Symmetric Perceptrons, Number Partitioning and Lattices

Neekon Vafa
MIT
nvafa@mit.edu

Vinod Vaikuntanathan
MIT
vinodv@mit.edu

Abstract

The *symmetric binary perceptron* (SBP_κ) problem with parameter $\kappa : \mathbb{R}_{\geq 1} \to [0,1]$ is an average-case search problem defined as follows: given a random Gaussian matrix $\mathbf{A} \sim \mathcal{N}(0,1)^{n \times m}$ as input where $m \geq n$, output a vector $\mathbf{x} \in \{-1,1\}^m$ such that

$$\|\mathbf{A}\mathbf{x}\|_\infty \leq \kappa(m/n) \cdot \sqrt{m}\,.$$

The *number partitioning problem* (NPP_κ) corresponds to the special case of setting $n = 1$. There is considerable evidence that both problems exhibit large computational-statistical gaps.

In this work, we show (nearly) tight *average-case* hardness for these problems, assuming the *worst-case* hardness of standard approximate shortest vector problems on lattices.

- For SBP_κ, statistically, solutions exist with $\kappa(x) = 2^{-\Theta(x)}$ (Aubin, Perkins and Zdeborová, Journal of Physics 2019). For large n, the best that efficient algorithms have been able to achieve is a far cry from the statistical bound, namely $\kappa(x) = \Theta(1/\sqrt{x})$ (Bansal and Spencer, Random Structures and Algorithms 2020). The problem has been extensively studied in the TCS and statistics communities, and Gamarnik, Kızıldağ, Perkins and Xu (FOCS 2022) conjecture that Bansal-Spencer is tight: namely, $\kappa(x) = \widetilde{\Theta}(1/\sqrt{x})$ is the optimal value achieved by computationally efficient algorithms.

  We prove their conjecture assuming the worst-case hardness of approximating the shortest vector problem on lattices.

- For NPP_κ, statistically, solutions exist with $\kappa(m) = \Theta(2^{-m})$ (Karmarkar, Karp, Lueker and Odlyzko, Journal of Applied Probability 1986). Karmarkar and Karp's classical differencing algorithm achieves $\kappa(m) = 2^{-O(\log^2 m)}$.

  We prove that Karmarkar-Karp is nearly tight: namely, no polynomial-time algorithm can achieve $\kappa(m) = 2^{-\Omega(\log^3 m)}$, once again assuming the worst-case subexponential hardness of approximating the shortest vector problem on lattices to within a subexponential factor.

Our hardness results are versatile, and hold with respect to different distributions of the matrix $\mathbf{A}$ (e.g., i.i.d. uniform entries from $[0,1]$) and weaker requirements on the solution vector $\mathbf{x}$.

1 Introduction

Symmetric Binary Perceptrons. The *symmetric binary perceptron* (SBP_κ) problem, also called the symmetric Ising perceptron problem [JH60, Win61, Cov65, APZ19, BDVLZ20, PX21, ALS21, ALS22, GKPX22, GKPX23, BEAKZ24], is a search problem defined as follows: given a random matrix $\mathbf{A} \sim \mathcal{N}(0,1)^{n \times m}$ with entries chosen i.i.d. from the normal distribution where $m \geq n$, find a binary vector $\mathbf{x} \in \{-1,1\}^m$ such that $\|\mathbf{A}\mathbf{x}\|_\infty$ is minimized. More formally, SBP with parameter $\kappa : \mathbb{R}_{\geq 1} \to [0,1]$ asks us to find an $\mathbf{x} \in \{-1,1\}^m$ such that

$$\|\mathbf{A}\mathbf{x}\|_\infty \leq \kappa(m/n) \cdot \sqrt{m}\,.$$

Here, the quality of the solution is parameterized as a function of m/n, the so-called (inverse) aspect ratio of the problem.[1]

The problem is versatile, and can be defined with respect to different distributions of the matrix $\mathbf{A}$ (i.i.d. uniform $[0,1]$ entries is another popular choice) and different requirements on the solution vector $\mathbf{x}$. The problem also has rich connections to several other fields including the classical subset sum problem and its variants and the problem of discrepancy minimization.

Two natural questions arise: a statistical question and a computational one. The statistical question asks for which κ do solutions exist (with high probability over the choice of the matrix $\mathbf{A}$). Recent works by Aubin, Perkins and Zdeborová [APZ19], Perkins and Xu [PX21], and Abbe, Li and Sly [ALS21] showed sharp statistical thresholds for this problem. In particular, they showed that the threshold for the existence of solutions is

$$\kappa_{\mathsf{stat}}(x) = O(2^{-x}).$$

On the other hand, the best solutions found by polynomial-time algorithms satisfy

$$\kappa_{\mathsf{comp}}(x) = \Omega\left(\frac{1}{\sqrt{x}}\right).^{[2]}$$

This comes from the breakthrough work of Bansal [Ban10] and Bansal and Spencer [BS20] from the closely related field of combinatorial discrepancy theory.[3] Gamarnik, Kızıldağ, Perkins and Xu [GKPX22, GKPX23] studied the large statistical-computational gap scenario in detail and conjectured that no polynomial-time algorithms can achieve a guarantee much better than $\kappa_{\mathsf{comp}}(x)$. In particular, they show that the so-called overlap gap property [MMZ05, AR06, Gl16], which rules out a class of stable algorithms, sets in at

$$\kappa_{\mathsf{overlap}}(x) = O\left(\frac{1}{\sqrt{x \log x}}\right).$$

[1] The notation commonly used in the literature to parameterize SBP (and NPP) is slightly different than the notation we choose to use. The aspect ratio is given by $\alpha = n/m$, and instead of writing κ as a function of α (or really, $x = 1/\alpha$, as we do), the roles are flipped, where α is a function of κ. For example, $\kappa(x) = 2^{-x}$ and $\kappa(x) = 1/\sqrt{x}$, in our notation, correspond to $\alpha(\kappa) = 1/\log_2(1/\kappa)$ and $\alpha(\kappa) = \kappa^2$ in the notation of [GKPX22], respectively.

[2] In an extreme parameter regime where $n = O\left(\sqrt{\log m}\right)$, [TMR20] gives a polynomial-time algorithm achieving discrepancy $2^{-\Omega(\log^2(m)/n)}$, but throughout our paper, for SBP, we will only consider the regime where $n = \omega(\log m)$.

[3] Their result is established for the case of matrices $\mathbf{A}$ with i.i.d. Rademacher entries. Nevertheless, [GKPX22] conjecture that the same guarantee remains true for the case of i.i.d. standard normal entries.

We refer the reader to [GKPX22] for an extensive discussion of the rationale behind their conjecture.

A survey of Gamarnik on the overlap gap property [Gam21] points to the question of whether average-case hardness of perceptron problems (and more) can be based on *worst-case* hardness assumptions. Gamarnik explicitly states that worst-case to average-case reductions for these problems "would be ideal for our setting, as they would provide the most compelling evidence of hardness of these problems" [Gam21].

The first contribution of this work is a proof of the conjecture of [GKPX22] upto lower order terms, under the assumption that standard, well-studied, lattice problems are hard to approximate in the worst case.

Theorem 1 (Informal version of Corollary 2)**.** *Let $\varepsilon > 0$ be any constant. Assuming that γ-approximate lattice problems in n dimensions with $\gamma(n) = n^{O(1/\varepsilon)}$ are worst-case hard for polynomial-time algorithms,* SBP_κ *with*

$$\kappa(x) = \frac{1}{x^{1/2+\varepsilon}}$$

is hard for polynomial-time algorithms. Additionally, assuming near-optimal worst-case hardness of lattice problems, we obtain near-optimal hardness of SBP. In particular, assuming that $\gamma(n)$-approximate lattice problems require $2^{\omega(n^{1/2-\varepsilon})}$ time to solve in the worst case with $\gamma(n) = 2^{n^{1/2-\varepsilon}}$, we have that SBP_κ *with*

$$\kappa(x) = \frac{1}{\sqrt{x} \cdot (\log x)^c}$$

is hard for some constant $c > 1$.

Lattice problems such as the shortest vector problem and the shortest independent vectors problem have been extensively studied for decades largely for their implications to combinatorial optimization [LLL82, Kan83, Kan87, RR23] and even more so, to cryptography [Ajt96, MR07, Reg09]. Time-approximation tradeoffs for lattice problems are well-known [Sch87]. The best known algorithms for these lattice problems employ *lattice reduction* techniques, based on the LLL and BKZ algorithms [LLL82, Sch87]. They currently all have the following time-approximation trade-off: For a tunable parameter k, in dimension n, it is possible to solve lattice problems with approximation factor $\gamma = 2^{\widetilde{O}(n/k)}$ in time $2^{\widetilde{O}(k)}$, where $\widetilde{O}(\cdot)$ hides polylog factors in n. Equalizing these terms gives a $2^{\widetilde{O}(\sqrt{n})}$-time algorithm to solve $2^{\widetilde{O}(\sqrt{n})}$-approximate lattice problems. Despite extensive study in the lattice literature, no better algorithms are known (that improve the exponent by more than a polylog factor). This motivates the assumptions in our theorem statement, both the conservative one and the near-optimal one. (For more discussion on this, see Section 2.2).

Our reduction has several additional features: first, our reduction is versatile and works with respect to different distributions of the matrix $\mathbf{A}$ (e.g., i.i.d. uniform entries from $[0, 1]$); and secondly, it shows the hardness not just of computing an SBP solution with $1 - o(1)$ probability, but indeed with any non-trivial inverse polynomial probability. For more implications of our reduction, we refer the reader to Section 3.2. Moreover, our full reduction is conceptually simple and direct. We discuss the possibility of using other intermediate problems to establish these same results in Section 1.1.3.

Number Partitioning (or Number Balancing). The (average-case) *number partitioning* problem is a special case of SBP and corresponds to setting $n = 1$ in SBP. Given m random numbers $a_1, \dots, a_m \sim \mathcal{N}(0,1)$ with entries chosen i.i.d. from the standard normal distribution, the goal is to find a binary vector $\mathbf{x} \in \{-1,1\}^m$ such that $|\sum_{i=1}^m x_i a_i|$ is minimized. More formally, NPP with parameter $\kappa : \mathbb{Z}_{\geq 1} \to [0,1]$ asks us to find an $\mathbf{x} \in \{-1,1\}^m$ such that

$$\left| \sum_{i=1}^{m} x_i a_i \right| \leq \kappa(m) \cdot \sqrt{m}$$

Number partitioning, as a worst-case problem, was one of the six original NP-complete problems in the classic book of Garey and Johnson [GJ79]. The worst-case version of the problem, where $a_1, \dots, a_m$ are arbitrary, is closely related to discrepancy problems and has been extensively studied; see [Spe85, Ban10, LM15, LRR17, Rot14, HRRY17]. In particular, [HRRY17] show that there are (worst-case to worst-case) reductions between NPP and worst-case lattice problems.[4] Their reduction from worst-case lattice problems to NPP [HRRY17, Theorem 9] shows hardness for $\kappa(m) \leq 2^{-m/2}$. Looking ahead, we show computational hardness for the average-case version of NPP and for the much tighter range of $\kappa(m) = 2^{-\mathsf{polylog}(m)}$.

An application of the pigeonhole principle shows that solutions exist, both in the worst case and on average (even with high probability [KKLO86]), for

$$\kappa_{\mathsf{stat}}(m) = 2^{-m}.$$

However, the best solutions found by polynomial-time algorithms satisfy

$$\kappa_{\mathsf{comp}}(m) = 2^{-O(\log^2 m)}.$$

This comes from the beautiful "differencing algorithm" of Karmarkar and Karp [KK82] which starts with a list; sorts it; replaces the largest and second largest element with their absolute difference; and repeats until a single element is left which the algorithm outputs as the discrepancy of the set of numbers. (An informed reader might already have observed the analogy of Karmarkar-Karp with the Blum-Kalai-Wasserman [BKW03] algorithm, which came much later.) A subsequent work of Yakir [Yak96] proved that their algorithm indeed achieves the claimed discrepancy of $\kappa_{\mathsf{comp}}(m) = 2^{-O(\log^2 m)}$. This remains the best algorithm known to date.

Gamarnik and Kızıldağ [GK21] studied the statistical-computational gap in depth, demonstrated an overlap gap property at

$$\kappa^*(m) = 2^{-\omega\left(\sqrt{m \log m}\right)},$$

implying that the class of stable algorithms will fail to find solutions for such small κ. This leaves open the question of the ground truth: are there improvements to Karmarkar-Karp, stable or not, that efficiently solve NPP_κ with $\kappa(m) = 2^{-m^{\Omega(1)}}$?

[4] Technically, the problem they consider is the number balancing problem, where they require the weaker condition on the solution $\mathbf{x}$ that $\mathbf{x} \in \{-1,0,1\}^m \setminus \{\mathbf{0}\}$ instead of $\mathbf{x} \in \{-1,1\}^m$. As we explain later (Section 3.2), our reduction works in this setting as well.

Our second contribution is proving that the answer to this question is *no*, in a quantitatively strong way. In particular, under the assumption that standard, well-studied, lattice problems are sub-exponentially hard to approximate in the worst case, we prove that Karmarkar-Karp is tight, up to a logarithmic factor in the exponent.

Theorem 2 (Informal version of Corollary 3)**.** *Suppose $\kappa(m) = 2^{-\log^{3+\varepsilon} m}$ for some constant $\varepsilon > 0$. Assuming near-optimal hardness of worst-case lattice problems, then NPP_κ is hard for polynomial-time algorithms. In particular, assuming that $\gamma(n)$-approximate lattice problems in dimension n require $2^{\omega(n^{1/2-\varepsilon})}$ time to solve in the worst case with $\gamma(n) = 2^{n^{1/2-\varepsilon}}$, we have that NPP_κ (in dimension m) is hard for $\mathsf{poly}(m)$-time algorithms.*

Similar to the case of our SBP result, this theorem is quite versatile. We refer the reader to the technical overview and Section 4 for more details. Like in the case SBP, our full reduction is conceptually simple and direct, and we discuss the possibility of using other intermediate problems to establish these same results in Section 1.1.3.

Adaptive Robustness of Johnson-Lindenstrauss. The Johnson-Lindenstrauss lemma [JL84] states that for all fixed, small, finite sets $S \subseteq \mathbb{R}^m$, for random $\mathbf{A} \sim \mathcal{N}(0,1)^{n\times m}$, the linear map given by $\mathbf{A}$ embeds S into $\mathbb{R}^n$ in a way that approximately preserves all ℓ_2 norms (up to a $\sqrt{n}$ normalization term). Slightly more concretely,

$$\forall \text{ small, finite } S \subseteq \mathbb{R}^m, \quad \Pr_{\mathbf{A}\sim\mathcal{N}(0,1)^{n\times m}}\left[\forall \mathbf{x} \in S,\ \|\mathbf{A}\mathbf{x}\|_2 \approx \sqrt{n}\,\|\mathbf{x}\|_2\right] = 1 - o(1).$$

This statement crucially relies on the fact the set S that is defined independently of $\mathbf{A}$. For example, even considering only singleton sets S, one can ask whether the order of quantifiers can be switched so that $\mathbf{x}$ can be chosen adaptively based on $\mathbf{A}$, in the following sense:

$$\Pr_{\mathbf{A}\sim\mathcal{N}(0,1)^{n\times m}}\left[\forall \mathbf{x} \in \mathbb{R}^m,\ \|\mathbf{A}\mathbf{x}\|_2 \approx \sqrt{n}\,\|\mathbf{x}\|_2\right] \stackrel{?}{=} 1 - o(1).$$

Or even weaker, for a function $\kappa : \mathbb{R}_{>1} \to (0,1]$,

$$\Pr_{\mathbf{A}\sim\mathcal{N}(0,1)^{n\times m}}\left[\forall \mathbf{x} \in \mathbb{R}^m,\ \|\mathbf{A}\mathbf{x}\|_2 \geq \|\mathbf{x}\|_2\,\kappa(m/n)\sqrt{n}\right] \stackrel{?}{=} 1 - o(1).$$

However, this is impossible. For $m > n$, one can find a vector $\mathbf{x} \in \ker(\mathbf{A})$ (so $S = \{\mathbf{x}\}$), making $\|\mathbf{A}\mathbf{x}\|_2 = 0$ while $\|\mathbf{x}\|_2$ can be arbitrarily large.

A natural question is whether this phenomenon could hold if we constrain $\mathbf{x} \in \mathbb{R}^m$ to some structured set, e.g., $\{-1,1\}^m$:

$$\Pr_{\mathbf{A}\sim\mathcal{N}(0,1)^{n\times m}}\left[\forall \mathbf{x} \in \{-1,1\}^m,\ \|\mathbf{A}\mathbf{x}\|_2 \geq \kappa(m/n)\sqrt{nm}\right] \stackrel{?}{=} 1 - o(1).$$

This question is exactly the statistical capacity of SBP_κ, with the exception that the norm on $\mathbf{A}\mathbf{x}$ has changed from ℓ_∞ to ℓ_2 (which differ by only a $\sqrt{n}$ factor at most).

Therefore, we view SBP_κ as defining a natural adaptive robustness question about a certain discretized version of the Johnson-Lindenstrauss lemma. In particular, since SBP_κ exhibits a computational-statistical gap, so does this variant of the Johnson-Lindenstrauss lemma. Given the utility of the Johnson-Lindenstrauss lemma in compressed sensing, dimensionality reduction, and more, we believe that this interpretation may inspire connections between worst-case lattice problems and adaptive robustness of downstream applications of the Johnson-Lindenstrauss lemma.

Open Questions and Future Directions. A direct open question raised by our results is to better understand the gap between Theorem 2 and Karmarkar-Karp [KK82]. Specifically, our result shows hardness for $\kappa(m) = 2^{-\log^{3+\epsilon} m}$, but Karmarkar-Karp gives a polynomial time algorithm that achieves $\kappa(m) = 2^{-\Theta(\log^2 m)}$. Can Karmarkar-Karp be improved to $\kappa(m) = 2^{-\Theta(\log^3 m)}$, can the hardness shown in Theorem 2 be improved, or is the truth somewhere in the middle?

For SBP, one can ask what happens in the setting where $m = \Theta(n)$, i.e., so the ratio m/n is $\Theta(1)$. The proof of our hardness result crucially needs $m \geq n^{\Theta(1/\epsilon)}$, with no setting of our parameters yielding $m = \Theta(n)$. Can one extend hardness like in Theorem 1 to this proportional setting?

Another question one can ask is related to the *asymmetric* binary perceptron (ABP_κ) problem, also called the asymmetric Ising perceptron problem, which for $\mathbf{A} \sim \mathcal{N}(0,1)^{n \times m}$, asks to find $\mathbf{x} \in \{-1, 1\}^m$ such that

$$\mathbf{A}\mathbf{x} \geq \kappa(m/n)\sqrt{m} \cdot \mathbf{1},$$

in the sense that for every row $\mathbf{a}_j \in \mathbb{R}^m$ for $j \in [n]$, we want $\mathbf{a}_j^\top \mathbf{x} \geq \kappa(m/n)\sqrt{m}$. (For more details, see e.g., [GKPX22].) Note that unlike SBP_κ, setting $\kappa(x) = 0$ still defines a meaningful problem. Does ABP share similar hardness results (from worst-case problems)? If so, are lattice problems the source of such hardness?

More generally, we can ask the *converse* questions of the ones raised in our paper. Can lattice algorithms be used to improve algorithms for NPP, SBP or ABP? Recent work has given some hope along these lines, showing that lattice-based methods outperform sum-of-squares [ZSWB22].

We leave all of these as fascinating open questions and future directions of our work.

Related work. Recently, various optimization and learning problems have been shown to have lattice-based hardness. The formative work of Bruna, Regev, Song and Tang [BRST21] defined a variant of the standard cryptographic Learning With Errors (LWE) problem called *Continuous Learning With Errors* (CLWE), which is an average-case, periodic, linear regression task, which they showed is as hard as worst-case lattice problems. This and subsequent works have used CLWE as an intermediate problem to show worst-case lattice hardness of learning mixtures of Gaussians, learning single periodic neurons, learning halfspaces, detecting planted backdoors in certain machine learning models, and more [BRST21, SZB21, GVV22, DKMR22, DKR23, Tie23, GKVZ22].

1.1 Technical Overview

For both of our reductions, we take direct inspiration from worst-case to average-case reductions in the lattice literature [Ajt96, MR07]. Our exposition and proofs closely follow [MR07]. We

view our work as bringing techniques from worst-case to average-case reductions in the lattice literature to closely related problems in statistics and discrete optimization.

1.1.1 Reduction to SBP

We outline the main ideas behind the reduction from the worst-case lattice problems to SBP. For an invertible matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$, let $\mathcal{L}(\mathbf{B})$ denote the lattice generated by (the columns of) $\mathbf{B}$, i.e.,

$$\mathcal{L}(\mathbf{B}) = \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^n\}.$$

Let $\mathcal{P}(\mathbf{B})$ denote the *fundamental parallelepiped* of $\mathbf{B}$, given by the set $\mathbf{B}[0,1)^n$.

To illustrate the ideas in our reduction, consider the following (informal) worst-case lattice problem: given some invertible lattice basis $\mathbf{B} \in \mathbb{R}^{n \times n}$, output a (non-zero) vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ slightly smaller than the columns in $\mathbf{B}$. (While this is not quite the worst-case lattice problem we reduce from, it is simpler to describe this way and captures all of the main ideas. See Definition 4 for more precise details.)

The key idea we employ is *smoothing* using Gaussian measures, as introduced by Micciancio and Regev [MR07]. Specifically, here is the critical point: For an *arbitrary* invertible $\mathbf{B} \in \mathbb{R}^{n \times n}$ and for $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$, as long as σ is large enough, $\mathbf{u} \pmod{\mathcal{L}(\mathbf{B})}$ looks *statistically* close to uniform over $\mathcal{P}(\mathbf{B})$. This holds as long as σ is larger than the *smoothing parameter* of the lattice $\mathcal{L}(\mathbf{B})$, defined by [MR07], which is basis-independent. Transforming by $\mathbf{B}^{-1}$, we see that

$$\mathbf{B}^{-1}\mathbf{u} \pmod{\mathbb{Z}^n} \approx U([0,1)^n),$$

where $\approx$ denotes small total variation distance, and $U(\cdot)$ denotes the uniform distribution. Thus, we have converted worst-case structure into average-case structure (that is independent of $\mathbf{B}$).

We can repeat this process m times as follows: sample $\mathbf{U} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)^m$, where now

$$\mathbf{B}^{-1}\mathbf{U} \pmod{\mathbb{Z}^{n \times m}} \approx U([0,1)^{n \times m}).$$

We then set $\mathbf{A} = \mathbf{B}^{-1}\mathbf{U} \pmod{\mathbb{Z}^{n \times m}}$ and feed $\mathbf{A}$ into the SBP solver. (For simplicity, in this overview, we assume that SBP allows input matrices that are uniform over $[0,1)^{n \times m}$ instead of Gaussian, but in the proof, we resolve this distinction by sampling from an appropriate discrete Gaussian distribution.) We get back some $\mathbf{x} \in \{\pm 1\}^m$ so that $\|\mathbf{Ax}\|_\infty \leq \kappa(m/n)\sqrt{m}$, or in other words,

$$\mathbf{Ax} - \mathbf{e} = \mathbf{0},$$

where $\|\mathbf{e}\|_\infty \leq \kappa(m/n)\sqrt{m}$. Since $\mathbf{x} \in \mathbb{Z}^m$, this implies

$$\mathbf{B}^{-1}\mathbf{Ux} - \mathbf{e} = \mathbf{0} \mod \mathbb{Z}^{n \times m}.$$

Multiplying by $\mathbf{B}$ on the left gives

$$\mathbf{Ux} - \mathbf{Be} = \mathbf{0} \mod \mathcal{L}(\mathbf{B}),$$

or in other words, $\mathbf{Ux} - \mathbf{Be} \in \mathcal{L}(\mathbf{B})$. To see that $\mathbf{Ux} - \mathbf{Be}$ is slightly smaller than the vectors in $\mathbf{B}$, note that $\|\mathbf{Ux}\|_2$ can be upper-bounded by a basis-independent quantity, since σ was chosen

basis-independently. The bottleneck term is $\|\mathbf{Be}\|_2$, which is smaller than columns of $\mathbf{B}$ as long as $\|\mathbf{e}\|_2$ is sufficiently small. If $\kappa(x) = 1/x^{1/2+\varepsilon}$ for $\varepsilon > 0$, then

$$\|\mathbf{e}\|_\infty \le \kappa(m/n)\sqrt{m} = \frac{n^{1/2+\varepsilon}}{m^\varepsilon},$$

so as long as we set m large enough so that $m^\varepsilon \gg n^{1/2+\varepsilon}$, we can force $\|\mathbf{Be}\|_2$ to be small enough to produce a smaller lattice vector than anything in $\mathbf{B}$, as desired.

Allowing $\mathbf{x} \in \{-1, 0, 1\}^m$ instead of $\mathbf{x} \in \{\pm 1\}^m$. One could define a variant of SBP where we just need $\mathbf{x} \in \{-1, 0, 1\}^m \setminus \{\mathbf{0}\}$ instead of $\mathbf{x} \in \{\pm 1\}^m$. This is an easier problem, as including zero entries in $\mathbf{x}$ is a simple way to decrease $\|\mathbf{Ax}\|_\infty$. However, our reduction idea above actually *still* works in this setting. For more details, see Section 3.2.

1.1.2 Reduction to NPP

The reduction to NPP is quite similar to the reduction to SBP, but with one additional trick that converts between vectors and scalars. This trick has been used in [Reg04, BV15] for a similar reason, and we follow their footsteps. Explicitly, it is the *Chinese remainder theorem*: for distinct primes $p_1, \dots, p_n$ and $q = \prod_{i\in[n]} p_i$, there is a group isomorphism

$$\varphi : \bigoplus_{i\in[n]} \mathbb{Z}/p_i\mathbb{Z} \longrightarrow \mathbb{Z}/q\mathbb{Z}.$$

We will scale this isomorphism so that

$$\widetilde{\varphi} : \bigoplus_{i\in[n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z} \longrightarrow 1/q \cdot \mathbb{Z}/q\mathbb{Z}$$

is (a) invariant to integer shifts in the input and (b) $\mathbb{Z}$-linear, in the sense that $\widetilde{\varphi}(\mathbf{x}) = \mathbf{c}^\top \mathbf{x} \pmod 1$ for some $\mathbf{c} \in \mathbb{Z}^n$.

Following the SBP reduction above, we can set $\mathbf{A} = \mathbf{B}^{-1}\mathbf{U} \pmod{\mathbb{Z}^{n\times m}}$ and then appropriately "round" $\mathbf{A}$ to get uniformly random $\mathbf{A}' = \lfloor \mathbf{A} \rceil_{\mathbf{p}} \in \left(\bigoplus_{i\in[n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z}\right)^m$. We then apply $\widetilde{\varphi}$ column-wise to $\mathbf{A}'$ to get $\mathbf{a}' \in (1/q\cdot\mathbb{Z}/q\mathbb{Z})^m \subseteq [0,1)^m$. By adding small uniform noise to $\mathbf{a}'$, we can get some $\mathbf{a} \sim [0,1)^m$. We feed this into our NPP solver to get some $\mathbf{x} \in \{\pm 1\}^m$ such that $|\mathbf{a}^\top \mathbf{x}| \le \kappa(m)\sqrt{m}$. By using $\mathbb{Z}$-linearity of $\widetilde{\varphi}$, and assuming κ is sufficiently small so that there are no "wraparound" errors in $\widetilde{\varphi}^{-1}$, we can recover a somewhat smaller lattice vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ than we started with, just like in the SBP reduction.

1.1.3 Alternate Reduction Paths

We briefly mention other paths to reduce worst-case lattice problems to SBP and NPP, using existing intermediate problems and known worst-case to average-case reductions.

SBP. Instead of starting from worst-case lattice problems, we could start with a noisy version of the *short integer solutions* (SIS) problem in the ℓ_∞ norm, defined roughly as follows: Given random $\mathbf{A} \sim (\mathbb{Z}/q\mathbb{Z})^{n\times m}$, output $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $\|\mathbf{x}\|_\infty$ and $\|\mathbf{A}\mathbf{x} \pmod q\|_\infty$ are small.[5] The original worst-case to average-case reductions for lattice problems indeed reduce worst-case lattice problems to (average-case) SIS [Ajt96, MR07]. One can reduce (noisy) SIS to SBP by adding small noise $\mathbf{E} \sim U([0,1/q)^{n\times m})$ and setting

$$\mathbf{A}' = \frac{1}{q}\mathbf{A} + \mathbf{E} \sim U([0,1)^{n\times m}).$$

(For simplicity, we assume here that SBP allows matrices with i.i.d. $U([0,1))$ entries instead of standard normal, but we can remove this assumption by sampling from the appropriate discrete Gaussian and scaling.) Feeding $\mathbf{A}'$ into the SBP solver yields $\mathbf{x} \in \{\pm 1\}^m$ such that $\|\mathbf{A}\mathbf{x}\|_\infty$ is small, assuming $q\|\mathbf{E}\mathbf{x}\|_\infty$ is also sufficiently small.

Another approach for SBP is to reduce directly from Continuous Learning With Errors (CLWE) [BRST21]. This average-case problem, which is known to be as hard as worst-case approximate lattice problems [BRST21, GVV22], asks to distinguish $(\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top \pmod 1)$ from $(\mathbf{A}, \mathbf{b}^\top)$, where $\mathbf{A} \sim \mathcal{N}(0,1)^{n\times m}$, $\mathbf{s}$ is drawn spherically from sufficiently large sphere, $\mathbf{e}$ is drawn from a Gaussian with small variance, and $\mathbf{b}$ is a uniformly random vector mod 1. By using integrality of $\mathbf{x}$ and simultaneous smallness of $\mathbf{x}$ and $\mathbf{A}\mathbf{x}$, one can simply multiply the second argument on the right by $\mathbf{x}$ to distinguish from the uniform distribution modulo 1. However, we find this CLWE approach somewhat unsatisfactory since it would show hardness of SBP assuming at least one of (a) *quantum* worst-case hardness of *search* lattice problems or (b) *classical* worst-case hardness of *decisional* lattice problems [BRST21, GVV22]. On the other hand, our main result shows hardness of SBP under only *classical* worst-case hardness of *search* lattice problems. As search lattice problems are known to be harder than decisional ones, our main approach in this work yields a stronger result than the CLWE approach. Indeed, for those familiar with lattice-based cryptography, this should not come as a surprise: the difference between SBP and CLWE is reminiscent of the difference between SIS and the Learning With Errors (LWE) problems in lattice-based cryptography.

NPP. Brakerski and Vaikuntanathan [BV15] define a problem called the *one-dimensional short integer solutions* problem (1D-SIS), defined roughly as follows: Given $\mathbf{a} \sim (\mathbb{Z}/q\mathbb{Z})^m$, output $\mathbf{x} \in \mathbb{Z}^m \setminus \{\mathbf{0}\}$ such that $\|\mathbf{x}\|_\infty$ and $|\mathbf{a}^\top\mathbf{x} \pmod q|$ are small, where q is a product of n primes.[6] [BV15] shows a worst-case to average-case reduction from worst-case lattice problems in dimension n to 1D-SIS. One can reduce 1D-SIS to NPP by adding small noise $\mathbf{e} \sim U([0,1/q)^m)$ and setting

$$\mathbf{a}' = \frac{1}{q}\mathbf{a} + \mathbf{e} \sim U([0,1)^m).$$

(For simplicity, we similarly assume here that NPP works for vectors with i.i.d. $U([0,1))$ entries.) Feeding $\mathbf{a}'$ into the NPP solver yields $\mathbf{x} \in \{\pm 1\}^m$ such that $|\mathbf{a}^\top\mathbf{x}|$ is small, assuming $q|\mathbf{e}^\top\mathbf{x}|$ is also

[5] One can remove the $\pmod q$ constraint to reduce from SIS over $\mathbb{Z}$, and this would work as well.

[6] Similarly here, one can remove the $\pmod q$ constraint and reduce from 1D-SIS over $\mathbb{Z}$.

sufficiently small.

While these approaches would work, we view this introduction of parameter $q \in \mathbb{Z}$ or modulus reduction as extraneous and misleading. To show hardness of a discrete optimization problem in continuous Euclidean space (SBP and NPP), we should ideally start from a problem that is itself a discrete optimization problem in continuous Euclidean space (worst-case lattice problems). There is no need to add premature discretization by introducing the parameter q or unnecessary modular reduction.

2 Preliminaries

For a predicate φ, we use the notation $\mathbf{1}[\varphi] \in \{0, 1\}$ to denote the indicator variable of whether φ is true (1) or false (0). We use $\ln(\cdot)$ to denote the natural logarithm (base e) and $\log(\cdot)$ to denote $\log_2(\cdot)$. We use $\mathbb{R}_{>0}$ to refer to the set of all positive real numbers, and we use $\mathbb{R}_{\geq 1}$ to refer to the set of all real numbers that are at least 1. We say a function $f : \mathbb{N} \to \mathbb{R}_{>0}$ is *negligible* if for all $c \in \mathbb{N}$,

$$\lim_{n \to \infty} n^c \cdot f(n) = 0.$$

We say a function $f : \mathbb{N} \to \mathbb{R}$ is *non-negligible* if there exists some $c \in \mathbb{N}$ such that $f(n) \geq 1/n^c$ for all sufficiently large n.

For two distributions $\mathcal{D}_1$, $\mathcal{D}_2$, we use the notation $\Delta(\mathcal{D}_1, \mathcal{D}_2)$ to denote the total variation distance between $\mathcal{D}_1$ and $\mathcal{D}_2$, which we refer to simply as the *statistical distance* between the two distributions. We say that two distributions are *statistically close* if their statistical distance is negligible (in some implicit parameter, typically n or m for us).

For a set S, we let $U(S)$ denote the uniform distribution over S. (If S is not finite and $S \subseteq \mathbb{R}^n$ is Lebesgue measurable, this will be uniform with respect to the standard Lebesgue measure.)

2.1 Norms & Matrices

We use the notation $\mathbf{0}$ to denote the all 0s vector in $\mathbb{R}^n$, where the dimension n is clear from context. We use $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ to denote the n-dimensional identity matrix. We use the standard $\ell_1, \ell_2, \ell_\infty$ norms on $\mathbb{R}^n$. For a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, we use the notation $\sigma_{\max}(\mathbf{A})$ to denote the spectral norm, or maximum singular value, of $\mathbf{A}$. Explicitly,

$$\sigma_{\max}(\mathbf{A}) = \max_{\mathbf{v} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{v}\|_2}{\|\mathbf{v}\|_2}.$$

For a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$, we often write $\mathbf{A}$ by its columns, as in $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_m]$ for $\mathbf{a}_i \in \mathbb{R}^n$. We sometimes abuse notation and move interchangeably between matrices $\mathbf{A} \in \mathbb{R}^{n \times m}$ and tuples of m vectors in $\mathbb{R}^n$ (as defined by the columns of $\mathbf{A}$). For a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ given by $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_m]$, we use the notation

$$\|\mathbf{A}\| = \max_{j \in [m]} \|\mathbf{a}_j\|_2 .$$

We emphasize that $\|\mathbf{A}\|$ does *not* refer to the standard spectral norm on matrices.

Lemma 1. *For $\mathbf{A} \in \mathbb{R}^{n\times m}$ and $\mathbf{v} \in \mathbb{R}^m$, we have the inequality $\|\mathbf{Av}\|_2 \le \|\mathbf{A}\| \|\mathbf{v}\|_1$.*

Proof. Let $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_m]$, and let $\mathbf{v}$ have entries $v_j \in \mathbb{R}$. We have

$$\|\mathbf{Av}\|_2 = \left\| \sum_{j=1}^{m} v_j \mathbf{a}_j \right\|_2 \le \sum_{j=1}^{m} \|v_j \mathbf{a}_j\|_2 = \sum_{j=1}^{m} |v_j| \|\mathbf{a}_j\|_2 \le \sum_{j=1}^{m} |v_j| \|\mathbf{A}\| = \|\mathbf{A}\| \|\mathbf{v}\|_1 .$$

□

We also use the following basic fact.

Lemma 2. *For all $\mathbf{v} \in \mathbb{R}^n$, $\|\mathbf{v}\|_1 \le n \|\mathbf{v}\|_\infty$.*

2.2 Lattices

For an invertible matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, an n-dimensional lattice generated by basis $\mathbf{B}$, denoted $\mathcal{L}(\mathbf{B})$, is given by all integer linear combinations of columns of $\mathbf{B}$. That is,

$$\mathcal{L}(\mathbf{B}) := \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^n\}.$$

We define the (half-open) parallelepiped $\mathcal{P}(\mathbf{B})$ to be the set

$$\mathcal{P}(\mathbf{B}) := \{\mathbf{Bv} : \mathbf{v} \in [0,1)^n\}.$$

Note that for all $\mathbf{x} \in \mathbb{R}^n$, there exists unique $\mathbf{y} \in \mathcal{P}(\mathbf{B})$ such that $\mathbf{x} - \mathbf{y} \in \mathcal{L}(\mathbf{B})$. We use the notation $\mathbf{y} = \mathbf{x} \pmod{\mathbf{B}}$ to denote the corresponding $\mathbf{y} \in \mathcal{P}(\mathbf{B})$ for a given $\mathbf{x} \in \mathbb{R}^n$. Note that $\mathbf{y}$ is computable in polynomial time given $\mathbf{B}$ and $\mathbf{x}$. For a lattice Λ, we denote the *dual lattice* of Λ as Λ^*, defined by

$$\Lambda^* = \{\mathbf{x} \in \mathbb{R}^n : \forall \mathbf{v} \in \Lambda, \langle \mathbf{x}, \mathbf{v} \rangle \in \mathbb{Z}\}.$$

For $i \in [n]$, we can define the ith successive minimum of a lattice Λ to be the smallest λ_i such that there exist i linearly independent lattice points of ℓ_2 norm at most λ_i. Letting B denote the unit ball, this can be phrased as

$$\lambda_i(\Lambda) := \min\{r : \dim(\mathrm{span}(\Lambda \cap rB)) \ge i\}.$$

Note that $\lambda_1(\Lambda)$ is the *minimum distance* of Λ.

We also define the *covering radius* $\nu(\Lambda)$ of a lattice Λ, defined by

$$\nu(\Lambda) = \max_{\mathbf{x}\in\mathbb{R}^n} \min_{\mathbf{v}\in\Lambda} \|\mathbf{x} - \mathbf{v}\|_2 .$$

That is, $\nu(\Lambda)$ is the smallest real number such that every element of $\mathbb{R}^n$ has distance at most $\nu(\Lambda)$ from (some point in) Λ.

We now define some fundamental (worst-case) lattice problems.

Definition 1 (Shortest Independent Vectors Problem)**.** *For a parameter* $\gamma : \mathbb{N} \to \mathbb{R}_{\geq 1}$*, the* shortest independent vectors problem (SIVP_γ) *is a (worst-case) search problem defined as follows. Given an invertible basis* $\mathbf{B} \in \mathbb{R}^{n \times n}$ *as input, output* n *vectors* $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_n] \in \mathbb{R}^{n \times n}$ *such that the following hold:*

- $\mathbf{S}$ *is linearly independent.*
- *For all* $i \in [n]$, $\mathbf{s}_i \in \mathcal{L}(\mathbf{B})$*;*
- $\|\mathbf{S}\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$.

Definition 2 (Covering Radius Problem)**.** *For a parameter* $\gamma : \mathbb{N} \to \mathbb{R}_{\geq 1}$*, the* gap covering radius problem (GapCRP_γ) *is a (worst-case) decision problem defined as follows. Given an invertible basis* $\mathbf{B} \in \mathbb{R}^{n \times n}$ *and threshold* $\theta \in \mathbb{R}_{>0}$ *as input, output* 1 *if* $\nu(\mathcal{L}(\mathbf{B})) \leq \theta$ *and* 0 *if* $\nu(\mathcal{L}(\mathbf{B})) > \gamma(n) \cdot \theta$.

Definition 3 (Guaranteed Distance Decoding)**.** *For a parameter* $\gamma : \mathbb{N} \to \mathbb{R}_{\geq 1}$*, the* guaranteed distance decoding (GDD_γ) *is a (worst-case) search problem defined as follows. Given an invertible basis* $\mathbf{B} \in \mathbb{R}^{n \times n}$ *and target vector* $\mathbf{t} \in \mathbb{R}^n$ *as input, output a lattice point* $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ *such that* $\|\mathbf{t} - \mathbf{x}\|_2 \leq \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$.

We note that GDD is often defined using $\nu(\cdot)$ instead of $\lambda_n(\cdot)$, as using $\nu(\cdot)$ guarantees solutions for all $\gamma \geq 1$. However, throughout this paper, we will be in the regime where $\gamma(n) \geq \sqrt{n}/2$, which guarantees a solution even with $\lambda_n(\cdot)$ because $\lambda_n(\Lambda) \geq \frac{2}{\sqrt{n}} \nu(\Lambda)$ for all full-rank lattices Λ [GMR04, Lemma 4.3].

We will also use an intermediate problem called Incremental Guaranteed Distance Decoding (IncGDD) defined by [MR07].

Definition 4 (Incremental GDD, Definition 5.6 of [MR07])**.** *For a parameter* $\gamma : \mathbb{N} \to \mathbb{R}_{\geq 1}$*, the* incremental guaranteed distance decoding (IncGDD_γ) *problem is a (worst-case) search problem defined as follows. Given as input:*

- *An invertible basis* $\mathbf{B} \in \mathbb{R}^{n \times n}$*,*
- *A set* $\mathbf{S}$ *of* n *linearly independent vectors in* $\mathcal{L}(\mathbf{B})$ *(represented as columns of* $\mathbf{S} \in \mathbb{R}^{n \times n}$*),*
- *A target point* $\mathbf{t} \in \mathbb{R}^n$*, and*
- *A parameter* $r > \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$*,*

output some vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ *such that* $\|\mathbf{s} - \mathbf{t}\|_2 \leq r + \dfrac{\|\mathbf{S}\|}{8}$.

Comparison to [MR07]. Definition 4 differs from [MR07, Definition 5.6] in two ways. First, instead of using $\lambda_n(\mathcal{L}(\mathbf{B}))$, [MR07] considers general functions ϕ mapping n-dimensional lattices to $\mathbb{R}_{>0}$. Second, instead of fixing the constant 8, [MR07] parametrizes this more generally by some constant g.

As shown in [MR07], there are reductions from these worst-case lattice problems to IncGDD.

Lemma 3 (Lemma 5.10 in [MR07]). *For any $\gamma(n) \geq 1$, there is a reduction from* $\mathrm{SIVP}_{8\gamma}$ *to* IncGDD_{γ}.

Lemma 4 (Combining Lemmas 5.11, 5.12 in [MR07]). *For any $\gamma(n) \geq 1$, there is a (randomized) reduction from* $\mathrm{GapCRP}_{12\gamma}$ *to* IncGDD_{γ}.

Lemma 5 (Lemma 5.11 in [MR07]). *For any $\gamma(n) \geq 1$, there is a reduction from* $\mathrm{GDD}_{3\gamma}$ *to* IncGDD_{γ}.

The best known algorithms for these lattice problems employ *lattice reduction* techniques, based on LLL and BKZ [LLL82, Sch87]. They currently all have the following time-approximation trade-off: For a tunable parameter k, in dimension n, it is possible to solve lattice problems with approximation factor $\gamma = 2^{\widetilde{O}(n/k)}$ in time $2^{\widetilde{O}(k)}$, where $\widetilde{O}(\cdot)$ hides polylog factors in n. Equalizing these terms gives a $2^{\widetilde{O}(\sqrt{n})}$-time algorithm to solve $2^{\widetilde{O}(\sqrt{n})}$-approximate lattice problems. Despite extensive study in the lattice literature, no better algorithms are known (that improve the exponent by more than a polylog factor). In particular, the following two assumptions stand:

Assumption 1 (Polynomial Hardness of Approximate Worst-case Lattice Problems). *For every polynomial $\gamma(n)$, at least one of* SIVP_{γ}, GapCRP_{γ}, *or* GDD_{γ} *requires super-polynomial time to solve.*

Assumption 2 (Subexponential Hardness of Approximate Worst-case Lattice Problems). *For all constants $\varepsilon > 0$, at least one of* SIVP_{γ}, GapCRP_{γ}, *or* GDD_{γ} *requires time $2^{\omega(n^{1/2-\varepsilon})}$ to solve, where $\gamma(n) = 2^{n^{1/2-\varepsilon}}$.*

2.3 Continuous and Discrete Gaussian Measures

For $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_{>0}$, we use the notation $\mathcal{N}(\mu, \sigma^2)$ to denote the (univariate) Normal distribution with mean μ and standard deviation σ. More generally, in $n \in \mathbb{N}$ variables, for $\boldsymbol{\mu} \in \mathbb{R}^n$ and positive semi-definite $\Sigma \in \mathbb{R}^{n \times n}$, we use the notation $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ to denote the multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix Σ. For the special case where $\Sigma = \sigma^2 \mathbf{I}_n$ for some $\sigma \in \mathbb{R}_{>0}$, let $\varphi_{\sigma,\boldsymbol{\mu}}(\cdot)$ denote the probability density function of $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_n)$, i.e.,

$$\varphi_{\sigma,\boldsymbol{\mu}}(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \cdot \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}\|_2^2}{2\sigma^2}\right).$$

We now recall standard Gaussian measure notions from the lattice literature (specifically, [MR07]). For an input $\mathbf{x} \in \mathbb{R}^n$, we define the Gaussian function $\rho_{s,\boldsymbol{\mu}}(\cdot)$ centered at $\boldsymbol{\mu} \in \mathbb{R}^n$ with scale parameter $s \in \mathbb{R}_{>0}$ to be

$$\rho_{s,\boldsymbol{\mu}}(\mathbf{x}) = \exp\left(-\frac{\pi \|\mathbf{x} - \boldsymbol{\mu}\|_2^2}{s^2}\right) = s^n \cdot \varphi_{s/\sqrt{2\pi},\boldsymbol{\mu}}(\mathbf{x}).$$

Therefore, $\rho_{s,\boldsymbol{\mu}}(\mathbf{x})/s^n$ is the density function for the probability distribution $\mathcal{N}(\boldsymbol{\mu}, \frac{s^2}{2\pi}\mathbf{I}_n)$.

For any discrete $S \subseteq \mathbb{R}^n$, let $\rho_{s,\boldsymbol{\mu}}(S)$ denote the sum

$$\sum_{\mathbf{v} \in S} \rho_{s,\boldsymbol{\mu}}(\mathbf{v}) \in \mathbb{R}.$$

Let Λ be a full-rank n-dimensional lattice. We define the *discrete Gaussian distribution* $D_{\Lambda+\boldsymbol{\mu},s}$ shifted by $\boldsymbol{\mu} \in \mathbb{R}^n$ and scale parameter $s \in \mathbb{R}_{>0}$ to be the distribution with probability mass function

$$D_{\Lambda+\boldsymbol{\mu},s}(\mathbf{x}) = \begin{cases} \dfrac{\rho_{s,0}(\mathbf{x})}{\rho_{s,0}(\Lambda+\boldsymbol{\mu})} = \dfrac{\rho_{s,0}(\mathbf{x})}{\rho_{s,-\boldsymbol{\mu}}(\Lambda)} & \text{if } \mathbf{x} \in \Lambda, \\ 0 & \text{otherwise,} \end{cases}$$
$$= \mathbf{1}[\mathbf{x} \in \Lambda] \cdot \frac{\exp\left(-\pi \left\|\mathbf{x}-\boldsymbol{\mu}\right\|_2^2 / s^2\right)}{\sum_{\mathbf{v}\in\Lambda} \exp\left(-\pi \left\|\mathbf{x}-\boldsymbol{\mu}\right\|_2^2 / s^2\right)}.$$

We will use the fact that there is an efficient sampler for the discrete Gaussian for the special case where $\Lambda = \mathbb{Z}^n$ (see, e.g., [BLP+13, Lemma 2.3]).

For an n-dimensional lattice Λ and $\epsilon > 0$, we define the *smoothing parameter* $\eta_\epsilon(\Lambda)$ to be the smallest $s \in \mathbb{R}_{>0}$ such that $\rho_{1/s,\mathbf{0}}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \epsilon$. We now recall standard results about the smoothing parameter.

Lemma 6 (Lemma 3.3 of [MR07])**.** *For any n-dimensional lattice Λ and $\epsilon > 0$,*

$$\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}} \cdot \lambda_n(\Lambda).$$

Lemma 7 (Lemma 4.1 of [MR07])**.** *For any lattice $\mathcal{L}(\mathbf{B})$ and any $\epsilon > 0$, $s \geq \eta_\epsilon(\mathcal{L}(\mathbf{B}))$, and $\boldsymbol{\mu} \in \mathbb{R}^n$, we have the statistical distance bound*

$$\Delta\left(\mathcal{N}\left(\boldsymbol{\mu}, \frac{s^2}{2\pi}\mathbf{I}_n\right) \mod \mathcal{P}(\mathbf{B}), U(\mathcal{P}(\mathbf{B}))\right) \leq \frac{\epsilon}{2}.$$

Corollary 1. *For any full-rank $\mathbf{B} \in \mathbb{R}^{n\times n}$, $\epsilon > 0$, and $\boldsymbol{\mu} \in \mathbb{R}^n$, if*

$$\sigma \geq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{2\pi^2}} \cdot \lambda_n(\mathcal{L}(\mathbf{B})),$$

then we have the statistical distance bound

$$\Delta\left(\mathcal{N}\left(\boldsymbol{\mu}, \sigma^2\mathbf{I}_n\right) \mod \mathcal{P}(\mathbf{B}), U(\mathcal{P}(\mathbf{B}))\right) \leq \frac{\epsilon}{2}.$$

Proof. This follows by combining Lemmas 6 and 7, where we set $\sigma = s/\sqrt{2\pi}$. □

Lemma 8. *For $\sigma > 0$, let $\mathcal{D}_\sigma$ denote the distribution of outputs when first sampling $\mathbf{v} \sim U([0,1)^n)$ and then outputting a sample from $D_{\mathbb{Z}^n+\mathbf{v},\sigma\sqrt{2\pi}}$. For any $\epsilon \in (0, 1/2)$, if*

$$\sigma \geq \sqrt{\frac{\ln(2n(1+1/\epsilon))}{2\pi^2}},$$

then we have the statistical distance bound

$$\Delta\left(\mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}_n\right), \mathcal{D}_\sigma\right) \leq 4\epsilon.$$

Proof sketch of Lemma 8. Let $s = \sigma\sqrt{2\pi}$. The analysis in [GVV22, Lemma 17] shows that the distributions have statistical distance at most

$$\sup_{\mathbf{v}\in[0,1)^n} \frac{\rho_{s,\mathbf{0}}(\mathbb{Z}^n)}{\rho_{s,\mathbf{0}}(\mathbb{Z}^n+\mathbf{v})} - 1 = \sup_{\mathbf{v}\in[0,1)^n} \frac{\rho_{s,\mathbf{0}}(\mathbb{Z}^n)}{\rho_{s,-\mathbf{v}}(\mathbb{Z}^n)} - 1.$$

Implicit in [MR07, Lemma 4.4] is that as long as $s \geq \eta_\epsilon(\mathbb{Z}^n)$, then

$$\frac{\rho_{s,\mathbf{0}}(\mathbb{Z}^n)}{\rho_{s,-\mathbf{v}}(\mathbb{Z}^n)} \in \left[1, \frac{1+\epsilon}{1-\epsilon}\right] \subseteq [1, 1+4\epsilon],$$

where the last inclusion comes from the bound $\epsilon < 1/2$. Subtracting by 1 and invoking Lemma 6 with $\Lambda = \mathbb{Z}^n$ gives the desired bound. □

We now recall some standard spectral bounds for Gaussian matrices.

Lemma 9 (As in [RV10])**.** *Let $\mathbf{A} \in \mathbb{R}^{n\times m}$ be such that $A_{i,j} \sim_{\text{i.i.d.}} \mathcal{N}(0,1)$. For all $t > 0$, we have*

$$\Pr\left[\sigma_{\max}(\mathbf{A}) \leq \sqrt{n} + \sqrt{m} + t\right] \geq 1 - 2e^{-t^2/2}.$$

In particular, for $m \geq 16n$, by setting $t = \sqrt{n}$, we have

$$\Pr\left[\sigma_{\max}(\mathbf{A}) \leq \frac{3\sqrt{m}}{2}\right] \geq 1 - 2e^{-n/2}.$$

We now recall standard tail bounds for the $\chi^2(n)$ distribution, corresponding to ℓ_2-norm bounds on Gaussian vectors.

Lemma 10 (As in [LM00], Corollary of Lemma 1)**.** *For any $t \geq 0$, we have*

$$\Pr_{\mathbf{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_n)}\left[\|\mathbf{v}\|_2^2 \geq n + 2\sqrt{tn} + 2t\right] \leq e^{-t}.$$

In particular, setting $t = n/4$, we get

$$\Pr_{\mathbf{v}\sim\mathcal{N}(\mathbf{0},\mathbf{I}_n)}\left[\|\mathbf{v}\|_2 \geq \sqrt{5/2}\cdot\sqrt{n}\right] \leq e^{-n/4}.$$

2.4 Symmetric Perceptrons and Number Partitioning

Definition 5 (Symmetric Binary Perceptron Problem)**.** *For a parameter $\kappa : \mathbb{R}_{\geq 1} \to [0,1]$, the* symmetric binary perceptron (SBP_κ) *problem is an average-case search problem defined as follows. Given a random Gaussian matrix $\mathbf{A} \sim \mathcal{N}(0,1)^{n\times m}$ as input where $m \geq n$, output a vector $\mathbf{x} \in \{-1,1\}^m$ such that $\|\mathbf{Ax}\|_\infty \leq \kappa(m/n)\cdot\sqrt{m}$.*

Definition 6 (Number Partitioning Problem)**.** *For a parameter $\kappa : \mathbb{N} \to [0,1]$, the* number partitioning problem (NPP_κ) *is an average-case search problem defined as follows. Given a random Gaussian vector $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_m)$, output a vector $\mathbf{x} \in \{-1,1\}^m$ such that $|\mathbf{a}^\top\mathbf{x}| \leq \kappa(m)\cdot\sqrt{m}$.*

We emphasize that NPP_κ is exactly a special case of SBP_κ (when setting $n = 1$).

2.5 Worst-case to Average-case Reductions

We recall basic notions of reductions between (search) worst-case and average-case computational problems. Let $A \in \mathsf{FNP}$ be a worst-case search problem, and let $B \in \mathsf{FNP}$ be an average-case search problem defined over some distribution family $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$. We say that there is a $T(n)$-time reduction from A to B if for all non-negligible functions μ, there exists a randomized $\mathsf{poly}(T)$-time oracle Turing machine $M^{(\cdot)}$ such that for all (possibly randomized) $\mathcal{O}$ such that

$$\Pr_{y\leftarrow\mathcal{D}_n}[(y,\mathcal{O}(y)) \in B] \geq \mu(n),$$

for all $n \in \mathbb{N}$, it holds that for all $x \in \{0,1\}^*$,

$$\Pr_r\left[\left(x, M^{\mathcal{O}}(x;r)\right) \in A\right] \geq \frac{2}{3},$$

where r denotes the internal randomness of M. We emphasize that the polynomial in the $\mathsf{poly}(T)$ run-time of $M^{(\cdot)}$ may depend on the non-negligible function μ. Since x is worst-case and $A \in \mathsf{FNP}$, standard amplification applies to make the success probability of M exponentially close to 1.

3 Reduction to Symmetric Binary Perceptrons

Theorem 3. *Suppose $\kappa(x) = 1/x^{1/2+\varepsilon}$ for some constant $\varepsilon > 0$. There exists some polynomial $\gamma(n) = n^{O(1/\varepsilon)}$ such that there is a polynomial-time reduction from IncGDD_γ to SBP_κ.*

We now state our main corollary for symmetric binary perceptrons.

Corollary 2. *Suppose there is a polynomial time algorithm for SBP_κ (on average) for $\kappa(x) = 1/x^{1/2+\varepsilon}$ for some constant $\varepsilon > 0$ that succeeds with non-negligible probability. Then, there are randomized polynomial time algorithms for the (worst-case) lattice problems SIVP_γ, GapCRP_γ, and GDD_γ for some polynomial $\gamma(n)$. In particular, Assumption 1 implies that there is no polynomial time algorithm for SBP_κ (on average) for $\kappa(x) = 1/x^{1/2+\varepsilon}$ for some constant $\varepsilon > 0$ that succeeds with non-negligible probability.*

More generally, if there is a $T(n,m)$-time algorithm for SBP_κ (on average) for $\kappa(x) = 1/x^{1/2+\varepsilon}$ for some constant $\varepsilon > 0$ that succeeds with non-negligible probability, then there are randomized $\mathsf{poly}(n, T(n, \mathsf{poly}(n)))$-time algorithms for the (worst-case) lattice problems SIVP_γ, GapCRP_γ, and GDD_γ for some polynomial $\gamma(n)$.

Proof of Corollary 2. This follows by directly composing Lemmas 3 to 5 and Theorem 3. □

Remark 1. *For a sharper bound on κ in Theorem 3 and Corollary 2, we can instead assume subexponential hardness of approximate worst-case lattice problems. Specifically, for $\kappa(x) = \frac{1}{\sqrt{x}\log^{1+c}(x)}$ where $c > 0$, we can set $m = 2^{O\left(n^{3/(2c)}\right)}$ and $\gamma(n) = 2^{O\left(n^{3/(2c)}\right)}$ in the reduction from IncGDD. In particular, for $c > 3$, Assumption 2 implies that there is no polynomial time algorithm for SBP_κ (on average) for $\kappa(x) = \frac{1}{\sqrt{x}\log^{1+c}(x)}$ that succeeds with non-negligible probability.*

3.1 Proof of Theorem 3

We now prove Theorem 3.

Proof of Theorem 3. Let $(\mathbf{B} \in \mathbb{R}^{n\times n}, \mathbf{S} \in \mathbb{R}^{n\times n}, \mathbf{t} \in \mathbb{R}^n, r \in \mathbb{R})$ be the given IncGDD instance. Let

$$\begin{aligned}
\sigma_2 &= \ln n, \\
m &= \left\lceil \left(8\sigma_2 n^{3/2+\varepsilon}\right)^{1/\varepsilon} \right\rceil = n^{\Theta(1/\varepsilon)}, \\
\sigma_1 &= \frac{r}{4m}, \\
\gamma(n) &= 4m \ln n = n^{\Theta(1/\varepsilon)}.
\end{aligned}$$

Sample $\mathbf{u}_1 \sim \mathcal{N}(\mathbf{t}, \sigma_1^2 \mathbf{I}_n)$, $\mathbf{u}_2, \cdots, \mathbf{u}_m \sim_{\text{i.i.d.}} \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}_n)$. Let $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$. Sample m uniformly random lattice vectors $\mathbf{v}_i \in \mathcal{L}(\mathbf{B}) \mod \mathcal{P}(\mathbf{S})$ (see [Mic04, Proposition 2.9]), and let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m] \in \mathbb{R}^{n\times m}$. Define

$$\widetilde{\mathbf{A}} = \mathbf{S}^{-1}(\mathbf{V} + \mathbf{U}) \mod \mathbb{Z}^n \in [0,1)^{n\times m}.$$

Proposition 1. *The distribution of $\widetilde{\mathbf{A}}$ is statistically close to $U([0,1)^{n\times m})$.*

Proof of Proposition 1. Recall that by definition of IncGDD, we know

$$r > \gamma(n) \cdot \lambda_n(\mathcal{L}(\mathbf{B})) = 4m \ln n \cdot \lambda_n(\mathcal{L}(\mathbf{B})).$$

In anticipation of applying Corollary 1, we observe that for sufficiently large n,

$$\sigma_1 = \frac{r}{4m} \geq \ln n \cdot \lambda_n(\mathcal{L}(\mathbf{B})) \geq \sqrt{\frac{\ln\left(2n\left(1 + 1/e^{-\ln^2(n)}\right)\right)}{2\pi^2}} \cdot \lambda_n(\mathcal{L}(\mathbf{B})).$$

Therefore, we can invoke Corollary 1 m times (once with $\boldsymbol{\mu} = \mathbf{t}$, $m-1$ times with $\boldsymbol{\mu} = \mathbf{0}$) and the triangle inequality to see that

$$\Delta\left(\mathbf{U} \mod \mathcal{P}(\mathbf{B}), U(\mathcal{P}(\mathbf{B}))^m\right) \leq m \cdot \frac{e^{-\ln^2(n)}}{2} = \mathsf{negl}(n).$$

Since $\mathbf{V}$ has columns that are uniform elements of $\mathcal{L}(\mathbf{B}) \mod \mathcal{P}(\mathbf{S})$, and since $\mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B})$, it follows that

$$\Delta\left(\mathbf{U} + \mathbf{V} \mod \mathcal{P}(\mathbf{S}), U(\mathcal{P}(\mathbf{S}))^m\right) \leq \mathsf{negl}(n).$$

Multiplying on the left by $\mathbf{S}^{-1}$ gives

$$\Delta\left(\widetilde{\mathbf{A}}, U(\mathcal{P}(\mathbb{Z}^n))^m\right) \leq \mathsf{negl}(n),$$

or equivalently, that $\widetilde{\mathbf{A}}$ is statistically close to uniform over $[0,1)^{n\times m}$. □

For $j \in [m]$, let $\widetilde{\mathbf{a}}_j \in [0,1)^n$ denote the jth column of $\widetilde{\mathbf{A}}$. For all $j \in [m]$, sample $\mathbf{w}_j \sim D_{\mathbb{Z}^n + \widetilde{\mathbf{a}}_j, \sigma_2\sqrt{2\pi}}$. Let $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_m]$. Note that by construction, $\mathbf{w}_j \equiv \widetilde{\mathbf{a}}_j \mod \mathbb{Z}^n$, i.e., $\mathbf{W} \equiv \widetilde{\mathbf{A}} \mod \mathbb{Z}^n$. In anticipation of applying Lemma 8, we observe that for sufficiently large n,

$$\sigma_2 = \ln(n) \geq \sqrt{\frac{\ln\left(2n\left(1 + 1/e^{-\ln^2(n)}\right)\right)}{2\pi^2}}.$$

Therefore, by invoking Lemma 8 m times, the triangle inequality, and Proposition 1, we see that

$$\Delta\left(\mathbf{W}, \mathcal{N}\left(0, \sigma_2^2\right)^{n\times m}\right) \leq 4me^{-\ln^2(n)} + \mathsf{negl}(n) = \mathsf{negl}(n).$$

Let $\mathbf{A} = \frac{1}{\sigma_2}\mathbf{W}$. It follows that $\mathbf{A}$ is statistically close to $\mathcal{N}(0,1)^{n\times m}$.

Feed $\mathbf{A}$ into the SBP_κ solver to receive some $\mathbf{x} \in \{-1,1\}^m$ such that $\|\mathbf{A}\mathbf{x}\|_\infty \leq \kappa(m/n) \cdot \sqrt{m}$. Let $\mathbf{e} = -\mathbf{A}\mathbf{x} \in \mathbb{R}^n$ with $\|\mathbf{e}\|_\infty \leq \kappa(m/n) \cdot \sqrt{m}$. Let $x_1 \in \{-1,1\}$ be the first entry of $\mathbf{x}$. Let $\mathbf{e}' = \sigma_2\mathbf{e} \in \mathbb{R}^n$. The reduction outputs $\mathbf{s} = x_1(\mathbf{U}\mathbf{x} + \mathbf{S}\mathbf{e}') \in \mathbb{R}^n$.

We first argue that $\mathbf{s} \in \mathcal{L}(\mathbf{B})$. Since $\mathbf{A}\mathbf{x} + \mathbf{e} = \mathbf{0}$, by scaling up by σ_2, we have $\mathbf{W}\mathbf{x} + \mathbf{e}' = \mathbf{0}$. Since $\mathbf{x} \in \mathbb{Z}^m$,

$$\mathbf{0} = \mathbf{W}\mathbf{x} + \mathbf{e}' \equiv \widetilde{\mathbf{A}}\mathbf{x} + \mathbf{e}' \equiv \mathbf{S}^{-1}(\mathbf{V} + \mathbf{U})\mathbf{x} + \mathbf{e}' \mod \mathbb{Z}^n,$$

meaning that $\mathbf{S}^{-1}(\mathbf{V} + \mathbf{U})\mathbf{x} + \mathbf{e}' \in \mathbb{Z}^n$, and thus

$$(\mathbf{V} + \mathbf{U})\mathbf{x} + \mathbf{S}\mathbf{e}' \in \mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B}).$$

Since $\mathbf{V}$ contains vectors in $\mathcal{L}(\mathbf{B})$ and $\mathbf{x} \in \mathbb{Z}^m$, we can subtract by $\mathbf{V}\mathbf{x}$ to get

$$\mathbf{U}\mathbf{x} + \mathbf{S}\mathbf{e}' \in \mathcal{L}(\mathbf{B}).$$

Multiplying by the sign $x_1 \in \{-1,1\}$ gives

$$\mathbf{s} = x_1(\mathbf{U}\mathbf{x} + \mathbf{S}\mathbf{e}') \in \mathcal{L}(\mathbf{B}),$$

as desired.

We next argue that the norm of $\mathbf{s} - \mathbf{t}$ is small. Decompose $\mathbf{x}$ as $\mathbf{x}^\top = [x_1 \| \mathbf{x}_{-1}^\top]$, and decompose $\mathbf{U}$ as $\mathbf{U} = [\mathbf{u}_1 \| \mathbf{U}_{-1}]$. We then have

$$\begin{aligned}
\mathbf{s} = x_1(\mathbf{U}\mathbf{x} + \mathbf{S}\mathbf{e}') &= x_1(\mathbf{u}_1 x_1 + \mathbf{U}_{-1}\mathbf{x}_{-1} + \mathbf{S}\mathbf{e}') \\
&= x_1^2\mathbf{u}_1 + x_1\mathbf{U}_{-1}\mathbf{x}_{-1} + x_1\mathbf{S}\mathbf{e}' \\
&= \mathbf{u}_1 + x_1\mathbf{U}_{-1}\mathbf{x}_{-1} + x_1\mathbf{S}\mathbf{e}' \\
&= \mathbf{t} + \mathbf{u}_1' + x_1\mathbf{U}_{-1}\mathbf{x}_{-1} + x_1\mathbf{S}\mathbf{e}',
\end{aligned}$$

where the distribution of $\mathbf{u}_1'$ is $\mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}_n)$. It follows that with all but negligible probability,

$$\begin{aligned}
\|\mathbf{s}-\mathbf{t}\|_2 &= \left\|\mathbf{u}_1' + x_1 \mathbf{U}_{-1}\mathbf{x}_{-1} + x_1 \mathbf{S}\mathbf{e}'\right\|_2 \\
&\le \left\|\mathbf{u}_1'\right\|_2 + \|\mathbf{U}_{-1}\mathbf{x}_{-1}\|_2 + \left\|\mathbf{S}\mathbf{e}'\right\|_2 \\
&\le \sigma_1\sqrt{\frac{5n}{2}} + \sigma_{\max}(\mathbf{U}_{-1})\,\|\mathbf{x}_{-1}\|_2 + \left\|\mathbf{S}\mathbf{e}'\right\|_2 && \text{(by Lemma 10)} \\
&\le \sigma_1\sqrt{\frac{5n}{2}} + \sigma_{\max}(\mathbf{U}_{-1})\sqrt{m-1} + \|\mathbf{S}\|\left\|\mathbf{e}'\right\|_1 && \text{(by Lemma 1)} \\
&\le \sigma_1\sqrt{\frac{5n}{2}} + \sigma_{\max}(\mathbf{U}_{-1})\sqrt{m-1} + n\,\|\mathbf{S}\|\left\|\mathbf{e}'\right\|_\infty && \text{(by Lemma 2)} \\
&\le \sigma_1\sqrt{\frac{5n}{2}} + \frac{3\sigma_1\sqrt{m}}{2}\cdot\sqrt{m-1} + n\,\|\mathbf{S}\|\left\|\mathbf{e}'\right\|_\infty && \text{(by Lemma 9)} \\
&\le 4\sigma_1 m + n\,\|\mathbf{S}\|\left\|\mathbf{e}'\right\|_\infty \\
&\le r + n\,\|\mathbf{S}\|\left\|\mathbf{e}'\right\|_\infty .
\end{aligned}$$

Therefore, it suffices to show that $\|\mathbf{e}'\|_\infty \le 1/(8n)$. Recall that we have

$$\left\|\mathbf{e}'\right\|_\infty = \sigma_2\,\|\mathbf{e}\|_\infty \le \sigma_2 \cdot \kappa(m/n)\cdot\sqrt{m} = \sigma_2 \cdot \left(\frac{n}{m}\right)^{1/2+\varepsilon}\sqrt{m} = \sigma_2 \cdot \frac{n^{1/2+\varepsilon}}{m^\varepsilon}.$$

Since $m \ge (8\sigma_2 n^{3/2+\varepsilon})^{1/\varepsilon}$, we have $\|\mathbf{e}\|_\infty \le 1/(8n)$, as desired.

Lastly, we note that the SBP_κ solver need only succeed with some non-negligible probability μ. As a result, we can repeat this whole process $O(1/\mu) = \mathsf{poly}(n, m)$ times, and since we can efficiently verify whether the SBP_κ solver succeeded, the reduction will still go through. □

3.2 Variants and Generalizations

We mention a few variants of SBP for which the reduction in Theorem 3 would also apply. As they are not critical to our main result, for simplicity, we only sketch the justifications.

1. **Uniform A.** Instead of having $\mathbf{A} \sim \mathcal{N}(0,1)^{n\times m}$, if we had a SBP solver that worked with $\mathbf{A} \sim U([0,1])^{n\times m}$, our reduction would actually be simpler and more direct. In particular, there would be no need for discrete Gaussian sampling.

2. **Zero entries in x.** Instead of requiring $\mathbf{x} \in \{\pm 1\}^m$ from the SBP solver, if we instead allowed $\mathbf{x} \in \{-1, 0, 1\}^m$ with $\mathbf{x} \ne \mathbf{0}$ from the SBP solver, a similar reduction to the one in Theorem 3 would work as well. The main difference is that the reduction would instead "guess" a coordinate $j \in [m]$ for which $x_j \ne 0$ (with success probability at least $1/m$) and put the vector $\mathbf{t}$ in the mean of that coordinate, instead of the first coordinate. (See [MR07] for more rigorous details.)

   A priori, the version of SBP that allows zero-entries in $\mathbf{x}$ is in fact a lot easier. In particular, for $\kappa(x) = 1/\sqrt{x}$, setting $\mathbf{x} = (1, 0, \ldots, 0)^\top$ would get $\|\mathbf{A}\mathbf{x}\|_\infty \le \widetilde{O}(1) \ll \kappa(m/n)\sqrt{m} = \sqrt{n}$

with high probability by standard Gaussian tail bounds. However, in our reduction, we have $\kappa(x) = 1/x^{1/2+\varepsilon}$. The bound on $\|\mathbf{Ax}\|_\infty$ is

$$\|\mathbf{Ax}\|_\infty \le \kappa(m/n)\sqrt{m} = \frac{\sqrt{m}}{(m/n)^{1/2+\varepsilon}} = \frac{n^{1/2+\varepsilon}}{m^\varepsilon}.$$

In our reduction, we set m so that $m^\varepsilon \gg n^{1/2+\varepsilon}$, making $\|\mathbf{Ax}\|_\infty \ll 1$, in particular, a stronger requirement than $\|\mathbf{Ax}\|_\infty \le \widetilde{O}(1)$.

3. **Larger $\mathbf{x} \in \mathbb{Z}^m$.** Instead of (in particular) requiring $\|\mathbf{x}\|_\infty \le 1$ from the SBP solver, one could relax this requirement to $\|\mathbf{x}\|_\infty \le B$ for some larger bound $B \in \mathbb{N}$. In addition to the modifications discussed in Item 2, we would put the vector $\mathbf{t}/z$ in the mean of that coordinate, where $z \sim U(\{-B, -B+1, \dots, B-1, B\} \setminus \{0\})$. The runtime, γ, and κ would now worsen by a factor of $\Theta(B)$. (See [MR07] for more rigorous details.)

4 Reduction to Number Partitioning

Lemma 11 (Chinese Remainder Theorem)**.** *Let $p_1, \dots, p_n \in \mathbb{N}$ be distinct positive prime numbers. For $q = \prod_{i\in[n]} p_i$, there is a group isomorphism*

$$\varphi : \bigoplus_{i\in[n]} \mathbb{Z}/p_i\mathbb{Z} \longrightarrow \mathbb{Z}/q\mathbb{Z}.$$

Moreover, this map can be written as

$$\varphi : (y_1, \cdots, y_n) \mapsto \sum_{i\in[n]} c_i y_i$$

for $c_i \in \mathbb{Z}$ such that q/p_i divides c_i for all $i \in [n]$ (so that this map is well-defined). Furthermore, the inverse map $\varphi^{-1} : \mathbb{Z}/q\mathbb{Z} \to \bigoplus_{i\in[n]} \mathbb{Z}/p_i\mathbb{Z}$ can be written as

$$\varphi^{-1} : z \mapsto (z, z, \dots, z),$$

where for all $i \in [n]$, $z \in \mathbb{Z}/q\mathbb{Z} = \{0, \dots, q-1\}$ is interpreted directly as an element of $\mathbb{Z}/p_i\mathbb{Z} = \{0, \dots, p_i - 1\}$ by reduction modulo p_i.

Lemma 12 (Normalized CRT)**.** *Let $p_1, \dots, p_n \in \mathbb{N}$ be distinct positive prime numbers. For $q = \prod_{i\in[n]} p_i$, there is a group isomorphism*

$$\widetilde{\varphi} : \bigoplus_{i\in[n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z} \longrightarrow 1/q \cdot \mathbb{Z}/q\mathbb{Z}.$$

Moreover, there exists an integer vector $\mathbf{c} \in \mathbb{Z}^n$ such that this map can be written as

$$\widetilde{\varphi} : (y_1, \cdots, y_n) \mapsto \mathbf{c}^\top \mathbf{y} = \sum_{i\in[n]} c_i y_i.$$

Furthermore, the inverse map $\widetilde{\varphi}^{-1} : 1/q \cdot \mathbb{Z}/q\mathbb{Z} \to \bigoplus_{i\in[n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z}$ can be written as

$$\widetilde{\varphi}^{-1} : z \mapsto \left(\frac{q}{p_1}z, \frac{q}{p_2}z, \dots, \frac{q}{p_n}z\right).$$

Proof of Lemma 12. This follows directly by Lemma 11, by setting

$$\widetilde{\varphi}(y_1, \dots, y_n) = 1/q \cdot \varphi(p_1 y_1, p_2 y_2, \dots, p_n y_n).$$

□

Letting $\mathbf{p} = (p_1, \dots, p_n) \in \mathbb{Z}^n$, we use the notation $\lfloor \cdot \rfloor_{\mathbf{p}} : [0,1)^n \to \bigoplus_{i \in [n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z}$ to denote the function

$$\lfloor \cdot \rfloor_{\mathbf{p}} : \mathbf{v} \mapsto \left(\frac{\lfloor v_1 p_1 \rfloor}{p_1}, \dots, \frac{\lfloor v_1 p_n \rfloor}{p_n} \right).$$

More generally, for elements in $[0,1)^{n \times m}$, we extend $\lfloor \cdot \rfloor_{\mathbf{p}} : [0,1)^{n \times m} \to \left(\bigoplus_{i \in [n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z} \right)^m$ to operate column-wise.

We will use the following basic fact.

Lemma 13. *For any* $\mathbf{A} \in [0,1)^{n \times m}$ *and* $\mathbf{x} \in \mathbb{R}^m$,

$$\left\| \left(\mathbf{A} - \lfloor \mathbf{A} \rfloor_{\mathbf{p}} \right) \mathbf{x} \right\|_1 \leq \frac{n}{\min_{i \in [n]} p_i} \|\mathbf{x}\|_1 .$$

Proof. Let $p^* = \min_{i \in [n]} p_i$. Letting $\mathbf{M} = \mathbf{A} - \lfloor \mathbf{A} \rfloor_{\mathbf{p}}$, by properties of the floor function, we have $\mathbf{M} \in \left[0, \frac{1}{p^*}\right)^{n \times m}$. Let $\mathbf{m}_1, \cdots, \mathbf{m}_n \in \left[0, \frac{1}{p^*}\right)^m$ be the rows of $\mathbf{M}$. We have

$$\|\mathbf{M}\mathbf{x}\|_1 = \sum_{i \in [n]} \left| \mathbf{m}_i^\top \mathbf{x} \right| \leq \sum_{i \in [n]} \|\mathbf{m}_i\|_\infty \|\mathbf{x}\|_1 \leq \frac{n}{p^*} \|\mathbf{x}\|_1 ,$$

as desired, where we have used Hölder's inequality to see that $|\mathbf{m}_i^\top \mathbf{x}| \leq \|\mathbf{m}_i\|_\infty \|\mathbf{x}\|_1$. □

We also use the following fact about the density of prime numbers.

Lemma 14. *For all sufficiently large* $N \in \mathbb{N}$*, there exist at least* $N/\ln(N)$ *distinct prime numbers in the interval* $[N, 10N]$.

Proof. For $N \in \mathbb{N}$, let $\pi(N) = |\{a \in [N] : a \text{ is prime}\}|$ denote the prime-counting function. By the prime number theorem, we know that for all sufficiently large N,

$$\frac{N}{2 \ln N} \leq \pi(N) \leq \frac{2N}{\ln N}.$$

In particular,

$$\pi(N) \leq \frac{2N}{\ln N}, \quad \pi(10N) \geq \frac{10N}{2\ln(10N)} = \frac{10N}{2\ln(N) + 2\ln(10)} > \frac{4N}{\ln N},$$

for sufficiently large N. Therefore, for sufficiently large N, by taking the difference of the two quantities, there are at least $N/\ln N$ primes in $[N, 10N]$. □

Theorem 4. *Suppose $\kappa(m) = 2^{-\log^{2+\varepsilon} m}$ for some constant $\varepsilon > 0$. Then there exists $\gamma(n) = 2^{O\left(n^{\frac{1}{1+\varepsilon}}\right)}$ such that there is a $\mathsf{poly}(m)$-time reduction from IncGDD_γ in dimension $n = \Omega((\log m)^{1+\varepsilon})$ to NPP_κ (in dimension m).*

We now state our main corollary for number partitioning.

Corollary 3. *Suppose there is a polynomial time algorithm for NPP_κ (on average) for $\kappa(m) = 2^{-\log^{2+\varepsilon} m}$ for some constant $\varepsilon > 0$ that succeeds with non-negligible probability. Then, there are randomized $2^{O\left(n^{\frac{1}{1+\varepsilon}}\right)}$-time algorithms for the (worst-case) lattice problems SIVP_γ, GapCRP_γ, and GDD_γ in dimension n for $\gamma(n) = 2^{O\left(n^{\frac{1}{1+\varepsilon}}\right)}$. In particular, Assumption 2 implies that for all constant $\varepsilon > 0$, there is no polynomial time algorithm for NPP_κ (on average) for $\kappa(m) = 2^{-\log^{3+\varepsilon} m}$ that succeeds with non-negligible probability.*

More generally, suppose there is a $T(m)$-time algorithm for NPP_κ (on average) for $\kappa(m) = 2^{-\log^{2+\varepsilon} m}$ for some constant $\varepsilon > 0$ that succeeds with non-negligible probability. Then, there are randomized $T\left(2^{O\left(n^{\frac{1}{1+\varepsilon}}\right)}\right)$-time algorithms for the (worst-case) lattice problems SIVP_γ, GapCRP_γ, and GDD_γ in dimension n for $\gamma(n) = 2^{O\left(n^{\frac{1}{1+\varepsilon}}\right)}$.

Proof of Corollary 3. This follows by directly composing Lemmas 3 to 5 and Theorem 4. □

We now prove Theorem 4.

Proof of Theorem 4. Let $(\mathbf{B} \in \mathbb{R}^{n\times n}, \mathbf{S} \in \mathbb{R}^{n\times n}, \mathbf{t} \in \mathbb{R}^n, r \in \mathbb{R})$ be the given IncGDD instance. Let

$$\begin{aligned} m &= 2^{10n^{\frac{1}{1+\varepsilon}}}, \\ \sigma_1 &= \frac{r}{4m}, \\ \sigma_2 &= \ln m, \\ \gamma(n) &= 4m \ln m = 40 \ln(2) 2^{10n^{\frac{1}{1+\varepsilon}}} n^{\frac{1}{1+\varepsilon}}. \end{aligned}$$

Let $p_1, \ldots, p_n$ be n distinct prime numbers in the range $[32nm, 320nm]$, which we know must exist for sufficiently large n, m by Lemma 14 (since $m \geq n$). Let $q = \prod_{i=1}^n p_i \leq (320nm)^n$. Sample $\mathbf{u}_1 \sim \mathcal{N}(\mathbf{t}, \sigma_1^2 \mathbf{I}_n)$, $\mathbf{u}_2, \cdots, \mathbf{u}_m \sim_{\text{i.i.d.}} \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}_n)$. Let $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$. Sample m uniformly random lattice vectors $\mathbf{v}_i \in \mathcal{L}(\mathbf{B}) \mod \mathcal{P}(\mathbf{S})$ (see [Mic04, Proposition 2.9]), and let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m] \in \mathbb{R}^{n\times m}$. Define

$$\mathbf{A} = \mathbf{S}^{-1}(\mathbf{V} + \mathbf{U}) \mod \mathbb{Z}^n \in [0,1)^{n\times m}.$$

Proposition 2. *We have the inequality*

$$\Delta(\mathbf{A}, U([0,1)^{n\times m})) \leq \mathsf{negl}(m).$$

Proof. This follows from the same analysis as in Proposition 1. Specifically, since for sufficiently large n and m,

$$\sigma_1 = \frac{r}{4m} \geq \ln m \cdot \lambda_n(\mathcal{L}(\mathbf{B})) \geq \sqrt{\frac{\ln\left(2n\left(1+1/e^{-\ln^2(m)}\right)\right)}{2\pi^2}} \cdot \lambda_n(\mathcal{L}(\mathbf{B})),$$

by Corollary 1 and the triangle inequality, we get a total statistical distance of $m \cdot e^{-\ln^2(m)} = \mathsf{negl}(m)$. □

Let $\widetilde{\varphi}$ be the isomorphism guaranteed by Lemma 12, with $\mathbf{c} \in \mathbb{Z}^n$ being the coefficients defining the linear map $\widetilde{\varphi}$. Let $\mathbf{y} \in \mathbb{R}^m$ be defined by

$$\mathbf{y} = \widetilde{\varphi}(\lfloor \mathbf{A} \rceil_p) + \mathbf{f} \mod \mathbb{Z}^m \in [0,1)^m$$

where $\mathbf{f} \in \mathbb{R}^m$ is sampled as $\mathbf{f} \sim U([0,1/q)^m)$.

We argue that $\Delta(\mathbf{y}, U([0,1)^m)) \leq \mathsf{negl}(m)$ as follows. Since $\mathbf{A}$ is (close to) $U([0,1)^{n\times m})$, it follows that $\lfloor \mathbf{A} \rceil_p$ is (close to) $U(\bigoplus_{i\in[n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z})^m$. Moreover, since $\widetilde{\varphi}$ is a bijection, it follows that $\widetilde{\varphi}(\lfloor \mathbf{A} \rceil_p)$ is (close to) $U(1/q \cdot \mathbb{Z}/q\mathbb{Z})^m$. Since $\mathbf{f} \sim U([0,1/q)^m)$, we can see that $\Delta(\mathbf{y}, U([0,1)^m)) \leq \mathsf{negl}(m)$.

Sample $\mathbf{w} \sim D_{\mathbb{Z}^m+\mathbf{y},\sigma_2\sqrt{2\pi}}$. Note that by construction, $\mathbf{w} \equiv \mathbf{y} \mod \mathbb{Z}^m$. In anticipation of applying Lemma 8, we observe that for sufficiently large m,

$$\sigma_2 = \ln m \geq \sqrt{\frac{\ln\left(2m\left(1+1/e^{-\ln^2(m)}\right)\right)}{2\pi^2}}.$$

Therefore, by invoking Lemma 8, $\Delta(\mathbf{y}, U([0,1)^m)) \leq \mathsf{negl}(m)$, and the triangle inequality, we see that

$$\Delta\left(\mathcal{N}(\mathbf{0}, \sigma_2^2\mathbf{I}_m), \mathbf{w}\right) \leq e^{-\ln^2(m)} + \mathsf{negl}(m) = \mathsf{negl}(m).$$

Let $\mathbf{a} = \frac{1}{\sigma_2}\mathbf{w}$. It follows that $\mathbf{a}$ is statistically close to $\mathcal{N}(\mathbf{0}, \mathbf{I}_m)$. Feed $\mathbf{a}$ into the NPP solver to receive some $\mathbf{x} \in \{-1,1\}^m$ such that $|\mathbf{a}^\top\mathbf{x}| \leq \kappa(m)\sqrt{m}$. Let $e = -\mathbf{a}^\top\mathbf{x} \in \mathbb{R}$ with $|e| \leq \kappa(m)\sqrt{m}$. Let $x_1 \in \{-1,1\}$ be the first entry of $\mathbf{x}$. Let $e' = \sigma_2 e \in \mathbb{R}$, and let $e'' = \mathbf{f}^\top\mathbf{x} + e' \in \mathbb{R}$.

The reduction outputs

$$\mathbf{s} = x_1\left(\mathbf{U}\mathbf{x} - \mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rceil_p)\mathbf{x} + \mathbf{S}\widetilde{\varphi}^{-1}(e'')\right).$$

We first argue that $\mathbf{s} \in \mathcal{L}(\mathbf{B})$. Since $\mathbf{a}^\top\mathbf{x} + e = 0$, by scaling up, we have $\mathbf{w}^\top\mathbf{x} + e' = 0$. Since $\mathbf{x} \in \mathbb{Z}^m$,

$$\begin{aligned} 0 = \mathbf{w}^\top\mathbf{x} + e' \equiv \mathbf{y}^\top\mathbf{x} + e' &= \left(\widetilde{\varphi}(\lfloor \mathbf{A} \rceil_p) + \mathbf{f}\right)^\top \mathbf{x} + e' \mod 1, \\ &\equiv \mathbf{c}^\top \lfloor \mathbf{A} \rceil_p \mathbf{x} + \mathbf{f}^\top\mathbf{x} + e' \mod 1, \\ &\equiv \widetilde{\varphi}(\lfloor \mathbf{A} \rceil_p \mathbf{x}) + e'' \mod 1. \end{aligned}$$

By closure, we know $e'' \in 1/q \cdot \mathbb{Z}/q\mathbb{Z}$, so we have

$$0 \equiv \widetilde{\varphi}(\lfloor \mathbf{A} \rceil_p \mathbf{x}) + e'' \equiv \widetilde{\varphi}(\lfloor \mathbf{A} \rceil_p \mathbf{x}) + \widetilde{\varphi}\left(\widetilde{\varphi}^{-1}(e'')\right) \equiv \widetilde{\varphi}\left(\lfloor \mathbf{A} \rceil_p \mathbf{x} + \widetilde{\varphi}^{-1}(e'')\right) \mod 1.$$

By applying $\widetilde{\varphi}^{-1}$, it follows that

$$\mathbf{A}\mathbf{x} - (\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + \widetilde{\varphi}^{-1}(e'') = \lfloor \mathbf{A} \rfloor_p \mathbf{x} + \widetilde{\varphi}^{-1}(e'') \in \mathbb{Z}^n.$$

Plugging in the definition of $\mathbf{A}$ and since $\mathbf{x} \in \mathbb{Z}^m$,

$$\mathbf{S}^{-1}(\mathbf{V} + \mathbf{U})\mathbf{x} - (\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + \widetilde{\varphi}^{-1}(e'') \in \mathbb{Z}^n.$$

Multiplying by $\mathbf{S}$ on the left gives

$$(\mathbf{V} + \mathbf{U})\mathbf{x} - \mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + \mathbf{S}\widetilde{\varphi}^{-1}(e'') \in \mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B}).$$

Since $\mathbf{V}$ contains vectors in $\mathcal{L}(\mathbf{B})$ and $\mathbf{x} \in \mathbb{Z}^m$, we can subtract by $\mathbf{V}\mathbf{x}$ to get

$$\mathbf{U}\mathbf{x} - \mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + \mathbf{S}\widetilde{\varphi}^{-1}(e'') \in \mathcal{L}(\mathbf{B}).$$

Multiplying by the sign $x_1 \in \{-1, 1\}$ gives

$$\mathbf{s} = x_1 \left(\mathbf{U}\mathbf{x} - \mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + \mathbf{S}\widetilde{\varphi}^{-1}(e'') \right) \in \mathcal{L}(\mathbf{B}),$$

as desired.

We next argue that the norm of $\mathbf{s} - \mathbf{t}$ is small. Decompose $\mathbf{x}$ as $\mathbf{x}^\top = [x_1 \| \mathbf{x}_{-1}^\top]$, and decompose $\mathbf{U}$ as $\mathbf{U} = [\mathbf{u}_1 \| \mathbf{U}_{-1}]$. We then have

$$\begin{aligned}
\mathbf{s} &= x_1\mathbf{U}\mathbf{x} - x_1\mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + x_1\mathbf{S}\widetilde{\varphi}^{-1}(e'') \\
&= x_1(\mathbf{u}_1 x_1 + \mathbf{U}_{-1}\mathbf{x}_{-1}) - x_1\mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + x_1\mathbf{S}\widetilde{\varphi}^{-1}(e'') \\
&= \mathbf{u}_1 + x_1\mathbf{U}_{-1}\mathbf{x}_{-1} - x_1\mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + x_1\mathbf{S}\widetilde{\varphi}^{-1}(e'') \\
&= \mathbf{t} + \mathbf{u}_1' + x_1\mathbf{U}_{-1}\mathbf{x}_{-1} - x_1\mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + x_1\mathbf{S}\widetilde{\varphi}^{-1}(e''),
\end{aligned}$$

where the distribution of $\mathbf{u}_1'$ is $\mathcal{N}(\mathbf{0}, \sigma_1^2\mathbf{I}_n)$. It follows that

$$\begin{aligned}
\|\mathbf{s} - \mathbf{t}\|_2 &= \left\| \mathbf{u}_1' + x_1\mathbf{U}_{-1}\mathbf{x}_{-1} - x_1\mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} + x_1\mathbf{S}\widetilde{\varphi}^{-1}(e'') \right\|_2 \\
&\le \|\mathbf{u}_1'\|_2 + \|\mathbf{U}_{-1}\mathbf{x}_{-1}\|_2 + \left\| \mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} \right\|_2 + \left\| \mathbf{S}\widetilde{\varphi}^{-1}(e'') \right\|_2 \\
&\le 4\sigma_1 m + \left\| \mathbf{S}(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p)\mathbf{x} \right\|_2 + \left\| \mathbf{S}\widetilde{\varphi}^{-1}(e'') \right\|_2 && \text{(by Lemmas 9 and 10)} \\
&\le 4\sigma_1 m + \|\mathbf{S}\| \left(\left\| \left(\mathbf{A} - \lfloor \mathbf{A} \rfloor_p\right) \mathbf{x} \right\|_1 + \left\| \widetilde{\varphi}^{-1}(e'') \right\|_1 \right) && \text{(by Lemma 1)} \\
&\le 4\sigma_1 m + \|\mathbf{S}\| \left(\frac{n}{\min_{i \in [n]} p_i} \|\mathbf{x}\|_1 + \left\| \widetilde{\varphi}^{-1}(e'') \right\|_1 \right) && \text{(by Lemma 13)} \\
&= r + \|\mathbf{S}\| \left(\frac{nm}{\min_{i \in [n]} p_i} + \left\| \widetilde{\varphi}^{-1}(e'') \right\|_1 \right). \\
&\le r + \|\mathbf{S}\| \left(\frac{1}{16} + \left\| \widetilde{\varphi}^{-1}(e'') \right\|_1 \right).
\end{aligned}$$

It suffices to upper bound $\|\widetilde{\varphi}^{-1}(e'')\|_1$ by $1/16$. Recall from Lemma 12 that

$$\widetilde{\varphi}^{-1}(e'') = \left(\frac{q}{p_1}e'', \frac{q}{p_2}e'', \ldots, \frac{q}{p_n}e''\right) \in \bigoplus_{i\in[n]} 1/p_i \cdot \mathbb{Z}/p_i\mathbb{Z}.$$

For of these entries to be small when viewed in $\mathbb{R}$, we want to ensure that there's no "wraparound." In particular, if

$$e'' \leq \frac{\min_{i\in[n]} p_i}{16qn},$$

then $\|\widetilde{\varphi}^{-1}(e'')\|_1 \leq 1/16$, as desired. Since $\min_{i\in[n]} p_i \geq 32nm$, it suffices to show that

$$e'' \leq \frac{2m}{q}.$$

Recall that $e'' = \mathbf{f}^\top\mathbf{x} + e' = \mathbf{f}^\top\mathbf{x} + \sigma_2 e$, where $\mathbf{f} \sim U([0,1/q)^m)$ and $|e| \leq \kappa(m)\sqrt{m}$. It follows that

$$|e''| \leq \left|\mathbf{f}^\top\mathbf{x}\right| + |e'| \leq \|\mathbf{f}\|_\infty \|\mathbf{x}\|_1 + \sigma_2|e| \leq \frac{m}{q} + \sigma_2 \cdot \kappa(m)\sqrt{m}.$$

Thus, it suffices to show $\sigma_2 \cdot \kappa(m)\sqrt{m} \leq m/q$, or equivalently, $\kappa(m) \leq \sqrt{m}/(q \ln m)$. We have

$$\begin{aligned}
\frac{\sqrt{m}}{q\ln m} = \frac{2^{5n^{\frac{1}{1+\varepsilon}}}}{q10\ln(2)n^{\frac{1}{1+\varepsilon}}} &\geq \frac{2^{5n^{\frac{1}{1+\varepsilon}}}}{10\ln(2)(320nm)^n n^{\frac{1}{1+\varepsilon}}} \\
&= \frac{2^{5n^{\frac{1}{1+\varepsilon}}}}{10\ln(2)(320n)^n 2^{10n^{\frac{2+\varepsilon}{1+\varepsilon}}} n^{\frac{1}{1+\varepsilon}}} \\
&\geq \frac{1}{2^{11n^{\frac{2+\varepsilon}{1+\varepsilon}}}}
\end{aligned}$$

for sufficiently large n. On the other hand,

$$\kappa(m) = \frac{1}{2^{(\log m)^{2+\varepsilon}}} = \frac{1}{2^{\left(10n^{\frac{1}{1+\varepsilon}}\right)^{2+\varepsilon}}} = \frac{1}{2^{10^{2+\varepsilon}n^{\frac{2+\varepsilon}{1+\varepsilon}}}} \leq \frac{1}{2^{100n^{\frac{2+\varepsilon}{1+\varepsilon}}}}.$$

Therefore, $\kappa(m) \leq \sqrt{m}/(q\ln m)$, as desired.

Lastly, we note that the NPP_κ solver need only succeed with some non-negligible probability $\mu(m)$. As a result, we can repeat this whole process $O(1/\mu) = \mathsf{poly}(m)$ times, and since we can efficiently verify whether the NPP_κ solver succeeded, the reduction will still go through. □

Moreover, as NPP is a special case of SBP, all of the generalizations and variants discussed in Section 3.2 apply here to NPP as well.

Acknowledgements. The authors were supported in part by DARPA under Agreement No. HR00112020023, NSF CNS-2154149 and a Simons Investigator Award. The first author is also supported in part by NSF DGE-2141064. We are particularly thankful to David Gamarnik for a stimulating conversation where he described the symmetric perceptron and number partitioning problems to us and posed the question of showing computational hardness for them. We are grateful to Alon Rosen, Kiril Bangachev, Stefan Tiegel and Riccardo Zecchina for valuable discussions. We also thank the anonymous reviewers for their valuable feedback.

References

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 99–108. ACM, 1996. 3, 6, 9

[ALS21] Emmanuel Abbe, Shuangping Li, and Allan Sly. Proof of the contiguity conjecture and lognormal limit for the symmetric perceptron, 2021. 2

[ALS22] Emmanuel Abbe, Shuangping Li, and Allan Sly. Binary perceptron: efficient algorithms can find solutions in a rare well-connected cluster. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 860–873. ACM, 2022. 2

[APZ19] Benjamin Aubin, Will Perkins, and Lenka Zdeborová. Storage capacity in symmetric binary perceptrons. *Journal of Physics A: Mathematical and Theoretical*, 52(29):294003, June 2019. 2

[AR06] Dimitris Achlioptas and Federico Ricci-Tersenghi. On the solution-space geometry of random constraint satisfaction problems. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 130–139. ACM, 2006. 2

[Ban10] Nikhil Bansal. Constructive algorithms for discrepancy minimization. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 3–10. IEEE Computer Society, 2010. 2, 4

[BDVLZ20] Carlo Baldassi, Riccardo Della Vecchia, Carlo Lucibello, and Riccardo Zecchina. Clustering of solutions in the symmetric binary perceptron. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(7):073303, 2020. 2

[BEAKZ24] Damien Barbier, Ahmed El Alaoui, Florent Krzakala, and Lenka Zdeborová. On the atypical solutions of the symmetric binary perceptron. *Journal of Physics A: Mathematical and Theoretical*, 57(19):195202, April 2024. 2

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003. 4

[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, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013. 14

[BRST21] Joan Bruna, Oded Regev, Min Jae Song, and Yi Tang. Continuous LWE. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM*

SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021, pages 694–707. ACM, 2021. 6, 9

[BS20] Nikhil Bansal and Joel H. Spencer. On-line balancing of random inputs. *Random Struct. Algorithms*, 57(4):879–891, 2020. 2

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic prfs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 1–30. Springer, 2015. 8, 9

[Cov65] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965. 2

[DKMR22] Ilias Diakonikolas, Daniel Kane, Pasin Manurangsi, and Lisheng Ren. Cryptographic hardness of learning halfspaces with massart noise. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 6

[DKR23] Ilias Diakonikolas, Daniel Kane, and Lisheng Ren. Near-optimal cryptographic hardness of agnostically learning halfspaces and relu regression under gaussian marginals. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 7922–7938. PMLR, 2023. 6

[Gam21] David Gamarnik. The overlap gap property: A topological barrier to optimizing over random structures. *Proceedings of the National Academy of Sciences*, 118(41):e2108492118, 2021. 3

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. 4

[GK21] David Gamarnik and Eren C. Kızıldağ. Algorithmic obstructions in the random number partitioning problem, 2021. 4

[GKPX22] David Gamarnik, Eren C. Kizildag, Will Perkins, and Changji Xu. Algorithms and barriers in the symmetric binary perceptron model. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 576–587. IEEE, 2022. 2, 3, 6

[GKPX23] David Gamarnik, Eren C. Kizildag, Will Perkins, and Changji Xu. Geometric barriers for stable and online algorithms for discrepancy minimization. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 3231–3263. PMLR, 2023. 2

[GKVZ22] Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. Planting undetectable backdoors in machine learning models : [extended abstract]. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 931–942. IEEE, 2022. 6

[Gl16] David Gamarnik and Quan li. Finding a large submatrix of a gaussian random matrix. *Annals of Statistics*, 46, 02 2016. 2

[GMR04] Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The complexity of the covering radius problem on lattices and codes. In *19th Annual IEEE Conference on Computational Complexity (CCC 2004), 21-24 June 2004, Amherst, MA, USA*, pages 161–173. IEEE Computer Society, 2004. 12

[GVV22] Aparna Gupte, Neekon Vafa, and Vinod Vaikuntanathan. Continuous LWE is as hard as LWE & applications to learning gaussian mixtures. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1162–1173. IEEE, 2022. 6, 9, 15

[HRRY17] Rebecca Hoberg, Harishchandra Ramadas, Thomas Rothvoss, and Xin Yang. Number balancing is as hard as minkowski's theorem and shortest vector. In Friedrich Eisenbrand and Jochen Könemann, editors, *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, volume 10328 of *Lecture Notes in Computer Science*, pages 254–266. Springer, 2017. 4

[JH60] Roger David Joseph and Louise Hay. The number of orthants in n-space intersected by an s-dimensional subspace. 1960. 2

[JL84] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984. 5

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo, and Joel I. Seiferas, editors, *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 193–206. ACM, 1983. 3

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987. 3

[KK82] Narendra Karmarkar and Richard M. Karp. The differencing method of set partitioning. 1982. 4, 6

[KKLO86] Narendra Karmarkar, Richard M Karp, George S Lueker, and Andrew M Odlyzko. Probabilistic analysis of optimum partitioning. *Journal of Applied probability*, 23(3):626–645, 1986. 4

[LLL82] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261:515–534, 1982. 3, 13

[LM00] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000. 15

[LM15] Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM J. Comput.*, 44(5):1573–1582, 2015. 4

[LRR17] Avi Levy, Harishchandra Ramadas, and Thomas Rothvoss. Deterministic discrepancy minimization via the multiplicative weight update method. In Friedrich Eisenbrand and Jochen Könemann, editors, *Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings*, volume 10328 of *Lecture Notes in Computer Science*, pages 380–391. Springer, 2017. 4

[Mic04] Daniele Micciancio. Almost perfect lattices, the covering radius problem, and applications to ajtai's connection factor. *SIAM J. Comput.*, 34(1):118–169, 2004. 17, 22

[MMZ05] M. Mézard, T. Mora, and R. Zecchina. Clustering of solutions in the random satisfiability problem. *Physical Review Letters*, 94(19), May 2005. 2

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. 3, 6, 7, 9, 12, 13, 14, 15, 19, 20

[PX21] Will Perkins and Changji Xu. Frozen 1-rsb structure of the symmetric ising perceptron. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1579–1588. ACM, 2021. 2

[Reg04] Oded Regev. Lattices in computer science - average-case hardness. Lecture Notes for Class (scribe: Elad Verbin). https://cims.nyu.edu/regev/teaching/latticesfall2004/ln/averagecase.pdf, 2004. 8

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009. 3

[Rot14] Thomas Rothvoß. Constructive discrepancy minimization for convex sets. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 140–145. IEEE Computer Society, 2014. 4

[RR23] Victor Reis and Thomas Rothvoss. The subspace flatness conjecture and faster integer programming. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 974–988. IEEE, 2023. 3

[RV10] Mark Rudelson and Roman Vershynin. Non-asymptotic theory of random matrices: extreme singular values. In *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*, pages 1576–1602. World Scientific, 2010. 15

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987. 3, 13

[Spe85] Joel Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289(2):679–706, 1985. 4

[SZB21] Min Jae Song, Ilias Zadik, and Joan Bruna. On the cryptographic hardness of learning single periodic neurons. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29602–29615, 2021. 6

[Tie23] Stefan Tiegel. Hardness of agnostically learning halfspaces from worst-case lattice problems. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 3029–3064. PMLR, 2023. 6

[TMR20] Paxton Turner, Raghu Meka, and Philippe Rigollet. Balancing gaussian vectors in high dimension. In Jacob D. Abernethy and Shivani Agarwal, editors, *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, volume 125 of *Proceedings of Machine Learning Research*, pages 3455–3486. PMLR, 2020. 2

[Win61] Robert O. Winder. Single stage threshold logic. In *2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*, pages 321–332, 1961. 2

[Yak96] Benjamin Yakir. The differencing algorithm ldm for partitioning: A proof of a conjecture of karmarkar and karp. *Mathematics of Operations Research*, 21(1):85–99, 1996. 4

[ZSWB22] Ilias Zadik, Min Jae Song, Alexander S. Wein, and Joan Bruna. Lattice-based methods surpass sum-of-squares in clustering. In Po-Ling Loh and Maxim Raginsky, editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 1247–1248. PMLR, 2022. 6