Mind the Gap? Not for SVP Hardness Under ETH!

Divesh Aggarwal
National University of Singapore, Singapore

Rishav Gupta
National University of Singapore, Singapore

Aditya Morolia
Centre for Quantum Technologies, Singapore, Singapore

Chuanqi Zhang
Monash University, Melbourne, Australia

Abstract

We prove new hardness results for fundamental lattice problems under the Exponential Time Hypothesis (ETH). Building on a recent breakthrough by Bitansky et al. [BHIRW24], who gave a polynomial-time reduction from 3SAT to the (gap) MAXLIN problem – a class of CSPs with linear equations over finite fields – we derive ETH hardness for several lattice problems.

First, we show that for any $p \in [1, \infty)$, there exists an explicit constant $\gamma > 1$ such that $\mathsf{CVP}_{p,\gamma}$ (the ℓ_p-norm approximate Closest Vector Problem) does not admit a $2^{o(n)}$-time algorithm unless ETH is false. Our reduction is deterministic and proceeds via a direct reduction from (gap) MAXLIN to $\mathsf{CVP}_{p,\gamma}$.

Our main contribution is a randomized ETH hardness result for $\mathsf{SVP}_{p,\gamma}$ (the ℓ_p-norm approximate Shortest Vector Problem) for all $p \in (2, \infty)$. This result relies on a novel geometric property of the integer lattice $\mathbb{Z}^n$ in the ℓ_p norm, which says that for any $p \in (2, \infty)$, the number of lattice vectors close to $\frac{1}{2}\mathbf{1}_n$ (in the ℓ_p norm) is exponentially larger than the number of short vectors (namely those close to the origin). We establish this property via a new inequality for the Theta function, which we use to get a randomized reduction from $\mathsf{CVP}_{p,\gamma}$ to $\mathsf{SVP}_{p,\gamma'}$.

Finally, we also use our ideas to give some minor improvements over prior reductions from 3SAT to $\mathsf{BDD}_{p,\alpha}$ (the Bounded Distance Decoding Problem), yielding better ETH hardness results for $\mathsf{BDD}_{p,\alpha}$ for any $p \in [1, \infty)$ and $\alpha > \alpha_p^\ddagger$, where $\alpha_p^\ddagger$ is an explicit threshold depending on p.

2012 ACM Subject Classification Theory of computation → Computational complexity and cryptography

Keywords and phrases Lattices, Fine-Grained Complexity, Exponential Time Hypothesis, Post-Quantum Cryptography

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.8

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2504.02695` [7]

Funding This work was supported by NRF grant NRF-NRFI09-0005.

1 Introduction

A lattice in d dimensions is a discrete subgroup of $\mathbb{R}^d$. Formally, given a set of vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{Q}^d$, the lattice generated by these vectors is defined as

$$\mathcal{L} = \mathcal{L}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) := \left\{ \sum_{i=1}^{n} a_i \boldsymbol{v}_i : a_i \in \mathbb{Z} \right\}.$$

Several algorithmic problems on lattices are of particular importance. The **Shortest Vector Problem** (SVP) asks for the shortest nonzero vector in a given lattice – typically measured in an ℓ_p norm. The **Closest Vector Problem** (CVP) asks for the lattice vector closest to a given target vector in the ambient space $\mathbb{R}^d$. A related variant is the **Bounded Distance Decoding** (BDD) problem, which can be viewed as CVP under the promise that the target point lies sufficiently close to the lattice.

Algorithms for solving these problems have led to impactful applications across multiple domains. These include polynomial factoring [51], integer programming [52, 47, 34, 67], and cryptanalysis [69, 62, 46, 59]. At the same time, the conjectured hardness of lattice problems has enabled the design of powerful cryptographic primitives – particularly in the context of *post-quantum cryptography*. These schemes are not only conjectured to be secure against quantum adversaries but also support advanced functionalities such as fully homomorphic encryption [65, 39, 31, 14]. Remarkably, many of these constructions rely on the *worst-case hardness* of lattice problems, in contrast to traditional schemes based on the *average-case hardness* of factoring or discrete logarithm [12, 65].

The security of most finalists in the NIST post-quantum cryptography standardization process rests on the assumed difficulty of solving lattice problems [61].

Algorithmic Progress on Lattice Problems. The algorithmic study of CVP and SVP in the Euclidean norm (ℓ_2) began with the LLL algorithm [51], which provides a $2^{O(n)}$-approximation, and continued through foundational works of Babai [19], Kannan [47], Schnorr [68], and Ajtai, Kumar, and Sivakumar [13]. The fastest known algorithms for exact SVP are based on randomized sieving [13], extended to all ℓ_p norms [30], general norms [18], and even asymmetric convex bodies [34], yielding $2^{O(n)} \cdot \mathsf{poly}(d)$-time algorithms for SVP_p. In the case $p = 2$, a long line of work [13, 64, 58] culminated in a $2^{n+o(n)}$-time algorithm [5, 6, 10] for both exact SVP and exact CVP, a $2^{0.835n}$ quantum algorithm for SVP [3], while constant-factor approximation for SVP is achievable in $2^{n/2+o(n)}$ time [5]. Even faster heuristic sieving algorithms are known under plausible assumptions [60, 73, 50, 22]. For polynomial approximation factors, the best known algorithms for SVP_2 run in time 2^{Cn}, where the constant C depends critically on the approximation factor [38, 8, 9], and small improvements in C have significant implications for cryptographic security. Recently, Eisenbrand and Venzin [36] showed that the fastest constant approximation factor SVP_2 algorithm can be adapted to solve constant approximation factor SVP_p and CVP_p for any p with essentially the same runtime. Building on this [2] developed tight, rank- and dimension-preserving reductions between SVP_p and CVP_p across all norms, handling both constant and polynomial approximation factors, thereby expanding the algorithmic toolkit for lattice problems in general ℓ_p spaces.

Computational Hardness of Lattice Problems. A series of efforts [72, 15, 32, 57, 48, 66, 42] have shown that SVP_p and CVP_p are NP-hard to approximate to within any constant factor, and hard to approximate to within $n^{c/\log\log n}$ for a constant $c > 0$, under reasonable complexity theoretic assumptions. See [23] for a recent survey on the hardness of SVP. These results however do not rule out the existence of sub-exponential time algorithms for SVP. This question is of immense interest from a theoretical point of view, as well as from a practical (cryptographic) point of view. For instance, to break the minimally secure post-quantum cryptographic schemes currently being standardized (for example, [20]), one would need to solve SVP_2 in roughly 400 dimensions. At this stage, a $2^{n/\log n}$ time algorithm would be sufficient to break these schemes in practice. We need stronger and more *fine-grained* hardness assumptions to rule out the existence of such algorithms.

Fine Grained Complexity. The PCP theorem [17, 16] was a breakthrough result that states that every language in NP has a polynomial-size proof that can be verified by a probabilistic verifier that reads only a constant number of bits from the proof. This implies NP hardness of approximation for problems such as MAX3SAT and MAXLIN [41]. This rules out polynomial time algorithms for these problems (unless P = NP), but does not provide any guarantees about the existence of sub-exponential time algorithms, which are of interest especially in cryptography. The analogous (to P ≠ NP) assumption to start from would be the Exponential Time Hypothesis (ETH) [44], which states that 3SAT on n variables cannot be solved in time $2^{o(n)}$. However, it is not known whether this can be used to rule out the existence of sub-exponential time approximation algorithms for problems such as MAX3SAT or MAXLIN. More recently, [35, 55] formulated this into a yet stronger hardness assumption called the GapETH which states that *approximating* MAX3SAT on n variables requires $2^{\Omega(n)}$ time.

Gap-ETH vs ETH. The Exponential Time Hypothesis (ETH) and its stronger variant, Gap-ETH, have both played central roles in establishing fine-grained complexity and inapproximability results. Gap-ETH, introduced in [35], states that for some constant $0 < \eta < 1$, no algorithm can distinguish, in $2^{o(n)}$ time, between a 3SAT instance in which all m clauses are satisfiable and one in which no assignment satisfies more than an η-fraction of the clauses. This stronger assumption has enabled sharp inapproximability results for a wide range of problems, including 2CSP [35], Densest k-Subgraph and k-Biclique [33], parameterized SVP [28], and TSP [49]. More recently, however, breakthrough works have shown how to derive similar gap-producing reductions under the weaker ETH assumption, leading to inapproximability results for problems such as 2CSP, Gap k-Clique [40], gapMAXLIN [29], and parameterized SVP [53]. These results raise the intriguing possibility that ETH may in fact imply Gap-ETH – a connection that could be established, for instance, via a PCP for 3SAT with linear proof blowup, though the existence of such PCPs remains a long-standing open question. In light of this, a promising intermediate goal is to demonstrate that hardness results previously known only under Gap-ETH can in fact be obtained under ETH alone. Our work takes this approach: we use the gap-producing reduction from [29] to show the ETH hardness for the approximate lattice problems $\mathrm{SVP}_{p,\gamma}$, $\mathrm{CVP}_{p,\gamma}$, and $\mathrm{BDD}_{p,\alpha}$, whose hardness was previously established only under Gap-ETH. Prior ETH-based hardness for these fundamental problems was unknown, though a sequence of works [25, 11, 4, 27, 1] established their hardness under the stronger Gap-ETH assumption. Our results thus provide new evidence for the power of ETH and take a step toward bridging the gap between ETH and Gap-ETH in the context of lattice-based inapproximability.

1.1 Our Results

We study the hardness of lattice problems under the Exponential Time Hypothesis. Our results are summarized in Table 1.

At the heart of our results, is a recent breakthrough result of Bitansky et. al. [29] which shows that there is a polynomial time reduction from 3SAT in, say, n variables to gap version of MAXLIN problem with $\mathcal{O}(n)$ variables and $\mathcal{O}(n)$ clauses, which is a constraint satisfaction problem (CSP) where each clause is a linear equation over a finite field. It follows that (gap) MAXLIN is ETH hard.

ETH Hardness of CVP. We define $\mathrm{CVP}_{p,\gamma}$ to be a decision version of the Closest Vector Problem, where given a matrix B over, say, integers, a target vector $\boldsymbol{t}$, and a radius r, the goal is to decide if there exists a lattice vector $\boldsymbol{v} \in \mathcal{L}(B)$ such that $\|\boldsymbol{v} - \boldsymbol{t}\|_p \leq r$, or for all

Table 1 Summary of known fine-grained upper and lower bounds for $\mathsf{SVP}_{p,\gamma}$ and $\mathsf{CVP}_{p,\gamma}$ for various values of p and some constant $\gamma > 1$, under various assumptions, with our results in blue.

Problem	ℓ_p-norm	Gap-ETH	ETH	Notes
$\mathsf{SVP}_{p,\gamma}$	$2 < p < \infty$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	
	$1 \le p \le 2$	–	–	
	$p = \infty$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	[25]
$\mathsf{CVP}_{p,\gamma}$	$1 \le p < \infty$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	
	$p = \infty$	$2^{\Omega(n)}$	$2^{\Omega(n)}$	[25]

lattice vectors $\boldsymbol{v}$, $\|\boldsymbol{v} - \boldsymbol{t}\|_p > \gamma r$. Here $\mathcal{L}(B)$ is the lattice generated by the columns of B. Our first result says that for any $p \in [1, \infty)$, there is no sub-exponential (in the dimension of the lattice) algorithm for $\mathsf{CVP}_{p,\gamma}$ for some (explicit) constant $\gamma > 1$, unless the Exponential Time Hypothesis is false.

▶ **Theorem 1.1** (ETH hardness of $\mathsf{CVP}_{p,\gamma}$). *For any $p \in [1, \infty)$, there exists a constant $\gamma > 1$ such that for all $n \in \mathbb{Z}^+$, there is no $2^{o(n)}$ time algorithm for $\mathsf{CVP}_{p,\gamma}$ over $\mathbb{R}^n$, unless the Exponential Time Hypothesis is false.*

To prove this, we give a *deterministic* Karp reduction from MAXLIN over, say, n variables and m equations to $\mathsf{CVP}_{p,\gamma}$ in a lattice in m dimensions.

Randomized ETH Hardness of SVP. We define $\mathsf{SVP}_{p,\gamma}$ to be a decision version of the approximate shortest vector problem, where given a matrix B over, say, integers, and a radius r, the goal is to decide if there exists a lattice vector $\boldsymbol{v}$ such that $\|\boldsymbol{v}\|_p \le r$, or if for all lattice vectors $\boldsymbol{v}$, we have that $\|\boldsymbol{v}\|_p > \gamma r$. Our main result is that for any $p > 2$, unless the randomized Exponential Time Hypothesis is false, there is no sub-exponential time algorithm for $\mathsf{SVP}_{p,\gamma}$ for an explicit constant $\gamma > 1$.

▶ **Theorem 1.2** (ETH hardness of $\mathsf{SVP}_{p,\gamma}$). *For any $p \in (2, \infty)$, there exists a constant $\gamma > 1$ such that for all sufficiently large $n \in \mathbb{Z}^+$, there is no $2^{o(n)}$ time algorithm for $\mathsf{SVP}_{p,\gamma}$ over $\mathbb{R}^n$ unless the randomized Exponential Time Hypothesis is false.*

To prove this theorem we show a novel property of the integer lattice $\mathbb{Z}^n$, which says that for any $p > 2$ there are exponentially more lattice vectors close (in the ℓ_p norm) to $\frac{1}{2}\mathbf{1}_n$, than the number of short vectors, i.e., vectors around the origin. Then we borrow techniques from [11] to show that we can use the $\mathsf{CVP}_{p,\gamma}$ instances constructed in the proof of Theorem 1.1 to generate an $\mathsf{SVP}_{p,\gamma'}$ instance for some $\gamma' > 1$ via a Karp reduction. Chaining together the efficient reductions from 3SAT to (gap) MAXLIN, from (gap) MAXLIN to $\mathsf{CVP}_{p,\gamma}$ and from $\mathsf{CVP}_{p,\gamma}$ to $\mathsf{SVP}_{p,\gamma'}$, we get our result.

$p = \infty$ case. We note that the reduction from k-SAT to CVP_∞ and SVP_∞ in [25, Corollary 6.7] achieves a gap of $\gamma_p = 1 + 2/(k-1)$. Therefore for $k = 3$, they get a gap of 2, and hence we know the ETH hardness of $\mathsf{CVP}_{\infty,\gamma}$ and $\mathsf{SVP}_{\infty,\gamma}$ for $\gamma = 2$ since their work.

A reduction from CVP to BDD, and hence ETH Hardness of BDD. We define $\mathsf{BDD}_{p,\alpha}$ to be the following search problem. The input is a matrix B over, say, integers and a target vector $\boldsymbol{t}$ under the promise that there exists a lattice vector $\boldsymbol{v}$ at a distance at most $\alpha \cdot \lambda_1^{(p)}(\mathcal{L}(B))$, where $\lambda_1^{(p)}(\mathcal{L}(B))$ is the length of the shortest non-zero vector in $\mathcal{L}(B)$ in the

ℓ_p norm. The goal is to find a lattice vector closest to $\boldsymbol{t}$. We show that for any $p \in [1, \infty)$, there is an efficient decision-to-search reduction from $\mathsf{CVP}_{p,\gamma}$ over a lattice over integers to $\mathsf{BDD}_{p,\alpha}$, for any constant γ and $\alpha > \alpha_p^\ddagger$, where $\alpha_p^\ddagger$ is an explicit constant defined in Equation (4), such that $\alpha_p^\ddagger = 1$ for $p \in [1, 2]$, $\alpha_p^\ddagger < 1$ for $p > 2$, and $\alpha_p^\ddagger \to 1/2$ as $p \to \infty$.

▶ **Theorem 1.3** ($\mathsf{CVP}_{p,\gamma}$ reduces to $\mathsf{BDD}_{p,\alpha}$). *For any $\gamma' > 1$, $c > 0$, and $p \in [1, \infty)$, the following holds for all $\alpha > \alpha_p^\ddagger$ and sufficiently large $m \in \mathbb{Z}^+$. There is a decision-to-search reduction from any $\mathsf{CVP}_{p,\gamma'}$ to $\mathsf{BDD}_{p,\alpha}$, where the $\mathsf{CVP}_{p,\gamma'}$ instance $(B', \boldsymbol{t}', r')$ is such that $B' \in \mathbb{Z}^{m \times m'}$, $\boldsymbol{t}' \in \mathbb{Z}^m$ and $r' = c \cdot m^{1/p}$.*

As a consequence, combining this result with our reduction from (gap) MAXLIN to $\mathsf{CVP}_{p,\gamma}$, we get that for any $p \in [1, \infty)$, $\alpha > \alpha_p^\ddagger$, there is no sub-exponential time algorithm for $\mathsf{BDD}_{p,\alpha}$, unless the randomized Exponential Time Hypothesis is false.

▶ **Theorem 1.4** (ETH hardness of $\mathsf{BDD}_{p,\alpha}$). *For any $p \in [1, \infty)$, $\alpha > \alpha_p^\ddagger$, there is no $2^{o(n)}$ time algorithm for $\mathsf{BDD}_{p,\alpha}$ over $\mathbb{R}^n$, unless the randomized Exponential Time Hypothesis is false.*

ETH hardness of Minimum Distance Problem for Linear Codes. A linear code is a subspace of $\mathbb{F}_q^m$, for some prime power q. Given a full rank generator matrix $C \in \mathbb{F}_q^{m \times n}$, such that $1 \leq n \leq m$, the q-ary code generated by C is

$$\mathcal{C} := C\mathbb{F}_q^n = \left\{ C\boldsymbol{z} : z \in \mathbb{F}_q^n \right\}.$$

The elements of $\mathcal{C}$ are known as *codewords*. The nearest codeword problem (NCP) asks to find the minimal Hamming distance between a given target vector and a codeword in a given linear code. This problem can be thought of as the code equivalent of the closest vector problem (CVP) over lattices. Similarly, the SVP equivalent over codes is the minimum distance problem for linear codes (MDP), that asks to find the minimal Hamming weight of a non-zero code word in a given code. Observe that the MAXLIN problem is essentially the NCP problem in disguise.

In [70], the authors studied the SETH and GapETH hardness of MDP and NCP. They give a reduction from k-SAT (Max-k-SAT) to (γ-approximate) NCP, establishing SETH (GapETH) hardness of (γ approximate) NCP. Here $\gamma > 1$ is a constant. Moreover, they give a reduction from γ-NCP in rank n to γ'-MDP in rank Cn for constants C, γ', thereby showing GapETH hardness of γ'-MDP. We note that this reduction can be thought of as a reduction from $\mathsf{MAXLIN}_\varepsilon$ for a constant ε in n variables to γ-MDP in rank n, for another constant γ. Thus, we get ETH hardness of γ-MDP for some constant γ for free.

1.2 Our Techniques

From Linear Equations to CVP. MAXLIN is one of the classical NP-hard approximation problems [41]. An instance of MAXLIN is of the form $(M, \boldsymbol{v})$, where $M \in \mathbb{F}^{m \times n}$, $\boldsymbol{v} \in \mathbb{F}^m$ have entries from a finite field $\mathbb{F}$. The goal is to find a vector $\boldsymbol{x} \in \mathbb{F}^n$ that satisfies as many linear equations $M_i \cdot \boldsymbol{x} = v_i$ as possible. The gap version $\mathsf{gap}_{c,s}\mathsf{MAXLIN}$ is a promise problem where an instance $(M, \boldsymbol{v})$ is a YES instance if there exists a vector $\boldsymbol{x} \in \mathbb{F}^n$ that satisfies at least cm of the linear equations, and a NO instance if every vector $\boldsymbol{x} \in \mathbb{F}^n$ satisfies at most sm of the linear equations. [29] showed that there is a Karp reduction from 3SAT to $\mathsf{gap}_{c,s}\mathsf{MAXLIN}$ for $c = 5/8$ and $s = 5/8 - \varepsilon$ for an explicit constant $\varepsilon > 0$. This implies that $\mathsf{gap}_{c,s}\mathsf{MAXLIN}$ is ETH hard. From now on, we write $\mathsf{MAXLIN}_\varepsilon$ to denote $\mathsf{gap}_{c,s}\mathsf{MAXLIN}$ over $\mathbb{F}_2$, for $c = 5/8$ and $s = 5/8 - \varepsilon$.

To prove Theorem 1.1, we show a deterministic Karp reduction from $\mathsf{MAXLIN}_\varepsilon$ to $\mathsf{CVP}_{p,\gamma}$ for a constant $\gamma := (1 + 8\varepsilon/3)^{1/p}$. Consider the following matrix-vector pair

$$B := [M \quad 2I_m]; \quad \boldsymbol{t} := \boldsymbol{v}, \tag{1}$$

where I_m is the identity matrix. Consider the lattice generated by B. Appending $2I_m$ to M ensures that distances between the lattice points in $\mathcal{L}(B)$ to the target vector $\boldsymbol{t}$ are computed modulo 2. Precisely, the presence of $2I_m$ enforces that for any lattice point $B\boldsymbol{x}$, the coordinates of the difference $B\boldsymbol{x} - \boldsymbol{t}$ remains within an equivalence class mod 2, effectively reducing the computations to over $\mathbb{F}_2$. Therefore, we are able to show that if the input was a YES instance of $\mathsf{MAXLIN}_\varepsilon$, then there exists a lattice vector in $\mathcal{L}(B)$ that is at distance at most $r := (3m/8)^{1/p}$ from the target vector $\boldsymbol{t}$; and if the input was a NO instance of $\mathsf{MAXLIN}_\varepsilon$, then for all lattice vectors in $\mathcal{L}(B)$, their distance to the target vector $\boldsymbol{t}$ is at least γr. Thus, $(B, \boldsymbol{t}, r)$ is an instance of $\mathsf{CVP}_{p,\gamma}$.

Enter Sparsification: From MAXLIN to SVP. Aggarwal and Stephens-Davidowitz [11] showed GapETH hardness of $\mathsf{SVP}_{p,\gamma}$ for all $p > 2$. Using ideas from Khot [48], they reduced an instance of (gap) ExactSetCover over m sets and a universe of size k, to $\mathsf{SVP}_{p,\gamma}$, using an auxiliary *gadget*, a lattice-target pair $(B^\dagger, \boldsymbol{t}^\dagger)$ of dimension $d^\dagger$. Precisely, they used the input ExactSetCover instance to define a lattice $\widehat{\mathcal{L}} \subset \mathbb{R}^{m+k}$ generated by a matrix $\widehat{B}$, and a vector $\widehat{\boldsymbol{t}}$. Define

$$B := \begin{pmatrix} \widehat{B} & 0 \\ 0 & B^\dagger \end{pmatrix}; \quad \boldsymbol{t} := \begin{pmatrix} \widehat{\boldsymbol{t}} \\ \boldsymbol{t}^\dagger \end{pmatrix}.$$

Denote the number of vectors at a radius at most r around $\boldsymbol{t}$ in the lattice $\mathcal{L}(B)$ by $N_p(\mathcal{L}(B), r, \boldsymbol{t})$. They show that for a certain choice of $B^\dagger$, if the ExactSetCover instance is a YES instance, then $N_p(\mathcal{L}(B), r, \boldsymbol{t})$ is exponentially larger than $N_p(\mathcal{L}(B), \gamma r, \boldsymbol{t})$ if it were a NO instance. Then they define a (generating set of a) lattice

$$B' := \begin{pmatrix} B & \boldsymbol{t} \\ 0 & s \end{pmatrix},$$

for some s, and use a lattice *sparsification* algorithm (Section 2.5) to sample a random (sparser) sub-lattice $\mathcal{L}'' \subseteq \mathcal{L}(B')$ such that if the input ExactSetCover instance was a YES instance, at least one lattice vector of length at most r survives in $\mathcal{L}''$, and if it were a NO instance, then all lattice vectors of length at least γr die. Together with a reduction from Gap3SAT to ExactSetCover, they get the GapETH hardness of $\mathsf{SVP}_{p,\gamma}$ for all $p > 2$.

They pick the gadget $(B^\dagger, \boldsymbol{t}^\dagger)$ so that if the input ExactSetCover instance was a YES instance, then it blows up the number of short lattice vector exponentially, whereas if it were a NO instance, the number of short lattice vectors remains small. This corresponds to a gadget that has exponentially more close vectors than short vectors. (We usually call such a lattice-target pair a *locally dense* gadget.) They show that for all $p > 2$, the integer lattice $\mathbb{Z}^d$, along with the vector $\boldsymbol{t} := t \cdot \mathbf{1}_d$ for some $t \in (0, 1/2]$ satisfies this property. To see this, for any $\tau > 0, t \in [0, 1/2]$ define the theta function

$$\Theta_p(\tau, t) := \sum_{z \in \mathbb{Z}} \exp\left(-\tau \left|z - t\right|^p\right).$$

Notice that

$$\begin{aligned}\Theta_p(\tau, \boldsymbol{t}) := \Theta_p(\tau, t)^d &= \sum_{z_1,\ldots,z_d \in \mathbb{Z}} \exp\left(-\tau \sum_{i=1}^{d} |z_i - t|^p\right) \\ &\geq \sum_{\substack{\boldsymbol{z} \in \mathbb{Z}^d \\ \|\boldsymbol{z}-\boldsymbol{t}\|_p \leq r}} \exp\left(-\tau \|\boldsymbol{z} - \boldsymbol{t}\|_p^p\right) \\ &\geq \exp\left(-\tau r^p\right) \cdot N_p\left(\mathbb{Z}^d, r, \boldsymbol{t}\right).\end{aligned}$$

This implies that

$$N_p\left(\mathbb{Z}^d, r, \boldsymbol{t}\right) \leq \exp\left(\tau r^p\right) \cdot \Theta_p(\tau, t)^d. \tag{2}$$

Therefore, the theta function can be used to find an upper bound to the number of lattice vectors close to $\boldsymbol{t}$ in the ℓ_p norm. In fact, the above inequality is quite strict. It has been shown [56, 37, 11] that $\Theta_p(\tau, \boldsymbol{t})$ can be used to approximate the number of integer points in an ℓ_p ball up to sub-exponential factors. Thus, there exists a vector of the form $t \cdot \mathbf{1}_d$ for some $t \in (0, 1/2]$ such that there are exponentially more close lattice vectors in the integer lattice than short lattice vectors if and only if there exists a $\tau > 0$ and a $t \in (0, 1/2]$ such that $\Theta_p(\tau, t) > \Theta_p(\tau, 0)$. They show that this is true for all $p > 2$ [11, Section 6].

Unfortunately, a similar reduction breaks down if we start from $\mathsf{MAXLIN}_\varepsilon$ instead of ExactSetCover. This is because while counting the number of short (*annoying*) vectors in the no case (say A), they [11] exploit the fact that for any non-zero integer ℓ, $\mathrm{dist}_p(\widehat{\mathcal{L}}, \widehat{\ell \boldsymbol{t}})$ is sufficiently large. Therefore they only have to count the number of vectors in $\mathcal{L}(B^\dagger)$ around $\ell \boldsymbol{t}^\dagger$ in a much *smaller* radius. When we define the corresponding CVP instance in Equation (1), for every even integer 2ℓ, the vector $2\ell \boldsymbol{t} \in \mathcal{L}(B)$, and therefore we cannot use a similar trick.

We can, however, construct a gadget that has a stronger property: for every *even* multiple of the target, $2\ell \boldsymbol{t}^\dagger$, it holds that the number of lattice points in $\mathcal{L}(B^\dagger)$ around $\boldsymbol{t}^\dagger$ are exponentially more than those around $2\ell \boldsymbol{t}^\dagger$. This would compensate for the increase in radius incurred in our situation. In Section 4.1 we show that for all $p > 2$ the integer lattice ($\mathcal{L}(B^\dagger) = \mathbb{Z}^d$) and the target $\boldsymbol{t}^\dagger = \frac{1}{2}\mathbf{1}_d$ satisfies this property. This is because with this choice, every even multiple would be an integer vector. Thus, for some $r' < r$, if the gadget satisfies the property that $N_p\left(\mathcal{L}(B^\dagger), r', \boldsymbol{t}^\dagger\right)$ is exponentially more than $N_p\left(\mathcal{L}(B^\dagger), r, \mathbf{0}\right)$, then for every even integer 2ℓ, it holds that $N_p\left(\mathcal{L}(B^\dagger), r', \boldsymbol{t}^\dagger\right)$ is exponentially more than $N_p\left(\mathcal{L}(B^\dagger), r, 2\ell \boldsymbol{t}^\dagger\right)$.

One of the main technical ingredients, which could be of independent interest, is Theorem 4.1, where we use Fourier analysis to show that for any $p > 2$, there always exists $\tau > 0$ such that

$$\Theta_p(\tau, 1/2) > \Theta_p(\tau, 0).$$

Since $\Theta_p(\tau, 1/2)$ and $\Theta_p(\tau, 0)$ are absolutely convergent, this can be proved by analyzing the difference series

$$f_p(\tau) := \Theta_p(\tau, 1/2) - \Theta_p(\tau, 0) = -1 + 2\sum_{z=1}^{\infty}(e^{-\tau(z-1/2)^p} - e^{-\tau z^p}).$$

In other words, it suffices to show that for all $p > 2$, there always exists $\tau > 0$ such that $f_p(\tau) > 0$. When p is significantly bounded away from 2, e.g., for $p \geq 2.2$, we can simply take a constant $\tau = 2$, which yields

$$f_p(2) > -1 + 2(e^{-2(1/2)^p} - e^{-2}) \geq -1 + 2(e^{-2(1/2)^{2.2}} - e^{-2}) > 0,$$

since $e^{-2(z-1/2)^p} - e^{-2z^p} > 0$ for all $z \in \mathbb{Z}^+$. However, as p gets closer to 2, if we still fix some constant τ, we will need more than just the first term of the series, but the required number of terms will depend on p, making such an analysis no longer feasible.

Before illustrating our proof idea for p close to 2, let us first discuss the case for $p = 2$. Interestingly, the above conclusion is known to be false for $p = 2$ by [21, Corollary 1.2], i.e., $f_2(\tau) < 0$ for all $\tau > 0$. We now introduce a novel approach to handle the $p = 2$ case, which will also be instructive for the $p > 2$ case. The fourth Jacobi theta function is defined by

$$\vartheta_4(x, q) \coloneqq \sum_{z\in\mathbb{Z}} (-1)^z q^{z^2} e^{2zix}.$$

Note that $f_2(\tau) = -\vartheta_4(0, e^{-\tau/4})$. It is well-known that the fourth Jacobi theta function $\vartheta_4(x, q)$ can be expressed in terms of the second Jacobi theta function

$$\vartheta_2(x, q) := \sum_{z\in\mathbb{Z}} q^{(z+1/2)^2} e^{(2z+1)ix}$$

by Jacobi's imaginary transformation [45]. Specifically, we can obtain

$$\vartheta_4(0, e^{-\tau/4}) = \sqrt{\frac{4\pi}{\tau}}\vartheta_2(0, e^{-\frac{4\pi^2}{\tau}}) = \sqrt{\frac{4\pi}{\tau}} \cdot 2\sum_{z=0}^{\infty} e^{-(4\pi^2/\tau)\cdot(z+1/2)^2} > 0,$$

which implies $f_2(\tau) < 0$ for all $\tau > 0$ and concludes the $p = 2$ case. Essentially, the trick relies on Jacobi's imaginary transformation, which turns an alternating series into a non-alternating series with all positive terms. So, this motivates us to find a transformation that can also rewrite $f_p(\tau)$ for $p > 2$ in the spirit of the principles behind Jacobi's imaginary transformation. Note that Jacobi's imaginary transformation is obtained by the Poisson summation formula [63], which states that for any function[1] $s(z) \in L^1(\mathbb{R})$, if its Fourier transform $\hat{s}(x) \in L^1(\mathbb{R})$, then

$$\sum_{z\in\mathbb{Z}} s(z) = \sum_{x\in\mathbb{Z}} \hat{s}(x).$$

Following this new perspective, we outline the proof for the $p > 2$ case as follows. We first slightly rewrite the difference series $f_p(\tau) := \Theta_p(\tau, 1/2) - \Theta_p(\tau, 0)$ to match the Poisson summation formula and let $p = 2 + \varepsilon$ for $\varepsilon > 0$:

$$\begin{aligned} f_{2+\varepsilon}(\tau) &= -1 + 2\sum_{z=1}^{\infty}(e^{-\tau(z-1/2)^p} - e^{-\tau z^p}) \\ &= -\sum_{z\in\mathbb{Z}} \underbrace{e^{(-\tau/2^{2+\varepsilon})|z|^{2+\varepsilon}+\pi iz}}_{=:s_{\tau,\varepsilon}(z)}. \end{aligned}$$

By Lemmas 4.2 and 4.3, we show that both $s_{\tau,\varepsilon}(z), \hat{s}_{\tau,\varepsilon}(x) \in L^1(\mathbb{R})$, to ensure that the Poisson summation formula $\sum_{z\in\mathbb{Z}} s_{\tau,\varepsilon}(z) = \sum_{x\in\mathbb{Z}} \hat{s}_{\tau,\varepsilon}(x)$ can be applied. Then it suffices to show that for any $\varepsilon > 0$, there exists some sufficiently small $\tau > 0$ such that for every $x \in \mathbb{Z}$, $\hat{s}_{\tau,\varepsilon}(x) < 0$. To find such τ, we consider the asymptotic expansion for the Fourier transform of the super-Gaussian $g(z) := e^{-|z|^{2+\varepsilon}}$, i.e., $\hat{g}_\varepsilon(x) := \int_{-\infty}^{\infty} e^{-|z|^{2+\varepsilon}-2\pi ixz} dz$. This has been studied in [37] (restated as Lemma 2.2), which demonstrates that $\hat{g}_\varepsilon(x) < 0$ for all sufficiently

[1] $L^1(\mathbb{R})$ denotes the space of Lebesgue integrable functions on $\mathbb{R}$, i.e., $f \in L^1(\mathbb{R})$ if $\int_{-\infty}^{\infty} |f(x)|\, dx < \infty$.

large x. We then relate $\hat{s}_{\tau,\varepsilon}(x)$ to $\hat{g}_{\varepsilon}(x')$ where x' is obtained from x by a translation and a scaling that depend on τ and ε. Thus, for every $\varepsilon > 0$, we can choose an appropriate τ such that x' is large enough for all $x \in \mathbb{Z}$, and thereby conclude that $\hat{s}_{\tau,\varepsilon}(x) < 0$ for all $x \in \mathbb{Z}$. This implies that for every $\varepsilon > 0$, the difference series $f_{2+\varepsilon}(\tau) = -\sum_{x\in\mathbb{Z}} \hat{s}_{\tau,\varepsilon}(x) > 0$ for some $\tau > 0$, which completes the proof for Theorem 4.1.

Putting them together, we get Theorem 4.4, which says that for any $p > 2$, we can find a radius r such that for any d, the integer lattice contains exponentially many points within a distance r around $1/2 \cdot \mathbf{1}_d$ than around any of its *even multiple*. Using these gadgets in our reduction, Theorem 1.2 follows.

We leave $p = 2$ as an open problem. Note that the ETH hardness of $\mathsf{SVP}_{p,\gamma}$ for $p \in [1, 2)$ follows from the ETH hardness of $p = 2$ by the norm embedding techniques of [66].

Open Problem 1. Prove that for any $p \in [1, 2]$, there exists a constant $\gamma > 1$ such that assuming ETH there are no sub-exponential time algorithms for $\mathsf{SVP}_{p,\gamma}$.

On to BDD. Bounded distance decoding is a lattice problem that has found applications in showing hardness results for important cryptographic primitives such as *Learning With Errors* (LWE). It can be thought of as CVP under a promise that the closest lattice vector to a target $\boldsymbol{t}$ is not too far away from the lattice $\mathcal{L}(B)$, relative to the length of the shortest non-zero vector. Alternatively, it can be thought of as a *decoding* problem over lattices, analogous to decoding noisy code words over finite fields. Quantitatively, a $\mathsf{BDD}_{p,\alpha}$ instance promises that there is a closest lattice vector at a distance at most $\alpha\lambda_1^{(p)}(\mathcal{L}(B))$ from the lattice. (See Definition 2.8 for the formal definition.) Regev [65], in a seminal work, gave a reduction from worst case BDD to (an average case problem) LWE, with polynomial (in the dimension of the lattice) α. It is easy to see that there would be a unique solution to the problem if $\alpha < 1/2$. If $\alpha_1 > \alpha_2 > 0$, it is easy to see that $\mathsf{BDD}_{p,\alpha_2}$ reduces to $\mathsf{BDD}_{p,\alpha_1}$. Therefore, when showing hardness results for BDD, a lower α corresponds to a stronger result. NP-hardness of BDD for a constant α was first shown by [54], with $\alpha = \min\{2^{-1/2}, 2^{-1/p}\}$ for $p \geq 1$. This reduction however incurs a polynomial blowup in the rank of the lattice. Recently [26] studied the quantitative hardness of BDD for the first time and show that for all $p > 1$, there is no $2^{\Omega(n)}$ time algorithm for $\mathsf{BDD}_{p,\alpha}$ for all α greater than a certain constant that approaches $1/2$ as $p \to \infty$, unless randomized ETH is false. In a followup work [27], they show a similar result for improved values of α, under the GapETH assumption. Quantitatively, they define a constant $\alpha_p^{\ddagger}$ Equation (4) that depends on p, and show that for all $p > 1$, for all $\alpha > \alpha_p^{\ddagger}$ there is no sub-exponential time algorithm for $\mathsf{BDD}_{p,\alpha}$ unless GapETH is false. To get this result, they show a decision-to-search reduction from $\mathsf{CVP}'_{p,\gamma}$ to the decision version of $\mathsf{BDD}_{p,\alpha}$, where the goal is to decide if there is a vector at a distance at most $\alpha \cdot \lambda_1^{(p)}(\mathcal{L}(B))$. Here, CVP′ is a special case of CVP where in the YES case, there is a *binary* vector $\boldsymbol{x} \in \{0, 1\}^n$ such that $\|B\boldsymbol{x} - \boldsymbol{t}\|_p \leq r$, as compared to an arbitrary integer vector $\boldsymbol{x}$. For this they make use a $\mathsf{CVP}'_{p,\gamma}$ instance constructed from 3SAT in [25]. Precisely, given a rank n' instance of CVP′ $(B', \boldsymbol{t}', r')$, for some scalar parameters $s, l > 0$, they define a (generating set of a) lattice and target pair

$$B := \begin{pmatrix} sB' & 0 \\ I_{n'} & 0 \\ 0 & lB^{\dagger} \end{pmatrix}; \qquad \boldsymbol{t} := \begin{pmatrix} s\boldsymbol{t}' \\ \frac{1}{2}I_{n'} \\ l\boldsymbol{t}^{\dagger} \end{pmatrix},$$

where $(B^{\dagger}, \boldsymbol{t}^{\dagger})$ is again a locally dense lattice gadget. Similar to what we saw in case of SVP above, they are able to bound the number of lattice vectors close to the target $\boldsymbol{t}$ in the YES

case by, say G, which is exponentially larger than the number of lattice vectors close to $\boldsymbol{t}$ in the NO case. However, there is a technical difficulty here. We also need the fact that if the output BDD instance $(B, \boldsymbol{t}, r)$ is a YES instance, it satisfies the promise that the nearest lattice vector is at a distance at most $\alpha \cdot \lambda_1^{(p)}(\mathcal{L}(B))$. This corresponds to also bounding the number of very short vectors in the gadget in the YES case by some A. As before, they can then sparsify the lattice (Section 2.5), to get a sparse random sub-lattice such that if the input CVP′ instance was a yes instance then there is a lattice vector at a distance at most $\alpha \cdot \lambda_1^{(p)}(\mathcal{L}(B))$ from the target, with high probability.

In this work we are able to achieve the same lower bound of $\alpha_p^{\ddagger}$ from [27], *but under* ETH, by starting from a $\mathsf{CVP}_{p,\gamma}$ instance constructed from gap MAXLIN. To achieve this, we show a reduction from arbitrary $\mathsf{CVP}_{p,\gamma}$ instances over lattices over integers, $\mathcal{L} \subseteq \mathbb{Z}^n$. In fact, we are able to get a *simpler* reduction, in that we don't need to embed an integer lattice into our BDD instance anymore. Given a $\mathsf{CVP}_{p,\gamma}$ instance $(B', \boldsymbol{t}', r)$ we can define the following lattice

$$B := \begin{pmatrix} B' & 0 \\ 0 & sB^{\dagger} \end{pmatrix}; \qquad \boldsymbol{t} := \begin{pmatrix} \boldsymbol{t}' \\ s\boldsymbol{t}^{\dagger} \end{pmatrix},$$

where $(B^{\dagger}, \boldsymbol{t}^{\dagger})$ is again a locally dense lattice gadget. Using an analysis similar to that in case of $\mathsf{SVP}_{p,\gamma}$ in Section 4, and a locally dense integer gadget from [27], we get a desired reduction from $\mathsf{CVP}_{p,\gamma}$ to $\mathsf{BDD}_{p,\alpha}$, for all $p \geq 1$ and for all $\alpha > \alpha_p^{\ddagger}$. When we instantiate this reduction with the $\mathsf{CVP}_{p,\gamma}$ instance from Section 3, we find that there is no sub-exponential time algorithm for $\mathsf{BDD}_{p,\alpha}$, for all $p \geq 1$ and for all $\alpha > \alpha_p^{\ddagger}$, unless the (randomized) ETH is false.

2 Preliminaries

Whenever we say that certain constants are *efficiently computable*, we mean that they can be efficiently *approximated* to high precision.

For any set $\mathcal{L}$ we define $\mathcal{B}_p(\mathcal{L}, r, \boldsymbol{t}) := \mathcal{B}_p(r, \boldsymbol{t}) \cap \mathcal{L}$, where $\mathcal{B}_p(r, \boldsymbol{t})$ denotes a ball in $\mathbb{R}^n$ centered at $\boldsymbol{t}$ of radius r in the ℓ_p norm, i.e., $\mathcal{B}_p(r, \boldsymbol{t}) := \left\{ \boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{t}\|_p \leq r \right\}$. For a discrete set $\mathcal{L}$, we define

$$N_p(\mathcal{L}, r, \boldsymbol{t}) := |\mathcal{B}_p(r, \boldsymbol{t}) \cap \mathcal{L}|.$$

For any matrix $B \in \mathbb{R}^{m \times n}$, we write $\mathcal{L}(B)$ to denote the lattice generated by the columns of B. We write $\mathbf{1}_n$ and $\mathbf{0}_n$ to denote the vector all 1s and all 0s vectors in n dimensions respectively. We write $\lambda_1^{(p)}(\mathcal{L})$ for the length of the *shortest non-zero vector with respect to ℓ_p-norm* in the lattice $\mathcal{L}$. For vectors $\boldsymbol{v}_1 \in \mathbb{R}^n$, $\boldsymbol{v}_2 \in \mathbb{R}^m$, we write $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ for their concatenation: $(\boldsymbol{v}_1^{\top}, \boldsymbol{v}_2^{\top})^{\top}$. Unless otherwise specified, all logarithms are base e. For a discrete set $\mathcal{L} \subset \mathbb{R}^n$ and a vector $\boldsymbol{t} \in \mathbb{R}^n$ we define the distance between them to be the minimum distance between the vector $\boldsymbol{t}$ and any point in the set:

$$\mathrm{dist}_p(\boldsymbol{t}, \mathcal{L}) := \min_{\boldsymbol{x} \in \mathcal{L}} \|\boldsymbol{x} - \boldsymbol{t}\|_p.$$

2.1 Asymptotic Expansion

We now define the notion of Poincaré Asymptotic Expansion, which we will use in Section 4.1. An asymptotic expansion describes the asymptotic behavior of a function in terms of a sequence of (gauge) functions [43, Definition 2.3].

▶ **Definition 2.1** (Poincaré Asymptotic Expansion). *A sequence of functions $\varphi_n : \mathbb{R} \setminus 0 \to \mathbb{R}$, where $n = 0, 1, 2, \ldots$, is a asymptotic sequence as $x \to \infty$ if for each $n = 0, 1, 2, \ldots$ we have*

$$\varphi_{n+1} = o(\varphi_n) \quad \text{as } x \to \infty.$$

Moreover, if $\{\varphi_n\}$ is an asymptotic sequence and $f : \mathbb{R} \setminus 0 \to \mathbb{R}$ is a function, we write

$$f(x) \sim \sum_{n=0}^{\infty} a_n \varphi_n(x) \quad \text{as } x \to \infty \tag{3}$$

if for each $N = 0, 1, 2, \ldots$ we have

$$f(x) - \sum_{n=0}^{N} a_n \varphi_n(x) = o(\varphi_N) \quad \text{as } x \to \infty.$$

We call Equation (3) the asymptotic expansion of f with respect to $\{\varphi_n\}$ as $x \to \infty$.

For instance, the functions $\varphi_n(x) = x^{-n}$ form an asymptotic sequence as $x \to \infty$. We will use the asymptotic expansion of the Fourier transform of super-Gaussian functions as shown in [37].

▶ **Lemma 2.2** ([37], Lemma 10 (i)). *For fixed $\varepsilon > 0$, the asymptotic expansion of*

$$\hat{g}_\varepsilon(x) := \int_{-\infty}^{\infty} e^{-|z|^{2+\varepsilon} - 2\pi i x z} dz$$

as $|x| \to \infty$ is

$$\hat{g}_\varepsilon(x) \sim 2 \sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m!} \sin\left(\frac{m\pi(2+\varepsilon)}{2}\right) \Gamma(m(2+\varepsilon)+1)(2\pi|x|)^{-m(2+\varepsilon)-1}.$$

2.2 Computational Problems

For an integer $k \geq 2$, a k-SAT formula over n boolean variables is the conjunction of clauses, where each clause is the disjunction of k literals. That is, k-SAT formulas have the form $\bigwedge_{i=1}^{m} \bigvee_{j=1}^{k} b_{i,j}$, where $b_{i,j} = x_k$ or $b_{i,j} = \neg x_k$ for some boolean variable x_k.

▶ **Definition 2.3.** *For any $k \geq 2$, the decision problem k-SAT is defined as follows. The input is a k-SAT formula. It is a YES instance if there exists an assignment to the variables that makes the formula evaluate to true and a NO instance otherwise.*

▶ **Definition 2.4.** *For any $k \geq 2$, the decision problem Max-k-SAT is defined as follows. The input is a k-SAT formula and an integer $S \geq 1$. It is a YES instance if there exists an assignment to the variables such that at least S of the clauses evaluate to true and a NO instance otherwise.*

Notice that k-SAT is a special case of Max-k-SAT. We write $\mathsf{3SAT}_C$ for a 3SAT instance such that it contains at most Cn clauses.

▶ **Definition 2.5.** *An instance of a $\mathsf{gap}_{(c,s)}$MAXLIN over $\mathbb{F}_2$ consists of an $m \times n$ matrix $A \in \{0,1\}^{m \times n}$ and a vector $\boldsymbol{b} = \{0,1\}^m$ constraints. It is a YES instance if there exists an $\boldsymbol{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$ that satisfies at least $c \cdot m$ of the m constraints*

$$\sum_{j=1}^{n} A_{i,j} \cdot x_j = b_i \pmod 2,$$

for $i = 1$ to m. It is a NO instance if for all $\boldsymbol{x} \in \mathbb{Z}^n$,[2] at most $s \cdot m$ constraints are satisfied.

[2] Note that this is equivalent to making the same statement for all $\boldsymbol{x} \in \{0,1\}^n$ since the equations are considered modulo 2.

▶ **Definition 2.6** (Shortest Vector Problem ($\mathsf{SVP}_{p,\gamma}$)). *For any $p \in [1,\infty]$ and any $\gamma \geq 1$,* the γ-approximate Shortest Vector Problem in the ℓ_p norm ($\mathsf{SVP}_{p,\gamma}$) *is a promise problem defined as follows. The input is a matrix $B \in \mathbb{Q}^{d\times n}$ generating a lattice $\mathcal{L} \subset \mathbb{R}^d$ of rank n and a length $r > 0$. It is a* YES *instance if $\lambda_1^{(p)}(\mathcal{L}) \leq r$ and a* NO *instance if $\lambda_1^{(p)}(\mathcal{L}) > \gamma r$.*

▶ **Definition 2.7** (Closest Vector Problem ($\mathsf{CVP}_{p,\gamma}$)). *For any $p \in [1,\infty]$ and any $\gamma \geq 1$,* the γ-approximate Closest Vector Problem in the ℓ_p norm ($\mathsf{CVP}_{p,\gamma}$) *is a promise problem defined as follows. The input is a matrix $B \in \mathbb{Q}^{d\times n}$ generating a lattice $\mathcal{L} \subset \mathbb{R}^d$ of rank n, a target $\boldsymbol{t} \in \mathbb{R}^d$, and a distance $r > 0$. It is a* YES *instance if $\mathrm{dist}_p(\boldsymbol{t},\mathcal{L}) \leq r$ and a* NO *instance if $\mathrm{dist}_p(\boldsymbol{t},\mathcal{L}) > \gamma r$.*

Note that in Definition 2.6 and Definition 2.7, the input is a matrix B that *generates* the lattice $\mathcal{L}$. This is equivalent to the more standard definition where the input is a basis, i.e. a set of linearly independent vectors that generates the lattice. Given a generating set B, a basis can be efficiently (in the bit length of the representation of B) computed from the generating set using the LLL algorithm [51] (c.f. [24, Algorithm 1].)

▶ **Definition 2.8** (Bounded Distance Decoding ($\mathsf{BDD}_{p,\alpha}$)). *For $p \in [1,\infty]$ and $\alpha = \alpha(n) > 0$, the search problem $\mathsf{BDD}_{p,\alpha}$ is defined as follows. The input is a (generating set for a) lattice $\mathcal{L} \subset \mathbb{R}^d$ and a target $\boldsymbol{t} \in \mathbb{R}^d$ satisfying*

$$\mathrm{dist}_p(\boldsymbol{t},\mathcal{L}) \leq \alpha \cdot \lambda_1^{(p)}(\mathcal{L}),$$

and the goal is to find a closest lattice vector $\boldsymbol{v} \in \mathcal{L}$ to $\boldsymbol{t}$ such that

$$\|\boldsymbol{t} - \boldsymbol{v}\|_p = \mathrm{dist}_p(\boldsymbol{t},\mathcal{L}).$$

2.3 Fine-grained Complexity

▶ **Theorem 2.9** ([29], Theorem 6.3). *For some $\varepsilon \in (0,1)$ there exists a polynomial time reduction from $\mathsf{3SAT}_C$ with n variables to $\mathsf{gap}_{c,s}\mathsf{MAXLIN}$ over $\mathbb{F}_2$ with $\mathcal{O}(n)$ variables and $\mathcal{O}(n)$ equations, where $c = 5/8$ and $s = 5/8 - \varepsilon$.*

Throughout this text, we write $\mathsf{MAXLIN}_\varepsilon$ for $\mathsf{gap}_{c,s}\mathsf{MAXLIN}$ over $\mathbb{F}_2$ with $c = 5/8$ and $s = 5/8 - \varepsilon$. Impagliazzo and Paturi introduced the following celebrated and well-studied hypothesis concerning the fine-grained complexity of k-SAT [44]. We will also need the randomized variant, which talks about the existence of randomized algorithms instead of deterministic ones.

▶ **Definition 2.10** (Exponential Time Hypothesis (ETH)). *The (randomized) Exponential Time Hypothesis ((randomized) ETH) asserts that, there exists $\delta > 0$ such that any (randomized) algorithm which solves* 3-SAT *must take $2^{\delta n}$ time.*

▶ **Definition 2.11** (GapETH). *The (randomized)* GapETH *asserts that there exists $\delta > 0$ and $0 < \eta < 1$ such that given a* 3-SAT *instance with n variables and m clauses, any (randomized) algorithm which can distinguish between the cases if all m clauses are satisfiable and one in which no assignment satisfies more than η-fraction of the clauses, must take $2^{\delta n}$ time.*

▶ **Lemma 2.12** (Sparsification Lemma [44]). *Let $\varepsilon > 0$, $k \geq 3$ be constants. There is a $2^{\varepsilon n} \cdot poly(n)$ time algorithm that takes a k-CNF F on n variables and produces $F_1, \ldots, F_{2^{\varepsilon n}}$, $2^{\varepsilon n}$ k-CNFs such that F is satisfied if and only if $\bigvee_i F_i$ is satisfied and each F_i has n variables and $n \cdot \left(\frac{k}{\varepsilon}\right)^{O(k)}$ clauses. In fact, each variable is in at most poly$\left(\frac{1}{\varepsilon}\right)$ clauses, and the F_i are over the same variables as F.*

The sparsification Lemma 2.12 and Tovey's reduction [71] together tell us that if ETH is true, then $3SAT_4$ over n variables can't be solved in $2^{o(n)}$ time. Together with Theorem 2.9, we find that if ETH holds, then for some $\varepsilon > 0$, any algorithm which solves $\mathsf{MAXLIN}_\varepsilon$ with n variables, $m = O(n)$ clauses, must take $2^{\delta n}$ time for some $\delta > 0$. We state it as the following corollary.

▶ **Corollary 2.13** ($\mathsf{MAXLIN}_\varepsilon$ is ETH Hard). *There exists constants $\varepsilon > 0, C > 0$ such that unless* ETH *is false, there is no $2^{o(n)}$-time algorithm for $\mathsf{MAXLIN}_\varepsilon$ with n variables and $m = Cn$ equations.*

2.4 Counting Lattice Points

We now define the Θ and the μ functions, and show that they can be used to approximate the number of lattice points within a given radius.

For any $p \in [1, \infty), \tau > 0$, and $t \in \mathbb{R}$ define the theta function to be

$$\Theta_p(\tau, t) := \sum_{z \in \mathbb{Z}} \exp(-\tau |z - t|^p).$$

Notice that without loss of generality, we can assume $t \in [0, 1/2]$. For a vector $\boldsymbol{t} \in \mathbb{R}^n$ we can analogously define

$$\Theta_p(\tau, \boldsymbol{t}) := \prod_{i \in [n]} \Theta_p(\tau, t_i)$$

Clearly the theta function[3]

$$\Theta_p(\tau, \boldsymbol{t}) = \sum_{\boldsymbol{v} \in \mathbb{Z}^n} \exp(-\tau \left\| \boldsymbol{v} - \boldsymbol{t} \right\|_p^p)$$

The Theta function acts as a smooth proxy for the point counting function, where the parameter τ plays the role of inverse radius.

For any $p \in [1, \infty), \tau > 0$ and $t \in [0, 1/2]$, define

$$\mu_p(\tau, t) := \mathbb{E}_{X \sim D_p(\tau, t)}[|X|^p] = \frac{1}{\Theta_p(\tau, t)} \cdot \sum_{z \in \mathbb{Z}} |z - t|^p \cdot \exp(-\tau |z - t|^p)$$

where, $D_p(\tau, t)$ is the probability distribution over $\mathbb{Z} - t$ that assigns probability $\exp(-\tau |x|^p)/\Theta_p(\tau, t)$ to $x \in \mathbb{Z} - t$. For $\boldsymbol{t} \in \mathbb{R}^n$ this extends as following

$$\mu_p(\tau, \boldsymbol{t}) := \sum_{i=1}^{n} \mathop{\mathbb{E}}_{X \sim D_p(\tau, t_i)}[|X|^p]$$

Notice that if $\boldsymbol{t} = t \cdot \mathbf{1}_n$ for some $t \in [0, 1/2]$, we have that $\mu(\boldsymbol{t}) = n\mu(t)$.

▶ **Lemma 2.14.** *For any $p \in [1, \infty)$, $r > 0$,$\tau > 0$ and $\boldsymbol{t} \in \mathbb{R}^n$, we have that*

$$N_p(\mathbb{Z}^n, r, \boldsymbol{t}) \leq \exp(\tau r^p) \Theta_p(\tau, \boldsymbol{t})$$

We defer the proof to the full version [7]. The upper bound in the previous lemma is quite tight. In fact, we know the following theorem.

[3] Notice that this is closely related to the discrete Gaussian function $\rho_s(\mathbb{Z} - \boldsymbol{t}) := \sum_{\boldsymbol{z} \in \mathbb{Z}^n} \exp\left(-\pi \left\| \boldsymbol{z} - \boldsymbol{t} \right\|^2 / s^2\right)$.

▶ **Theorem 2.15** ([11], Theorem 6.1). *For any constants $p \geq 1$ and $\tau > 0$, there is another constant $C^* > 0$ such that for any $\boldsymbol{t} \in \mathbb{R}^n$ and any positive integer n.*

$$\exp\left(\tau\mu_p(\tau,\boldsymbol{t}) - C^*\sqrt{n}\right)\cdot\Theta_p(\tau,\boldsymbol{t}) \;\leq\; N_p\left(\mathbb{Z}^n, \mu_p(\tau,\boldsymbol{t})^{1/p}, \boldsymbol{t}\right) \;\leq\; \exp\left(\tau\mu_p(\tau,\boldsymbol{t})\right)\cdot\Theta_p(\tau,\boldsymbol{t})$$

Next we define the β function, which we will use to define the threshold $\alpha_p^\ddagger$ above which we are able to show hardness results for $\mathsf{BDD}_{p,\alpha}$.

▶ **Definition 2.16** ([27]). *For $p \in [1,\infty)$, $t \in [0,1/2]$, and $a \geq 0$, we define $\beta_{p,t}(a)$ as follows.*

1. *For $a < t$, define $\beta_{p,t}(a) := 0$.*
2. *For $a = t$, define $\beta_{p,1/2}(1/2) := 2$ and for $t \neq 1/2$ define $\beta_{p,t}(t) := 1$.*
3. *For $a > t$, define*

$$\beta_{p,t}(a) := \exp(\tau^* a^p)\cdot\Theta_p(\tau^*, t),$$

where $\tau^* > 0$ is the unique solution to $\mu_p(\tau^*, t) = a^p$.

▶ **Definition 2.17** ([27]). *For $p \in [1,\infty)$, define*

$$\alpha_p^\ddagger := \inf_{\substack{t\in[0,1/2]\\ a\geq t}} \frac{a}{\beta_{p,0}^{-1}(\beta_{p,t}(a))}\,. \tag{4}$$

We note that the functions Θ, β and μ can be efficiently approximated to within high precision. Throughout this text, we deal with constants A, G, which are the number of vectors in the lattice of a particular length, and will functions of Θ. Thus they will be computable efficiently to a high precision.

2.5 Lattice Sparsification

Khot introduced the idea of lattice sparsification [48], which is a randomized process that given a lattice $\mathcal{L}$ lets us sample a sub-lattice $\mathcal{L}' \subseteq \mathcal{L}$ that has a lot fewer points in any fixed radius, large enough ℓ_p-ball. It works by taking a random hyperplane over a finite field $\mathbb{F}_q$, for some prime power q, and restricting the coefficients of the lattice vectors to belong to the particular hyperplane. Formally, we prove and use the following statement.

▶ **Lemma 2.18.** *For any $p \in [1,\infty)$, there is an efficient algorithm that takes as input a (basis for a) full rank lattice $\mathcal{L} \subset \mathbb{R}^n$ of rank n and a prime number q, and outputs a (basis for a) sub-lattice $\mathcal{L}' \subseteq \mathcal{L}$ of rank n such that for any finite set $\mathcal{S} \subset \mathcal{L}$ if $\forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathcal{S}$ it holds that $B^{-1}\boldsymbol{v}_1 \mod q$ and $B^{-1}\boldsymbol{v}_2 \mod q$ are non-zero and pairwise linearly independent over $\mathbb{F}_q$, then*

$$1 - \frac{q}{|\mathcal{S}|} \leq \Pr\left[\exists \boldsymbol{v} \in \mathcal{S} : \boldsymbol{v} \in \mathcal{L}'\right] \leq \frac{|\mathcal{S}|}{q}.$$

We defer the proof to the full version [7]. We also use the following treatment of lattice sparsification from [27].

▶ **Lemma 2.19** ([27], Proposition 2.5). *Let $p \in [1,\infty)$, let $\mathcal{L}$ be a lattice of rank n with basis B, let $\boldsymbol{t} \in \mathrm{span}(\mathcal{L})$, let q be a prime, and let $r \geq 0$. Let $\boldsymbol{x}, \boldsymbol{z} \sim \mathbb{F}_q^n$ be sampled uniformly at random, and define*

$$\mathcal{L}' := \{\boldsymbol{v} \in \mathcal{L} : \langle B^+\boldsymbol{v}, \boldsymbol{x}\rangle \equiv 0 \pmod q\}, \quad \boldsymbol{t}' := \boldsymbol{t} - B\boldsymbol{z}.$$

1. *If* $r \le q\lambda_1^{(p)}(\mathcal{L})$, *then*

$$\Pr[\lambda_1(\mathcal{L}') \le r] \le \frac{N_p(\mathcal{L}, r, \mathbf{0})}{q}. \tag{5}$$

2. *If* $r < q\lambda_1^{(p)}(\mathcal{L})/2$, *then* ,

$$\Pr[\mathrm{dist}_p(\boldsymbol{t}', \mathcal{L}') > r] \le \frac{q}{N_p(\mathcal{L}, r, \boldsymbol{t})} + \frac{1}{q^n}. \tag{6}$$

3.

$$\Pr[\mathrm{dist}_p(\boldsymbol{t}', \mathcal{L}') \le r] \le \frac{N_p(\mathcal{L}, r, \boldsymbol{t})}{q} + \frac{1}{q^n}. \tag{7}$$

3 ETH hardness of $\mathsf{CVP}_{p,\gamma}$

In the full version of this paper [7], we show a reduction from the gap MAXLIN problem to the $\mathsf{CVP}_{p,\gamma_p}$ and thereby conclude that $\mathsf{CVP}_{p,\gamma_p}$ is hard under ETH. Precisely, we have the following two results.

▶ **Theorem 3.1.** *For any* $p \in [1, \infty)$, *there exists a constant* $\gamma_p > 1$ *such that for all* $n \in \mathbb{Z}^+$, *there is a polynomial time Karp reduction from* $\mathsf{MAXLIN}_\varepsilon$ *in* n *variables and* m *equations to* $\mathsf{CVP}_{p,\gamma_p}$ *over* $\mathbb{R}^m$.

Together with Corollary 2.13, we find the following.

▶ **Theorem 1.1** (ETH hardness of $\mathsf{CVP}_{p,\gamma}$)**.** *For any* $p \in [1, \infty)$, *there exists a constant* $\gamma > 1$ *such that for all* $n \in \mathbb{Z}^+$, *there is no* $2^{o(n)}$ *time algorithm for* $\mathsf{CVP}_{p,\gamma}$ *over* $\mathbb{R}^n$, *unless the Exponential Time Hypothesis is false.*

4 ETH hardness of $\mathsf{SVP}_{p,\gamma}$

In this section we sketch our reduction from 3SAT in n variables to $\mathsf{SVP}_{p,\gamma}$ in a lattice of rank $\mathcal{O}(n)$. We do this by a reduction from $\mathsf{CVP}_{p,\gamma'}$ for a constant γ' (an instance obtained in Section 3) to $\mathsf{SVP}_{p,\gamma}$ for a constant γ. The result then follows from combining the reduction from 3SAT to MAXLIN in Theorem 2.9, and from MAXLIN to CVP in Theorem 3.1. We first show that for $p > 2$, the integer lattice with the all half vector as the target forms a (family of) locally dense lattice gadgets with certain quantitative properties that will be useful in our reduction. We will defer the proofs to the full version [7].

4.1 Locally Dense Integer Gadget

We show that in the integer lattice $\mathbb{Z}^n$, there are exponentially more vectors close to $\frac{1}{2}\mathbf{1}_n$ than the number of *short* vectors. We first show the following property of the Theta function. This might be of independent interests.

▶ **Theorem 4.1.** *For every* $p \in (2, \infty)$, *there always exists* $\tau > 0$ *such that*

$$\Theta_p(\tau, 0) < \Theta_p(\tau, 1/2). \tag{8}$$

Proof. From the definition,

$$\Theta_p(\tau,0)=\sum_{z\in\mathbb{Z}}e^{-\tau|z|^p}=1+2\sum_{z=1}^{\infty}e^{-\tau\cdot z^p},$$

and

$$\Theta_p(\tau,1/2)=\sum_{z\in\mathbb{Z}}e^{-\tau|z-1/2|^p}=2\sum_{z=1}^{\infty}e^{-\tau\cdot(z-1/2)^p}.$$

It is straightforward to see $\Theta_p(\tau,0)$ and $\Theta_p(\tau,1/2)$ are absolutely convergent; therefore we can define the following function $f_p(\tau)$ on $\tau\in(0,\infty)$ that represents the difference between the two series, i.e.,

$$f_p(\tau):=\Theta_p(\tau,1/2)-\Theta_p(\tau,0)=-1+2\sum_{z=1}^{\infty}\left[e^{-\tau(z-1/2)^p}-e^{-\tau z^p}\right]=-1+2\sum_{z=1}^{\infty}(-1)^{z-1}e^{-\tau(z/2)^p}.$$

Let $p=2+\varepsilon$ for $\varepsilon>0$, and write $f_p(\tau)$ as

$$f_{2+\varepsilon}(\tau)=-1+2\sum_{z=1}^{\infty}(-1)^{z-1}e^{-\tau(z/2)^{2+\varepsilon}}=-\sum_{z\in\mathbb{Z}}(-1)^z e^{-\tau|z/2|^{2+\varepsilon}}=-\sum_{z\in\mathbb{Z}}e^{(-\tau/2^{2+\varepsilon})|z|^{2+\varepsilon}+\pi iz}.$$

It suffices to prove that for all $\varepsilon>0$, there always exists $\tau>0$ such that $f_p(\tau)>0$.

Case 1: ($\varepsilon\in(0,2)$). We will use the Poisson summation formula, which states that for any function $s(z)\in L^1(\mathbb{R})$, if its Fourier transform $\hat{s}(x):=\int_{-\infty}^{\infty}s(z)e^{-2\pi ixz}dz\in L^1(\mathbb{R})$, then

$$\sum_{z\in\mathbb{Z}}s(z)=\sum_{x\in\mathbb{Z}}\hat{s}(x).$$

Define $C_{\tau,\varepsilon}:=\tau/2^{2+\varepsilon}>0$ and $s_{\tau,\varepsilon}(z):=e^{-C_{\tau,\varepsilon}|z|^{2+\varepsilon}+\pi iz}$. In other words, $f_{2+\varepsilon}(\tau)=-\sum_{z\in\mathbb{Z}}s_{\tau,\varepsilon}(z)$. Now we consider the Fourier transform of $s_{\tau,\varepsilon}(z)$, i.e.,

$$\hat{s}_{\tau,\varepsilon}(x):=\int_{-\infty}^{\infty}s_{\tau,\varepsilon}(z)e^{-2\pi ixz}dz=\int_{-\infty}^{\infty}\exp\left(-C_{\tau,\varepsilon}|z|^{2+\varepsilon}-2\pi i(x-1/2)z\right)dz.$$

Recall that the Fourier transform of $e^{-|z|^{2+\varepsilon}}$, $\hat{g}_\varepsilon(x):=\int_{-\infty}^{\infty}e^{-|z|^{2+\varepsilon}-2\pi ixz}dz$, has been well-studied in [37]. This motivates us to relate $\hat{s}_{\tau,\varepsilon}(x)$ to $\hat{g}_\varepsilon(x)$ by some translation and scaling as follows:

$$\begin{aligned}
\hat{s}_{\tau,\varepsilon}(x)&=\int_{-\infty}^{\infty}\exp\left(-C_{\tau,\varepsilon}|z|^{2+\varepsilon}-2\pi i(x-1/2)z\right)dz\\
&=\int_{-\infty}^{\infty}\exp\left(-|C_{\tau,\varepsilon}^{1/(2+\varepsilon)}z|^{2+\varepsilon}-2\pi i\frac{x-1/2}{C_{\tau,\varepsilon}^{1/(2+\varepsilon)}}C_{\tau,\varepsilon}^{1/(2+\varepsilon)}z\right)dz\\
&=\frac{1}{C_{\tau,\varepsilon}^{1/(2+\varepsilon)}}\int_{-\infty}^{\infty}\exp\left(-|C_{\tau,\varepsilon}^{1/(2+\varepsilon)}z|^{2+\varepsilon}-2\pi i\frac{x-1/2}{C_{\tau,\varepsilon}^{1/(2+\varepsilon)}}C_{\tau,\varepsilon}^{1/(2+\varepsilon)}z\right)d\left(C_{\tau,\varepsilon}^{1/(2+\varepsilon)}z\right)\\
&=\frac{1}{C_{\tau,\varepsilon}^{1/(2+\varepsilon)}}\int_{-\infty}^{\infty}\exp\left(-|z|^{2+\varepsilon}-2\pi i\frac{x-1/2}{C_{\tau,\varepsilon}^{1/(2+\varepsilon)}}z\right)dz=\frac{1}{C_{\tau,\varepsilon}^{1/(2+\varepsilon)}}\cdot\hat{g}_\varepsilon\left(\frac{x-1/2}{C_{\tau,\varepsilon}^{1/(2+\varepsilon)}}\right).
\end{aligned}$$

To be eligible for the Poisson summation formula $\sum_{z\in\mathbb{Z}}s_{\tau,\varepsilon}(z)=\sum_{x\in\mathbb{Z}}\hat{s}_{\tau,\varepsilon}(x)$, we need to show that for any $\tau>0$ and $\varepsilon>0$, $s_{\tau,\varepsilon},\hat{s}_{\tau,\varepsilon}\in L^1(\mathbb{R})$, which is given by the following lemmas. We defer the proof to the full version [7].

▶ **Lemma 4.2.** *For any $\tau > 0$ and $\varepsilon > 0$, $s_{\tau,\varepsilon} \in L^1(\mathbb{R})$.*

▶ **Lemma 4.3.** *For any $\tau > 0$ and $\varepsilon > 0$, $\hat{s}_{\tau,\varepsilon} \in L^1(\mathbb{R})$.*

To finish the proof of this case, recall from Lemma 2.2 that

$$\hat{g}_\varepsilon(x) \sim 2\sum_{m=1}^{\infty} \frac{(-1)^{m+1}}{m!} \sin\left(\frac{m\pi(2+\varepsilon)}{2}\right) \Gamma(m(2+\varepsilon)+1)(2\pi|x|)^{-m(2+\varepsilon)-1},$$

and observe that for any $x \in \mathbb{R}$ and $\varepsilon \in (0,2)$, the first term in the asymptotic expansion of $\hat{g}_\varepsilon(x)$ is negative as $\sin(\pi(2+\varepsilon)/2) < 0$. By Definition 2.1, there exists $N_0 \in \mathbb{R}$ such that for all $|x| > N_0$, $\hat{g}_\varepsilon(x) < 0$. Therefore, we can always choose $\tau_0 \in (0, N_0^{-(2+\varepsilon)})$, which satisfies that for all $x \in \mathbb{Z}$,

$$N_0 < \frac{1}{\tau_0^{1/(2+\varepsilon)}} = \frac{1}{2C_{\tau_0,\varepsilon}^{1/(2+\varepsilon)}} \leq \left|\frac{x-1/2}{C_{\tau_0,\varepsilon}^{1/(2+\varepsilon)}}\right|.$$

Since $C_{\tau_0,\varepsilon}^{1/(2+\varepsilon)} > 0$, we have that for all $x \in \mathbb{Z}$,

$$\hat{s}_{\tau_0,\varepsilon}(x) = \frac{1}{C_{\tau_0,\varepsilon}^{1/(2+\varepsilon)}} \cdot \hat{g}_\varepsilon\left(\frac{x-1/2}{C_{\tau_0,\varepsilon}^{1/(2+\varepsilon)}}\right) < 0.$$

By Lemmas 4.2 and 4.3, we can apply the Poisson summation formula to conclude that

$$f_{2+\varepsilon}(\tau_0) = -\sum_{z\in\mathbb{Z}} s_{\tau_0,\varepsilon}(z) = -\sum_{x\in\mathbb{Z}} \hat{s}_{\tau_0,\varepsilon}(x) > 0.$$

This concludes the proof of **Case 1**.

Case 2: $(\varepsilon \geq 2)$. In this case, we simply choose $\tau_0 = 1$, which yields that

$$f_{2+\varepsilon}(1) = f_p(1) = -1 + 2\sum_{z=1}^{\infty}\left(e^{-(z-1/2)^p} - e^{-z^p}\right).$$

Since $e^{-(z-1/2)^p} - e^{-z^p} > 0$ for all $z \in \mathbb{Z}^+$ and $p \geq 4$, it follows that

$$f_p(1) > -1 + 2(e^{-1/2^p} - e^{-1}) \geq -1 + 2(e^{-1/2^4} - e^{-1}) > 0,$$

which completes our proof. ◀

The above theorem allows us to prove the following striking property of the integer lattice.

▶ **Theorem 4.4.** *For any $p \in (2,\infty)$, $\sigma > 1$, there exists constants $\delta \in (0,1/2)$, $\phi_0, \phi_1 > 1$ and $C_r > 0$, such that for any $n \in \mathbb{Z}^+$, if $r := C_r \cdot n^{1/p}$ then the following holds for the integer lattice.*

$$N_p\left(\mathbb{Z}^n, (1-\delta)^{1/p} r, \frac{1}{2}\mathbf{1}_n\right) \geq \max\left\{\phi_0^{n-o(n)} N_p\left(\mathbb{Z}^n, r, \mathbf{0}\right), \right.$$
$$\left. \phi_1^{n-o(n)} N_p\left(\mathbb{Z}^n, (1-\delta\sqrt{\sigma})^{1/p} r, \frac{1}{2}\mathbf{1}_n\right)\right\}. \tag{9}$$

The constants $\delta, \phi_0, \phi_1, C_r$ only depend upon p and can be efficiently computed with required precision, from p.

4.2 From MAXLIN$_\varepsilon$ to SVP$_{p,\gamma}$

In the following, we start with a CVP$_{p,\gamma'}$ instance with the special property that twice the target vector is in the lattice, and find a lattice $\mathcal{L}$ and a parameter r such that the number of vectors in $\mathcal{L}$ of length at most r are $2^{\Omega(n)}$ times more for a YES instance than for a NO instance.

▶ **Lemma 4.5.** *For any $p \in [1,\infty)$, suppose $(B', \boldsymbol{t}', r')$ is a CVP$_{p,\gamma'}$ instance, such that $B' \in \mathbb{Z}^{m\times m'}, \boldsymbol{t}' \in \mathcal{L}(B')/2$ and $\boldsymbol{t}' \in \mathbb{Z}^m$. Then for any $d \in \mathbb{Z}^+$, given a lattice $\mathcal{L}^\dagger \subseteq \mathbb{R}^d$ with basis $B^\dagger \in \mathbb{R}^{d\times d'}$, and a target $\boldsymbol{t}^\dagger \in \mathbb{R}^d$, there exists an efficiently computable matrix B that generates a lattice $\mathcal{L}(B)$ in $m+d+1$ dimensions such that for any constants $\gamma' > 1$ and any radii r_G,r_A,*

1. *If $\operatorname{dist}_p(\boldsymbol{t}', \mathcal{L}(B')) \le r'$ then,*

$$N_p(\mathcal{L}, r_G, \mathbf{0}) \ge N_p\left(\mathcal{L}^\dagger, (r_G^p - 1 - r'^p)^{1/p}, \boldsymbol{t}^\dagger\right), \tag{10}$$

2. *If $\operatorname{dist}_p(\boldsymbol{t}', \mathcal{L}(B')) > \gamma' r'$ then,*

$$N_p(\mathcal{L}, r_A, \mathbf{0}) \le (r_A+4)\cdot N_p(\mathbb{Z}^m, r_A, \mathbf{0}) \cdot \left(\max_{\ell_1\in\mathbb{Z}}\left\{N_p\left(\mathcal{L}^\dagger,\ (r_A^p - (\gamma' r')^p)^{1/p}, \ell_1\cdot\boldsymbol{t}^\dagger\right)\right\} + \max_{\ell_0\in 2\mathbb{Z}}\left\{N_p\left(\mathcal{L}^\dagger, r_A, \ell_0\cdot\boldsymbol{t}^\dagger\right)\right\}\right). \tag{11}$$

This lets us prove the following lemma, that says that for any MAXLIN$_\varepsilon$ instance, we can efficiently compute a lattice such that the number of short vectors in the YES case are exponentially more than the number of short vectors in the NO case.

▶ **Lemma 4.6.** *For all $p \in (2,\infty)$, there is an efficient algorithm that takes as an input a MAXLIN$_\varepsilon$ instance $(M, \boldsymbol{v})$ over $\mathbb{F}_2$ in n variables and $m = \mathcal{O}(n)$ equations, and outputs (B, r, γ, A, G) where B generates a lattice $\mathcal{L}$ in $\mathcal{O}(n)$ dimensions, $\gamma > 1$ is a constant, $r > 0$ is a radius, and constants A, G are such that $G \ge 2^m A$ and*

1. *if it was a YES instance of MAXLIN$_\varepsilon$ then, $N_p(\mathcal{L}, r, \mathbf{0}) \ge G$;*
2. *if it was a NO instance of MAXLIN$_\varepsilon$ then, $N_p(\mathcal{L}, \gamma r, \mathbf{0}) \le A$.*

From here, we are able to reach our final theorem.

▶ **Theorem 4.7.** *For any $p \in (2,\infty)$, there exists a constant $\gamma > 1$ such that for all $n \in \mathbb{Z}^+$, there is a polynomial time randomized Karp reduction from MAXLIN$_\varepsilon$ in n variables and $\mathcal{O}(n)$ equations to SVP$_{p,\gamma}$ in a lattice of rank $\mathcal{O}(n)$.*

Together with Corollary 2.13, we get the following theorem. We refer the reader to the full version [7] for the proofs.

▶ **Theorem 1.2** (ETH hardness of SVP$_{p,\gamma}$)**.** *For any $p \in (2,\infty)$, there exists a constant $\gamma > 1$ such that for all sufficiently large $n \in \mathbb{Z}^+$, there is no $2^{o(n)}$ time algorithm for SVP$_{p,\gamma}$ over $\mathbb{R}^n$ unless the randomized Exponential Time Hypothesis is false.*

5 A reduction from CVP$_{p,\gamma'}$ to BDD$_{p,\alpha}$

In this section we give a randomized Karp reduction from CVP$_{p,\gamma'}$ in a lattice in m dimensions, for any constant $\gamma' > 1$ to BDD$_{p,\alpha}$ in a lattice in $\mathcal{O}(m)$ dimensions, for all $\alpha > \alpha_p^\ddagger$, where $\alpha_p^\ddagger$ is as defined in Equation (4). We generally follow the reduction from [27, Section 3], but give simpler proofs that work for *any* CVP instance. Our reduction will use the following property of the integer lattice.

▶ **Lemma 5.1** ([27], Lemma 3.13). *For any $p \in [1,\infty)$, $\alpha_p^\ddagger < \alpha_G$ and $\alpha_A < \alpha_G$, there exist $t \in [0,1/2]$, $C_r \geq t$ and $\phi_0, \phi_1 > 1$, such that for any $d \in \mathbb{Z}^+$ and for $r^\dagger = C_r \cdot (d)^{1/p}$, $\boldsymbol{t}^\dagger = t \cdot \mathbf{1}_d$,*

$$N_p(\mathbb{Z}^d, \alpha_G \cdot r^\dagger, \boldsymbol{t}^\dagger) \geq \max\left\{\phi_0^{d-o(d)} \cdot N_p(\mathbb{Z}^d, r^\dagger, 0),\ \phi_1^{d-o(d)} \cdot N_p(\mathbb{Z}^d, \alpha_A \cdot r^\dagger, \boldsymbol{t}^\dagger)\right\}. \tag{12}$$

Furthermore[4]*, the constants t, C_r, ϕ_0, ϕ_1 only depends upon p, and can be efficiently computed from p.*

Then we can prove separate bounds on the number of lattice points within a ball of appropriate radii for the YES and the NO cases.

▶ **Lemma 5.2.** *For any $p \in [1,\infty)$, suppose $(B', \boldsymbol{t}', r')$ is a $\mathsf{CVP}_{p,\gamma'}$ instance over an integer lattice such that $B \in \mathbb{Z}^{m \times m'}, \boldsymbol{t}' \in \mathbb{Z}^m$. Then for any $d \in \mathbb{Z}^+$, there exists an efficiently computable matrix-vector pair $(B, \boldsymbol{t})$ such that B generates a lattice $\mathcal{L}(B)$ in $m+d$ dimensions, and for any radius $r > 0$ and constants $\alpha, s > 0$,*

$$N_p\left(\mathcal{L}(B), r/\alpha, \mathbf{0}\right) \leq N_p\left(\mathbb{Z}^m, r/\alpha, \mathbf{0}\right) \cdot N_p\left(\mathcal{L}^\dagger, \frac{r}{\alpha s}, \mathbf{0}\right), \tag{13}$$

and

- *if $\mathrm{dist}_p(\boldsymbol{t}', \mathcal{L}(B')) \leq r'$ then,*

$$N_p\left(\mathcal{L}(B), r, \boldsymbol{t}\right) \geq N_p\left(\mathcal{L}^\dagger, \frac{(r^p - r'^p)^{1/p}}{s}, \boldsymbol{t}^\dagger\right). \tag{14}$$

- *if $\mathrm{dist}_p(\boldsymbol{t}', \mathcal{L}(B')) \geq \gamma' r'$ then,*

$$N_p\left(\mathcal{L}(B), r, \boldsymbol{t}\right) \leq N_p(\mathbb{Z}^m, r, \mathbf{0}) \cdot N_p\left(\mathcal{L}^\dagger, \frac{(r^p - (\gamma' r')^p)^{1/p}}{s}, \boldsymbol{t}^\dagger\right). \tag{15}$$

This allows us to complete the reduction.

▶ **Lemma 5.3.** *For all $p \in [1,\infty)$, $\alpha > \alpha_p^\ddagger$, $c > 0$ and $\gamma' > 1$ the following holds for all sufficiently large $m \in \mathbb{Z}^+$. There exists an efficient algorithm that takes as an input a $\mathsf{CVP}_{p,\gamma'}$ instance $(B', \boldsymbol{t}', r')$, such that $B' \in \mathbb{Z}^{m \times m'}, \boldsymbol{t}' \in \mathbb{Z}^m$ and $r' = cm^{1/p}$, and returns $(B, \boldsymbol{t}, r, A, G)$, where B generates a lattice in $\mathcal{O}(m)$ dimensions, and A, G are integers such that $G > 2^m A$ and $r > 0$,*

- *If $\mathrm{dist}_p(\boldsymbol{t}', \mathcal{L}(B')) \leq r'$, then $N_p(\mathcal{L}(B), r/\alpha, \mathbf{0}) \leq A$ and $N_p(\mathcal{L}(B), r, \boldsymbol{t}) \geq G$.*
- *If $\mathrm{dist}_p(\boldsymbol{t}', \mathcal{L}(B')) \geq \gamma' r'$, then $N_p(\mathcal{L}(B), r, \boldsymbol{t}) \leq A$.*

▶ **Theorem 1.3** ($\mathsf{CVP}_{p,\gamma}$ reduces to $\mathsf{BDD}_{p,\alpha}$). *For any $\gamma' > 1$, $c > 0$, and $p \in [1,\infty)$, the following holds for all $\alpha > \alpha_p^\ddagger$ and sufficiently large $m \in \mathbb{Z}^+$. There is a decision-to-search reduction from any $\mathsf{CVP}_{p,\gamma'}$ to $\mathsf{BDD}_{p,\alpha}$, where the $\mathsf{CVP}_{p,\gamma'}$ instance $(B', \boldsymbol{t}', r')$ is such that $B' \in \mathbb{Z}^{m \times m'}$, $\boldsymbol{t}' \in \mathbb{Z}^m$ and $r' = c \cdot m^{1/p}$.*

Together with the reduction from $\mathsf{MAXLIN}_\varepsilon$ to $\mathsf{CVP}_{p,\gamma}$ in Theorem 3.1, we get the following corollary.

▶ **Corollary 5.4.** *For all $p \in [1,\infty)$, $\alpha > \alpha_p^\ddagger$, and for all sufficiently large m, there exists a polynomial time randomized Karp reduction from $\mathsf{MAXLIN}_\varepsilon$ over n variables and m equations $\mathsf{BDD}_{p,\alpha}$ in rank $\mathcal{O}(m)$.*

[4] In [27] the right hand side appears in terms of N_p°; our inspection of their proof shows that it works for N_p as well.

From Corollary 2.13 and Corollary 5.4 we get the following,

▶ **Theorem 1.4** (ETH hardness of $\mathsf{BDD}_{p,\alpha}$)**.** *For any $p \in [1,\infty)$, $\alpha > \alpha_p^{\ddagger}$, there is no $2^{o(n)}$ time algorithm for $\mathsf{BDD}_{p,\alpha}$ over $\mathbb{R}^n$, unless the randomized Exponential Time Hypothesis is false.*

References

1 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 *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '21, pages 1816–1835, USA, 2021. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611976465.109.

2 Divesh Aggarwal, Yanlin Chen, Rajendra Kumar, Zeyong Li, and Noah Stephens-Davidowitz. Dimension-preserving reductions between svp and cvp in different p-norms. In *Proceedings of the Thirty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '21, pages 2444–2462, USA, 2021. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611976465.145.

3 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. doi:10.1137/22M1486959.

4 Divesh Aggarwal and Eldon Chung. A note on the concrete hardness of the shortest independent vector in lattices. *Information Processing Letters*, 167:106065, 2021. doi:10.1016/J.IPL.2020.106065.

5 Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the Shortest Vector Problem in 2^n time via discrete Gaussian sampling. In *STOC*, 2015.

6 Divesh Aggarwal, Daniel Dadush, and Noah Stephens-Davidowitz. Solving the Closest Vector Problem in 2^n time – The discrete Gaussian strikes again! In *FOCS*, 2015.

7 Divesh Aggarwal, Rishav Gupta, Aditya Morolia, and Chuanqi Zhang. Mind the gap? not for svp hardness under eth!, 2026. arXiv:2504.02695.

8 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*, pages 274–295, Cham, 2020. Springer International Publishing. doi:10.1007/978-3-030-56880-1_10.

9 Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. A 2n/2-time algorithm for n-svp and n-hermite svp, and an improved time-approximation tradeoff for (h)svp. In *Advances in Cryptology – EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I*, pages 467–497, Berlin, Heidelberg, 2021. Springer-Verlag. doi:10.1007/978-3-030-77870-5_17.

10 Divesh Aggarwal and Noah Stephens-Davidowitz. Just take the average! An embarrassingly simple 2^n-time algorithm for SVP (and CVP), 2017. arXiv:1709.01535.

11 Divesh Aggarwal and Noah Stephens-Davidowitz. (gap/s)eth hardness of svp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 228–238, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3188745.3188840.

12 Miklós Ajtai. Generating hard instances of lattice problems. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 1–32. Dept. Math., Seconda Univ. Napoli, Caserta, 2004. Preliminary version in STOC'96.

13 Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610, 2001. doi:10.1145/380752.380857.

14 Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange – A new hope. In *USENIX Security Symposium*, 2016.

15 Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *FOCS*, 1993. doi:10.1109/SFCS.1993.366815.

16 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

17 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of np. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

18 Vikraman Arvind and Pushkar S Joglekar. Some sieving algorithms for lattice problems. In *FSTTCS*, pages 25–36, 2008. doi:10.4230/LIPIcs.FSTTCS.2008.1738.

19 L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986. doi:10.1007/BF02579403.

20 Shi Bai, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: Algorithm specifications and supporting documentation (version 3.1). https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf, 2021.

21 W. Banaszczyk. Inequalities for convex bodies and polar reciprocal lattices in $\mathbb{R}^n$. *Discrete Comput. Geom.*, 13(2):217–231, 1995. doi:10.1007/BF02574039.

22 Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*, 2016.

23 Huck Bennett. The complexity of the shortest vector problem. *SIGACT News*, 54(1):37–61, 2023. doi:10.1145/3586165.3586172.

24 Huck Bennett, Atul Ganju, Pura Peetathawatchai, and Noah Stephens-Davidowitz. Just how hard are rotations of z^n? algorithms and cryptography the simplest lattice. In *Advances in Cryptology – EUROCRYPT 2023: 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*, pages 252–281, Berlin, Heidelberg, 2023. Springer-Verlag. doi:10.1007/978-3-031-30589-4_9.

25 Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. On the quantitative hardness of CVP. In *FOCS*, 2017.

26 Huck Bennett and Chris Peikert. Hardness of bounded distance decoding on lattices in lp norms. In *Proceedings of the 35th Computational Complexity Conference*, CCC '20, Dagstuhl, DEU, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2020.36.

27 Huck Bennett, Chris Peikert, and Yi Tang. Improved Hardness of BDD and SVP Under Gap-(S)ETH. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:12, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2022.19.

28 Arnab Bhattacharyya, Édouard Bonnet, László Egri, Suprovat Ghoshal, Karthik C. S., Bingkai Lin, Pasin Manurangsi, and Dániel Marx. Parameterized intractability of even set and shortest vector problem. *J. ACM*, 68(3), 2021. doi:10.1145/3444942.

29 Nir Bitansky, Prahladh Harsha, Yuval Ishai, Ron D. Rothblum, and David J. Wu. Dot-product proofs and their applications. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 806–825, 2024. doi:10.1109/FOCS61266.2024.00057.

30 Johannes Blömer and Stefanie Naewe. Sampling methods for shortest vectors, closest vectors and successive minima. *Theoret. Comput. Sci.*, 410(18):1648–1665, 2009. doi:10.1016/j.tcs.2008.12.045.

31 Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, 2011. doi:10.1109/FOCS.2011.12.

32 J-Y Cai and Ajay Nerurkar. Approximating the SVP to within a factor $(1 + 1/\dim^{\varepsilon})$ is NP-hard under randomized conditions. In *CCC*. IEEE, 1998.

33 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From Gap-ETH to FPT-Inapproximability: Clique, Dominating Set, and More . In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754, Los Alamitos, CA, USA, October 2017. IEEE Computer Society. doi:10.1109/FOCS.2017.74.

34 Daniel Dadush, Chris Peikert, and Santosh Vempala. Enumerative lattice algorithms in any norm via M-ellipsoid coverings. In *FOCS*, 2011. doi:10.1109/FOCS.2011.31.

35 Irit Dinur. Mildly exponential reduction from gap 3SAT to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:128, 2016. URL: http://eccc.hpi-web.de/report/2016/128.

36 Friedrich Eisenbrand and Moritz Venzin. Approximate CVP in Time $2^{\{}0.802n\}$. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2020.43.

37 N. D. Elkies, A. M. Odlyzko, and J. A. Rush. On the packing densities of superballs and other bodies. *Inventiones mathematicae*, 105(1):613–639, December 1991.

38 Nicolas Gama and Phong Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC*, 2008. doi:10.1145/1374376.1374408.

39 Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

40 Venkatesan Guruswami, Bingkai Lin, Xuandi Ren, Yican Sun, and Kewen Wu. Parameterized inapproximability hypothesis under exponential time hypothesis. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 24–35, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649771.

41 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

42 Ishay Haviv and Oded Regev. On the Lattice Isomorphism Problem. In *SODA*, 2014.

43 John K. Hunter. Asymptotic analysis and singular perturbation theory. Technical report, Department of Mathematics, University of California, Davis, 2004. Lecture Notes, available online. URL: http://www.mat.unimi.it/users/scacchi/didattica_2017/biomat2/asy.pdf.

44 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

45 Carl G. J. Jacobi. Suite des notices sur les fonctions elliptiques. *J. reine angew. Math.*, 3:403–404, 1828. Reprinted in *Gesammelte Werke, Vol. 1.* Providence, RI: Amer. Math. Soc., pp. 264-265, 1969. URL: http://eudml.org/doc/183124.

46 Antoine Joux and Jacques Stern. Lattice reduction: A toolbox for the cryptanalyst. *Journal of Cryptology*, 11(3):161–185, 1998. doi:10.1007/S001459900042.

47 Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987. doi:10.1287/moor.12.3.415.

48 Subhash Khot. Hardness of approximating the Shortest Vector Problem in lattices. *Journal of the ACM*, 52(5):789–808, September 2005. Preliminary version in FOCS'04. doi:10.1145/1089023.1089027.

49 Sandor Kisfaludi-Bak, Jesper Nederlof, and Karol Wegrzycki. A Gap-ETH-Tight Approximation Scheme for Euclidean TSP . In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 351–362, Los Alamitos, CA, USA, February 2022. IEEE Computer Society. doi:10.1109/FOCS52979.2021.00043.

50 Thijs Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In *CRYPTO*, 2015.

51 A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982. doi:10.1007/BF01457454.

52 H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983. doi:10.1287/moor.8.4.538.

53 Shuangle Li, Bingkai Lin, and Yuwei Liu. Improved Lower Bounds for Approximating Parameterized Nearest Codeword and Related Problems Under ETH. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 107:1–107:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2024.107.

54 Yi-Kai Liu, Vadim Lyubashevsky, and Daniele Micciancio. On bounded distance decoding for general lattices. In *Proceedings of the 9th International Conference on Approximation Algorithms for Combinatorial Optimization Problems, and 10th International Conference on Randomization and Computation*, APPROX'06/RANDOM'06, pages 450–461, Berlin, Heidelberg, 2006. Springer-Verlag. doi:10.1007/11830924_41.

55 Pasin Manurangsi and Prasad Raghavendra. A Birthday Repetition Theorem and Complexity of Approximating Dense CSPs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 78:1–78:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2017.78.

56 J. E. Mazo and A. M. Odlyzko. Lattice points in high-dimensional spheres. *Monatsh. Math.*, 110(1):47–61, 1990. doi:10.1007/BF01571276.

57 Daniele Micciancio. The Shortest Vector Problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, March 2001. Preliminary version in FOCS 1998.

58 Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the Shortest Vector Problem. In *SODA*, 2010.

59 Phong Q Nguyen and Jacques Stern. The two faces of lattices in cryptology. In *Cryptography and lattices*, pages 146–180. Springer, 2001. doi:10.1007/3-540-44670-2_12.

60 Phong Q. Nguyen and Thomas Vidick. Sieve algorithms for the Shortest Vector Problem are practical. *J. Math. Cryptol.*, 2(2):181–207, 2008. doi:10.1515/JMC.2008.009.

61 NIST post-quantum standardization call for proposals. http://csrc.nist.gov/groups/ST/post-quantum-crypto/cfp-announce-dec2016.html, 2016. Accessed: 2017-04-02.

62 Andrew M Odlyzko. The rise and fall of knapsack cryptosystems. *Cryptology and computational number theory*, 42:75–88, 1990.

63 Siméon-Denis Poisson. Mémoire sur le calcul numérique des intégrales définies. *Mém. de l'Acad. des Sci.*, 6:571–602, 1827.

64 Xavier Pujol and Damien Stehlé. Solving the Shortest Lattice Vector Problem in time $2^{2.465n}$. *IACR Cryptology ePrint Archive*, 2009:605, 2009. URL: http://eprint.iacr.org/2009/605.

65 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):Art. 34, 40, 2009. doi:10.1145/1568318.1568324.

66 Oded Regev and Ricky Rosen. Lattice problems and norm embeddings. In *STOC*, 2006.

67 Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 974–988, 2023. doi:10.1109/FOCS57990.2023.00060.

68 C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53(2):201–224, June 1987. doi:10.1016/0304-3975(87)90064-8.

69 Adi Shamir. A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Trans. Inform. Theory*, 30(5):699–704, 1984. doi:10.1109/TIT.1984.1056964.

70 Noah Stephens-Davidowitz and Vinod Vaikuntanathan. Seth-hardness of coding problems. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 287–301, 2019. doi:10.1109/FOCS.2019.00027.

71 Craig A. Tovey. A simplified np-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.

72 Peter van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical report, University of Amsterdam, Department of Mathematics, Netherlands, 1981. Technical Report 8104.

73 Xiaoyun Wang, Mingjie Liu, Chengliang Tian, and Jingguo Bi. Improved Nguyen-Vidick heuristic sieve algorithm for shortest vector problem. In *ASIACCS*, 2011.