Time vs Success Probability Tradeoff for SVP and BDD with Implications to LWE and SIS

Divesh Aggarwal[1] and Haoxiang Jin[2]

[1]National University of Singapore, dcsdiva@nus.edu.sg
[2]University of Illinois Urbana-Champaign, jin56@illinois.edu

Abstract

Worst-case to average-case reductions from lattice problems such as *GapSVP* and *Bounded Distance Decoding* (BDD) to the Learning with Errors (LWE) problem form the backbone of the security guarantees for lattice-based cryptography. However, these classic reductions are notoriously lossy: even assuming exponential hardness for worst-case lattice problems, they yield only subexponential lower bounds on the hardness of LWE. Recent work by Aggarwal, Leong, and Veliche (AMV, TCC'24) proposed a new perspective, quantifying hardness in terms of the *maximum success probability* achievable by any efficient (PPT) algorithm, and provided nearly tight reductions for LWE in the polynomial-time regime. Nevertheless, their framework is inherently limited to polynomial-time adversaries, leaving open the question of how the tradeoff between running time and success probability for lattice problems governs the concrete security of LWE and SIS against powerful, time-rich adversaries.

In this work, we address this gap by systematically analyzing and tightly characterizing the *time-success probability tradeoff* for SVP and BDD, focusing on algorithms that exploit the fine-grained structure of slide-reduced bases. We present new blockwise guessing algorithms for SVP and BDD that utilize small-dimension SVP and CVP oracles; by leveraging the consecutive-product properties of Slide Reduction, we obtain the tightest known lower bounds on the success probability as a function of time. Assuming that we cannot do much better than this, we conjecture that no algorithm can outperform this tradeoff—for any subexponential time bound $T(n) = 2^{o(n)}$, the success probability of solving worst-case SVP or BDD cannot exceed $2^{-\frac{n^2 \log\log T(n)}{c \log T(n)}}$ for some constant $c > 1$, up to polynomial factors.

Applying this conjecture, we derive sharply improved, modular worst-case to average-case reductions for LWE and SIS that are robust against all time-bounded adversaries, not just those restricted to polynomial time. Our results provide the first fine-grained, quantitative foundation for the bit-security of lattice-based cryptography across the full spectrum of adversarial resources, closing a key gap in both the theory and practice of cryptographic security reductions.

Keywords: Lattice problems · SVP · BDD · Slide Reduction · Time-Success Tradeoff · Worst-Case to Average-Case Reduction · LWE · SIS

1 Introduction

LATTICE CRYPTOGRAPHY AND THE LIMITS OF REDUCTIONS. Lattice problems such as the Shortest Vector Problem (SVP) and Bounded Distance Decoding (BDD) are foundational to both the theory and practice of post-quantum cryptography [Ajt96, MR04, Reg05, BV14, Gen09]. Their presumed hardness underpins the security of critical cryptographic primitives, especially the Learning With Errors (LWE) [Reg05] and Short Integer Solution

(SIS) [Ajt96, MR04] problems. Since Regev's breakthrough reduction [Reg05], a rich body of work has established worst-case to average-case connections from GapSVP and BDD to LWE, providing a theoretical bedrock for lattice-based cryptography [Reg05, Pei09, BLP+13, MR04].

Yet, a longstanding and pressing concern is that these reductions are *lossy*: even if we assume that no algorithm solves GapSVP or BDD in time 2^{cn}, the best known reductions only rule out LWE attacks running in time $2^{\sqrt{n}}$ or $2^{o(n)}$ for polynomial modulus [Reg05, Pei09, BLP+13]. In practical terms, this means that even strong conjectures about the hardness of lattice problems yield only mild security guarantees for the actual cryptosystems we hope to deploy. The gap between our confidence in the worst-case hardness of lattice problems and the average-case security of cryptosystems presents a fundamental challenge.

From Runtime to Success Probability. A recent advance by Aggarwal, Leong, and Veliche [AMV24] tackled this problem by shifting the measure of security from runtime to *maximum success probability*. Rather than asking whether an algorithm can solve a problem within a given time bound, they asked: "What is the highest success probability achievable by any probabilistic polynomial-time (PPT) algorithm?" This *success probability* viewpoint is closely related to unpredictability entropy [AM11], witness compression, and oracle complexity, and aligns with how "bit security" is interpreted in real-world cryptography [MW18]. For instance, when a cryptosystem is said to offer 256-bit security, what is typically meant is that no feasible adversary—say, running in 2^{60} steps—can succeed with probability exceeding 2^{-200}.

The framework of [AMV24] led to nearly tight reductions: assuming no PPT algorithm solves BDD with probability greater than $2^{-\Omega(n^2/\log n)}$, the same bound holds for LWE, up to polynomial factors. This brought the theory closer to practice, providing a much stronger and more meaningful foundation for cryptographic security.

Beyond Polynomial-Time: The Need for Fine-Grained Hardness. Despite this progress, an important limitation remained: [AMV24] and nearly all prior works restrict their analysis to polynomial-time adversaries. However, cryptography must protect against adversaries with far greater computational resources. The highest levels of cryptographic assurance require us to understand the optimal *tradeoff between time and success probability*: for any feasible time budget—even superpolynomial or subexponential—how much advantage can an adversary gain? The challenge is to move beyond the PPT-only lens, and to provide robust, concrete security guarantees that scale with adversarial resources.

Indeed, a majority of contemporary works on the fine-grained hardness of lattice problems [Kle00, HR07, Mic12, BGS17, ASD18, EK20, ABGS21, ALS21, ALNS20] have revealed that even modest improvements over known algorithms for SVP or BDD would have dramatic consequences for cryptographic security. However, explicitly quantifying this time-success tradeoff—and translating it into worst-case to average-case reductions for LWE and SIS—remains an open and central challenge.

Our Approach and Main Contributions. In this work, we address this challenge by extending the success-probability viewpoint to *all* time-bounded algorithms, and by providing a new, fine-grained analysis of lattice problems through the lens of slide-reduced bases [GN08, LN14, ALNS20, MW16]. Our approach is guided by the following steps:

- **Blockwise BKZ-Based Guessing Algorithm.** We design a new algorithm for solving SVP and BDD on n-dimensional lattices that leverages the structural properties of slide-reduced bases. By partitioning the basis into blocks and strategically invoking k-dimensional SVP and CVP (or BDD) oracles, our algorithm maximizes the overall success probability, exploiting the consecutive-product property of Gram-Schmidt norms unique to slide reduction [GN08, ALNS20].
- **Tight Time-Success Analysis.** We provide a tight quantitative analysis of the achievable success probability as a function of block size k and the time to solve k-dimensional problems. Our results yield the best known lower bounds for the SVP and BDD success probability in the worst case, and clarify the fundamental bottleneck for all attack strategies based on basis reduction.
- **A Time-Success Probability Conjecture.** Motivated by our analysis and the state of the art in lattice algorithms [Kan87, MV10a, MV10b, ADRS15, ADS15, ALNS20, ALS21, ACKS25], we conjecture—in the spirit of [BGS17, ASD18, ABGS21]—that this BKZ-based tradeoff is essentially optimal: that is, no algorithm running in (subexponential) time 2^k, where $k = o(n)$, can solve n-dimensional BDD or SVP with significantly better

success probability than our algorithm achieves. This leads to an explicit *time-success probability conjecture* for lattice problems, extending the unpredictability framework of [AMV24] and giving a modular, quantitative foundation for analyzing time-success tradeoffs. We acknowledge that our conjecture may seem optimistic or even too strong at first glance. However, this only serves to motivate future work: should an algorithm be discovered that outperforms our proposed bound, it would likely imply a breakthrough in lattice algorithms. Thus, whether the conjecture stands or falls, the process will deepen our understanding of the algorithmic landscape, making it a win-win for the community.

IMPLICATIONS FOR LWE AND SIS. With this conjecture in hand, we revisit the classical worst-case to average-case reductions [Reg05, Pei09, BLP+13, MR04], and show that our framework yields vastly improved and modular conclusions: the best achievable success probability for solving LWE or SIS is tightly governed by the time-success curve for BDD or SVP. Thus, even subexponential-time LWE attacks are ruled out, provided no such improvement is possible for BDD or SVP. Our results yield robust, transparent, and explicit bit-security guarantees for LWE and SIS against all time-bounded adversaries.

IMPACT AND BROADER SIGNIFICANCE. Our contributions close a key gap in the lattice cryptography literature. By integrating time-success tradeoffs into the analysis of lattice problems and by exploiting the fine structure of slide-reduced bases, we offer a new foundation for assessing and justifying the security of lattice-based cryptography at both the conceptual and practical levels. Our approach is fully modular: it separates cryptographic security from the performance of specific algorithms and instead ties it to evidence-based, sharply quantified hardness assumptions. This not only clarifies what is needed to threaten cryptosystem parameters, but also guides future research on both lattice algorithms and cryptographic reductions.

SUMMARY OF CONTRIBUTIONS.

- We introduce a new, blockwise BKZ-based algorithm for SVP/BDD that achieves the tightest known time-success tradeoff using k-dimensional oracles, and we realize the consecutive-product structure that matches BKZ's geometric guarantees.
- We formalize and motivate a time-success probability conjecture for worst-case lattice problems, and quantify the best possible tradeoff between adversarial time and success probability.
- We derive fine-grained, robust worst-case to average-case reductions for LWE and SIS, yielding explicit and strong security guarantees for all adversarial time budgets.

ORGANIZATION. Section 2 reviews the structure and properties of slide-reduced bases. Section 3 describes our blockwise guessing algorithm. Section 4 presents our time-success probability conjecture, and describes the consequences for LWE and SIS. We conclude with open problems and directions for further research.

1.1 Technical Overview

We first recall some details in Gama and Nguyen's slide reduction [GN08]. This technique partitions the basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{R}^{m \times n}$ into equal-sized, non-overlapping segments called "primal blocks" with each of length k (which divides n), denoted as the projected block $\mathbf{B}_{[ik+1,(i+1)k]}$. In addition, this method defines "dual blocks" $\mathbf{B}_{[ik+2,(i+1)k+1]}$, which are the primal blocks shifted one position to the right. The goal of the slide reduction is to enforce specific conditions on both primal and dual blocks simultaneously, i.e. make each primal block $\mathbf{B}_{[ik+1,(i+1)k]}$ HKZ-reduced and each dual block $\mathbf{B}_{[ik+2,(i+1)k+1]}$ DSVP-reduced at the same time. (For the definitions of HKZ and DSVP, please refer to Section 2.3.) We will call such basis $\mathbf{B}$ slide-reduced.

A slide-reduced basis $\mathbf{B}$ exhibits several useful relationships between the successive minima λ_1 and the norms of the Gram-Schmidt orthogonalized vectors $\mathbf{b}_i^*$. Since each block $\mathbf{B}_{[ik+1,(i+1)k]}$ is HKZ-reduced, we obtain a lower bound on the ratio $\mathrm{vol}(\mathbf{B}_{[ik+1,p]})^{1/(p-ik)} / \|\mathbf{b}_{ik+1}^*\|$ for any $p \in [ik+1, (i+1)k]$. This yields a connection between the *consecutive* product of the Gram-Schmidt norms $\|\mathbf{b}_j^*\|$ from $j = ik+1$ to p and the norm of the first vector in the block, $\|\mathbf{b}_{ik+1}^*\|$. Moreover, Theorem 1 from [GN08] shows that we can derive a lower bound on the ratio $\|\mathbf{b}_{(i+1)k+1}^*\| / \|\mathbf{b}_{ik+1}^*\|$ by combining the properties of the HKZ-reduced block $\mathbf{B}_{[ik+1,(i+1)k]}$ with those of the DSVP-reduced block $\mathbf{B}_{[ik+2,(i+1)k+1]}$, for all $i \leqslant n/k - 2$. This, in turn, leads to a lower bound on $\|\mathbf{b}_{ik+1}^*\| / \lambda_1$ for all $i \leqslant n/k - 1$ (see Section 2.3 for more

details). As a result, we can derive a desirable lower bound on the ratio

$$\frac{\|\mathbf{b}^*_{ik+1}\| \cdot \|\mathbf{b}^*_{ik+2}\| \cdots \|\mathbf{b}^*_p\|}{\lambda_1^{p-ik}} \tag{1}$$

for any $i \leqslant n/k-1$ and $p \in [ik+1,(i+1)k]$. In contrast, for a basis reduced only by the LLL algorithm [LLL82], the worst-case lower bound for this ratio is significantly weaker than that achieved by a slide-reduced basis. This bound plays a crucial role in our analysis of the success probability of our new algorithm for solving SVP or BDD instances.

We start guessing the SVP solution with a slide-reduced basis $\mathbf{B}$ as input.

We first would like to argue that a naive attempt won't work here. This naive attempt is that we sample a lattice vector $\mathbf{v}$ from the Gaussian distribution over the lattice $\mathcal{L}(\mathbf{B})$ with width $\sigma = O(\|\tilde{\mathbf{B}}\| \cdot \log(n))$ by the Gaussian sampling algorithm in [BLP+13], and compute the probability that $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$, which is

$$\frac{\rho_\sigma(\mathbf{v})}{\rho_\sigma(\mathcal{L})} \approx \frac{1}{\rho_\sigma(\mathcal{L})} \approx \frac{\mathrm{vol}(\mathcal{L})}{\sigma^n} = (C\log n)^{-n} \cdot \frac{\mathrm{vol}(\mathcal{L})}{\|\tilde{\mathbf{B}}\|^n}.$$

A slide-reduced basis $\mathbf{B}$ can only guarantee that $\|\mathbf{b}_1\| \leqslant k^{\frac{n}{k}\cdot(\frac{1}{2}+o(1))} \cdot \mathrm{vol}(\mathcal{L})^{\frac{1}{n}}$, and we can also assume $\|\mathbf{b}^*_n\| \leqslant \|\mathbf{b}_1\| \leqslant k^{\frac{n}{k}\cdot(\frac{1}{2}+o(1))} \cdot \mathrm{vol}(\mathcal{L})^{\frac{1}{n}}$ (if $\|\mathbf{b}^*_n\| > \|\mathbf{b}_1\|$, then the shortest vector won't involve $\mathbf{b}_n$). However, we cannot obtain the same constraint for any other $\|\mathbf{b}^*_i\|$, so we cannot have a desired upper bound on $\|\tilde{\mathbf{B}}\|$.

Now we describe our new guessing algorithm. Let the vector $\mathbf{v} = \alpha_1\mathbf{b}_1 + \cdots + \alpha_n\mathbf{b}_n$ be the targeted non-zero shortest vector of lattice $\mathcal{L}(\mathbf{B})$. Our algorithm is iterative, via guessing each coefficient $\alpha_j \in \mathbb{Z}$ from back to front (from $j = n$ to 1). In the iteration labeled j (i.e., the $n-j+1$-th iteration), assume we have already guessed the last $n-j$ coefficients $\tilde{\alpha}_p = \alpha_p$ $(p > j)$ correctly. A very straightforward way to guess α_j, is by uniformly choose $\tilde{\alpha}_j$, such that

$$\|\pi_j(\mathbf{v})\| = \left\|\tilde{\alpha}_j \cdot \pi_j(\mathbf{b}_j) + \alpha_{j+1} \cdot \pi_j(\mathbf{b}_{j+1}) + \cdots + \alpha_n \cdot \pi_j(\mathbf{b}_n)\right\| \leqslant \lambda_1(\mathbf{B}), \tag{2}$$

since applying the projection map π_j to $\mathbf{v}$ won't increase the norm. The inequality (2) implies that we can restrict the coefficient $\tilde{\alpha}_j$ within a range of length $2\lambda_1/\|\mathbf{b}^*_j\|$, which indicates that the probability of choosing $\tilde{\alpha}_j = \alpha_j$ successfully is approximately $\|\mathbf{b}^*_j\|/2\lambda_1$. However, if $\|\mathbf{b}^*_j\| > 2\lambda_1$, there is a unique choice of $\tilde{\alpha}$, and so the probability is exactly 1. This analysis shows that

$$\Pr[\text{we guess } \mathbf{v} \text{ successfully}] \geqslant \prod_{j:\|\mathbf{b}^*_j\| \leqslant 2\lambda_1} \frac{\|\mathbf{b}^*_j\|}{2\lambda_1}.$$

This partial product of terms makes it difficult to directly exploit the properties of the slide-reduced basis $\mathbf{B}$ since we only have reasonable lower bounds for the product of consecutive terms such as (1). To avoid this issue, we make the following modifications to this guessing algorithm.

Let the iterative algorithm arrive at the i-th block $[ik+1,(i+1)k]$. Due to the following three properties:

- The "breakpoint" occurs when $\|\mathbf{b}^*_j\| > 2\lambda_1$;
- $\mathbf{b}^*_j$ is the shortest vector in the projected block $\mathbf{B}_{[j,(i+1)k]}$;
- $\sum_{p=j}^{(i+1)k} \alpha_j \cdot \pi_j(\mathbf{b}_p)$ is a vector in the projected block $\mathbf{B}_{[j,(i+1)k]}$ within a distance no more than λ_1 from a point $-\sum_{p=(i+1)k+1}^{n} \pi_j(\mathbf{b}_p)$;

We observed that $(\mathbf{B}_{[j,(i+1)k]}, -\sum_{p=(i+1)k}^{n} \pi_j(\mathbf{b}_p))$ is a valid BDD instance with dimension no more than k. Therefore, we can apply a k-dim CVP (or BDD) oracle to this instance to fully recover the linear combination $\alpha_j, \cdots, \alpha_{(i+1)k}$. Moreover, we can always find the minimal $j \in [ik+1,(i+1)k]$, then the probability of guessing all the integer coefficients α_p $(p \in [ik+1,(i+1)k])$ correctly with probability

$$\Pr[\text{we guess coefficients in block } [ik+1,(i+1)k] \text{ successfully}] \geqslant \prod_{p=ik+1}^{j-1} \frac{\|\mathbf{b}^*_p\|}{2\lambda_1}, \tag{3}$$

which is a consecutive product, and we can bound it using the lower bound of (2).

The algorithm we described above needs knowledge of the non-zero shortest length $\lambda_1 := \lambda_1(\mathbf{B})$ of the lattice. Since a slide-reduced basis $\mathbf{B}$ is always LLL-reduced, we can have the length λ_1 within a range of ratio 2^n and guess an approximate value of λ_1 within a factor of 2 with probability $1/n$.

At last, we compute the product of (3), and obtain the approximate result $2^{-\frac{n^2 \log k}{2k}}$ as the success probability.

Guessing the BDD solution follows the same paradigm as that of guessing the SVP solution, since the goal of BDD is also to find an integer linear combination $\alpha_1, \cdots, \alpha_n \in \mathbb{Z}$ such that $\left\| \sum_i \alpha_i \mathbf{b}_i + \mathbf{c} \right\| \leqslant \frac{\lambda_1}{\gamma}$ for some target $\mathbf{c}$ and approximation factor γ. Guessing the SVP solution is a special case where we set $\gamma = 1$ and $\mathbf{c} = \mathbf{0}$.

2 Preliminaries

We denote column vectors $\mathbf{x} \in \mathbb{R}^m$ by bold lower-case letters. Matrices $\mathbf{B} \in \mathbb{R}^{m \times n}$ are denoted by bold upper-case letters, and we often think of a matrix as a list of column vectors, $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$. For a matrix $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ with n linearly independent columns, we write $\mathcal{L}(\mathbf{B}) := \{z_1\mathbf{b}_1 + \cdots + z_n\mathbf{b}_n \ : \ z_i \in \mathbb{Z}\}$ for the lattice generated by $\mathbf{B}$ and $\|\mathbf{B}\| = \max\{\|\mathbf{b}_1\|, \ldots, \|\mathbf{b}_n\|\}$ for the maximum norm of a matrix. We often implicitly assume that $m \geqslant n$ and that a basis matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ has rank n (i.e., that the columns of $\mathbf{B}$ are linearly independent). We use the notation $\log := \log_2$ to mean the logarithm with base two.

2.1 Lattices

The **minimum distance** of a lattice $\mathcal{L}$, denoted $\lambda_1(\mathcal{L})$, is the length of the non-zero shortest nonzero lattice vector:

$$\lambda_1(\mathcal{L}) = \min_{\mathbf{x} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{x}\|.$$

The **i-th successive minimum** is defined as the smallest λ_i such that $\lambda_i \mathcal{B}$ ($\mathcal{B}$ is the unit ball) contains i linearly independent lattice points:

$$\lambda_i(\mathcal{L}) = \min\{r : \dim(\mathrm{span}(\mathcal{L} \cap r\mathcal{B})) \geqslant i\}.$$

For any lattice $\mathcal{L}$, its **dual lattice** is

$$\mathcal{L}^\vee = \{\mathbf{w} \in \mathrm{span}(\mathcal{L}) : \ \langle \mathbf{w}, \mathbf{y} \rangle \in \mathbb{Z} \text{ for all } \mathbf{y} \in \mathcal{L}\} \ .$$

If $\mathbf{B} \in \mathbb{R}^{m \times n}$ is a basis of $\mathcal{L}$, then $\mathcal{L}^\vee$ has basis $\mathbf{B}^\vee := \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$, called the **dual basis** of $\mathbf{B}$. The **reversed dual basis** $\mathbf{B}^{-s}$ of $\mathbf{B}$ is simply $\mathbf{B}^\vee$ with its columns in reversed order [GHN06].

The **Shortest Vector Problem** (SVP) is the computational search problem in which the input is (a basis for) a lattice $\mathcal{L} \subseteq \mathbb{Z}^m$, and the goal is to output a non-zero lattice vector $\mathbf{y} \in \mathcal{L}$ with minimal length, $\|\mathbf{y}\| = \lambda_1(\mathcal{L})$. For $\delta \geqslant 1$, the δ-approximate variant of SVP (δ-SVP) is the relaxation of this problem in which any non-zero lattice vector $\mathbf{y} \in \mathcal{L} \setminus \{\mathbf{0}\}$ with $\|\mathbf{y}\| \leqslant \delta \cdot \lambda_1(\mathcal{L})$ is a valid solution.

δ-Hermite SVP (δ-HSVP, sometimes also called Minkowski SVP) is another computational search problem, which asks us to find a non-zero lattice vector $\mathbf{y} \in \mathcal{L} \setminus \{\mathbf{0}\}$ with $\|\mathbf{y}\| \leqslant \delta \cdot \mathrm{vol}(\mathcal{L})^{1/n}$, where $\mathrm{vol}(\mathcal{L}) := \det(\mathbf{B}^\top\mathbf{B})^{1/2}$ is the covolume of the lattice. **Hermite's constant** γ_n is (the square of) the minimal possible approximation factor that can be achieved in the worst case, i.e.,

$$\gamma_n := \sup \frac{\lambda_1(\mathcal{L})^2}{\mathrm{vol}(\mathcal{L})^{2/n}} \ ,$$

where the supremum is over lattices $\mathcal{L} \subset \mathbb{R}^n$ with full rank n. Hermite's constant is only known exactly for $1 \leqslant n \leqslant 8$ and $n = 24$, but it is known to be asymptotically linear in n, i.e., $\gamma_n = \Theta(n)$. The best asymptotical bounds known are from [MH13, CSB87]:

$$\frac{n + \log(\pi n)}{2\pi e} \leqslant \gamma_n \leqslant \frac{1.744n}{2\pi e} + o(n).$$

We now give the precise definitions for two other computational search problems. The **δ-Bounded Distance Decoding** (δ-BDD) for approximation factor $\delta > 1$ is the computational search problem that, on input a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$

and a target vector $\mathbf{c} \in \mathbb{R}^m$ such that $\mathrm{dist}(\mathcal{L}(\mathbf{B}), \mathbf{c}) \leqslant \frac{\lambda_1(\mathbf{B})}{2\delta}$, finds the lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ which is closest to $\mathbf{c}$. The **δ-Closest Vector Problem** (δ-CVP) for approximation factor $\delta \geqslant 1$ is the computational search problem where, on input a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and a target vector $\mathbf{c}$, the goal is to find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{c}\| \leqslant \delta \cdot \mathrm{dist}(\mathbf{B}, \mathbf{c})$.

2.2 Gram-Schmidt orthogonalization

For a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{R}^{m \times n}$, we associate a sequence of projections $\pi_i := \pi_{\{\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}\}^\perp}$. Here, for any set W $\pi_{W^\perp}$ means the orthogonal projection onto the subspace $W^\perp$ orthogonal to the subspace spanned by W. As in [GN08], $\mathbf{B}_{[i,j]}$ denotes the projected block $(\pi_i(\mathbf{b}_i), \pi_i(\mathbf{b}_{i+1}), \ldots, \pi_i(\mathbf{b}_j))$.

We also associate with $\mathbf{B}$ its Gram-Schmidt orthogonalization (GSO) $\mathbf{B}^* := (\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*)$, where $\mathbf{b}_i^* := \pi_i(\mathbf{b}_i) = \mathbf{b}_i - \sum_{j<i} \mu_{i,j} \mathbf{b}_j^*$, and $\mu_{i,j} = \langle \mathbf{b}_i, \mathbf{b}_j^* \rangle / \|\mathbf{b}_j^*\|^2$.

We say that $\mathbf{B}$ is **size-reduced** if $|\mu_{i,j}| \leq \frac{1}{2}$ for all $i \neq j$: then $\|\mathbf{B}\| \leq \sqrt{n}\|\mathbf{B}^*\|$. Transforming a basis into this form without modifying $\mathcal{L}(\mathbf{B})$ or $\mathbf{B}^*$ is called **size reduction**, and this can be done easily and efficiently.

2.3 Lattice basis reduction

LLL **reduction.** Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a size-reduced basis. For $\varepsilon \in [0, 1]$, we say that $\mathbf{B}$ is ε- LLL**-reduced** [LLL82] if every rank-two projected block $\mathbf{B}_{[i,i+1]}$ satisfies Lovász's condition: $\|\mathbf{b}_i^*\|^2 \leqslant (1+\varepsilon)\|\mu_{i+1,i}\mathbf{b}_i^* + \mathbf{b}_{i+1}^*\|^2$ for $1 < i \leqslant n$. For $\varepsilon \geqslant 1/\mathrm{poly}(n)$, one can efficiently compute an ε-LLL-reduced basis for a given lattice. Furthermore, an $1/3$-LLL-reduced basis $\mathbf{B}$ satisfies $2^{-\frac{n-1}{2}} \cdot \|\mathbf{b}_i\| \leqslant \lambda_i(\mathbf{B}) \leqslant 2^{\frac{i-1}{2}} \cdot \|\mathbf{b}_i^*\|$ for $1 < i \leqslant n$.

SVP **reduction and its extensions.** Let $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ be a basis of a lattice $\mathcal{L}$ and $\delta \geq 1$ be an approximation factor.

We say that $\mathbf{B}$ is δ-SVP**-reduced** if $\|\mathbf{b}_1\| \leqslant \delta \cdot \lambda_1(\mathcal{L})$. If the basis is a projected block $\mathbf{B}_{[i,j]}$, we have

$$\|\mathbf{b}_i^*\| \leqslant \delta\sqrt{\gamma_{j-i+1}} \cdot \mathrm{vol}(\mathbf{B}_{[i,j]})^{\frac{1}{j-i+1}}.$$

Similarly, we say that $\mathbf{B}$ is δ-HSVP**-reduced** if $\|\mathbf{b}_1\| \leqslant \delta \cdot \mathrm{vol}(\mathcal{L})^{1/n}$. It is easy to see that if $\mathbf{B}$ is a δ-SVP-reduced basis, then $\mathbf{B}$ is also a $\delta\sqrt{\gamma_n}$-HSVP-reduced basis.

We say that $\mathbf{B}$ is δ-HKZ**-reduced** if it is size-reduced, and $\|\mathbf{b}_i^*\| \leqslant \delta \cdot \lambda_1(\mathbf{B}_{[i,n]})$.

$\mathbf{B}$ is δ-DSVP**-reduced** [GN08] (where D stands for dual) if the reversed dual basis $\mathbf{B}^{-s}$ is δ-SVP-reduced and $\mathbf{B}$ is $\frac{1}{3}$-LLL-reduced. If the basis is a projected block $\mathbf{B}_{[i,j]}$, we have

$$\mathrm{vol}(\mathbf{B}_{[i,j]})^{\frac{1}{j-i+1}} \leqslant \delta\sqrt{\gamma_{j-i+1}} \cdot \|\mathbf{b}_j^*\|.$$

Similarly, we say that $\mathbf{B}$ is δ-DHSVP**-reduced** if $\mathbf{B}^{-s}$ is δ-HSVP-reduced. It is easy to see that if $\mathbf{B}$ is a δ-DSVP-reduced basis, then $\mathbf{B}$ is also a $\delta\sqrt{\gamma_n}$-DHSVP-reduced basis.

The existence of such δ-DSVP-reduced bases is guaranteed by a classical property of LLL that $\|\mathbf{b}_n^*\|$ never decreases during the LLL-reduction process [LLL82].

We can efficiently compute a δ-(D)SVP-reduced basis for a given rank n lattice $\mathcal{L} \subseteq \mathbb{Z}^m$ with access to an oracle for δ-SVP on lattices with rank at most n. Furthermore, given a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n) \in \mathbb{Z}^{m \times n}$ of $\mathcal{L}$ and an index $i \in [1, n-k+1]$, we can use a δ-SVP oracle for lattices with rank at most k to efficiently compute a size-reduced basis $\mathbf{C} = (\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, \mathbf{c}_i, \ldots, \mathbf{c}_{i+k-1}, \mathbf{b}_{i+k}, \ldots, \mathbf{b}_n)$ of $\mathcal{L}$ such that the block $\mathbf{C}_{[i,i+k-1]}$ is δ-SVP-reduced or δ-DSVP-reduced:

- If $\mathbf{C}_{[i,i+k-1]}$ is δ-SVP-reduced, the procedures in [GN08, MW16, LN19] equipped with δ-SVP-oracle ensure that $\|\mathbf{C}^*\| \leq \|\mathbf{B}^*\|$;
- If $\mathbf{C}_{[i,i+k-1]}$ is δ-DSVP-reduced, the inherent LLL reduction implies $\|\mathbf{C}^*\| \leq 2^k\|\mathbf{B}^*\|$. Indeed, the GSO of $\mathbf{C}_{[i,i+k-1]}$ satisfies
$$\|(\mathbf{C}_{[i,i+k-1]})^*\| \leq 2^{k/2}\lambda_k(\mathcal{L}(\mathbf{C}_{[i,i+k-1]}))$$
(by [LLL82, p. 518, Line 27]) and $\lambda_k(\mathcal{L}(\mathbf{C}_{[i,i+k-1]})) \leq \sqrt{k}\|\mathbf{B}^*\|$. Here, $\lambda_k(\cdot)$ denotes the k-th minimum.

With size-reduction, we can iteratively perform $\mathrm{poly}(n, \log \|\mathbf{B}\|)$ many such operations efficiently. In particular, doing so will not increase $\|\mathbf{B}^*\|$ by more than a factor of $2^{\mathrm{poly}(n,\log\|\mathbf{B}\|)}$, and therefore the same is true of $\|\mathbf{B}\|$. That is, all intermediate entries and the total cost during execution (excluding oracle queries) remain polynomially bounded in the initial input size; See, e.g., [GN08, LN14] for the evidence. Therefore, to bound the running time of basis reduction, it suffices to bound the number of calls to these block reduction subprocedures.

Twin reduction and gluing. We recall the definition of twin reduction from [GN08, ALNS20]: $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_{d+1}) \in \mathbb{Z}^{m\times(d+1)}$ is δ-**twin-reduced** if $\mathbf{B}_{[1,d]}$ is δ-HSVP**-reduced** and $\mathbf{B}_{[2,d+1]}$ is δ- DHSVP**-reduced**.

Fact 2.1 (Fact 2.1 [ALNS20])**.** *If* $\mathbf{B} := (\mathbf{b}_1, \ldots, \mathbf{b}_{d+1}) \in \mathbb{Z}^{m\times(d+1)}$ *is* δ*-twin-reduced, then*

$$\|\mathbf{b}_1\| \le \delta^{2d/(d-1)} \|\mathbf{b}^*_{d+1}\| \,. \tag{4}$$

Furthermore,

$$\delta^{-d/(d-1)} \|\mathbf{b}_1\| \le \mathrm{vol}(\mathbf{B})^{1/(d+1)} \le \delta^{d/(d-1)} \|\mathbf{b}^*_{d+1}\| \,. \tag{5}$$

Slide **reduction.** We say a basis $\mathbf{B} \in \mathbb{Z}^{m\times n}$, where $n = kt$, is (δ, k)-slide-reduced if $\mathbf{B}_{[k(i-1)+1,ki]}$ is δ-HKZ-reduced for any $i \in [1, t]$ and $\mathbf{B}_{[k(i-1)+2,ki+1]}$ is δ-DSVP-reduced for any $i \in [1, t-1]$. Due to the definition of Hermite factor and the relationship between SVP reduction and HSVP reduction, between DSVP reduction and DHSVP reduction, we have the following fact:

Fact 2.2. *Let* $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n) \in \mathbb{Z}^{m\times n}$ *be a integer basis and* $\lambda_1 = \lambda_1(\mathbf{B})$ *be the shortest length of this lattice. If* $\mathbf{B}$ *is* (δ, k)-slide*-reduced and* $n = kt$*, we have*

$$\lambda_1 \leqslant \|\mathbf{b}^*_1\| \leqslant (\delta^2\gamma_k)^{\frac{k}{k-1}} \cdot \|\mathbf{b}^*_{k+1}\| \leqslant \cdots \leqslant (\delta^2\gamma_k)^{\frac{k(t-1)}{k-1}} \cdot \|\mathbf{b}^*_{k(t-1)+1}\|.$$

Proof. For any $i \in [t]$, since $\mathbf{B}_{[k(i-1)+1,ki]}$ is δ-HKZ-reduced, we have $\mathbf{B}_{[k(i-1)+1,ki]}$ is δ-SVP-reduced, then $\mathbf{B}_{[k(i-1)+1,ki]}$ is also $\delta\sqrt{\gamma_k}$-HSVP-reduced. For any $i \in [t-1]$, since $\mathbf{B}_{[k(i-1)+2,ki+1]}$ is δ-DSVP-reduced, $\mathbf{B}_{[k(i-1)+2,ki+1]}$ is also $\delta\sqrt{\gamma_k}$-DHSVP-reduced. Therefore, for any $i \in [t-1]$, $\mathbf{B}_{k(i-1)+1,ki+1}$ is δ-**twin-reduced**, which proves the $i+1$-th inequality of the statement. □

2.4 Gaussian Measures

We define a Gaussian function centered in $\mathbf{c} \in \mathbb{R}^n$ with Gaussian parameter σ to be $\rho_{\sigma,\mathbf{c}} : \mathbb{R}^n \to (0, 1]$, such that

$$\rho_{\sigma,\mathbf{c}}(\mathbf{x}) = \exp\left(-\pi \|(\mathbf{x}-\mathbf{c})/\sigma\|^2\right).$$

Since the total measure associated to $\rho_{\sigma,\mathbf{c}}$ is $\int_{\mathbf{x}\in\mathbb{R}^n} \rho_{\sigma,\mathbf{c}}(\mathbf{x})d\mathbf{x} = \sigma^n$, we can define the continuous Gaussian distribution centered in $\mathbf{c}$ with Gaussian parameter σ by its probability density function $D_{\sigma,\mathbf{c}} : \mathbb{R}^n \to (0, 1]$, such that

$$D_{\sigma,\mathbf{c}}(\mathbf{x}) = \sigma^{-n} \cdot \exp\left(-\pi \|(\mathbf{x}-\mathbf{c})/\sigma\|^2\right).$$

For any countable subset $A \subseteq \mathbb{R}^n$, we write $\rho_{\sigma,\mathbf{c}}(A) = \sum_{\mathbf{x}\in A} \rho_\sigma(A)$. For any countable subset A for which $\rho_s(A)$ converges, we define the **discrete Gaussian distribution** over A via its probability density function $D_{\sigma,\mathbf{c}} : A \to (0, 1]$, such that

$$D_{A,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{x})}{\rho_{\sigma,\mathbf{c}}(A)}.$$

We will omit σ and $\mathbf{c}$ in the subscript when $\sigma = 1$ and $\mathbf{c} = \mathbf{0}$ respectively.

2.5 Smoothing Parameter

For lattice $\mathcal{L} \subseteq \mathbb{R}^n$ and any $\varepsilon > 0$, the **smoothing parameter** [MR04] $\eta_\varepsilon(\mathcal{L})$ is defined as the smallest real $\sigma > 0$ such that $\rho_{1/\sigma}(\mathcal{L}^\vee) \leqslant 1+\varepsilon$. Regarding smoothing parameters, we have the following useful bounds and properties:

Lemma 2.3 (Lemma 3.3 in [MR04])**.** *For an n-dimensional lattice $\mathcal{L}$, $\varepsilon > 0$, we have*

$$\eta_\varepsilon(\mathcal{L}) \leqslant \lambda_n(\mathcal{L}) \cdot \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}}.$$

Lemma 2.4 (Claim 2.13 in [Reg05])**.** *For an n-dimensional lattice $\mathcal{L}$, $\varepsilon > 0$, we have*

$$\eta_\varepsilon(\mathcal{L}) \geqslant \frac{\lambda_n(\mathcal{L})}{n} \cdot \sqrt{\frac{\ln(1/\varepsilon)}{\pi}}.$$

Lemma 2.5 (A Particular Case of Lemma 2.9 in [MP12])**.** *Let $\mathcal{L} \in \mathbb{R}^n$ be a lattice and $\sigma \geqslant \eta_\varepsilon(\mathcal{L})$ for some $\varepsilon \in (0.1)$. Let $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_m) \in \mathbb{R}^{n\times m}$ be a random matrix where each column vector $\mathbf{x}_i$ follows $D_{\mathcal{L}+\mathbf{y}_i,\sigma}$ with arbitrary center $\mathbf{y}_i$ independently. There exists a universal constant $C > 0$, for any $t \geqslant 0$, we have*

$$\Pr_{\mathbf{X}}\left[s_1(\mathbf{X}) \leqslant C \cdot \sigma \cdot (\sqrt{m}+\sqrt{n}+t)\right] \geqslant 1 - 2 \cdot \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^m \cdot \exp(-\pi t^2).$$

2.6 Average-Case Lattice Problems

Here we recall two major average-case problems in lattices.

The **Short Integer Solution** $\mathsf{SIS}_{n,m,q,\beta}$ problem [Ajt96, MR04] is defined as follows: given an integer q, a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a real $\beta > 0$, find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m/\{\mathbf{0}\}$ such that $\mathbf{Az} = \mathbf{0} \bmod q$ and $\|\mathbf{z}\| \leqslant \beta$.

The **Learning with Errors** $\mathsf{LWE}_{n,m,q,\chi}$ problem [Reg05, Pei09, BLP+13] is defined as follows: given an integer q, a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and a error distribution χ with support $\mathsf{Supp}(\chi) \subseteq \mathbb{Z}$, find $\mathbf{s} \in \mathbb{Z}_q^n$ from LWE samples $(\mathbf{A}, \mathbf{b}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)$ where $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{e} \leftarrow \chi^m$.

3 Solving Lattice Problems in Limited Time

In this section, we will present two blockwise algorithms for guessing the SVP and BDD solutions in polynomial time, given an (approximate) SVP and CVP (or BDD) oracle. As we stated in the section Technical Overview, these two algorithms actually follow the same strategy, which is to take full advantage of the blockwise property of a slide-reduced basis, and guess the coefficients of the solution block by block from back to front.

3.1 Time-probability Tradeoff for Solving SVP

We now present our blockwise algorithm for guessing the shortest vector in an n-dimensional lattice, given input on a (δ_1,k)-slide-reduced basis and a δ_2-CVP (or δ_2-BDD) oracle.

Theorem 3.1. *Let n and $m \geqslant n$ be dimension parameters, $k = o(n)$ and $k = \Omega(\log n)$ be the number of block size and $\delta_1, \delta_2 \geqslant 1$ be some approximation factor. Let $\mathbf{B} \in \mathbb{Z}^{m\times n}$ be a lattice basis. Given a δ_1-SVP oracle and a δ_2-CVP (or δ_2-BDD) oracle both for lattices with rank at most k, there exists a probabilistic polynomial time algorithm, such that it outputs a shortest vector in lattice $\mathcal{L}(\mathbf{B})$ with probability at least $\delta_1^{-\frac{n^2}{k}\cdot(1+o(1))} \cdot \delta_2^{-n} \cdot k^{-\frac{n^2}{k}\cdot(\frac{1}{2}+o(1))}$.*

Specifically, if the SVP oracle and CVP oracle are non-approximate oracles, i.e. $\delta_1 = \delta_2 = 1$, there exists a ppt algorithm which outputs a shortest vector in lattice $\mathcal{L}(\mathbf{B})$ with probability at least $2^{-\frac{n^2\log k}{k}\cdot(\frac{1}{2}+o(1))}$.

Algorithm 1 A Blockwise BKZ-Based Algorithm for Guessing the SVP Solution

Input: A block size $k \geq 2$, number of blocks t, a (δ_1, k)-slide-reduced basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{Z}^{m \times n}$, and access to a δ_2-CVP (or δ_2-BDD) oracle for lattices with rank no more than k.

Output: A shortest vector of $\mathcal{L}(\mathbf{B})$.

Sample $\tilde{\lambda}_1 \leftarrow \left\{2^{-n+1}\|\mathbf{b}_1\|, 2^{-n+2}\|\mathbf{b}_1\|, \cdots, \|\mathbf{b}_1\|\right\}$ uniformly at random.
Sample $\tilde{t} \leftarrow \{1, 2, \cdots, t\}$ uniformly at random.
Initiate $\tilde{\mathbf{v}} \leftarrow \mathbf{0} \in \mathbb{Z}^m$.
for $\ell = \tilde{t}$ to 1 **do**
  if $\exists$ integer $j \in [k(\ell-1)+1, k\ell]$, s.t. $\|\mathbf{b}_j^*\| > 2\delta_1\delta_2 \cdot \tilde{\lambda}_1$ **then**
    Let j' be the smallest integer among such j.
    Call the CVP oracle on input lattice $\mathcal{L}(\mathbf{B}_{[j',k\ell]})$ and target vector $-\pi_{j'}(\tilde{\mathbf{v}})$, and let $(\tilde{\alpha}_{j'}, \tilde{\alpha}_{j'+1}, \cdots, \tilde{\alpha}_{k\ell})$ be the output linear combination.
    Compute $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + \tilde{\alpha}_{j'}\mathbf{b}_{j'} + \tilde{\alpha}_{j'+1}\mathbf{b}_{j'+1} + \cdots + \tilde{\alpha}_{k\ell}\mathbf{b}_{k\ell}$.
    $c \leftarrow j' - 1$.
  else
    $c \leftarrow k\ell$.
  end if
  for $i = c$ to $k(\ell-1)+1$ **do**
    Sample integer $\tilde{\alpha}_i \leftarrow \left[-\frac{\tilde{\lambda}_1}{\|\mathbf{b}_i^*\|} - \frac{\langle\tilde{\mathbf{v}}, \mathbf{b}_i^*\rangle}{\|\mathbf{b}_i^*\|^2}, \frac{\tilde{\lambda}_1}{\|\mathbf{b}_i^*\|} - \frac{\langle\tilde{\mathbf{v}}, \mathbf{b}_i^*\rangle}{\|\mathbf{b}_i^*\|^2}\right]$ uniformly at random.
    Compute $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + \tilde{\alpha}_i\mathbf{b}_i$.
  end for
end for
return $\mathbf{v}$.

Proof. W.l.o.g., we can assume $k \mid n$. Otherwise, we can always pad the basis $\mathbf{B}$ so that it has rank a multiple of k. Furthermore, changing n to $k \cdot \lceil n/k \rceil$ does not influence the asymptotic time complexity.

From [GN08, ALNS20], we can first call the k-dim δ_1-SVP oracle polynomial times to transform the original basis to a (δ_1, k)-slide-reduced basis. Therefore, we can always assume the basis $\mathbf{B}$ is (δ_1, k)-slide-reduced.

With a slide-reduced basis, we next run and analyse Algorithm 1.

Let $\mathbf{v} = \sum_{i \in [n]} \alpha_i \mathbf{b}_i$, where $\alpha_i \in \mathbb{Z}$ for $i \in [n]$, be the shortest vector in the whole lattice $\mathcal{L}(\mathbf{B})$, and $\lambda_1 := \lambda_1(\mathbf{B})$ be the shortest length, s.t. $\|\mathbf{v}\| = \lambda_1$.

In step 1, we guess an approximate value of the shortest length λ_1. Since $\mathbf{B}$ is a 1/3-LLL-reduced basis, we have $2^{-\frac{n-1}{2}} \|\mathbf{b}_1\| \leqslant \lambda_1 \leqslant \|\mathbf{b}_1\|$, and this indicates that there exists unique $\lambda' \in \{2^{-i}\|\mathbf{b}_1\| : i \in [0, n-1] \cap \mathbb{Z}\}$, s.t. $\lambda_1 \leqslant \lambda' < 2\lambda_1$. Therefore, we have $\Pr\left[\lambda_1 \leqslant \tilde{\lambda}_1 < 2\lambda_1\right] = \Pr\left[\tilde{\lambda}_1 = \lambda'_1\right] = 1/n$.

In step 2, we guess the maximal index $t' \in [t]$ such that $\pi_{k(t'-1)+1}(\mathbf{v}) \neq 0$. The probability of guessing correctly is exactly $1/t$. Guessing correctly $\tilde{t} = t'$ indicates that the coefficients in the last $t - \tilde{t}$ blocks are all 0, i.e.

$$\alpha_{\tilde{t}k+1} = \tilde{\alpha}_{\tilde{t}k+1} = \cdots = \alpha_n = \tilde{\alpha}_n = 0.$$

Next, we analyze the probability of guessing each α_i correctly, conditioned on $\lambda_1 \leqslant \tilde{\lambda}_1 < 2\lambda_1$, and $\tilde{t}$ is the maximal index such that $\pi_{k(\tilde{t}-1)+1}(\mathbf{v}) \neq 0$.

In the following steps, we guess the coefficients block by block and compute the probability of guessing all coefficients correctly as:

$$\begin{aligned}
\Pr[\tilde{\mathbf{v}} = \mathbf{v}] &\geqslant \frac{1}{n^2} \cdot \Pr\left[\mathsf{E}_1 \wedge \mathsf{E}_2 \wedge \cdots \wedge \mathsf{E}_{\tilde{t}} \mid \tilde{\lambda}_1 \in [\lambda_1, 2\lambda_1) \wedge \tilde{t} = t'\right] \\
&= \frac{1}{n^2} \cdot \prod_{\ell \in [\tilde{t}]} \Pr\left[\mathsf{E}_\ell \mid \mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_{\tilde{t}} \wedge \left(\tilde{\lambda}_1 \in [\lambda_1, 2\lambda_1)\right) \wedge (\tilde{t} = t')\right] \qquad (6)
\end{aligned}$$

where the E_ℓ is defined as the event of guessing all the coefficients in ℓ-th block successfully, i.e.

$$\left(\tilde{\alpha}_{k(\ell-1)+1} = \alpha_{k(\ell-1)+1}\right) \wedge \left(\tilde{\alpha}_{k(\ell-1)+2} = \alpha_{k(\ell-1)+2}\right) \wedge \cdots \wedge \left(\tilde{\alpha}_{k\ell} = \alpha_{k\ell}\right).$$

We next omit the condition of $\tilde{\lambda}_1 \in [\lambda_1, 2\lambda_1)$ and $\tilde{t} = t'$ when computing probability.

When the large loop comes to the ℓ-th block, we first check whether there exists a GS-orthogonalized vector $\mathbf{b}_j^*$ in the ℓ-th block, such that the length far exceeds $\tilde{\lambda}_1$. We let j' be the smallest index among such j, then we use the following claim to prove that the integers returned by the δ_2-CVP call in step 7 always satisfies $\tilde{\alpha}_i = \alpha_i$ for all $i = j', \cdots, k\ell$.

Claim 3.2. *Let the projected block $\mathbf{B}_{[k(\ell-1)+1,k\ell]}$ be δ_1-HKZ-reduced and $\|\mathbf{b}_j^*\| > 2\delta_1\delta_2 \cdot \lambda_1$ for some $j \in [k(\ell-1)+1, k\ell]$. Conditioned on $\tilde{\alpha}_i = \alpha_i$ for $i = k\ell+1, \cdots, n$, the δ_2-CVP (or δ_2-BDD) oracle, on input input lattice $\mathbf{B}_{[j,k\ell]}$ and target vector $-\pi_j\left(\sum_{i=k\ell+1}^n \alpha_i \mathbf{b}_j\right)$, returns linear combination $\alpha_j, \cdots, \alpha_{k\ell}$ with success probability 1.*

Proof. We only need to prove that $(\mathbf{B}_{[j,k\ell]}, -\pi_j(\sum_{i=k\ell+1}^n \alpha_i \mathbf{b}_i))$ is a valid δ_2-BDD instance, and the closest vector to the target is $\pi_j(\sum_{i=j}^{k\ell} \alpha_i \mathbf{b}_i) \in \mathcal{L}(\mathbf{B}_{[j,k\ell]})$.

Due to the property of δ_1-HKZ-reduced basis $\mathbf{B}_{[k(\ell-1)+1,k\ell]}$, $\|\mathbf{b}_j^*\| \leqslant \delta_1 \cdot \lambda_1\left(\mathbf{B}_{[j,k\ell]}\right)$, then we have

$$\left\| \sum_{i=j}^{k\ell} \alpha_i \pi_j(\mathbf{b}_i) + \sum_{i=k\ell+1}^{n} \alpha_i \pi_j(\mathbf{b}_i) \right\| = \|\pi_j(\mathbf{v})\| \leqslant \lambda_1 < \frac{\|\mathbf{b}_j^*\|}{2\delta_1\delta_2} \leqslant \frac{\lambda_1\left(\mathbf{B}_{[j,k\ell]}\right)}{2\delta_2},$$

which completes the proof. □

Claim 3.2 shows that in step 7,

$$\Pr[\tilde{\alpha}_{c+1} = \alpha_{c+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell} \mid \mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_t] = 1.$$

Next, we compute the probability of guessing each $\tilde{\alpha}_i = \alpha_i$ correctly conditioned on $\tilde{\alpha}_{i+1} = \alpha_{i+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell}$ and $\mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_t$, i.e. in step 14, $\tilde{\mathbf{v}} = \sum_{j=i+1}^n \alpha_j \mathbf{b}_j$, for all $i \in [k(\ell-1)+1, c]$.

Claim 3.3. *For all $i \in [k(\ell-1)+1, c]$, we have*

$$\Pr[\alpha_i = \tilde{\alpha}_i \mid \alpha_{i+1} = \tilde{\alpha}_{i+1} \wedge \cdots \wedge \alpha_n = \tilde{\alpha}_n] \geqslant \frac{\|\mathbf{b}_i^*\|}{2\tilde{\lambda}_1 + \|\mathbf{b}_i^*\|}.$$

Proof. Project $\mathbf{v}$ onto the direction of $\mathbf{b}_i^*$, we get

$$\pi_{\mathbf{b}_i^*}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2} \cdot \mathbf{b}_i^* = \left(\alpha_i + \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}\right) \cdot \mathbf{b}_i^*$$

And we know that the length of this projection is not larger than λ_1, so we have

$$\left|\alpha_i + \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}\right| \cdot \|\mathbf{b}_i^*\| \leqslant \lambda_1 \leqslant \tilde{\lambda}_1$$

which indicates that

$$\alpha_i \in \left[-\frac{\tilde{\lambda}_1}{\|\mathbf{b}_i^*\|} - \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}, \frac{\tilde{\lambda}_1}{\|\mathbf{b}_i^*\|} - \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}\right].$$

Therefore, the probability of correcting guessing $\tilde{\alpha}_i = \alpha_i$ is no less than $\frac{1}{2\tilde{\lambda}_1/\|\mathbf{b}_i^*\|+1}$, which completes the proof. □

Therefore, we have (here we omit the condition $\mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_{\tilde{\ell}}$),

$$\begin{aligned}
&\Pr[\mathsf{E}_\ell] \\
&= \Pr[\tilde{\alpha}_{c+1} = \alpha_{c+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell}] \cdot \prod_{i=k(\ell-1)+1}^{c} \Pr[\tilde{\alpha}_i = \alpha_i \mid \tilde{\alpha}_{i+1} = \alpha_{i+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell}] \\
&\geqslant \prod_{i=k(\ell-1)+1}^{c} \frac{\|\mathbf{b}_i^*\|}{2\tilde{\lambda}_1 + \|\mathbf{b}_i^*\|} \geqslant \prod_{i=k(\ell-1)+1}^{c} \frac{\|\mathbf{b}_i^*\|}{4\delta_1\delta_2\tilde{\lambda}_1} \geqslant \prod_{i=k(\ell-1)+1}^{c} \frac{\|\mathbf{b}_i^*\|}{8\delta_1\delta_2\lambda_1} \\
&\geqslant \left(\frac{\|\mathbf{b}_{k(\ell-1)+1}^*\|}{8\delta_1^2\delta_2\lambda_1\sqrt{\gamma_{c-k(\ell-1)}}}\right)^{c-k(\ell-1)} \qquad (7) \\
&\geqslant \left(8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2\gamma_k^{\frac{k(\ell-1)}{k-1}}\gamma_{c-k(\ell-1)}^{\frac{1}{2}}\right)^{-(c-k(\ell-1))} \qquad (8) \\
&\geqslant \left(8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2\gamma_k^{\frac{k(\ell-1)}{k-1}}\gamma_{c-k(\ell-1)}^{\frac{1}{2}}\right)^{-k} \\
&\geqslant \left(8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}\right)^{-k}. \qquad (9)
\end{aligned}$$

for some constant $C > 0$. The inequality in (7) is from the fact that the block $\mathbf{B}_{[k(\ell-1)+1,k\ell]}$ is δ_1-HKZ-reduced, which indicates that $\|\mathbf{b}_{k(\ell-1)+1}^*\| \leqslant \delta_1 \cdot \lambda_1(\mathbf{B}_{[k(\ell-1)+1,c]})$, i.e. $\mathrm{vol}\left(\mathbf{B}_{[k(\ell-1)+1,c]}\right) \geqslant \left(\frac{\|\mathbf{b}_{k(\ell-1)+1}^*\|}{\delta_1\sqrt{\gamma_{c-k(\ell-1)}}}\right)^{c-k(\ell-1)}$. The inequality (8) is due to Fact 2.2, $\lambda_1 \leqslant (\delta_1^2 \cdot \gamma_k)^{\frac{k(\ell-1)}{k-1}} \cdot \|\mathbf{b}_{k(\ell-1)+1}^*\|$. The inequality (9) is from the fact that $\gamma_i = \Theta(i)$ for every positive integer i and $c - k(\ell-1) \leqslant k$.

Finally, we compute the probability 6 as follows:

$$\begin{aligned}
\Pr[\tilde{\mathbf{v}} = \mathbf{v}] &> \frac{1}{n^2} \cdot \prod_{\ell \in [\tilde{t}]} \Pr[\mathsf{E}_\ell \mid \mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_{\tilde{t}}] \\
&\geqslant \frac{1}{n^2} \cdot \prod_{\ell \in [\tilde{t}]} \left(8\delta_1^{2\ell + \frac{2(\ell-1)}{k-1}} \delta_2 (Ck)^{\frac{k(\ell-1)}{k-1} + \frac{1}{2}} \right)^{-k} \\
&= \frac{1}{n^2} \cdot \left(8^{\tilde{t}} \delta_1^{\tilde{t}(\tilde{t}+1) + \frac{\tilde{t}(\tilde{t}-1)}{k-1}} \delta_2^{\tilde{t}} (Ck)^{\frac{k\tilde{t}(\tilde{t}-1)}{2(k-1)} + \frac{\tilde{t}}{2}} \right)^{-k} \\
&\geqslant n^{-2} \cdot 8^{-n} \cdot \delta_1^{-n(\frac{n}{k}+1) - \frac{n(\frac{n}{k}-1)}{k-1}} \cdot \delta_2^{-n} \cdot (Ck)^{-\frac{n(n-k)}{2(k-1)} - \frac{n}{2}} \\
&= \delta_1^{-\frac{n^2}{k} \cdot (1+o(1))} \cdot \delta_2^{-n} \cdot k^{-\frac{n^2}{k} \cdot (\frac{1}{2} + o(1))}. \qquad (10)
\end{aligned}$$

This, together with the fact that $k = \Omega(\log n)$, we obtain that the probability of guessing correctly in Algorithm 1 is at least $\delta_1^{-\frac{n^2}{k} \cdot (1+o(1))} \cdot \delta_2^{-n} \cdot k^{-\frac{n^2}{k} \cdot (\frac{1}{2} + o(1))}$. □

3.2 Time-probability Tradeoff for Solving BDD

Similarly to the time-probability tradeoff for solving SVP in Section 3.1, we also present a blockwise algorithm for guessing the γ-BDD solution in an n-dimensional lattice, given input on a (δ_1, k)-slide-reduced basis and a δ_2-CVP (or δ_2-BDD) oracle. Different from our SVP algorithm, we can take advantage of the approximation factor $\gamma \geqslant 1$ in a BDD problem, since the factor γ can narrow down the range of the coefficients we can guess.

Algorithm 2 A Blockwise BKZ-Based Algorithm for Guessing the γ-BDD Solution

Input: A block size $k \geq 2$, number of blocks t, a (δ_1, k)-slice-reduced basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{Z}^{m \times n}$, a target vector $\mathbf{c} \in \mathbb{R}^m$ s.t. $\mathbf{c} \in \mathrm{span}(\mathbf{B})$ and $\mathrm{dist}(\mathbf{B}, \mathbf{c}) \leqslant \frac{\lambda_1(\mathbf{B})}{\gamma}$, and access to a δ_2-CVP oracle for lattices with rank no more than k.

Output: A vector of $\mathcal{L}(\mathbf{B})$ closest to $\mathbf{c}$.

Sample $\tilde{\lambda}_1 \leftarrow \left\{2^{-n+1}\|\mathbf{b}_1\|, 2^{-n+2}\|\mathbf{b}_1\|, \cdots, \|\mathbf{b}_1\|\right\}$ uniformly at random.
Sample $\tilde{t} \leftarrow \{1, 2, \cdots, t\}$ uniformly at random.
Initiate $\tilde{\mathbf{v}} \leftarrow -\mathbf{c} \in \mathbb{Z}^m$.
for $\ell = \tilde{t}$ to 1 **do**
  if $\exists$ integer $j \in [k(\ell-1)+1, k\ell]$, s.t. $\|\mathbf{b}_j^*\| > \frac{\delta_1\delta_2}{\gamma} \cdot \tilde{\lambda}_1$ **then**
    Let j' be the smallest integer among such j.
    Call the δ_2-CVP oracle on input lattice $\mathcal{L}(\mathbf{B}_{[j',k\ell]})$ and target vector $-\pi_{j'}(\tilde{\mathbf{v}})$, and let $(\tilde{\alpha}_{j'}, \tilde{\alpha}_{j'+1}, \cdots, \tilde{\alpha}_{k\ell})$ be the output linear combination.
    Compute $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + \tilde{\alpha}_{j'}\mathbf{b}_{j'} + \tilde{\alpha}_{j'+1}\mathbf{b}_{j'+1} + \cdots + \tilde{\alpha}_{k\ell}\mathbf{b}_{k\ell}$.
    $c \leftarrow j' - 1$.
  else
    $c \leftarrow k\ell$.
  end if
  for $i = c$ to $k(\ell-1)+1$ **do**
    Sample integer $\tilde{\alpha}_i \leftarrow \left[-\frac{\tilde{\lambda}_1}{\gamma\|\mathbf{b}_i^*\|} - \frac{\langle\tilde{\mathbf{v}}, \mathbf{b}_i^*\rangle}{\|\mathbf{b}_i^*\|^2}, \frac{\tilde{\lambda}_1}{\gamma\|\mathbf{b}_i^*\|} - \frac{\langle\tilde{\mathbf{v}}, \mathbf{b}_i^*\rangle}{\|\mathbf{b}_i^*\|^2}\right]$ uniformly at random.
    Compute $\tilde{\mathbf{v}} \leftarrow \tilde{\mathbf{v}} + \tilde{\alpha}_i\mathbf{b}_i$.
  end for
end for
return $\mathbf{v}$.

Theorem 3.4. *Let n and $m \geqslant n$ be dimension parameters, $k = o(n)$ be the number of block size and $\gamma, \delta_1, \delta_2 \geqslant 1$ be some approximation factors such that $\gamma \leqslant 8\delta_1^{\frac{n}{k}+\frac{n/k-2}{k-1}}\delta_2(Ck)^{\frac{n/2-k}{k-1}+\frac{1}{2}}$ for some constant $C > 0$. Let $\mathbf{B} \in \mathbb{Z}^{m\times n}$ be a lattice basis. Given a δ_1-SVP oracle and a δ_2-CVP (or δ_2-BDD) oracle both for lattices with rank at most k, there exists a probabilistic polynomial time algorithm, such that on input a valid γ-BDD instance $(\mathbf{B}, \mathbf{c})$, it outputs a correct solution with probability at least $\left(\delta_1^{O(n/k)}\delta_2/\gamma\right)^{-n} \cdot k^{-O(n^2/k)}$.*

Specifically, if the SVP oracle and CVP oracle are non-approximate oracles, i.e. $\delta_1 = \delta_2 = 1$, and the target BDD instance is also non-approximate, i.e. $\gamma = 1$, there exists a ppt algorithm which outputs a shortest vector in lattice $\mathcal{L}(\mathbf{B})$ with probability at least $2^{-\frac{n^2\log k}{k}\cdot(\frac{1}{2}+o(1))}$.

Because the proof is almost identical to the proof of Theorem 3.1 for guessing the SVP solution, we put the proof of Theorem 3.4 in Appendix A.1.

4 From Time-Success Tradeoffs for BDD and SVP to Hardness of LWE and SIS

In this section, we formalize a conjecture that captures the hardness of BDD and SVP in terms of both time and success probability. We then derive a new worst-case to average-case reduction from BDD to search-LWE that leverages this conjecture, yielding improved lower bounds on the success probability of any adversary solving LWE in subexponential time.

4.1 A Time-Success Probability Conjecture for BDD and SVP

In light of the results from the previous section, we propose the following conjecture, which quantifies the difficulty of solving approximate BDD, SVP, or CVP in terms of both runtime and success probability:

Conjecture 1 (Time-Success Tradeoff Hardness of BDD and SVP)**.** *Let $c > 0$, and let $T(n) = 2^{o(n)} \geqslant n^c$ be any subexponential function representing the time bound of an algorithm. Then there exists a constant $\kappa > 0$ (that depends on c) such that no algorithm running in time $T(n)$ can solve γ-BDD, γ-SVP, or γ-CVP on arbitrary lattices with success probability better than*

$$2^{-\frac{n^2 \log\log T(n)}{\kappa \log T(n)}},$$

where $\gamma = \mathrm{poly}(n)$.

We believe this conjecture is reasonable in light of the structure of our BKZ-based algorithms (Sections 3.1 and 3.2), which achieve success probability $k^{-O(n^2/k)}$ in the worst-case lattice when given access to SVP and CVP oracles in rank k. Since the best known algorithms for these oracle problems themselves require exponential or superpolynomial time in k, the overall time required to achieve even modest success probability becomes superpolynomial in n. This justifies our belief that no polynomial-time (or even mildly exponential-time) algorithm can significantly improve on this success rate.

Moreover, the formulation in terms of an arbitrary time bound $T(n)$ allows the conjecture to extend beyond polynomial time, making it applicable to a wide range of cryptanalytic settings including subexponential-time adversaries.

4.2 Consequence for LWE

We now use Conjecture 1 to derive hardness guarantees for the Learning with Errors problem.

We recall a known reduction from γ-BDD to search-LWE. The following is adapted from the reduction in [AMV24], but now applied with finer-grained success probabilities:

Lemma 4.1 (γ-BDD $\Rightarrow$ search-LWE, Theorem 3 in [AMV24])**.** *Let $\alpha = \alpha(n) \in (0,1)$, and $p = \mathrm{poly}(n)$. Suppose there exists a PPT algorithm $\mathcal{B}$ that solves $\mathsf{LWE}_{5n^2,p,\Psi_\alpha}$ with n^l samples and success probability q, where*

$$\alpha^2 \geqslant \frac{10\ln 2 \cdot n(n^2 + 2\log n + 8)}{\pi 2^{2n}} + \frac{4\ln(2n(7 \cdot 2^{n^2 + k\log n + 1} + 1))}{\pi n^{2l-1}}.$$

Then there exists a PPT algorithm $\mathcal{A}$, given oracle access to $\mathcal{B}$, that solves any γ-BDD instance for $\gamma = O(\sqrt{n\ln(n/\varepsilon)}/\alpha) = \mathrm{poly}(n)$ with advantage at least

$$\frac{1}{2n} \cdot \left(\left(\frac{1}{24n^l} q - 2^{-n^2+2} \right) \cdot \frac{1}{n^l \sqrt{e}} - 6n^l \varepsilon \right),$$

for some negligible $\varepsilon \in (2^{-n^2}, 1/2)$.

Applying Conjecture 1 to Lemma 4.1, we obtain:

Theorem 4.2 (LWE Hardness from BDD Time-Success Conjecture)**.** *Let $\alpha \in (0,1)$ and $p = \mathrm{poly}(n)$ as above. Assume Conjecture 1 holds. Then for all sufficiently large n, any algorithm running in subexponential time $T(n) = 2^{o(n)}$ and solving $\mathsf{LWE}_{5n^2,p,\Psi_\alpha}$ must have success probability at most*

$$2^{-\frac{n^2 \log\log T(n)}{\kappa \log T(n)}}.$$

4.3 Consequence for SIS

We next re-interpret the seminal reduction from γ-SVP$'$ to SIS by Micciancio and Regev [MR04, Reg04]. γ-SVP$'$ is an SVP-like computational problem, where given a lattice $\mathcal{L}$, the goal is to find a lattice vector $\mathbf{v} \in \mathcal{L}$, such that $\|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathcal{L})$. We modify it into a fine-grained and tight reduction regarding the success probability, like Lemma 4.1. Since we require only one call to the SIS oracle, we cannot expect will get more than one short vector or a vector with length close to $\lambda_1(\mathcal{L})$, even $\lambda_{n-1}(\mathcal{L})$ in the worst-case lattice.

Lemma 4.3 (γ-SVP$'$ $\Rightarrow$ SIS)**.** *Let $p \geqslant 2^{\frac{n-1}{2}}$ be a modulus, $n, m \in \mathbb{Z}$ be dimension parameters, $\beta > 0$ be the* SIS *parameter, and $\gamma = 6\beta n\sqrt{m}$ be the approximation factor. Suppose there exists a polynomial-time algorithm $\mathcal{B}$ that solves $\mathsf{SIS}_{n,m,p,\beta}$ with success probability q. Then there exists a polynomial time algorithm $\mathcal{A}$ that, given oracle access to $\mathcal{B}$, solves γ-SVP$'$ with probability at least $\frac{q - m\varepsilon/2}{40n}$ for $\varepsilon = 2^{-n^2}$.*

Proof. Now we give the reduction algorithm, which follows the worst-case to average-case reduction paradigm by Micciancio and Regev [MR04, Reg04].

Algorithm 3 Reduction from γ-SVP$'$ to $\mathsf{SIS}_{n,m,q,\beta}$

Input: A lattice $\mathcal{L}$ with a basis $\mathbf{B} \in \mathbb{Z}^{n \times n}$, and access to a $\mathsf{SIS}_{n,m,q,\beta}$ oracle on input uniformly random matrix from $\mathbb{Z}_q^{n \times m}$.

Output: A nonzero vector of $\mathcal{L}(\mathbf{B})$ with length no more than $\gamma \cdot \lambda_n(\mathbf{B})$.

1: LLL-reduce the basis $\mathbf{B}$.
2: Sample $\tilde{\eta} \leftarrow \{2^i \|\mathbf{b}_n\| \mid i \in (-n, n] \cap \mathbb{Z}\}$ uniformly at random.
3: **for** $\ell = 1$ to m **do**
4: Sample $\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n$.
5: Let $\mathbf{y}_i \leftarrow \mathbf{x}_i \bmod \mathcal{P}(\mathbf{B})$, $\mathbf{a}_i \leftarrow \lfloor p\mathbf{B}^{-1}\mathbf{y}_i \rfloor$ and $\mathbf{z}_i \leftarrow \mathbf{B}\mathbf{a}_i/p$.
6: **end for**
7: Write $\mathbf{A} = (\mathbf{a}_1, \cdots, \mathbf{a}_m)$, $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_m)$, $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_m)$ and $\mathbf{Z} = (\mathbf{z}_1, \cdots, \mathbf{z}_m)$
8: Call $\mathsf{SIS}_{n,m,q,\beta}$ oracle on input $\mathbf{A}$, and let $\mathbf{w} \in \mathbb{Z}^m$ be its output.
9: **return** $\mathbf{v} = (\mathbf{X} - \mathbf{Y} + \mathbf{Z})\mathbf{w}$.

In the first step of this reduction, we run LLL algorithm on the initial basis and get an LLL-reduced basis $\mathbf{B}$, which has the following property on the bound of its n-th successive minima $\lambda_n(\mathbf{B})$:

$$2^{-\frac{n-1}{2}} \|\mathbf{b}_n\| \leqslant \lambda_n(\mathbf{B}) \leqslant 2^{\frac{n-1}{2}} \|\mathbf{b}_n^*\| \leqslant 2^{\frac{n-1}{2}} \|\mathbf{b}_n\|.$$

Combining this with two bounds on the smoothing parameter $\eta_\varepsilon(\mathbf{B})$ in Lemma 2.3 and Lemma 2.4, we can derive bounds of $\eta_\varepsilon(\mathbf{B})$ on both sides in terms of known $\|\mathbf{b}_i\|$:

$$\eta_\varepsilon(\mathbf{B}) \in \|\mathbf{b}_n\| \cdot \left[\frac{\ln 2}{\sqrt{\pi}} \cdot 2^{-\frac{n-1}{2}}, n \ln 2 \cdot 2^{\frac{n-1}{2}} \right] \subseteq \|\mathbf{b}_n\| \cdot (2^{-n}, 2^n].$$

In step 2, we guess an approximate value of the smoothing parameter $\eta_\varepsilon(\mathbf{B})$. From the previous analysis, we guess $\tilde{\eta}$ within the range $[2\eta_\varepsilon(\mathbf{B}), 4\eta_\varepsilon(\mathbf{B}))$ with probability exactly $1/2n$. We obtain

$$\begin{aligned} &\Pr[0 < \|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B}) \wedge \mathbf{v} \in \mathcal{L}(\mathbf{B})] \\ &\geqslant \frac{1}{2n} \cdot \Pr[0 < \|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B}) \wedge \mathbf{v} \in \mathcal{L}(\mathbf{B}) \mid \tilde{\eta} \in [2\eta_\varepsilon(\mathbf{B}), 4\eta_\varepsilon(\mathbf{B}))] \end{aligned} \tag{11}$$

In the following probability analysis, we assume that we guess $\tilde{\eta} \in [2\eta_\varepsilon(\mathbf{B}), 4\eta_\varepsilon(\mathbf{B}))$ successfully and we omit this event in the condition part.

Claim 4.4 analyses the statistical distance between $\mathbf{A}$ in Algorithm 3 and the uniformly random distribution $U(\mathbb{Z}_q^{n \times m})$ via the smoothing technique [MR04], and use this to bound the success probability of the SIS oracle in Step 8.

Claim 4.4 (Implicit in Claim 13 [Reg04] and Theorem 5.9 [MR04]). *Let the* SIS *instance* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *be generated as stated in Algorithm 3, we have* $\Pr[\mathbf{Aw} = \mathbf{0} \wedge \|\mathbf{w}\| \leqslant \beta] \geqslant q - \frac{m\varepsilon}{2}$, *where the probability is taken from sampling* $\mathbf{x}_i \sim D_{\tilde{\eta}}^n$.

With Claim 4.4, we have

$$\begin{aligned}
&\Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}}[0 < \|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B}) \wedge \mathbf{v} \in \mathcal{L}(\mathbf{B})] \\
\geqslant &\Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}}[0 < \|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B}) \wedge \mathbf{v} \in \mathcal{L}(\mathbf{B}) \mid \mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \\
&\cdot \Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}}[\mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \\
\geqslant &\left(q - \frac{m\varepsilon}{2}\right) \cdot \Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}}[0 < \|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B}) \wedge \mathbf{v} \in \mathcal{L}(\mathbf{B}) \mid \mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \qquad (12)
\end{aligned}$$

Claim 4.5 states that $\mathbf{v}$ belongs to the lattice $\mathcal{L}(\mathbf{B})$ under the condition $\mathbf{Aw} = \mathbf{0}$.

Claim 4.5 (Lemma 5.8 in [MR04] and Claim 11 in [Reg04]). *If the* $\mathsf{SIS}_{n,m,q,\beta}$ *returns a vector* $\mathbf{w}$ *such that* $\mathbf{Aw} = 0 \bmod q$, *the output of Algorithm 3 is in the lattice* $\mathcal{L}(\mathbf{B})$.

With Claim 4.5, we have

$$\begin{aligned}
&\Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n}[0 < \|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B}) \wedge \mathbf{v} \in \mathcal{L}(\mathbf{B}) \mid \mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \\
= &\Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n}[0 < \|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B}) \mid \mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \\
\geqslant &1 - \Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n}[\mathbf{v} = \mathbf{0} \mid \mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \\
&- \Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n}[\|\mathbf{v}\| > \gamma \cdot \lambda_n(\mathbf{B}) \mid \mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \qquad (13)
\end{aligned}$$

Claim 4.6 tells us the probability of the final solution $\mathbf{v}$ to be non-zero is at most 0.9, where the probability is taken on $\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n$ with any fixed $\mathbf{y}_i, \mathbf{z}_i \in \mathbb{R}^n$. From a simple conditioned Gaussian analysis, $\mathbf{x}_i$ actually follows the discrete Gaussian distribution $D_{\mathcal{L}+\mathbf{y}_i,\tilde{\eta}}$. Since the condition $(\mathbf{Aw} = \mathbf{0} \wedge \|\mathbf{w}\| \leqslant \beta)$ depends on $\mathbf{Y}$ totally, we can always analyze the probabilities under the condition that $\mathbf{Y}$ and $\mathbf{Z}$ are fixed.

Claim 4.6 (Lemma 14 in [Reg04]). *If* $\tilde{\eta} \geqslant \sqrt{2}\eta_\varepsilon(\mathbf{B})$, *any* $y \in \mathcal{P}(\mathbf{B})$ *and any* $n-1$*-dimensional hyperplane* H, *we have* $\Pr_{\mathbf{x} \leftarrow D_{\tilde{\eta}}^n}[\mathbf{x} \in H \mid \mathbf{y} = \mathbf{x} \bmod \mathcal{P}(\mathbf{B})] = \Pr_{\mathbf{x} \leftarrow D_{\mathcal{L}+\mathbf{y},\tilde{\eta}}}[\mathbf{x} \in H] < 0.9$.

Furthermore, in Algorithm 3, if $\mathbf{w} \neq \mathbf{0}$, *we have* $\Pr_{\mathbf{X} \leftarrow D_{\tilde{\eta}}^{n\times m}}[\mathbf{v} \neq \mathbf{0} \mid \mathbf{Y}, \mathbf{Z}] > 0.1$ *for any fixed matrix* $\mathbf{Y}, \mathbf{Z} \in \mathbb{R}^{n\times m}$.

From Claim 4.6, we can take the law of total probability over $\mathbf{Y}$ and $\mathbf{Z}$ that satisfy this condition $(\mathbf{Aw} = \mathbf{0} \wedge \|\mathbf{w}\| \leqslant \beta)$, and obtain that

$$\Pr_{\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n}[\mathbf{v} = \mathbf{0} \mid \mathbf{Aw} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta] \leqslant 0.9. \qquad (14)$$

Claim 4.7. *Let* $\tilde{\eta} \geqslant \eta_\varepsilon(\mathbf{B})$. *If the* $\mathsf{SIS}_{n,m,q,\beta}$ *returns a vector* $\mathbf{w}$ *which satisfies that* $\|\mathbf{w}\| \leqslant \beta$, *then the probability that* $\|\mathbf{v}\| \leqslant \gamma \cdot \lambda_n(\mathbf{B})$ *is at least* $1 - 2^{-n}$. *The probability is taken on* $\mathbf{x}_i \leftarrow D_{\tilde{\eta}}^n$ *conditioned on any fixed* $\mathbf{y}_i$ *and* $\mathbf{z}_i$ *for* $i \in [m]$.

Proof. First, by the triangle inequality, we have

$$\|\mathbf{v}\| \leqslant \|\mathbf{Xw}\| + \|(\mathbf{Y} - \mathbf{Z})\mathbf{w}\|$$

To deal with the second term, we have

$$\begin{aligned}\|(\mathbf{Y}-\mathbf{Z})\mathbf{w}\| &\leqslant \sum_{i=1}^{m} \|\mathbf{y}_i-\mathbf{z}_i\|\cdot|w_i| \\ &= \sum_{i=1}^{m} \left\|\mathbf{B}\left(p\mathbf{B}^{-1}\mathbf{y}_i - \lfloor p\mathbf{B}^{-1}\mathbf{y}_i\rfloor\right)\right\|\cdot|w_i|/p \\ &\leqslant n\|\mathbf{B}\|/p\cdot\sum_{i=1}^{m}|w_i| \\ &\leqslant n\|\mathbf{B}\|/p\cdot\sqrt{m}\|\mathbf{w}\| \\ &\leqslant \beta n\sqrt{m}\cdot\lambda_n(\mathbf{B}) = \frac{\gamma}{6}\cdot\lambda_n(\mathbf{B}).\end{aligned}$$

where the last inequality is due to the fact that for a LLL-reduced basis $\mathbf{B}$, $\|\mathbf{B}\| \leqslant 2^{\frac{n-1}{2}}\lambda_n(\mathbf{B})$. The condition $\mathbf{y} = \mathbf{x} \bmod \mathcal{P}(\mathbf{B})$ means that $\mathbf{x}$'s conditional support is the coset $\mathbf{y}+\mathcal{L}(\mathbf{B})$, so the conditional distribution of $\mathbf{x}$ is exactly $D_{\mathcal{L}+\mathbf{y}_i,\sigma}$. From Lemma 2.5, we have

$$s_1(\mathbf{X}) \leqslant \frac{\tilde{\eta}\cdot(\sqrt{m}+2\sqrt{n})}{\sqrt{2\pi}} \leqslant \frac{3\sqrt{m}}{\sqrt{2\pi}}\cdot 4\eta_\varepsilon(\mathbf{B}) \leqslant \frac{12n\sqrt{m}}{\sqrt{2\pi}}\cdot\lambda_n(\mathbf{B}) < 5n\sqrt{m}\cdot\lambda_n(\mathbf{B}),$$

except with probability $4e^{-\pi n}$. Under this condition, we have

$$\|\mathbf{X}\mathbf{w}\| \leqslant s_1(\mathbf{X})\cdot\|\mathbf{w}\| \leqslant 5\beta n\sqrt{m}\cdot\lambda_n(\mathbf{B}) = \frac{5\gamma}{6}\cdot\lambda_n(\mathbf{B}),$$

which completes the proof of Claim 4.7. □

From Claim 4.7, as stated before, we can take the law of total probability over $\mathbf{Y}$ and $\mathbf{Z}$ that satisfy this condition ($\mathbf{A}\mathbf{w} = \mathbf{0} \wedge \|\mathbf{w}\| \leqslant \beta$), and obtain that

$$\Pr_{\mathbf{x}_i\leftarrow D_{\tilde{\eta}}^n}\left[\|\mathbf{v}\| > \gamma\cdot\lambda_n(\mathbf{B}) \mid \mathbf{A}\mathbf{w} = \mathbf{0} \bmod p \wedge \|\mathbf{w}\| \leqslant \beta\right] \leqslant 2^{-n}. \tag{15}$$

Combining the inequalities (11), (12), (13), (14) and (15) together, we obtain that

$$\Pr\left[0 < \|\mathbf{v}\| \leqslant \gamma\cdot\lambda_n(\mathbf{B}) \wedge \mathbf{v}\in\mathcal{L}(\mathbf{B})\right] \geqslant \frac{q-m\varepsilon/2}{40n},$$

which completes the proof. □

Applying Conjecture 1 to Lemma 4.3, we obtain:

Theorem 4.8 (SIS Hardness from SVP′ Time-Success Conjecture)**.** *Let $p \geqslant 2^{\frac{n-1}{2}}, n, m, \beta$ be the* SIS *parameters as Lemma 4.3. Assume Conjecture 1 holds. Then there exists a constant $\kappa > 1$, for all sufficiently large n, any algorithm running in subexponential time $T(n) = 2^{o(n)}$ and solving $\mathsf{SIS}_{n,m,p,\beta}$ must have success probability at most*

$$2^{-\frac{n^2\log\log T(n)}{\kappa\log T(n)}}.$$

Theorem 4.2 (Theorem 4.8, respectively) yields a concrete lower bound on the **bit security** of LWE (SIS, respectively) under fine-grained hardness assumptions for worst-case lattice problems. It shows that even algorithms running in subexponential time cannot solve LWE (SIS, respectively) with success probability above negligible, as long as they cannot beat the stated bound for BDD (SVP, respectively).

Moreover, this approach provides a **modular perspective**: rather than assuming LWE is secure directly, we make a conjecture about the hardness of BDD (SVP, respectively) that can be supported by known algorithmic limitations

(e.g., barriers for BKZ and sieving), and deduce strong implications for the average-case hardness of LWE (SIS, respectively).

Acknowledgment. Both Divesh Aggarwal and Haoxiang Jin are supported by the research grant "Computational Hardness of Lattice Problems and Applications" with the project number "NRF award NRF-NRFI09-0005". Haoxiang Jin is also supported by NSF CNS-2247727.

References

[ABGS21] Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. Fine-grained hardness of CVP(P) - everything that we can prove (and nothing else). In Dániel Marx, editor, *32nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1816–1835, Virtual Conference, January 10–13, 2021. ACM-SIAM. 2

[ACKS25] Divesh Aggarwal, Yanlin Chen, Rajendra Kumar, and Yixin Shen. Improved classical and quantum algorithms for the shortest vector problem via bounded distance decoding. *SIAM Journal on Computing*, 54(2):233–278, 2025. 2

[ADRS15] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2^n time using discrete Gaussian sampling: Extended abstract. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing*, pages 733–742, Portland, OR, USA, June 14–17, 2015. ACM Press. 2

[ADS15] Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the closest vector problem in 2^n time - the discrete Gaussian strikes again! In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 563–582, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press. 2

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th Annual ACM Symposium on Theory of Computing*, pages 99–108, Philadephia, PA, USA, May 22–24, 1996. ACM Press. 1, 2, 8

[ALNS20] Divesh Aggarwal, Jianwei Li, Phong Q. Nguyen, and Noah Stephens-Davidowitz. Slide reduction, revisited - filling the gaps in SVP approximation. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 274–295, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Cham, Switzerland. 2, 7, 10, 20

[ALS21] Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. A $2^{n/2}$-time algorithm for $\sqrt{n}$-SVP and $\sqrt{n}$-Hermite SVP, and an improved time-approximation tradeoff for (H)SVP. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 467–497, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland. 2

[AM11] Divesh Aggarwal and Ueli Maurer. The leakage-resilience limit of a computational problem is equal to its unpredictability entropy. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 686–701, Seoul, South Korea, December 4–8, 2011. Springer Berlin Heidelberg, Germany. 2

[AMV24] Divesh Aggarwal, Leong Jin Ming, and Alexandra Veliche. Worst-case to average-case hardness of LWE: An alternative perspective. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024: 22nd Theory of Cryptography Conference, Part II*, volume 15365 of *Lecture Notes in Computer Science*, pages 308–336, Milan, Italy, December 2–6, 2024. Springer, Cham, Switzerland. 2, 3, 14

[ASD18] Divesh Aggarwal and Noah Stephens-Davidowitz. Just Take the Average! An Embarrassingly Simple 2^n-Time Algorithm for SVP (and CVP). In Raimund Seidel, editor, *1st Symposium on Simplicity in Algorithms (SOSA 2018)*, volume 61 of *Open Access Series in Informatics (OASIcs)*, pages 12:1–12:19, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 2

[BGS17] Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the quantitative hardness of CVP. In Chris Umans, editor, *58th Annual Symposium on Foundations of Computer Science*, pages 13–24, Berkeley, CA, USA, October 15–17, 2017. IEEE Computer Society Press. 2

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 575–584, Palo Alto, CA, USA, June 1–4, 2013. ACM Press. 2, 3, 4, 8

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In Moni Naor, editor, *ITCS 2014: 5th Conference on Innovations in Theoretical Computer Science*, pages 1–12, Princeton, NJ, USA, January 12–14, 2014. Association for Computing Machinery. 1

[CSB87] J. H. Conway, N. J. A. Sloane, and E. Bannai. *Sphere-packings, lattices, and groups*. Springer-Verlag, Berlin, Heidelberg, 1987. 5

[EK20] Thomas Espitau and Paul Kirchner. The nearest-colattice algorithm. Cryptology ePrint Archive, Report 2020/694, 2020. 2

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Mitzenmacher [Mit09], pages 169–178. 1

[GHN06] Nicolas Gama, Nick Howgrave-Graham, and Phong Q. Nguyen. Symplectic lattice reduction and NTRU. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 233–253, St. Petersburg, Russia, May 28 – June 1, 2006. Springer Berlin Heidelberg, Germany. 5

[GN08] Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 207–216, Victoria, BC, Canada, May 17–20, 2008. ACM Press. 2, 3, 6, 7, 10, 20

[HR07] Ishay Haviv and Oded Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In David S. Johnson and Uriel Feige, editors, *39th Annual ACM Symposium on Theory of Computing*, pages 469–477, San Diego, CA, USA, June 11–13, 2007. ACM Press. 2

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987. 2

[Kle00] Philip N. Klein. Finding the closest lattice vector when it's unusually close. In David B. Shmoys, editor, *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 937–941, San Francisco, CA, USA, January 9–11, 2000. ACM-SIAM. 2

[LLL82] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982. 4, 6

[LN14] Jianwei Li and Phong Q. Nguyen. Approximating the densest sublattice from rankin's inequality. *LMS Journal of Computation and Mathematics*, 17(A):92–111, 2014. 2, 7

[LN19] Jianwei Li and Phong Q. Nguyen. Computing a lattice basis revisited. In *Proceedings of the 2019 International Symposium on Symbolic and Algebraic Computation*, ISSAC '19, page 275–282, New York, NY, USA, 2019. Association for Computing Machinery. 6

[MH13] J. Milnor and D. Husemoller. *Symmetric Bilinear Forms*. Ergebnisse der Mathematik und ihrer Grenzgebiete. 2. Folge. Springer Berlin Heidelberg, 2013. 5

[Mic12] Daniele Micciancio. Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory of Computing*, 8(22):487–512, 2012. 2

[Mit09] Michael Mitzenmacher, editor. *41st Annual ACM Symposium on Theory of Computing*, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press. 19, 20

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer Berlin Heidelberg, Germany. 8

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th Annual Symposium on Foundations of Computer Science*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press. 1, 2, 3, 8, 15, 16

[MV10a] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In Leonard J. Schulman, editor, *42nd Annual ACM Symposium on Theory of Computing*, pages 351–358, Cambridge, MA, USA, June 5–8, 2010. ACM Press.

2

[MV10b] Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In Moses Charika, editor, *21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1468–1480, Austin, TX, USA, January 17–19, 2010. ACM-SIAM. 2

[MW16] Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 820–849, Vienna, Austria, May 8–12, 2016. Springer Berlin Heidelberg, Germany. 2, 6

[MW18] Daniele Micciancio and Michael Walter. On the bit security of cryptographic primitives. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 3–28, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland. 2

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Mitzenmacher [Mit09], pages 333–342. 2, 3, 8

[Reg04] Oded Regev. Lecture 12. average-case hardness, 2004. Lattices in Computer Science, Lecture Notes. Available at: https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/averagecase.pdf, last accessed 01 May 2025. 15, 16

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press. 1, 2, 3, 8

A Missing Proofs

A.1 Proof of Theorem 3.4

Proof of Theorem 3.4. W.l.o.g., we can assume $k \mid n$. Otherwise, we can always pad zero vectors to the basis $\mathbf{B}$, to make k divide n. Furthermore, changing n to $k \cdot \lceil n/k \rceil$ does not influence the asymptotic value in the exponent since $k = o(n)$.

From [GN08, ALNS20], we can first call the k-dim δ-SVP oracle polynomial times to transform the original basis to a (δ_1, k)-slide-reduced basis. Therefore, we can always assume the basis $\mathbf{B}$ is (δ_1, k)-slide-reduced.

With a slide-reduced basis, we next run and analyse Algorithm 2.

Let $\mathbf{v} = \sum_{i\in[n]} \alpha_i \mathbf{b}_i$, where $\alpha_i \in \mathbb{Z}$ for $i \in [n]$, be the vector in the lattice $\mathcal{L}(\mathbf{B})$ which is closest to $\mathbf{c}$, and $\lambda_1 := \lambda_1(\mathbf{B})$ be the shortest length of $\mathcal{L}(\mathbf{B})$, s.t. $\|\mathbf{v}-\mathbf{c}\| \leqslant \frac{\lambda_1}{2\gamma}$.

In step 1, we guess an approximate value of the shortest length λ_1. Since $\mathbf{B}$ is a $1/3$-LLL-reduced basis, we have $2^{-\frac{n-1}{2}}\|\mathbf{b}_1\| \leqslant \lambda_1 \leqslant \|\mathbf{b}_1\|$, and this indicates that there exists unique $\lambda' \in \{2^{-i}\|\mathbf{b}_1\| : i \in [0, n-1] \cap \mathbb{Z}\}$, s.t. $\lambda_1 \leqslant \lambda' < 2\lambda_1$. Therefore, we have $\Pr\left[\lambda_1 \leqslant \tilde{\lambda}_1 < 2\lambda_1\right] = \Pr\left[\tilde{\lambda}_1 = \lambda'_1\right] = \frac{1}{n}$.

In step 2, we guess the maximal index $t' \in [t]$ such that $\pi_{k(t'-1)+1}(\mathbf{v}) \neq 0$. The probability of guessing correctly is exactly $1/t$. Guessing correctly $\tilde{t} = t'$ indicates that the coefficients in the last $t - \tilde{t}$ blocks are all 0, i.e.

$$\alpha_{\tilde{t}k+1} = \tilde{\alpha}_{\tilde{t}k+1} = \cdots = \alpha_n = \tilde{\alpha}_n = 0.$$

Next, we analyze the probability of guessing each α_i correctly, conditioned on $\lambda_1 \leqslant \tilde{\lambda}_1 < 2\lambda_1$, and $\tilde{t}$ is the maximal index such that $\pi_{k(\tilde{t}-1)+1}(\mathbf{v}) \neq 0$.

In the following steps, we guess the coefficients block by block and compute the probability of guessing all coefficients correctly as:

$$\begin{aligned}\Pr[\tilde{\mathbf{v}} = \mathbf{v}] &\geqslant \frac{1}{n^2} \cdot \Pr\left[\mathsf{E}_1 \wedge \mathsf{E}_2 \wedge \cdots \wedge \mathsf{E}_{\tilde{t}} \mid \tilde{\lambda}_1 \in [\lambda_1, 2\lambda_1) \wedge \tilde{t} = t'\right] \\ &= \frac{1}{n^2} \cdot \prod_{\ell\in[\tilde{t}]} \Pr\left[\mathsf{E}_\ell \mid \mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_{\tilde{t}} \wedge \left(\tilde{\lambda}_1 \in [\lambda_1, 2\lambda_1)\right) \wedge (\tilde{t} = t')\right] \end{aligned} \tag{16}$$

where the E_ℓ is defined as the event of guessing all the coefficients in ℓ-th block successfully, i.e.

$$\left(\tilde{\alpha}_{k(\ell-1)+1} = \alpha_{k(\ell-1)+1}\right) \wedge \left(\tilde{\alpha}_{k(\ell-1)+2} = \alpha_{k(\ell-1)+2}\right) \wedge \cdots \wedge \left(\tilde{\alpha}_{k\ell} = \alpha_{k\ell}\right).$$

We next omit the condition of $\tilde{\lambda}_1 \in [\lambda_1, 2\lambda_1)$ and $\tilde{t} = t'$ when computing probability.

When the large loop comes to the ℓ-th block, we first check whether there exists a GS-orthogonalized vector $\mathbf{b}_j^*$ in the ℓ-th block, such that the length far exceeds $\tilde{\lambda}_1$. We let j' be the smallest index among such j, then we use the following claim to prove that the integers returned by the δ_2-CVP call in step 7 always satisfies $\tilde{\alpha}_i = \alpha_i$ for all $i = j', \cdots, k\ell$.

Claim A.1. *Let the projected block $\mathbf{B}_{[k(\ell-1)+1,k\ell]}$ be δ_1-HKZ-reduced and $\|\mathbf{b}_j^*\| > \frac{\delta_1\delta_2}{\gamma} \cdot \lambda_1$ for some $j \in [k(\ell-1)+1, k\ell]$. Conditioned on $\tilde{\alpha}_i = \alpha_i$ for $i = k\ell+1, \cdots, n$, the δ_2-CVP oracle, on input input lattice $\mathbf{B}_{[j,k\ell]}$ and target vector $-\pi_j\left(\sum_{i=k\ell+1}^{n} \alpha_i \mathbf{b}_i - \mathbf{c}\right)$, returns linear combination $\alpha_j, \cdots, \alpha_{k\ell}$ with success probability 1.*

Proof. We only need to prove that $(\mathbf{B}_{[j,k\ell]}, -\pi_j(\sum_{i=k\ell+1}^{n} \alpha_i \mathbf{b}_i - \mathbf{c}))$ is a valid δ_2-BDD instance, and the closest vector to the target is $\pi_j(\sum_{i=j}^{k\ell} \alpha_i \mathbf{b}_i) \in \mathcal{L}(\mathbf{B}_{[j,k\ell]})$.

Due to the property of δ_1-HKZ-reduced basis $\mathbf{B}_{[k(\ell-1)+1,k\ell]}$, $\|\mathbf{b}_j^*\| \leqslant \delta_1 \cdot \lambda_1\left(\mathbf{B}_{[j,k\ell]}\right)$, then we have

$$\left\| \pi_j\left(\sum_{i=j}^{k\ell} \alpha_i \mathbf{b}_i\right) + \pi_j\left(\sum_{i=k\ell+1}^{n} \alpha_i \mathbf{b}_i - \mathbf{c}\right) \right\| = \|\pi(\mathbf{v} - \mathbf{c})\| \leqslant \frac{\lambda_1}{2\gamma} < \frac{\|\mathbf{b}_j^*\|}{2\delta_1\delta_2} \leqslant \frac{\lambda_1\left(\mathbf{B}_{[j,k\ell]}\right)}{2\delta_2},$$

which completes the proof. □

Claim 3.2 shows that in step 7,

$$\Pr[\tilde{\alpha}_{c+1} = \alpha_{c+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell} \mid \mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_t] = 1.$$

Next, we compute the probability of guessing each $\tilde{\alpha}_i = \alpha_i$ correctly conditioned on $\tilde{\alpha}_{i+1} = \alpha_{i+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell}$ and $\mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_t$, i.e. in step 14, $\tilde{\mathbf{v}} = \sum_{j=i+1}^{n} \alpha_j \mathbf{b}_j$, for all $i \in [k(\ell-1)+1, c]$.

Claim A.2. *For all $i \in [k(\ell-1)+1, c]$, we have*

$$\Pr[\alpha_i = \tilde{\alpha}_i \mid \alpha_{i+1} = \tilde{\alpha}_{i+1} \wedge \cdots \wedge \alpha_n = \tilde{\alpha}_n] \geqslant \frac{\gamma\|\mathbf{b}_i^*\|}{2\tilde{\lambda}_1 + \gamma\|\mathbf{b}_i^*\|}.$$

Proof. Project $\mathbf{v}$ onto the direction of $\mathbf{b}_i^*$, we get

$$\pi_{\mathbf{b}_i^*}(\mathbf{v}) = \frac{\langle \mathbf{v}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|} \cdot \mathbf{b}_i^* = \left(\alpha_i + \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}\right) \cdot \mathbf{b}_i^*$$

And we know that the length of this projection is not larger than λ_1, so we have

$$\left|\alpha_i + \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}\right| \cdot \|\mathbf{b}_i^*\| \leqslant \frac{\lambda_1}{\gamma} \leqslant \frac{\tilde{\lambda}_1}{\gamma}$$

which indicates that

$$\alpha_i \in \left[-\frac{\tilde{\lambda}_1}{\gamma\|\mathbf{b}_i^*\|} - \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}, \frac{\tilde{\lambda}_1}{\gamma\|\mathbf{b}_i^*\|} - \frac{\langle \tilde{\mathbf{v}}, \mathbf{b}_i^* \rangle}{\|\mathbf{b}_i^*\|^2}\right].$$

Therefore, the probability of correcting guessing $\tilde{\alpha}_i = \alpha_i$ is no more than $\frac{1}{2\tilde{\lambda}_1/(\gamma\|\mathbf{b}_i^*\|)+1}$, which completes the proof. □

In the following proof, we first assume $\gamma \leqslant 8\delta_1^2\delta_2 \cdot \sqrt{Ck}$. This is to guarantee that $\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}} \leqslant 1$ for every $\ell \geqslant 1$ and we can get a quick capture of the final result via a smooth computation. We have (here we omit the condition $\mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_{\tilde{t}}$),

$$
\begin{aligned}
&\Pr[\mathsf{E}_\ell] \\
&= \Pr[\tilde{\alpha}_{c+1} = \alpha_{c+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell}] \cdot \prod_{i=k(\ell-1)+1}^{c} \Pr[\tilde{\alpha}_i = \alpha_i \mid \tilde{\alpha}_{i+1} = \alpha_{i+1} \wedge \cdots \wedge \tilde{\alpha}_{k\ell} = \alpha_{k\ell}] \\
&\geqslant \prod_{i=k(\ell-1)+1}^{c} \frac{\gamma\|\mathbf{b}_i^*\|}{2\tilde{\lambda}_1 + \gamma\|\mathbf{b}_i^*\|} \geqslant \prod_{i=k(\ell-1)+1}^{c} \frac{\gamma\|\mathbf{b}_i^*\|}{4\delta_1\delta_2\tilde{\lambda}_1} \geqslant \prod_{i=k(\ell-1)+1}^{c} \frac{\gamma\|\mathbf{b}_i^*\|}{8\delta_1\delta_2\lambda_1} \\
&\geqslant \left(\frac{\gamma\|\mathbf{b}^*_{k(\ell-1)+1}\|}{8\delta_1^2\delta_2\lambda_1\sqrt{\gamma_{c-k(\ell-1)}}}\right)^{c-k(\ell-1)} && (17) \\
&\geqslant \left(\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2\gamma_k^{\frac{k(\ell-1)}{k-1}}\gamma_{c-k(\ell-1)}^{\frac{1}{2}}}\right)^{c-k(\ell-1)} && (18) \\
&\geqslant \left(\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}}\right)^{c-k(\ell-1)} && (19) \\
&\geqslant \left(\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}}\right)^{k} && (20)
\end{aligned}
$$

for some constant $C > 0$. The first inequality in (17) is from the fact that the block $\mathbf{B}_{[k(\ell-1)+1,k\ell]}$ is δ_1-HKZ-reduced, which indicates that $\|\mathbf{b}_{k(\ell-1)+1}\| \leqslant \delta_1 \cdot \lambda_1(\mathbf{B}_{[k(\ell-1)+1,c]})$, i.e. $\mathrm{vol}\left(\mathbf{B}_{[k(\ell-1)+1,c]}\right) \geqslant \left(\frac{\|\mathbf{b}^*_{k(\ell-1)+1}\|}{\delta_1\sqrt{\gamma_{c-k(\ell-1)}}}\right)^{c-k(\ell-1)}$. The inequality (18) is due to Fact 2.2, i.e. $\lambda_1 \leqslant \gamma_k^{\frac{k(\ell-1)}{k-1}} \cdot \|\mathbf{b}^*_{k(\ell-1)+1}\|$. The inequality (19) is from the fact that Hermite's constant $\gamma_n = O(n)$. The inequality (20) is based on the condition $\gamma \leqslant 8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}$ for all $\ell \geqslant 1$.

Finally, we compute the probability (16) as follows:

$$
\begin{aligned}
\Pr[\tilde{\mathbf{v}} = \mathbf{v}] &> \frac{1}{n^2} \cdot \prod_{\ell\in[\tilde{t}]} \Pr[\mathsf{E}_\ell \mid \mathsf{E}_{\ell+1} \wedge \cdots \wedge \mathsf{E}_{\tilde{t}}] \\
&\geqslant \frac{1}{n^2} \cdot \prod_{\ell\in[\tilde{t}]} \left(\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}}\right)^{k} \\
&\geqslant \frac{1}{n^2} \cdot \prod_{\ell\in[t]} \left(\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}}\right)^{k} \\
&= n^{-2} \cdot \left(\frac{\gamma^t}{8\delta_1^{n(\frac{n}{k}+1)+\frac{t(t-1)}{k-1}}\delta_2^t(Ck)^{\frac{tk(t-1)}{2(k-1)}+\frac{t}{2}}}\right)^{k} && (21) \\
&= n^{-2} \cdot \left(\frac{\gamma^n}{8\delta_1^{n(\frac{n}{k}+1)+\frac{n(\frac{n}{k}-1)}{k-1}}\delta_2^n(Ck)^{\frac{n(n-k)}{2(k-1)}+\frac{n}{2}}}\right) && (22)
\end{aligned}
$$

This, together with the fact that $\gamma_k = \Theta(k)$, we obtain that the probability of guessing correctly in Algorithm 2 is at least $\left(\delta_1^{\frac{n}{k}\cdot(1+o(1))}\delta_2/\gamma\right)^{-n}\cdot k^{-\frac{n^2}{k}\cdot(\frac{1}{2}+o(1))}$ under the constraint $\gamma \leqslant 8\delta_1^2\delta_2\cdot(Ck)^{\frac{1}{2}}$.

In order to deal with the general case $\gamma \leqslant 8\delta_1^{t+\frac{t-2}{k-1}}\delta_2(Ck)^{\frac{k(t/2-1)}{k-1}+\frac{1}{2}}$, we can use a somewhat extreme inequality: for $\ell \leqslant \min\{t/2, \tilde{t}\}$, we have

$$\begin{aligned}
\Pr[\mathsf{E}_\ell \mid \mathsf{E}_{\ell+1}\wedge\cdots\wedge\mathsf{E}_{\tilde{t}}] &\geqslant \left(\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}}\right)^{c-k(\ell-1)} \\
&\geqslant \left(\frac{\gamma}{8\delta_1^{t+\frac{t-2}{k-1}}\delta_2(Ck)^{\frac{k(t/2-1)}{k-1}+\frac{1}{2}}}\right)^{c-k(\ell-1)} \\
&\geqslant \left(\frac{\gamma}{8\delta_1^{t+\frac{t-2}{k-1}}\delta_2(Ck)^{\frac{k(t/2-1)}{k-1}+\frac{1}{2}}}\right)^{k},
\end{aligned}$$

where the first inequality is the same as the way to get (18), the second inequality is because $\ell \leqslant t/2$, and the third inequality is due to $\gamma \leqslant 8\delta_1^{t+\frac{t-2}{k-1}}\delta_2\gamma_k^{\frac{k(t/2-1)}{k-1}+\frac{1}{2}}$. Then, we can compute that

$$\begin{aligned}
\Pr[\tilde{\mathbf{v}} = \mathbf{v}] &> \frac{1}{n^2}\cdot\prod_{\ell\in[\tilde{t}]}\Pr[\mathsf{E}_\ell \mid \mathsf{E}_{\ell+1}\wedge\cdots\wedge\mathsf{E}_{\tilde{t}}] \\
&\geqslant \frac{1}{n^2}\cdot\prod_{\ell\leqslant t/2}\left(\frac{\gamma}{8\delta_1^{t+\frac{t-2}{k-1}}\delta_2(Ck)^{\frac{k(t/2-1)}{k-1}+\frac{1}{2}}}\right)^{k}\cdot\prod_{t/2<\ell\leqslant t}\left(\frac{\gamma}{8\delta_1^{2\ell+\frac{2(\ell-1)}{k-1}}\delta_2(Ck)^{\frac{k(\ell-1)}{k-1}+\frac{1}{2}}}\right)^{k} \\
&\geqslant n^{-2}\cdot\left(8\delta_1^{O(n/k)}\delta_2/\gamma\right)^{-n}\cdot(Ck)^{-\frac{n^2}{k}-\frac{n}{2}} \qquad (23)
\end{aligned}$$

which also achieves the probability $(\delta_1^{O(n/k)}\delta_2/\gamma)^{-n}\cdot k^{-O(n^2/k)}$. □