{b}_j^*\| \right)^{\frac{1}{n-i+1}}.$$

Definition 2.4 (Quasi-HKZ-reduction)**.** *A basis $(\vec{b}_1, \ldots, \vec{b}_n)$ is quasi-HKZ-reduced if it is size-reduced, if $\|\vec{b}_2^*\| \geq \|\vec{b}_1^*\|/2$ and if once projected orthogonally to $\vec{b}_1$, the other $\vec{b}_i$'s are HKZ-reduced.*

The Gaussian Heuristic is used to estimate the number of lattice points in measurable sets or the length of the shortest vector.

Heuristics 2.5 (Gaussian Heuristic)**.** *Let Λ be a full rank lattice in $\mathbb{R}^n$, and S is a measurable set in $\mathbb{R}^n$. Then, the number of lattice points in $S \cap \Lambda$ is approximately* $\mathrm{vol}(S)/\mathrm{vol}(\Lambda)$.

Lemma 2.6. *Suppose that a basis $(\vec{b}_1, \ldots, \vec{b}_n)$ is HKZ-reduced and satisfies the worst-case bound of the inequality in the Lemma 2.3. Then, it holds that*

$$\|\vec{b}_1^*\| = \|\vec{b}_i^*\| \cdot \sqrt{n}^{\ln\left(\frac{n}{n-i+1}\right)}.$$

Lemma 2.7 ([10, Thm. 3.]). *Let $\{\vec{b}_i\}_{1\le i\le d}$ be HKZ-reduced Then*

$$\frac{\|\vec{b}_1\|^i}{\prod_{j\ge d-i}\|\vec{b}_i^*\|} \le \sqrt{d}^{\,i\cdot\left(1+\ln\frac{d}{i}\right)}.$$

2.3 Module lattice backgrounds

Let $((I_i, \mathbf{b}_i))_{i=1}^n$ be a pseudo-basis of a module M, where $M = \sum_{i=1}^n I_i\mathbf{b}_i$ and $\mathbf{b}_i$ are $K_{\mathbb{R}}$-linearly independent vectors in $K_{\mathbb{R}}^m$. Here, *independent* means that the zero vector cannot be expressed as a non-trivial $K_{\mathbb{R}}$-linear combination of the $\mathbf{b}_i$. n is called the *rank* of the module M.

As in the number field case, for any vector $\mathbf{v} \in K_{\mathbb{R}}^m$, we can define the extended canonical embedding $\sigma(\mathbf{v})$. The definition is straightforward: $\sigma(\mathbf{v}) \in \mathbb{R}^{nd}$ is obtained by concatenating the embeddings of the components of $\mathbf{v}$. Consequently, M can be viewed as a lattice via the extended canonical embedding map. This allows us to define the volume of a module from a lattice perspective. More precisely, the volume of a module is given by $\det(M) = \Delta_K^{n/2} \cdot \mathcal{N}(\det_{K_{\mathbb{R}}}(M))$, where $\det_{K_{\mathbb{R}}}(M) = \det(\bar{\mathbf{B}}^T \cdot \mathbf{B})^{1/2} \cdot \prod_i I_i$. Here, $(I_i, \mathbf{B})$ is a pseudo-basis of M.[4]

Furthermore, we extend the inner products for $\mathbf{a}, \mathbf{b} \in K_{\mathbb{R}}^m$ as follows:

$$\langle \mathbf{a}, \mathbf{b}\rangle_{K_{\mathbb{R}}} = \sum_i a_i\bar{b}_i \in K_{\mathbb{R}}, \text{ and } \langle \mathbf{a}, \mathbf{b}\rangle = \mathrm{Tr}\left(\sum_i a_i\bar{b}_i\right) \in \mathbb{C}.$$

The inner product defined over $K_{\mathbb{R}}$ is also used to define some (algebraic) norms. For example, for any $\mathbf{v} \in K_{\mathbb{R}}^m$, we define $\|\mathbf{v}\|_{K_{\mathbb{R}}} = \sqrt{\langle \mathbf{v}, \mathbf{v}\rangle_{K_{\mathbb{R}}}}$. Similarly, we define $\mathcal{N}(\mathbf{v}) = \mathcal{N}(\|\mathbf{v}\|_{K_{\mathbb{R}}})$.

Note that lattice reduction algorithms, such as LLL [16], BKZ [4], and HKZ reduction, can be easily generalized since the inner product $\langle \mathbf{a}, \mathbf{b}\rangle$ is well-defined. Conceptually, this generalization is equivalent to applying reduction algorithms after identifying ring elements to the complex space via the canonical embedding.

With these preliminaries, we formally define the module-SVP and module-CVP problems:

Definition 2.8 (Module-SVP). *Let K be a number field and $K_{\mathbb{R}}$ as above. Given a module $M \subset K_{\mathbb{R}}^m$, the module-SVP is to find a nonzero vector $\mathbf{x} \in M$ such that $\|\mathbf{x}\|$ is minimized.*

Definition 2.9 (Module-CVP). *Let K be a number field and $K_{\mathbb{R}}$ as above. Given a module $M \subset K_{\mathbb{R}}^m$ and a target $\mathbf{t} \in K_{\mathbb{R}}^m$, the module-CVP is to find the vector $\mathbf{x} \in M$ closest to the target vector $\mathbf{t}$ with respect to the $\|\cdot\|$ norm.*

Let $\lambda_1(M)$ denote the norm of the shortest nonzero element of M, and let $\lambda_1^{\mathcal{N}}(M) = \inf\{\mathcal{N}(\mathbf{v}) : \mathbf{v} \in M \setminus \{\mathbf{0}\}\}$. Then, the following inequality holds:

[4] The determinant of a module is well-defined.

Lemma 2.10 ([15, Lem. 2.2]). *For any rank-n module M, we have:*

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

Gram-Schmidt Orthogonalization for Modules. The Gram-Schmidt Orthogonalization can be extended from real numbers to number fields using the inner products defined over $K_{\mathbb{R}}$.

More precisely, for $K_{\mathbb{R}}$-linearly independent vectors $(\mathbf{b}_1, \ldots, \mathbf{b}_n) \in K_{\mathbb{R}}^{m \times n}$, we define $\mathbf{b}_1^* = \mathbf{b}_1$, and for $i > 1$,

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

Using these definitions, one can compute the QR-factorization of the matrix over $K_{\mathbb{R}}$ by setting $r_{i,i} = \|\mathbf{b}_i^*\|_{K_{\mathbb{R}}}$. Then, we have $\mathbf{B} = \mathbf{Q} \cdot \mathbf{R}$, where $\mathbf{Q} \in K_{\mathbb{R}}^{m \times n}$ is a matrix satisfying $\bar{\mathbf{Q}} \cdot \mathbf{Q} = \mathbf{I}$, and $\mathbf{R} = (r_{i,j})$ is an upper triangular matrix. For a simple description, the orthogonal projection of $\mathbf{v}$ onto $(\mathbf{b}_1, \ldots, \mathbf{b}_i)^{\perp}$ is denoted by $\tau_i(\mathbf{v})$.

The following lemma can be directly obtained from QR-factorization.

Lemma 2.11 ([15, Lem. 2.6]). *Let $((I_i, \mathbf{b}_i))_{i \leq n}$ be a pseudo-basis of a module $M \subset K_{\mathbb{R}}^m$. Then, it holds*

$$\det_{K_{\mathbb{R}}} M = \prod_i r_{i,i} I_i \text{ and } \det M = \Delta_K^{n/2} \prod_i \mathcal{N}(r_{i,i} \cdot I_i).$$

Moreover, using the canonical embedding, we obtain an analogue result of the Minkowski' bound. The shortest vector $\mathbf{v}$ of a rank-n module M is bounded as follows:

$$\begin{aligned} \|\mathbf{v}\| &\leq \sqrt{n} \cdot \left(\Delta_K^{n/2} \prod_{i=1}^{n} \mathcal{N}(r_{i,i} \cdot I_i) \right)^{1/nd} \\ &= \sqrt{n} \cdot \Delta_K^{1/(2d)} \cdot \left(\prod_{i=1}^{n} \mathcal{N}(r_{i,i} \cdot I_i) \right)^{1/nd} \end{aligned} \tag{1}$$

Lemma 2.12 (Thm. 4, [7]). *Suppose a pseudo-basis $((I_i, \mathbf{b}_i))_{i \leq n}$ and a full-rank set of vectors $(\mathbf{s}_i)$ of $M \subset K_{\mathbb{R}}^m$ are given. Then, there exists an algorithm that returns a pseudo-basis $\{(J_i, \mathbf{c}_i)\}_{i \leq n}$ such that $\mathbf{c}_i \in M$ and $\mathbf{c}_i^* = \mathbf{s}_i^*$ for all i. Furthermore, if $M \subset K^m$, the time complexity is polynomial in $\log \Delta_K$ and the input bit-length.*

Handling bit-lengths. Lee et al. [15] proposed algorithms for handling the bit-length of ideals. All algorithms preserve the module space, but provide efficient representations of ideals and R-factor of pseudo-basis. For example, a

scaled pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ implies $\mathcal{N}(I_i) \le 1$ for all i. Fortunately, every algorithm terminates in polynomial time in $\log \Delta_K$ and the input bit-length.

We introduce the formal definitions and algorithms to transform the input pseudo-basis into a *good* pseudo-basis.

Definition 2.13 ([15, Def. 3.5])**.** *We say a pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ with $I_i \subset K$ and $\mathbf{b}_i \in K_{\mathbb{R}}^m$ for all i is scaled if for all $i \le n$,*

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

If $\|r_{i,j}/r_{i,i}\| \le (4d)^d \Delta_K^{1/2}$ for all $i < j \le n$, it is called a 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 $i = 1$ *to* n **do**
  Use LLL to find $s_i \in r_{i,i} \cdot I_i \setminus \{0\}$ such that $\|s_i\| \le 2^d \Delta_K^{1/(2d)} \mathcal{N}(r_{i,i} \cdot I_i)^{1/d}$.
  Find $x_i \in I_i$ such that $s_i = r_{i,i} \cdot x_i$ and update $I_i' = I_i \cdot \langle x_i \rangle^{-1}$ and $\mathbf{b}_i' = x_i \cdot \mathbf{b}_i$.
end for
return $((I_i', \mathbf{b}_i'))_{i\le n}$.

Lemma 2.14 ([15, Lem. 3.6])**.** *Given a pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ of a module M, Algorithm 1 returns a scaled pseudo-basis of the module M and preserves the $\mathcal{N}(r_{i,i} I_i)$ for all i. The algorithm terminates in polynomial time in $\log \Delta_K$ and the input bit-length.*

We further introduce a notion called size-reduced. To this end, we additionally introduce $\lfloor \cdot \rceil_R$ operation. Given input $y = \sum_i y_i x_i \in K_{\mathbb{R}}$ for some $y_i \in \mathbb{R}$, the operation $\lfloor y \rceil_R$ returns $\sum \lfloor y_i \rceil x_i$.

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

Lemma 2.15 ([15, Lem. 3.7]). *Given a pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ of a module M, there exists an algorithm (Algorithm 2) that returns a scaled size-reduced pseudo-basis of the module M and preserves the $\mathcal{N}(r_{i,i}I_i)$ for all i. The algorithm terminates in polynomial time in $\log \Delta_K$ and the input bit-length.*

3 Lattice Enumeration Algorithm over Modules

This section presents an analysis of the lattice enumeration algorithm for modules. We adapt the analysis from [10] to modules, demonstrating improvements in asymptotic performance.

First, recall Kannan's lattice enumeration algorithm (Algorithm 3). Given a basis $\{\vec{b}_1, \ldots, \vec{b}_d\}$ of a lattice and a bound $A \in \mathbb{Z}$, the goal of the enumeration algorithm is to find all lattice vectors $\sum_i x_i\vec{b}_i$ such that $\|\sum_i x_i\vec{b}_i\|^2 \le A$, where $x_i \in \mathbb{Z}$. Intuitively, the algorithm counts all integer points within a given bound.

More precisely, the condition $\|\sum_i x_i\vec{b}_i\|^2 \le A$ can be reformulated as

$$\sum_{i=1}^{d}\left(x_i + \sum_{j=i+1}^{d}\mu_{j,i}^{\mathbb{Q}}x_j\right)^2 \cdot \bar{r}_i \le A, \tag{2}$$

where $\mu_{j,i}^{\mathbb{Q}} = \frac{\langle\vec{b}_i,\vec{b}_j^*\rangle}{\|\vec{b}_j^*\|^2}$ and $\bar{r}_i = \|\vec{b}_i^*\|^2$. For each $i \in [d]$, define $c_i = -\sum_{j=i+1}^{d}\mu_{j,i}^{\mathbb{Q}}\cdot x_j$ and $y_i = x_i - c_i$. Then, finding integer points $(x_1, \ldots, x_d)$ is equivalent to finding all integer points $(y_1, \ldots, y_d) \in \mathbb{Z}^d$ that satisfy

$$\begin{aligned} y_d^2 \cdot \bar{r}_d &\le A, \\ y_{d-1}^2 \cdot \bar{r}_{d-1} + y_d^2 \cdot \bar{r}_d &\le A, \\ &\cdots \\ \sum_{i=1}^{d} y_i^2 \cdot \bar{r}_i &\le A. \end{aligned} \tag{3}$$

Let N_i represent the number of integer points satisfying

$$\{(y_i, \ldots, y_d) \mid \sum_{j\ge i} y_j^2 \cdot \bar{r}_j \le A\}.$$

As analyzed in [10], the time complexity of the enumeration algorithm is dominated by $\max_i N_i$, up to a polynomial factor. As a consequence, by combining Lemma 2.7 with the upper bound of N_i, [10] concludes the algorithm for solving SVP using the lattice enumeration algorithm terminates in time $d^{d/(2e)+o(d)}$ up to a polynomial factor.

The algorithm primarily involves inner product computations and QR-factorization of the input $\mathbb{Z}$-basis. Since inner product calculations and QR-factorization of a pseudo-basis for a module are already addressed in [15], extending the lattice enumeration algorithm to modules appears straightforward. However, there are

technical challenges, such as adapting definitions for modules and devising an algorithm to convert a given pseudo-basis of a module into a $\mathbb{Z}$-basis.

In the latter part of this section, we define relevant terms and propose an algorithm to derive a $\mathbb{Z}$-basis for a module. Finally, we provide a new analysis of the lattice enumeration algorithm over modules, showing that it outperforms its counterpart for Euclidean lattices. Specifically, Section 3.1 introduces definitions for module lattices, and Section 3.2 describes the module lattice enumeration algorithm. Lastly, Section 3.3 provides a detailed analysis of the module lattice enumeration algorithm.

Algorithm 3: Lattice Enumeration Algorithm

Input: An $\mathbb{Z}$-basis $(\vec{b}_1, \ldots, \vec{b}_d)$ and a bound $A \in \mathbb{Z}$.
Output: All vectors in $\Lambda(\vec{b}_1, \ldots, \vec{b}_d)$ of which 2-norm is bounded by A.

Compute $\mu^{\mathbb{Q}}_{i,j}$'s and $\|\vec{b}^*_i\|$'s of $(\vec{b}_1, \ldots, \vec{b}_d)$.
$x \leftarrow \vec{0}, l \leftarrow \vec{0}, i \leftarrow 1$ and $S \leftarrow \emptyset$
while $i \leq d$ **do**
  $l_i \leftarrow (x_i + \sum_{j>i} x_j \mu^{\mathbb{Q}}_{j,i})^2 \|\vec{b}^*_i\|^2$
  if $i = 1$ *and* $\sum_{j \geq i} l_j \leq A$ **then**
    $S \leftarrow S \cup \{\vec{x}\}$, $x_1 \leftarrow x_1 + 1$
  end if
  if $i \neq 1$ *and* $\sum_{j \geq i} l_j \leq A$ **then**
    $i \leftarrow i - 1$
    $x_i \leftarrow \left\lceil -\sum_{j>i}(x_j \mu^{\mathbb{Q}}_{j,i}) - \sqrt{\frac{A - \sum_{j>i} l_j}{\|\vec{b}^*_i\|^2}} \right\rceil$
  end if
  if $\sum_{j \geq i} l_j > A$ **then**
    $i \leftarrow i + 1$, and $x_i \leftarrow x_i + 1$.
  end if
end while
return S

3.1 Definitions for Modules

We first extend the notions of Euclidean lattices to module lattices in order to employ the enumeration algorithm over modules. The Hermite-Korkine-Zolotarev (HKZ)-reduced basis (Definition 2.2) is adapted to a module-HKZ-reduced *pseudo-basis* (Definition 3.1), as no basis exists in some modules. Similarly, the quasi-HKZ-reduced basis for Euclidean lattices (Definition 2.4) can be extended the quasi-module-HKZ-reduced pseudo-basis for any modules (Definition 3.2).

These definitions are mainly used in the analysis of module lattice enumeration algorithm in Section 3.3. Intuitively, we define all the necessary module-related concepts to make it easier to adapt existing algorithms from [8, 10, 12].

For example, computing the quasi-module-HKZ-reduced pseudo-basis in Definition 3.2 turns out to be simpler than expected. This is because the previous algorithms for computing quasi-HKZ-reduced bases can be directly adapted to the module setting. Detailed algorithms will be provided in later sections.

Definition 3.1 (Module-HKZ-reduced)**.** *A pseudo-basis* $((I_i, \mathbf{b}_i))_{1\leq i\leq n}$ *is said to be module-HKZ-reduced the vector* $\mathbf{b}_1$ *reaches the lattice minimum, and the projections of the* $((I_i, \mathbf{b}_i))_{2\leq i\leq n}$*'s orthogonally to the vector* $\mathbf{b}_1$ *are themselves a module-HKZ-reduced pseudo basis.*

Note that the projection of $((I_i, \mathbf{b}_i))_{2\leq i\leq n}$ to the vector $\mathbf{b}_1$ is identical to that of $\{\mathbf{b}_i\}_{2\leq i\leq n}$ to the vector $\mathbf{b}_1$. The projection of $\{\mathbf{b}_i\}_{2\leq i\leq n}$ to $\mathbf{b}_i$ means that $\mathbf{b}_i - \mu_{i,1}\mathbf{b}_1$.

Hanrot and Stehlé [10] proposed a notion, named quasi-HKZ-reduced basis to tightly analyze the Kannan's lattice enumeration algorithm. To analyze the algorithm to the module variant, we also required an extended definition of quasi-HKZ-reduced for modules.

Definition 3.2 (Quasi-Module-HKZ-reduction)**.** *A pseudo-basis* $\{(I_1, \mathbf{b}_1)\}_{1\leq i\leq n}$ *of module* M *is quasi-module-HKZ-reduced, if* $\{(I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2)\}$ *is module-HKZ-reduced pseudo-basis. and if once projected orthogonally to* $\mathbf{b}_1$*, the other* $(I_i, \mathbf{b}_i)$*'s are HKZ-reduced.*

When we deal with algebraic objects such as ideals, modules, the representation of the objects should be considered in order to instantiate algorithms for algebraic objects. Throughout this paper, we assume that every pseudo-basis is a scaled pseudo-basis (Definition 2.13). Sometimes, we employ a size-reduced algorithm to hand bit-lengths. Fortunately, there exists an algorithm which converts an input pseudo-basis into a scaled one and a size-reduced one (Algorithm 2).

3.2 Module Lattice Enumeration Algorithm

This section provides a module lattice enumeration algorithm (Algorithm 5). Similar to the previous analysis [10], we will use the property of quasi-module-HKZ-reduced pseudo-basis (Definition 3.2) to analyze the performance of the module lattice enumeration algorithm.

The core idea of this section is that the input to Algorithm 3 must be a $\mathbb{Z}$-basis of the lattice, rather than a pseudo-basis that we have. Therefore, it is necessary to convert the pseudo-basis into a $\mathbb{Z}$-basis of the lattice. The conversion algorithm is given in Algorithm 4, and we provide the correctness and complexity of the algorithm in Theorem 3.3.

Theorem 3.3 (Conversion to $\mathbb{Z}$-basis)**.** *Let* K *be a number field of an extension degree* d*, and let* $K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R}$*. Let* $M \subset K^m$ *be a rank-*n *module, and let* $((I_i, \mathbf{b}_i))_{1\leq i\leq n}$ *be its scaled and size-reduced pseudo-basis. Let* $\mathbf{c}_i$ *be a LLL-reduced* $\mathbb{Z}$*-basis of* I_i*. Then, Algorithm 4, given* $((I_i, \mathbf{b}_i))_{1\leq i\leq n}$ *as input, returns a* $\mathbb{Z}$*-basis* $\{\mathbf{w}_1, \ldots, \mathbf{w}_{nd}\}$ *for* M*. The time complexity of the algorithm is* $\mathsf{poly}(B, \log \Delta_K) \cdot n \cdot 2^{O(d)}$*, up to a polynomial factor, where* B *is the input size. Specifically, it holds that* $\prod_{j=1}^{d} \|\mathbf{w}^*_{(i-1)\cdot d+j}\| = \mathcal{N}(I_i \cdot \mathbf{b}_i^*) \cdot \sqrt{\Delta_K}$*.*

Algorithm 4: $\mathbb{Z}$-basis Conversion

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

Compute the Gram-Schmit orthogonalization of a pseudo-basis, $((I_i, \mathbf{b}_i))_{1 \leq i \leq n}$. Let $\{(I_i, \mathbf{r}_i)\}_{1 \leq i \leq n}$ be an output.
for $i = 1$ *to* n **do**
  Compute a set of matrices $\{\mathbf{U}_i\}$ such that $\mathbf{c}_i \cdot r_{i,k} \cdot \mathbf{U}_i$ is a HKZ-reduced, where $\mathbf{c}_i$ is a $\mathbb{Z}$-basis of an ideal I_i.
end for
Compute $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{nd}]$, which is a $\mathbb{Z}$-basis of a module M
return W

Proof of Theorem 3.3. We first prove the correctness of Algorithm 4, showing that it outputs a $\mathbb{Z}$-basis for the module M.

Let $((I_i, \mathbf{b}_i))_{1 \leq i \leq n}$ be a pseudo-basis of the rank-n module M. We assume that the pseudo-basis is a scaled sized reduced pseudo-basis using Algorithm 2. By applying the Gram-Schmidt orthogonalization (GSO) to the pseudo-basis, we obtain $\mathbf{Q}, \{(I_i, \mathbf{r}_i)\}_{1 \leq i \leq n}$, where $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_n]$ satisfies

$$\mathbf{R} = \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ & r_{2,2} & \cdots & r_{2,n} \\ & & \ddots & \vdots \\ & & & r_{n,n} \end{pmatrix} \in K_{\mathbb{R}}^{n \times n}.$$

Next, we construct a $\mathbb{Z}$-basis for the module $\mathbf{Q}^T \cdot M$. Let $\mathbf{c}_i = (c_{i,1}, \ldots, c_{i,d})$ denote an LLL-reduced $\mathbb{Z}$-basis of the ideal I_i. Since $((I_i, \mathbf{r}_i))_{i \leq n}$ is also a pseudo-basis of $\mathbf{Q}^T \cdot M$, any element of $\mathbf{Q}^T \cdot M$ can be expressed as a combination of $\mathbf{c}_i$ and $\mathbf{r}_i$. Hence, the $\mathbb{Z}$-basis of $\mathbf{Q}^T \cdot M$ can be expressed as

$$\mathbf{B}_{\mathbb{Z}} = \begin{pmatrix} \mathbf{c}_1 \cdot r_{1,1} & \mathbf{c}_2 \cdot r_{1,2} & \cdots & \mathbf{c}_n \cdot r_{1,n} \\ & \mathbf{c}_2 \cdot r_{2,2} & \cdots & \mathbf{c}_n \cdot r_{2,n} \\ & & \ddots & \vdots \\ & & & \mathbf{c}_n \cdot r_{n,n} \end{pmatrix} \in K_{\mathbb{R}}^{n \times nd},$$

where $\mathbf{c}_i \cdot r_{j,k} = (c_{i,1} \cdot r_{j,k}, \ldots, c_{i,d} \cdot r_{j,k}) \in K_{\mathbb{R}}^d$ is a row vector.

Although $\mathbf{B}_{\mathbb{Z}}$ is a valid $\mathbb{Z}$-basis of $\mathbf{Q}^T \cdot M$, we refine it to employ the complexity analysis of the module enumeration algorithm. For each $i \in [n]$, compute a matrix $\mathbf{U}_i \in \mathbb{Z}^{d \times d}$ such that $\mathbf{c}_i \cdot r_{i,i} \cdot \mathbf{U}_i$ is HKZ-reduced.[5] Intuitively, it is conducted as the followings:

[5] Later, we will just require the profile $\|\mathbf{b}_i^*\|$, so this computation suffices for our purpose.

1. Identify the d elements of $\mathbf{c}_i \cdot r_{i,i}$ as the complex space using the canonical embedding σ, resulting in $[\sigma(c_{i,1} \cdot r_{i,i}), \ldots, \sigma(c_{i,d} \cdot r_{i,i})] \in \mathbb{C}^{d \times d}$, which forms a basis for a d-dimensional lattice.
2. Compute an unimodular matrix $\mathbf{U}_i \in \mathbb{Z}^{d \times d}$ such that the product
$$[\sigma(\mathbf{c}_{i,1} \cdot r_{i,i}), \ldots, \sigma(\mathbf{c}_{i,d} \cdot r_{i,i})] \cdot \mathbf{U}_i$$
is HKZ-reduced.
3. Define $\mathbf{c}_i \cdot r_{i,i} \cdot \mathbf{U}_i$ as the HKZ-reduced basis for the original space.

We then construct a new basis of $\mathbf{Q}^T \cdot M$ by multiplying the block-diagonal matrix $\mathsf{diag}(\mathbf{U}_1, \ldots, \mathbf{U}_n)$ with $\mathbf{B}_{\mathbb{Z}}$, ensuring that for each i, $\mathbf{c}_i \cdot r_{i,i} \cdot \mathbf{U}_i$ is HKZ-reduced. Let
$$\mathbf{W} = \mathbf{Q} \cdot \mathbf{B}_{\mathbb{Z}} \cdot \mathsf{diag}(\mathbf{U}_1, \ldots, \mathbf{U}_n).$$
The resulting matrix $\mathbf{W}$ is expressed as $[\mathbf{w}_1, \ldots, \mathbf{w}_{nd}]$ and remains a $\mathbb{Z}$-basis for the module M.

In addition, by the construction, it directly implies that $\prod_{j=1}^{d} \|\mathbf{w}^*_{(i-1)\cdot d+j}\| = \prod_{j=1}^{d} \|\mathbf{Q}^T \cdot \mathbf{w}^*_{(i-1)\cdot d+j}\|$ is the volume of the lattice generated by $\mathbf{c}_i \cdot r_{i,i}$. It concludes that
$$\prod_{j=1}^{d} \|\mathbf{w}^*_{(i-1)\cdot d+j}\| = \mathcal{N}(I_i \cdot r_{i,i}) \cdot \sqrt{\Delta_K}.$$

Time complexity. The conversion algorithm mainly consists of two steps. First, compute the Gram-Schmidt orthogonalized pseudo-basis $\{(I_i, \mathbf{r}_i)\}_{i \in [n]}$. Next, compute a matrix $\mathbf{U}_i$ such that $\mathbf{c}_i \cdot r_{i,i} \cdot \mathbf{U}_i$ is HKZ-reduced, for all i. It is obvious that the complexity of the conversion algorithm is dominated by the second step. In order to compute $\mathbf{U}_i$, we compute HKZ-reduced basis of a certain d-dimensional lattice over $K_{\mathbb{R}}$, which terminates in $2^{O(d)} \cdot \mathsf{poly}(B')$, where B' is the maximum length of $\mathbf{c}_i \cdot r_{i,i}$. From the inner product in $K_{\mathbb{R}}$, the size B' corresponds to a polynomial in the input length of $\mathbf{c}_i \cdot r_{i,i}$ and $\log \Delta_K$. Thus, the total time complexity is $poly(B, \log \Delta_K) \cdot n \cdot 2^{O(d)}$. □

The output matrix $\mathbf{W}$ will be used as input for the lattice enumeration algorithm (Algorithm 3) with a specified size bound A. While $\mathbf{W}$ is not an integer matrix, the generalization of the enumeration algorithm to handle such inputs is natural and effective. This is because all operations required by the enumeration algorithm, such as computing norms and updating lattice points, are well-defined for any $\mathbb{Z}$-basis, though it originates from an integer lattice.

By substituting $\mathbf{W}$ as the input to Algorithm 3, the algorithm functions as intended in the generalized setting and successfully enumerates all points in the module M within the specified size bound A.

This natural generalization gives rise to the following module lattice enumeration algorithm (Algorithm 5).

The correctness of Algorithm 5 follows directly from the correctness of Algorithm 4 and Algorithm 3.

Algorithm 5: Module Lattice Enumeration Algorithm

Input: A pseudo-basis $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ of a module M and a bound A
Output: All vectors in M of which 2-norm is bounded by A.
1 Call $\mathbf{W} \leftarrow$ *Algorithm 4*$(\{\mathbf{c}_i\}_{i \le n}, ((I_i, \mathbf{b}_i))_{i \le n})$
2 Call $S \leftarrow$ *Algorithm 3*$(\mathbf{W}, A)$
3 **return** S

In the next section, we analyze the asymptotic performance of Algorithm 5 when the input pseudo-basis is a quasi-module-HKZ-reduced basis (Definition 3.2). For detailed results, refer to Theorem 3.4.

3.3 Analysis of the Lattice Enumeration Algorithm over Modules

Building on the definitions of modules introduced in Section 3.1, we present a new analysis of the lattice enumeration algorithm over modules.

The main goal of this subsection is to prove Theorem 3.4, which will be used to analyze the performance of the module-SVP and module-CVP algorithms. Although the theorem assumes that the input pseudo-basis of the module lattice enumeration algorithm is quasi-module-HKZ-reduced, as defined in Definition 3.2, the process for obtaining such a pseudo-basis will be discussed in the next section.

Throughout this section, we assume that the pseudo-basis $((I_i, \mathbf{b}_i))_{i \le n}$ is quasi-module-HKZ-reduced.

Theorem 3.4. *Let K be a number field of an extension degree d, and let $K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R}$. Let $M \subset K^n$ be a rank-n module. Suppose that a pseudo-basis $((I_i, \mathbf{b}_i))_{i \le n}$ of M is a quasi-module-HKZ-reduced pseudo-basis in Definition 3.2. Then, the module lattice enumeration algorithm in Algorithm 5 that takes as inputs $((I_i, \mathbf{b}_i))_{i \le n}$ and a bound A terminates in time polynomial in $O(\max_i N_i)$ up to a polynomial factor, where N_i is bounded by*

$$(2\sqrt{2\pi e})^i \cdot \sqrt{n}^{i' \ln\left(\frac{n-1}{ek}\right)} \cdot \sqrt{d}^{i' \cdot \left(\ln \frac{d}{i'}\right)} \cdot \sqrt{n}^{d \cdot k \cdot \ln \frac{n}{k}}.$$

Wosrt-bound assumption for the module-HKZ-reduced basis. As in the analysis of [10], we also assume the worst-case assumption that indicates the size of the shortest vector in the lattice equals to Minkowski's bound. This assumption provides the upper bound on the worst-case complexity of the algorithm.

We thus explain the worst-case bound for the module-HKZ-reduced basis. Let $((I_i, \mathbf{b}_i))_{i \in n}$ be the module-HKZ-reduced basis. We note that by definition of the module-HKZ-reduced, $\mathbf{b}_i^*$ is the shortest vector of the submodule generated by $\{(I_j, \mathbf{b}_j^*)\}_{j \ge i}$ and the rank-1 submodule generated by $(I_i, \mathbf{b}_i^*)$. By Minkowski's bound for two submodule, $\|\mathbf{b}_i^*\|$ satisfies two constraints:

$$\|\mathbf{b}_i^*\| \le \sqrt{(n-i+1) \cdot d} \cdot \Delta_K^{1/2d} \cdot \left(\prod_{j \ge i} \mathcal{N}(I_j \cdot \mathbf{b}_j^*)^{1/d} \right)^{\frac{1}{n-i+1}} \tag{4}$$

$$\|\mathbf{b}_i^*\| \leq \sqrt{d} \cdot \Delta_K^{1/2d} \cdot \mathcal{N}(I_i \cdot \mathbf{b}_i^*)^{1/d}. \tag{5}$$

Letting the first bound M_1 and the second bound M_2, respectively, we consider the following case: Suppose that $M_1 \geq M_2$. Then assuming the worst-bound for $\|\mathbf{b}_i^*\|$ means $\|\mathbf{b}_i^*\| = M_1$. At the same time, from the Equation (5), it holds that

$$M_1 = \|\mathbf{b}_i^*\| \leq M_2 \leq M_1.$$

Assuming $M_1 \leq M_2$ gives the same equality. Hence, assuming the worst-bound for module lattice means

$$\|\mathbf{b}_i^*\| = \sqrt{(n-i+1)\cdot d} \cdot \Delta_K^{1/2d} \cdot \left(\prod_{j \geq i} \mathcal{N}(I_j \cdot \mathbf{b}_j^*)^{1/d} \right)^{\frac{1}{n-i+1}} \tag{6}$$

$$= \sqrt{d} \cdot \Delta_K^{1/2d} \cdot \mathcal{N}(I_i \cdot \mathbf{b}_i^*)^{1/d}. \tag{7}$$

Assuming the worst-case bound, we prove the following result.

Lemma 3.5. *Let K be a number field of an extension degree d, and let $K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R}$. Let $M \subset K^n$ be a rank-n module. Suppose that a pseudo-basis $((I_i, \mathbf{b}_i))_{1 \leq i \leq n}$ is module-HKZ-reduced of a module M. Assuming the worst-case bound, we have*

$$\mathcal{N}(I_1 \cdot \mathbf{b}_1^*)^{1/d} \leq \mathcal{N}(I_i \cdot \mathbf{b}_i^*)^{1/d} \cdot \sqrt{n}^{\ln\left(\frac{n}{n-i+1}\right)}$$
$$\|\mathbf{b}_1^*\| \leq \|\mathbf{b}_i^*\| \cdot \sqrt{n}^{\ln\left(\frac{n}{n-i+1}\right)},$$

where $\{(I_i, \mathbf{b}_i^*)\}_{1 \leq i \leq n}$ is the Gram-Schmidt orthogonalization of the pseudo-basis. In particular, when $n = 2$, it holds that

$$\|\mathbf{b}_1^*\| = 2 \cdot \|\mathbf{b}_2^*\|.$$

Proof of Lemma 3.5. The proof is straightforward via definitions of module-HKZ-reduced and Minkowski's bound.

Rearranging the Equation (6), we have

$$\mathcal{N}(I_i \cdot \mathbf{b}_i^*)^{1/d} = \sqrt{(n-i+1)} \cdot \left(\prod_{j \geq i} \mathcal{N}(I_j \cdot \mathbf{b}_j^*)^{1/d} \right)^{\frac{1}{n-i+1}}. \tag{8}$$

This equation is similar to the relation of HKZ-reduced basis of the Euclidean lattice in Lemma 2.3, which states that $\|\vec{b}_i^*\| \leq \sqrt{\frac{n-i+5}{4}} \cdot \left(\prod_{j \geq i} \|\vec{b}_j^*\| \right)^{\frac{1}{n-i+1}}$ except for the constant. We adapt the previous analysis [10] in the modules in order to obtain $\mathcal{N}(I_1 \cdot \mathbf{b}_1^*)$ and $\mathcal{N}(I_i \cdot \mathbf{b}_i^*)$. More precisely, let $w_i = \ln \mathcal{N}(I_i, \mathbf{b}_i^*)$. Then, by the relation in Equation (8), it holds that

$$w_i = \sum_{j \geq i} \frac{w_j}{n-i+1} + \frac{\ln(n-i+1)^d}{2}.$$

Then, we have

$$(n-i+1)w_i = \sum_{j\geq i} w_j + \frac{(n-i+1)\cdot \ln(n-i+1)^d}{2}.$$

By substituting $i = i+1$, it gives another identity:

$$(n-i)w_{i+1} = \sum_{j\geq i+1} w_j + (n-i)\cdot \frac{\ln(n-i)^d}{2}.$$

Using the above two equations, we directly obtain the following relation:

$$w_i = w_{i+1} + \frac{d\ln\frac{n-i+1}{n-i}}{2} + \frac{\ln(n-i+1)^d}{2(n-i)}.$$

From the inductive definition of w_i, we have

$$w_i = w_n + \frac{d\ln(n-i+1)}{2} + \sum_{j=i}^{n-1} \frac{\ln(n-j+1)^d}{2(n-j)}$$

It means that

$$\frac{1}{d}w_1 - \frac{1}{d}w_i = \frac{\ln(\frac{n}{n-i+1})}{2} + \sum_{j=1}^{i-1} \frac{\ln(n-j+1)}{2(n-j)}.$$

Let $S_i = \frac{\ln(\frac{n}{n-i+1})}{2} + \sum_{j=1}^{i-1} \frac{\ln(n-j+1)}{2(n-j)}$. Then, we have $\exp(S_i) \leq \sqrt{n}^{\ln(\frac{n}{n-i+1})}$. We defer the computation at Appendix A. As a consequence, we have

$$\mathcal{N}(I_1\cdot\mathbf{b}_1^*)^{1/d} \leq \mathcal{N}(I_i\cdot\mathbf{b}_i^*)^{1/d}\cdot\sqrt{n}^{\ln(\frac{n}{n-i+1})},$$

which implies $\|\mathbf{b}_1^*\| \leq \|\mathbf{b}_i^*\|\cdot\sqrt{n}^{\ln(\frac{n}{n-i+1})}$ using Equation (7).

Note that the last equality is directly obtained by using the Equation (8) with $n=2$ and $i=1$.

$$\mathcal{N}(I_1\cdot\mathbf{b}_1^*)^{1/2d} = \sqrt{2}\cdot\mathcal{N}(I_2\cdot\mathbf{b}_2^*)^{1/2d}.$$

Applying Equation (7) gives the result $\|\mathbf{b}_1^*\| = 2\cdot\|\mathbf{b}_2^*\|$. This completes the proof. □

Proof of Theorem 3.4. We are now ready to prove the Theorem 3.4. It suffices to estimate the performance of the algorithm of which input is $\mathbb{Z}$-basis $\mathbf{W}$ and a size $\|\mathbf{b}_1\|$.

According to [10], given an input $(\mathbf{W}, \|\mathbf{w}_1^*\|)$ on Algorithm 3, the time complexity of the enumeration algorithm is dominated by $\max_i N_i$ up to a polynomial

factor, where N_i is the number of lattice points generated by $\tau_{n-i}(\mathbf{W})$ of size $\leq \|\mathbf{w}_1^*\|$. From the Gaussian heuristics, it is represented by

$$N_i := \frac{\sqrt{2\pi e}^i \|\mathbf{w}_1^*\|^i}{\sqrt{nd}^i \prod_{j \geq nd-i} \|\mathbf{w}_j^*\|}. \tag{9}$$

Therefore, the remaining part of this proof is to estimate an upper bound of N_i for every i.

We first remind that $\{(I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2)\}$ is module-HKZ reduced due to the definition of the quasi-module-HKZ-reduction (Definition 3.2). Since $\mathbf{w}_1$ and $\mathbf{w}_{d+1}$ is the shortest vector of rank-1 module $(I_1, \mathbf{b}_1)$ and $(I_2, \mathbf{b}_2^*)$, Lemma 3.5 also give the following relations:

$$\|\mathbf{w}_1^*\| = 2\|\mathbf{w}_{d+1}^*\|,$$

which directly gives another representation of N_i:

$$N_i = \frac{(2\sqrt{2\pi e})^i \cdot \|\mathbf{w}_{d+1}^*\|^i}{\sqrt{nd}^i \prod_{j \geq nd-i} \|\mathbf{w}_j^*\|}$$

Furthermore, $((I_i, \mathbf{b}_i^*))_{2 \leq i \leq n}$ is the module-HKZ-reduced with the relations $\|\mathbf{w}_{(i-1)d+1}\| = \|\mathbf{b}_i^*\|$. Applying the Lemma 3.5 with the worst-case bound directly gives the following relations:

$$\|\mathbf{w}_{d+1}^*\| \leq \|\mathbf{w}_{(i-1)d+1}^*\| \cdot \sqrt{n-1}^{\ln\left(\frac{n-1}{n-i}\right)} \leq \|\mathbf{w}_{(i-1)d+1}^*\| \cdot \sqrt{n}^{\ln\left(\frac{n-1}{n-i}\right)} \tag{10}$$

$$\|\mathbf{w}_{d+1}^*\| = \sqrt{d} \cdot \mathcal{N}(I_2 \cdot \mathbf{b}_2)^{1/d} \Delta_K^{1/2d} \tag{11}$$

Leveraging these relations, we compute the bound of N_i. Let $i = kd + i'$ for some k and i'. Then, it satisfies that

$$\begin{aligned} N_i &= \frac{(2\sqrt{2\pi e})^i \cdot \|\mathbf{w}_{d+1}^*\|^i}{\sqrt{nd}^i \prod_{j \geq nd-i} \|\mathbf{w}_j^*\|} \\ &= \underbrace{\frac{(2\sqrt{2\pi e})^i \cdot \|\mathbf{w}_{d+1}^*\|^{i'}}{\sqrt{nd}^i \prod_{nd-i \leq j < d\cdot(n-k)} \|\mathbf{w}_j^*\|}}_{A_i} \cdot \underbrace{\frac{\|\mathbf{w}_{d+1}^*\|^{d\cdot k}}{\prod_{j \geq d\cdot(n-k)} \|\mathbf{w}_j^*\|}}_{C_i}. \end{aligned}$$

That is, we split N_i into a product of two fractions. Intuitively, A_i indicates the profiles of a lattice generated by $\mathbf{c}_i \cdot r_{i,i} \cdot \mathbf{U}_i$ and C_i indicates the remaining term, respectively.

Using the Equation (10), A_i is replaced by

$$A_i = \frac{(2\sqrt{2\pi e})^i \cdot \|\mathbf{w}_{d\cdot(n-k-1)+1}^*\|^{i'} \cdot \sqrt{n}^{i' \ln\left(\frac{n-1}{k}\right)}}{\sqrt{nd}^i \prod_{nd-i \leq j < d\cdot(n-k)} \|\mathbf{w}_j^*\|}.$$

By the construction of $\mathbf{w}_j$, the basis $\{\mathbf{w}_j\}_{d\cdot(n-k-1)+1\le j<d\cdot(n-k)}$ is HKZ-reduced. We can thus apply Lemma 2.7 to the HKZ-reduced basis, which implies

$$A_i \le \frac{(2\sqrt{2\pi e})^i \cdot \sqrt{n}^{i'\ln\left(\frac{n-1}{k}\right)}}{\sqrt{nd}^i} \cdot \sqrt{d}^{i'\cdot\left(1+\ln\frac{d}{i'}\right)}.$$

On the other hand, due to the Theorem 3.3, we remind that

$$\mathcal{N}(I_i\cdot\mathbf{b}_i^*)\cdot\sqrt{\Delta_K} = \mathcal{N}(I_i\cdot r_{i,i})\cdot\sqrt{\Delta_K} = \prod_{(i-1)d+1\le j\le id}\|\mathbf{w}_j^*\|. \tag{12}$$

Plugging Equation (11), one can show that

$$\begin{aligned} C_i &= \frac{\|\mathbf{w}_{d+1}^*\|^{d\cdot k}}{\prod_{j\ge d\cdot(n-k)}\|\mathbf{w}_j^*\|} \\ &= \frac{\sqrt{d}^{d\cdot k}\cdot\Delta_K^{k/2}\cdot\mathcal{N}(I_2\cdot\mathbf{b}_2)^k}{\Delta_K^{k/2}\cdot\prod_{j\ge n-k}\mathcal{N}(I_j\cdot\mathbf{b}_j^*)} \\ &= \sqrt{d}^{d\cdot k}\cdot\left(\frac{\mathcal{N}(I_2\cdot\mathbf{b}_2)^{k/d}}{\prod_{j\ge n-k}\mathcal{N}(I_j\cdot\mathbf{b}_j^*)^{1/d}}\right)^d \end{aligned}$$

The results in Lemma 3.5 with $\{\mathcal{N}(I_i\cdot\mathbf{b}_i^*)^{1/d}\}_{2\le i\le n}$ show that $\{\mathcal{N}(I_i\cdot\mathbf{b}_i^*)^{1/d}\}_{2\le i\le n}$ follows the HKZ-reduced identity. It implies that applying the Lemma 2.7, this term is bounded by

$$C_i \le \sqrt{d}^{d\cdot k}\cdot\left(\sqrt{n}^{k\cdot\left(1+\ln\frac{n}{k}\right)}\right)^d.$$

Putting them together, we can give an upper bound of N_i:

$$\begin{aligned} N_i &\le \frac{(2\sqrt{2\pi e})^i\cdot\sqrt{n}^{i'\ln\left(\frac{n-1}{k}\right)}}{\sqrt{nd}^i}\cdot\sqrt{d}^{i'\cdot\left(1+\ln\frac{d}{i'}\right)}\cdot\sqrt{d}^{d\cdot k}\cdot\left(\sqrt{n}^{k\cdot\left(1+\ln\frac{n}{k}\right)}\right)^d \\ &= (2\sqrt{2\pi e})^i\cdot\sqrt{n}^{i'\ln\left(\frac{n-1}{ek}\right)}\cdot\sqrt{d}^{i'\cdot\left(\ln\frac{d}{i'}\right)}\cdot\sqrt{n}^{d\cdot k\cdot\ln\frac{n}{k}}. \end{aligned}$$

This completes the proof. □

4 Algorithm for Solving Module-SVP

In this section, we provide an algorithm that computes module-HKZ-reduced bases to solve Module-SVP. Then, based on the results of Theorem 3.4, we claim that our new analysis using module structures provides a more tight upper bound of the lattice enumeration algorithm over modules.

We first introduce a subroutine algorithm that obviously returns an HKZ-reduced basis.

Algorithm 6: HKZ-reduction over rank-2 module M

Input: A pseudo-basis $((I_i, \mathbf{b}_i))_{1\le i\le 2}$ of a module M
Output: An HKZ-reduced pseudo-basis of M

1 Call Algorithm 4 with the input $((I_i, \mathbf{b}_i))_{1\le i\le 2}$ and $\mathbb{Z}$-basis of I_i, and let $\mathbf{M}$ be an output.
2 HKZ-reduce on $\mathbf{M}$, and let $\mathbf{u}_1$ be an output.
3 Compute a projection of $\mathbf{M}$ onto $\mathbf{u}_1^{\perp}$, and let $\tau_{\mathbf{u}_1^{\perp}}(\mathbf{M})$ be an output.
4 HKZ-reduce on $\tau_{\mathbf{u}_1^{\perp}}(\mathbf{M})$, and let $\mathbf{u}_2$ be an output.
5 Extend the pseudo-basis $((I_2, \mathbf{u}_2))$ to the rank-2 module M using rational multiples of $\mathbf{u}_1$ satisfying $\lfloor \frac{\langle \mathbf{u}_1, \mathbf{u}_2\rangle_{K_{\mathbb{R}}}}{\langle \mathbf{u}_1, \mathbf{u}_1\rangle_{K_{\mathbb{R}}}} \rceil_R = 0$. Let $((I_i', \tilde{\mathbf{b}}_i))_{1\le i\le 2}$ be a new basis.
6 Call the algorithm of Lemma 2.12 with $((I_i, \tilde{\mathbf{b}}_i))_{1\le i\le 2}$ and $(\mathbf{u}_1, \mathbf{u}_2)$ as inputs, and let $((J_i, \overline{\mathbf{b}}_i))_{i\le 2}$ be an output.
7 **return** $((J_i, \overline{\mathbf{b}}_i))_{i\le 2}$.

To this end, we should clarify how we obtain the quasi-module-HKZ reduced pseudo-basis $((I_i, \mathbf{b}_i))_{i\le n}$ given the pseudo-basis $((J_i, \mathbf{d}_i))_{i\le n}$ of a module M. If possible, we directly apply Algorithm 5 with inputs $((I_i, \mathbf{b}_i))_{i\le n}$ and $\|\mathbf{b}_1\|$ to solve module-SVP. Indeed, the shortest vector of the outputs of Algorithm 5 would be a desired vector.

Fortunately, the process of converting to a quasi-module-HKZ-reduced pseudo-basis is inspired by the algorithm for computing a quasi-HKZ-reduced basis on Euclidean lattices. We emphasize again that the concept of a module-HKZ-reduced pseudo-basis, including Definition 3.2, is specifically designed to make it easier to adapt algorithms developed for similar concepts in Euclidean lattices.

The conversion algorithm is given in Algorithm 7. The important modification of the algorithm is to use the size-reduction of module vectors, corresponding to Line 4 of Algorithm 7, and call HKZ-reduction algorithm on a rank-2 module as a subroutine of algorithm.

Theorem 4.1. *Let K be a number field of an extension degree d. Let $M \subset K^n$ be a rank-n module. Let $((J_i, \mathbf{d}_i))_{1\le i\le n}$ be a pseudo-basis of a module M. Given the input pseudo-basis $((J_i, \mathbf{d}_i))_{1\le i\le n}$, Algorithm 7 returns a quasi-module-HKZ reduced basis. It terminates in $\#iter \cdot T_{\mathsf{MHKZ}}(n-1)$, where $T_{\mathsf{MHKZ}}(n-1)$ is the time complexity of Algorithm 8 with a rank $n-1$ module and $\#iter$ is the number of loop iterations is bounded by*

$$\frac{1}{\log\sqrt{3/2}} \cdot \log \frac{\Delta_K^{1/2d} \cdot \|\mathbf{b}_1\|}{\lambda_1(M)},$$

which is polynomial in $\log \Delta_K$, and the bit-length of inputs.

To prove the theorem, we require an additional inequality. Let $\lambda_1^I(M) = \inf\{\mathcal{N}(I \cdot \mathbf{v}) \ : \ I \cdot \mathbf{v} \in M \setminus \{0\}\}$. Then, the following inequality holds:

Algorithm 7: Quasi-Module-HKZ Conversion

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

1 Update module-HKZ-reduced pseudo-basis of $((I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2))$ by applying Algorithm 6 with an input $((I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2))$.
2 Compute the projections $\{\mathbf{b}'_i\}$, where $\mathbf{b}'_i = \mathbf{b}_i - \mu_{i,1} \cdot \mathbf{b}_1$ is orthogonal to $\mathbf{b}_1$.
3 Compute a module-HKZ-reduced basis:
$((I_2, \mathbf{b}'_2), \ldots, (I_n, \mathbf{b}'_n)) \leftarrow$ *Algorithm 8*$((I_2, \mathbf{b}'_2), \ldots, (I_n, \mathbf{b}'_n))$.
4 Extend the pseudo-basis $((I_2, \mathbf{b}'_2), \ldots, (I_n, \mathbf{b}'_n))$ to the module M using rational multiples of $\mathbf{b}_1$ satisfying $\lfloor \mu_{i,1} \rceil_R = 0$ for any $i > 1$. Let $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ be a new basis.
5 **if** $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ *is not quasi-module-HKZ-reduced* **then**
6 Update module-HKZ-reduced pseudo-basis of $((I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2))$ by applying Algorithm 6 with an input $((I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2))$.
7 Go back to Line 2.
8 **end if**
9 **return** $((I_i, \mathbf{b}_i))_{1 \le i \le n}$

Lemma 4.2. *For any rank-n module M, we have:*

$$d^{-d/2} \lambda_1(M)^d \Delta_K^{-1/2} \le \lambda_1^I(M) \le \lambda_1^{\mathcal{N}}(M).$$

Proof of Lemma 4.2. Let $\mathbf{v}$ be a module element of the minimal algebraic norm. Then it implies that $R \cdot \mathbf{v} \in M \setminus \{0\}$. Then by definition, we have

$$\lambda_1^I(M) \le \mathcal{N}(R \cdot \mathbf{v}) = \lambda^N(M).$$

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

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

The first equality follows by definition of the infimum. □

Proof of Theorem 4.1. We now prove a bound on the number of loop iterations after Line 4, which implies termination. For this proof, we use the worst-bound for the-module HKZ-reduced basis of a rank-2 module. We also note that without loss of generality, we assume that $\mathcal{N}(I_2 \cdot \mathbf{b}_2^*)^{1/d} \le \frac{1}{3}\mathcal{N}(I_1 \cdot \mathbf{b}_1^*)^{1/d}$ for basis after Line 4 at each iteration. If not, we can argue that

$$\begin{aligned} \|\mathbf{b}_1^*\| &\le \sqrt{d} \cdot \sqrt{\Delta_K}^{1/d} \cdot \mathcal{N}(I_1 \cdot \mathbf{b}_1^*)^{1/d} < 3\sqrt{d} \cdot \sqrt{\Delta_K}^{1/d} \cdot \mathcal{N}(I_2 \cdot \mathbf{b}_2^*)^{1/d} \\ &= 3\|\mathbf{b}_2^*\|. \end{aligned}$$

The last equality comes from the worst-bound of $\|\mathbf{b}_2^*\|$. More precisely, after the Line 4, $((I_i, \mathbf{b}'_i))_{2 \le i \le n}$ is the module-HKZ-reduced, we can ensure that $\sqrt{d} \cdot$

$\sqrt{\Delta_K}^{1/d} \cdot \mathcal{N}(I_2 \cdot \mathbf{b}_2^*)^{1/d} = \|\mathbf{b}_2^*\|$. Thus, we can conclude the shortest vector can be recovered through Algorithm 5, $((I_i, \mathbf{b}_i))_{1 \le i \le n}, 3\|\mathbf{b}_2^*\|)$. We note that $((I_i, \mathbf{b}_i^*))_{2 \le i \le n}$ is module-HKZ reduced, it gives the asymptotically same result with the Theorem 3.4. Since we aim at describing an algorithm for the module SVP through the quasi-module-HKZ, we assume that $\mathcal{N}(I_2 \cdot \mathbf{b}_2^*)^{1/d} \le \frac{1}{3}\mathcal{N}(I_1 \cdot \mathbf{b}_1^*)^{1/d}$.

We first suppose that the basis $M_2 = ((I_i, \mathbf{b}_i))_{1 \le i \le 2}$ after Line 4 is not module-HKZ lattice. It means that $\|\mathbf{b}_1\|$ is not the shortest vector of the rank-2 module lattice $M_2 := ((I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2))$. We now shot that $\lambda_1(M_2)$ decreases by a factor $\sqrt{\frac{2}{3}}$ at every iteration of the algorithm.

Let $\mathbf{c}_1$ be the shortest vector of M_2 and $((J_1, \mathbf{c}_1), (J_2, \mathbf{c}_2))$ be the module-HKZ-reduced basis of M_2. From the worst-bound argument, we have

$$\begin{aligned}\|\mathbf{c}_1\| &= \sqrt{2d} \cdot \sqrt{\Delta_K}^{1/d} \cdot \mathcal{N}(J_1 \cdot J_2 \cdot \mathbf{c}_1^* \cdot \mathbf{c}_2^*)^{1/2d} = \sqrt{d} \cdot \sqrt{\Delta_K}^{1/d} \cdot \mathcal{N}(J_1 \cdot \mathbf{c}_1^*)^{1/d} \\ &= \sqrt{2d} \cdot \sqrt{\Delta_K}^{1/d} \cdot \mathcal{N}(I_1 \cdot I_2 \cdot \mathbf{b}_1^* \cdot \mathbf{b}_2^*)^{1/2d}.\end{aligned}$$

The last equality is correct because $((I_1, \mathbf{b}_1), (I_2, \mathbf{b}_2))$ is also a pseudo-basis of M_2.

Rearranging the equation, we have

$$\begin{aligned}&\mathcal{N}(J_1 \cdot \mathbf{c}_1^*)^{1/d} = \sqrt{2} \cdot \mathcal{N}(J_1 \cdot J_2 \cdot \mathbf{c}_1^* \cdot \mathbf{c}_2^*)^{1/2d} \\ &= \sqrt{2} \cdot \mathcal{N}(I_1 \cdot I_2 \cdot \mathbf{b}_1^* \cdot \mathbf{b}_2^*)^{1/2d} \le \sqrt{\frac{2}{3}} \cdot \mathcal{N}(I_1 \cdot \mathbf{b}_1^*)^{1/d},\end{aligned}$$

where the last inequality comes from the above assumption $\mathcal{N}(I_2 \cdot \mathbf{b}_2^*)^{1/d} \le \frac{1}{3}\mathcal{N}(I_1 \cdot \mathbf{b}_1^*)^{1/d}$.

On the other hand, from the Lemma 4.2, this quantity at each stage has a lower bound:

$$\mathcal{N}(I_1 \cdot \mathbf{b}_1) \ge \lambda_1^I(M) \ge d^{-d/2}\lambda_1(M)^d \Delta_K^{-1/2}.$$

Combining the decrease rate with the lower bounds, this implies that the number of loop iterations is bounded by

$$\frac{1}{\log\sqrt{3/2}} \cdot \log \frac{\mathcal{N}(I_1 \cdot \mathbf{b}_1)^{1/d}}{d^{-1/2}\lambda_1(M)\Delta_K^{-1/2d}} \le \frac{1}{\log\sqrt{3/2}} \cdot \log \frac{\Delta_K^{1/2d} \cdot \|\mathbf{b}_1\|}{\lambda_1(M)}.$$

This completes the proof. □

The algorithm is also based on the module lattice enumeration algorithm. As in the previous algorithms, it is also inspired by the Kannan's HKZ reduced algorithm [10, 12].

Theorem 4.3. *Given a pseudo-basis $((I_i, \mathbf{b}_i))_{1 \le i \le n}$ of a module M, Algorithm 8 returns a module-HKZ-reduced basis $((I_i', \tilde{\mathbf{d}}_i))_{i \le n}$. This algorithm concludes after performing n iterations of Algorithm 9, Algorithm 1, and Algorithm 2.*

Algorithm 8: Module-HKZ-reduced pseudo-basis

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

1 Call $\mathbf{c}_1 \leftarrow$ *Algorithm 9*$(((I_i, \mathbf{b}_i))_{i\le n})$
2 Pick $n-1$ vectors, which is independent to $\mathbf{c}_1$ from $\{\mathbf{b}_i\}_{i\le n}$.
3 Letting these vectors as $\mathbf{c}_i$ for $2 \le i \le n$ with arbitrary order, call the algorithm of Lemma 2.12 with $((I_i, \mathbf{b}_i))_{i\le n}$ and $(\mathbf{c}_i)_{i\le n}$ as inputs, and let $((I'_i, \mathbf{d}_i))_{i\le n}$ denote the output.
4 Update the pseudo-basis by applying Algorithm 1 and Algorithm 2.
5 Compute the projections $\{\mathbf{d}'_i\}$, where $\mathbf{d}'_i = \mathbf{d}_i - \mu_{i,1}\cdot\mathbf{d}_1$ is orthogonal to $\mathbf{d}_1$ for every $i \ge 2$.
6 Update $((I'_i, \mathbf{d}'_i))_{2\le i\le n} \leftarrow$ Algorithm 8$(((I'_i, \mathbf{d}'_i))_{2\le i\le n})$.
7 Extend the basis $\{\mathbf{d}'_2, \dots, \mathbf{d}'_d\}$ to M using rational multiples of $\mathbf{d}_1$ satisfying $\lfloor \mu_{i,1} = \frac{\langle \mathbf{u}_i, \mathbf{u}_1^* \rangle_{K_{\mathbb{R}}}}{\langle \mathbf{u}_1^*, \mathbf{u}_1^* \rangle_{K_{\mathbb{R}}}} \rceil_R = 0$ for any $i > 1$. Let $\{\tilde{\mathbf{d}}_1, \dots, \tilde{\mathbf{d}}_d\}$ be a new basis.
8 **return** $(I'_i, \tilde{\mathbf{d}}_i)_{i\le n}$.

Proof. By the setup, $\mathbf{c}_1$ is the shortest vector in the module M. On the other hand, the pseudo basis of $((I'_i, \mathbf{d}_i))_{1\le i\le n}$, as guaranteed by the algorithm of Lemma 2.12, ensures that $\|\mathbf{d}_1\| = \|\mathbf{c}_1\|$ implying that $\mathbf{d}_1$ is also the shortest vector in the module $\mathbf{M}$. After update on line 6, $(I'_i, \mathbf{d}'_i)$ be the module-HKZ-reduced pseudo basis. Therefore, the resulting pseudo basis $(I'_i, \tilde{\mathbf{d}}_i)_{i\le n}$ is the module-HKZ-reduced pseudo basis by definition.

The number of iterations the algorithm is used is determined exactly by the dimension n. □

4.1 Module-SVP

This section provides a simple modification for solving module-SVP, rather than computing the module-HKZ-reduced basis. The algorithm is given by Algorithm 9.

Algorithm 9: Module-SVP

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

1 Compute $((I_i, \mathbf{b}_i))_{i\le n} \leftarrow$ *Algorithm 7*$(((I_i, \mathbf{b}_i))_{i\le n})$
2 Call $S \leftarrow$ *Algorithm 5*$(((I_i, \mathbf{b}_i))_{i\le n}, \|\mathbf{b}_1\|)$. Let $\mathbf{b}_0$ be the nonzero smallest vector in S.
3 **return** $\mathbf{b}_0$

Theorem 4.4. *Let $M \subset K^n$ be a module of a rank n, and $((I_i, \mathbf{b}_i))_{i\le n}$ be a pseudo-basis of M. Suppose that $\mathbb{Z}$-basis I_i and a $2d$-dimensional HKZ reduction algorithm are given. Setting $\alpha_K = 2^{2d}\cdot\Delta_K$ based on the HKZ reduction,*

Algorithm 9 returns the nonzero shortest vector of module M, and terminates in time

$$poly(n, d, B, \log \Delta_K) \cdot \left(\max_{0 \le i < n \cdot d} N_i + T_{HKZ} \right),$$

where B is the bit-length of inputs, T_{HKZ} is the time complexity of the given HKZ algorithm, and

$$N_i = (2\sqrt{2\pi e})^i \cdot \sqrt{n}^{i' \ln\left(\frac{n-1}{ek}\right)} \cdot \sqrt{d}^{i' \cdot \left(\ln \frac{d}{i'}\right)} \cdot \sqrt{n}^{d \cdot k \cdot \ln \frac{n}{k}} \quad \text{with} \quad i = d \cdot k + i'.$$

Proof of Theorem 4.4. Since Algorithm 5 returns all module elements of which size less than $\|\mathbf{b}_1\|$. Thus, by definition, $\mathbf{b}_0$ should be the nonzero smallest vector in M.

Moreover, it is also obvious that the Algorithm 6 equipped with the $2d$-dimensional HKZ algorithm guarantees $\gamma = 1$ in the result. Thus, with $\alpha_K = 2^{2 \cdot d} \cdot \Delta_K$, Algorithm 7 terminates in $\#iter \cdot (T_{MHKZ}(n-1) + T_{HKZ})$ with

$$\#iter = \frac{1}{\log \sqrt{3/2}} \cdot \log \frac{\Delta_K^{1/2d} \cdot \|\mathbf{b}_1\|}{\lambda_1(M)}$$

The time complexity of Algorithm 5 consists of Algorithm 4 and Algorithm 3. By the Theorem 3.3, the time complexity of Algorithm 4 is dominated by that of Algorithm 7.

On the other hand, because $T_{MHKZ}(n-1)$ employs the Algorithm 9 as a subroutine, $T_{MHKZ}(n-1)$ is bounded by *Algorithm* 3$(((I_i, \mathbf{b}_i))_{i \le n}, \|\mathbf{b}_1\|)$. It implies the asymptotic complexity of Algorithm 9 is exactly the same as that of Algorithm 5 except for $poly(n, d, B, \log \Delta_K) \cdot T_{HKZ}$. As a consequence, we conclude Algorithm 9 also terminates in time

$$poly(n, d, B, \log \Delta_K) \cdot \left(\max_{0 \le i < n \cdot d} N_i + T_{HKZ} \right)$$

using the proof of Theorem 3.4. □

Proposition 4.5. *Let $n \ln n = o(\ln d)$ and $R = \mathbb{Z}[x]/\langle x^d + 1 \rangle$, where d is a power of two and $M \subset K^n$, where $K = \mathbb{Q}[x]/\langle x^d + 1 \rangle$. Then, one can find the nonzero shortest vector $\mathbf{v} \in M$ in time $d^{\frac{d}{2e} + o(d)}$ up to a polynomial factor.*

Proof of Proposition 4.5. Since d is a power of two integer, the discriminant Δ_K of K equals to d^d. On the other hand, Theorem 4.4 shows the complexity of module-SVP is dominated by the term $\sqrt{d}^{i' \cdot \ln \frac{d}{i'}}$, which is maximized as $\sqrt{d}^{\frac{d}{e}}$ with $i' = \frac{d}{e}$. Consequently, the asymptotic time complexity of module-SVP is $d^{\frac{d}{2e} + o(d)}$. □

As a direct corollary of Proposition 4.5, we describe how to solve the ideal-SVP in the ring $\mathbb{Z}[x]/\langle x^t + 1 \rangle$, where t is a power of two. This setup is highly applicable in cryptographic contexts, including fully homomorphic encryption schemes [3].

Corollary 4.6. *Let I be an ideal over $\mathbb{Z}[x]/\langle x^t+1\rangle$, where t is a power of two. Then, one can solve the ideal SVP on I in time $e^{O(\frac{t}{2e}\ln\ln t)}$.*

Proof of Corollary 4.6. Solving ideal SVP on I can be treated as solving module-SVP in $(\mathbb{Z}[X]/\langle x^d+1\rangle)^{t/d}$. It allows to optimiz N_i through an appropriate choice of d. We first describe how to set i' and k. In the N_I, the terms $\sqrt{d}^{i'\cdot\ln\frac{d}{i'}}$ and $\sqrt{n}^{d\cdot k\cdot\ln\frac{n}{k}}$ are dominant of the cyclotomic ideal lattice. Under the setup, $i'=\frac{d}{e}$ and $k=\frac{n}{e}$ makes the N_i maximize with $N_i = d^{\frac{d}{2e}}\cdot n^{\frac{d\cdot n}{2e}}$.

We then set $d=\frac{t}{\log t}$ so that $d^{\frac{d}{2e}} = n^{\frac{d\cdot n}{2e}}$. Under the d setup, the complexity N_i becomes

$$e^{\left(\frac{t}{2e}\ln\ln t\right)+o(t)}.$$

This concludes the proof. □

For comparison, we briefly recall the result from [10]: Given an Euclidean lattice of dimension t, Kannan's SVP algorithm finds the shortest nonzero vector in the lattice in time $t^{t/(2e)+o(t)} = e^{t\ln t/(2e)+o(t)}$, up to a polynomial factor, while our algorithm terminates in $e^{t\ln\ln t/(2e)+o(t)}$.

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. Yuanmi Chen and Phong Q Nguyen. BKZ 2.0: Better lattice security estimates. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 1–20. Springer, 2011.
5. 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.
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. Ulrich Fincke and Michael Pohst. A procedure for determining algebraic integers of given norm. In *Computer Algebra: EUROCAL'83, European Computer Algebra Conference London, England, March 28–30, 1983 Proceedings*, pages 194–202. Springer, 1983.

9. 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.
10. Guillaume Hanrot and Damien Stehlé. Improved analysis of kannan's shortest lattice vector algorithm. In *Annual international cryptology conference*, pages 170–186. Springer, 2007.
11. 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.
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. Subhash Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM (JACM)*, 52(5):789–808, 2005.
14. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.
15. 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.
16. Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
17. 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.
18. 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.
19. 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.
20. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

A Upper bound of S_i

The purpose of this section is to present the estimation of S_i. Let $S_i = \frac{1}{2} \cdot \ln(\frac{n}{n-i+1}) + \sum_{j=1}^{i-1} \frac{\ln(n-j+1)}{2(n-j)}$ for $1 \leq i \leq n$. Our goal is to show that

$$\exp(S_i) \leq \sqrt{n}^{\ln(\frac{n}{n-i+1})}.$$

Consequently, we conclude

$$\mathcal{N}(I_1 \cdot \mathbf{b}_1^*)^{1/d} \leq \mathcal{N}(I_i \cdot \mathbf{b}_i^*)^{1/d} \cdot \sqrt{n}^{\ln\left(\frac{n}{n-i+1}\right)}.$$

By substituting $x = n - j$ to $\sum_{j=1}^{i-1} \frac{\ln(n-j+1)}{2(n-j)}$, it is approximately

$$\sum_{x=n-i+1}^{n-1} \frac{\ln(x)}{2x}.$$

Since the function $\frac{\ln(x)}{2x}$ is a decreasing function, the sum is less than

$$\frac{1}{2}\int_{x=n-i+1}^{n} \frac{\ln x}{x}\mathrm{d}x = \frac{1}{4}((\ln n)^2 - (\ln(n-i+1))^2).$$

Recall that

$$S_i = \frac{1}{2}\ln(\frac{n}{n-i+1}) + \sum_{j=1}^{i-1} \frac{\ln(n-j+1)}{2(n-j)}.$$

Let $\mathit{\Gamma} = \ln(n) - \ln(n-i+1) = \ln(\frac{n}{n-i+1})$. Then we have

$$(\ln n)^2 - (\ln(n-i+1))^2 = (\ln n + \ln(n-i+1)) \times \mathit{\Gamma}.$$

Hence, it holds

$$\sum_{j=1}^{i-1} \frac{\ln(n-j+1)}{2(n-j)} \leq \frac{1}{4}\left((\ln n + \ln(n-i+1))\mathit{\Gamma}\right) = \frac{1}{4}\ln(n(n-i+1))\mathit{\Gamma}.$$

Adding the extra term $\frac{1}{2}\mathit{\Gamma}$, one obtains

$$S_i \leq \mathit{\Gamma} \cdot \left(\frac{1}{2} + \frac{1}{4}\ln(n(n-i+1))\right).$$

Consequently, it satisfies

$$\exp(S_i) \leq \exp\left(\mathit{\Gamma} \cdot \left(\frac{1}{2} + \frac{1}{4}\ln(n(n-i+1))\right)\right).$$

On the other hand, $\ln(n(n-i+1)) \leq \ln n^2$. Thus, it is less than $\frac{1}{2}\ln(n)$. Hence,

$$\exp(S_i) \approx \exp(\frac{1}{2} \cdot \ln(n) \cdot \mathit{\Gamma}) = \exp(\frac{1}{2} \cdot \ln(n) \cdot \ln(\frac{n}{n-i+1})).$$

Since $\exp(\frac{1}{2}\ln(n)) = \sqrt{n}$, we finally get

$$\exp(S_i) \leq \sqrt{n}^{\ln(\frac{n}{n-i+1})},$$

establishing the claimed form.