Provable Dual Attacks on Learning with Errors

Amaury Pouly[1[0000−0002−2549−951X]] and Yixin Shen[2[0000−0002−8657−9337]]

[1] CNRS
amaury.pouly@cnrs.fr
[2] King's College London
yixin.shen@kcl.ac.uk

Abstract. Learning with Errors (LWE) is an important problem for post-quantum cryptography (PQC) that underlines the security of several NIST PQC selected algorithms. Several recent papers [6,23], [31,13] have claimed improvements on the complexity of so-called dual attacks on LWE. These improvements make dual attacks comparable to or even better than primal attacks in certain parameter regimes. Unfortunately, those improvements rely on a number of untested and hard-to-test statistical assumptions. Furthermore, a recent paper [18] claims that the whole premise of those improvements might be incorrect.
The goal of this paper is to improve the situation by proving the correctness of a dual attack without relying on any statistical assumption. Although our attack is greatly simplified compared to the recent ones, it shares many important technical elements with those attacks and can serve as a basis for the analysis of more advanced attacks.
Our main contribution is to clearly identify a set of parameters under which our attack (and presumably other recent dual attacks) can work. Furthermore, our analysis completely departs from the existing statistics-based analysis and is instead rooted in geometry. We also compare the regime in which our algorithm works to the "contradictory regime" of [18]. We observe that those two regimes are essentially complementary.
Finally, we give a quantum version of our algorithm to speed up the computation. The algorithm is inspired by [7] but is completely formal and does not rely on any heuristics.

Keywords: Learning with Errors, Dual attack, Lattice-based cryptography, Quantum algorithm

1 Introduction

The Learning With Errors (LWE) problem [38] has become central to the security of several cryptosystems. Most notably, Kyber (public-key encryption) and Dilithium (signature) have been selected by the NIST for the Post-Quantum Cryptography (PQC) Standardization and rely on algebraic version of LWE for their security proofs. Other advanced cryptographic primitives such as FHE can be built with LWE [12]. This makes LWE security estimates critical for the future of PQC. The *search* LWE *problem* asks to recover the secret $\mathbf{s}$ given one or more "LWE samples[3]." of the form $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b} = \mathbf{As} + \mathbf{e}$, $\mathbf{A}$ is chosen uniformly at random and $\mathbf{e}$ has small entries (more details in Section 2.2).

There are two main approaches to attack the LWE problem: so-called primal and dual attacks. In this paper, we will exclusively focus on dual attacks which have recent attracted some interest due to significant improvements in their complexity. Both primal and dual attacks rely on the BKZ lattice reduction algorithm [40] to obtain short vectors in lattices. The fundamental idea of dual attacks is to use short vectors in the dual of the lattice to detect whether points are close to the lattice or not, an idea that can be traced back to [4]. This allows us to solve the *distinguishing* LWE *problem* where one is asked to detect whether a sample comes from an LWE distribution, or a uniform distribution [34]. In conjunction with some guessing step, this allows one to recover part of the secret by trying several values until we get a point close to the lattice. By repeating this operation a couple of time, we can solve the search LWE problem.

Originally, the main limiting factor (on the complexity) of dual attacks was the need to compute one short vector (a very expensive operation) per LWE sample (more details in Section 3) and compute a score for each secret guess. Since then, a series of improvements have found their way into these attacks. First, a series of work on lattice sieving have shown [37,35,10] that those algorithms produce not only one but in fact exponentially many short vectors "for free". [8] suggested that this idea could be used in dual attacks but it appears that [21] was the first paper to try to analyze it. Independently, [6] used a "rerandomization" technique to produce many short vectors from a single BKZ reduced basis. All those technique greatly reduce the complexity of attack although its correctness relies on an unproven assumption about the quality of those many short vectors. Then [23] noted that instead of computing the score for each secret guess separately, all the scores can be computed at once using a discrete Fourier transform (DFT), essentially reducing the cost to that of a single guess. Following this work, a technical report by the MATZOV group [31] has claimed further improvements by the use of a "modulus switching" technique[4] that significantly reduces the size of the DFT. Two recent work have modified this attack to include a quantum speed up [7] and lattice coding speed up [13].

One issue with the papers above is that the number of statistical assumptions that are necessary to justify the correctness of the algorithms has grown significantly, notably in [31]. While certain assumptions could probably be justified (almost) formally, others are subject to more controversy [18]. In particular, the most controversial aspect of [23] and [31] is that the attack only uses a single[5] LWE sample and all the short vectors are derived from this single sample. When using a single LWE sample, the problem becomes very close to the *Bounded Distance Decoding* which has been extensively studied. The status of [6] is unclear

[3] Note that this is a matrix sample, so it contains several vector samples.

[4] A modulus switching technique was also suggested in [23] but it is unclear to us how it compares to [31], and [18] suggests that they are different.

[5] Recall that in this paper, an LWE sample is always of the form $(\mathbf{A}, \mathbf{b})$ where $\mathbf{A}$ is a matrix and $\mathbf{b}$ a vector.

since it also computes many (exponentially) short vectors for each LWE sample, but they also use many (exponentially) LWE samples. One could in principle model this as an algorithm operating on a single LWE sample but this sample would then have exponential size and the dual vectors would be extremely special. This makes it unclear whether an argument like that of [18] applies to such a case.

The purpose of this paper is encourage a more rigorous analysis of dual attacks on LWE to better understand under what set of parameters they provably work. We note in that regard that a recently accepted paper at TCC 2023 [32] has focused on similar problems in statistical decoding/"dual attacks" in coding theory. The authors claim in the conclusion that at least part of their results apply to lattice dual attack. We believe that it would indeed be interesting to see what this approach yields for lattices, however we point out that the notion of dual attack that the authors have in mind looks quite different from the one in this paper. In short, and with our notations, the "dual attack" of [32] would be akin to splitting $\mathbf{A}$ horizontally instead of vertically. This splitting would not correspond anymore to a decomposition of $L_q(\mathbf{A})$ as $L_q(\mathbf{A}_{\text{guess}}) + L_q(\mathbf{A}_{\text{dual}})$ and therefore looks incompatible with existing works on dual attacks on LWE. Furthermore, our understanding of [32] is that generating parity check vectors $\mathbf{h}$ corresponds to generating many short dual vectors in $L_q^{\perp}(\mathbf{A})$, independently of the splitting of $\mathbf{A}$. This is completely at odds with lattice dual attacks where we split $\mathbf{A}$ to generate dual vectors in $L_q^{\perp}(\mathbf{A}_{\text{dual}})$ which is much cheaper. Overall it looks like [32] might be a completely different kind of dual attack. See Appendix A for more details.

1.1 Contributions

The main contribution of this paper is to provide a *completely formal, non-asymptotic analysis* of a simplified dual attack. To simplify the presentation, we do not include elements such as the guessing complexity and modulus switching[6] to focus on the most controversial element, namely the fact that the attack only uses a single LWE sample and that all the short vectors are derived from this single sample.

Our approach completely departs from the existing statistics-based attacks and is instead rooted in geometry. This allows us to obtain a relatively short proof and leverage existing results on the geometry of lattices.

One of the most important technical contribution of this paper is to make completely clear (Theorem 5) under what choice of parameters the attack works, without any statistical assumption. As far as we are aware, no other dual attack has been formally analyzed in this way. We believe that this is important since virtually all algorithms in the literature rely on statistical assumptions that clearly cannot hold for all parameter regimes but without a proper analysis, it is impossible to tell when and why they hold.

[6] See Section 7 for more details about modulus switching.

We also provide some new results on random q-ary lattices in a similar spirit to that of Siegel, Rogers and Macbeath. This allows us to obtain some sharper bounds on λ_1 for random q-ary lattices and show that the Gaussian Heuristic is quite tight for such lattices. This heuristic is usually considered valid for "random" lattices and has been extensively tested. Up to our knowledge, the only formal analysis of λ_1 for random q-ary lattice is in [43, Lemma 7.9.2] which only analyzes the expected value and therefore provides a much weaker bound on λ_1. We refer to Section 2.5 for more details.

Finally, we give a quantum version of our algorithm to speed up the computation. The algorithm is inspired by [7] and reuses some technical lemmas to speed up the computation of sums of cosines that appear in the algorithm. Similarly to our classical algorithm, we prove that our quantum algorithm is correct without relying on any heuristics.

1.2 Comparison with [18]'s Contradictory Regime

A recent paper [18] has claimed that virtuall all recent dual attacks rely on an incorrect statistical assumption and that they are, therefore, probably incorrect. They do so by formalizing what they claim is the key statistical assumption of those paper, and show that for the parameter regime of the attacks, it falls into what they call the "contradictory regime", a regime where this assumption can be proven not to hold.

As a byproduct of our analysis, we are able to compare the regime in which our analysis works with the contradictory regime of [18]. Interestingly, the two are essentially complementary with a small gap inbetween. This suggests that our analysis and that of [18] are quite tight and provide an almost complete characterization of when dual attacks work in our simplified setting. However, we nuance this conclusion by noting that the statistical model used in [18] to argue about the contradiction does not seem to match what happens in our algorithm. We refer to Section 6 for more details.

1.3 Organisation of the paper

In Section 2, we introduce the various technical elements that are necessary to analyse the dual attack. In Section 3, we first present a basic dual attack whose purpose is to introduce the reader to the ideas of dual attacks without overwhelming them with technical details. This dual attack is very naive and computes one short vector per LWE sample, in the spirit of [4]. We emphasize that this attack and Theorem 4 are not new but that our analysis is significantly simpler than in previous papers. In Section 4, we introduce our simplified dual attack in the spirit of [31] and formally analyse its correctness without assumption. In Section 5, we give a quantum version of the algorithm from Section 4 and prove its correctness. In Section 6, we compare our regime with that of [18]. Finally, in Section 7, we describe what we believe is the main obstacle to develop a formal analysis of the full algorithm in [31].

2 Preliminaries

We denote vectors and matrices in bold case. We denote by $\mathbf{x}^T$ the transpose of the (column) vector $\mathbf{x}$, which is therefore a row vector. We denote by $\mathbf{I}_n$ the identity matrix of size $n \times n$. For any vector $\mathbf{x} \in \mathbb{R}^n$, we denote by $\|\mathbf{x}\|$ its Euclidean norm. For any integer q, we say that a vector $\mathbf{x} \in \mathbb{Z}_q^n$ is *totally nonzero* if all the coordinates of $\mathbf{x}$ are nonzero. We denote by $\langle \mathbf{x}, \mathbf{y} \rangle$ the scalar product between two vectors $\mathbf{x}$ and $\mathbf{y}$. For any function $f : \mathbb{R}^n \to \mathbb{C}$, we denote by $\widehat{f}$ its Fourier transform over $\mathbb{R}^n$ defined by $\widehat{f}(\mathbf{x}) = \int_{\mathbb{R}^n} f(\mathbf{y}) e^{-2i\pi\langle \mathbf{x}, \mathbf{y} \rangle} \, \mathrm{d}\mathbf{x}$. For any $n \in \mathbb{N}$ and $R > 0$, we denote by $B_n(R)$ (resp. $\overline{B}_n(R)$) the open (resp. closed) ball of radius R in $\mathbb{R}^n$. We also let $B_n^{\mathbb{Z}}(R) = B_n(R) \cap \mathbb{Z}^n$ be the set of integers points in this ball, and similarly for $\overline{B}_n^{\mathbb{Z}}(R)$. For any two distributions P and Q, we denote by $\mathrm{d}_{\mathrm{TV}}(P, Q)$ the statistical distance (or total variation distance) between P and Q.

2.1 Probabilities

For any finite set X, we denote by $\mathcal{U}(X)$ the uniform distribution over X. As usual, if P and Q are two probability distributions over X and Y respectively, we denote by PQ the product distribution over $X \times Y$. Recall that we have the following facts.

Theorem 1 (Hoeffding's inequality). *Let $X_1, \ldots, X_N$ be independent random variables such that $a_i \leqslant X_i \leqslant b_i$. Consider the sum $S_N = X_1 + \cdots + X_N$. Then $\Pr[S_N - \mathbb{E}[S_N] \geqslant t] \leqslant \exp\left(\frac{-2t^2}{\sum_{i=1}^N (b_i - a_i)^2}\right)$ and $\Pr[|S_N - \mathbb{E}[S_N]| \geqslant t] \leqslant 2\exp\left(\frac{-2t^2}{\sum_{i=1}^N (b_i - a_i)^2}\right)$ for all $t > 0$.*

2.2 LWE

Let $n, m, q \in \mathbb{N}$ and let χ_e be a distributions over $\mathbb{Z}_q$, which we call χ_e the *noise distribution*. For every vector $\mathbf{s} \in \mathbb{Z}_q^n$, we denote by $\mathrm{LWE}(m, \mathbf{s}, \chi_e)$ the probability distribution on $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ obtained by sampling a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ uniformly at random, sampling a vector $\mathbf{e} \in \mathbb{Z}_q^m$ according to χ_e^m, and outputting $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b} \coloneqq \mathbf{A}\mathbf{s} + \mathbf{e}$.

We have chosen the "matrix form" for the LWE distribution where each sample is really m LWE samples in the sense of [38]. We have chosen this formalism because it is simpler for dual attacks. The value of m is typically in the order of n and depends on the cryptosystem. In the rest of this paper, we will always refer to a sample of the form $(\mathbf{A}, \mathbf{b})$ as *one sample*.

The search LWE problem is to find $\mathbf{s}$ given oracle access to a sampler for $\mathrm{LWE}(m, \mathbf{s}, \chi_e)$. The decision LWE problem is to decide, given oracle access to either $\mathrm{LWE}(m, \mathbf{s}, \chi_e)$ or $\mathcal{U}(\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m)$, which one it is. In practical scenarios, the attacker may not have access to the sampler but rather only possess a limited number LWE samples. In this case, the search LWE problem asks, given those LWE samples, to recover $\mathbf{s}$ if possible.

The LWE secret $\mathbf{s}$ is usually generated according to a distribution χ_s over $\mathbb{Z}_q^n$. One can therefore, in principle, analyse the success probability of an algorithm for search/decision LWE on a distribution $\mathrm{LWE}(m, \mathbf{s}, \chi_e)$ where $\mathbf{s} \hookleftarrow\!\$\ \chi_s^n$. In this paper, we will not need to make any assumption on the distribution of the secret since our algorithms work for every secret.

2.3 Discrete Gaussian distribution

Let $n \in \mathbb{N}$ and $s > 0$. For any $\mathbf{x} \in \mathbb{R}^n$, we let $\rho_s(\mathbf{x}) := e^{-\pi\|\mathbf{x}\|^2/s^2}$. As usual, we extend to ρ_s to sets by $\rho_s(X) = \sum_{\mathbf{x}\in X} \rho_s(x)$ for any set X. For any lattice $L \subset \mathbb{R}^n$, we denote the *discrete Gaussian distribution* over L by

$$D_{L,s}(\mathbf{x}) = \frac{\rho_s(\mathbf{x})}{\rho_s(L)}$$

for any $\mathbf{x} \in L$. We denote $D_{L,1}$ by D_L for simplicity.

In general, the smaller s is, the harder it is to construct a sampler from $D_{L,s}$. The notion of smoothing parameter [33] captures the idea that sampling for a valuer of s above this threshold is significantly easier than sampling below because the distribution looks more like a continuous Gaussian. There are many algorithms to sample above the smoothing parameter [26,22,11], including a time-space trade-off [2]. Sampling below the smoothing parameter is much more challenging and usually inefficient [3]. At the extreme, sampling for sufficiently small values of s allows one to solve the Shortest Vector problem (SVP) [3] which is known to be NP-hard under randomized reduction [5].

For any $q \in \mathbb{N}$, we denote by $D_{\mathbb{Z}_q^n,s}$ the *modular discrete Gaussian distribution* over $\mathbb{Z}_q^n$ defined by

$$D_{\mathbb{Z}_q^n,s}(\mathbf{x}) = \frac{\rho_s(\mathbf{x} + q\mathbb{Z}^n)}{\rho_s(\mathbb{Z}^n)}$$

for any $\mathbf{x} \in \mathbb{Z}_q^n$. We define the periodic Gaussian function $f_{L,s} : \mathbb{R}^n \to \mathbb{R}$ by

$$f_{L,s}(\mathbf{t}) = \frac{\rho_s(L + \mathbf{t})}{\rho_s(L)}.$$

We have $f_{L/s,1}(\mathbf{t}/s) = f_{L,s}(\mathbf{t})$. In the following, we denote $f_{L,1}$ as f_L.

Lemma 1 ([15, Lemma 2.14]). *For any lattice L, $s > 0$ and $\mathbf{x} \in \mathbb{R}^n$, $f_{L,s}(\mathbf{x}) \geqslant \rho_s(\mathbf{x})$.*

Lemma 2 ([9, Lemma 7], see also [42, Theorem 1.3.4]). *For any lattice $L \subset \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$ and $u \geqslant 1/\sqrt{2\pi}$,*

$$\rho_s((L - \mathbf{x}) \setminus B_n(us\sqrt{n})) \leqslant \left(u\sqrt{2\pi e}e^{-\pi u^2}\right)^n \rho_s(L)$$

Corollary 1 ([42, Corollary 1.3.5]). *For any lattice $L \subset \mathbb{R}^n$, $\mathbf{t} \in \mathbb{R}^n$ and $r \geqslant \delta := s\sqrt{n/2\pi}$,*

$$\rho_s((L - \mathbf{t}) \setminus B_n(r)) \leqslant \rho_s(r - \delta)\rho_s(L)$$

Lemma 3 ([4, Claim 4.1]). *For any lattice L and $s > 0$, we have $\widehat{f_{L,s}} = D_{\widehat{L},1/s}$ which is a probability measure over the dual lattice $\widehat{L}$.*

2.4 Distinguisher

All dual attacks rely on an algorithm to distinguish between the uniform distribution and the modular discrete Gaussian over $\mathbb{Z}_q$. Specifically, we are given N independent samples $X_1, \dots, X_N$ either from the uniform distribution or modular discrete Gaussian and we want to decide from which it was sampled. Several algorithms exist for this task but the most common one is to compute the discrete Fourier transform of the samples (at 1) and to compare against a threshold. The informal argument usually goes as follows. Consider the sum

$$S = \sum_{j=1}^{N} \cos(2\pi X_j) = \Re\left(\sum_{j=1}^{N} e^{2i\pi X_j}\right).$$

It is well-known (see Lemma 5 below) that the expected value of S is 0 for the uniform distribution and we can bound the variance of S easily. For the modular discrete Gaussian, we note that S is a sum of many (N is typically exponential) independent various so by the central limit theorem is essential Gaussian. We can analyze the expected value and variance of each $e^{2i\pi X_j}$ by Lemma 5 which provides us with an estimate of the expected value and variance of S and therefore of the parameters of the "Gaussian approximation" of S. We then use classical tail bounds on Gaussian to bound the probability that S is far away from its expected value. For example, for a modular discrete Gaussian of parameter s, the expected value will be at least $Ne^{-\pi s^2 k^2/q^2}$.

The analysis above is quite informal since it requires the central limit theorem (CLT) to argue that the sum is close to a Gaussian. Unfortunately, a more formal analysis of this step is often lacking in the literature. Many papers refer to [28, Section 4] to identity the advantage of distinguishing although the argument quite informal. [23, Lemma 1] gives a more formal statement of this advantage but without proof. The analysis of [31] is essentially the one above with an explicit mention of the CLT. We find this usage of (a non-quantitative version of) the CLT problematic since the rate of convergence is usually quite slow. In fact a proper analysis shows that the minimum value of N needed for convergence is roughly the same as the value needed for distinguishing so this is no small detail. [21] contains a fully formal analysis that relies on the Berry–Esseen theorem but which requires a bound on the third moment of the variable. More puzzling is the fact that these analysis are both lacking and overkill since a simple application of Hoeffding's inequality immediately provides the same result. Although we will not use this lemma directly, we keep it for reference and also it helps understand the algorithm and proofs that we use in the dual attack.

Lemma 4 (Appendix B.1). *Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two distributions over $\mathbb{R}$. Let $E_i = \mathbb{E}_{X \leftarrow\$ \mathcal{X}_i}[\cos(X)]$ and assume that $E_1 \neq E_2$. There is an algorithm $\mathcal{A}$ that given N independent samples from $\mathcal{X}_j$ for $j \in \{1, 2\}$, correctly finds j in time $O(N)$ with probability at least $1 - 2\exp(-N|E_1 - E_2|^2)$ on the choice of the samples. Formally, for all $j \in \{1, 2\}$,*

$$\Pr_{X_1,\dots,X_N \leftarrow\$ \mathcal{X}_j}[\mathcal{A}(X_1, \dots, X_N) = j] \geqslant 1 - 2\exp(-\tfrac{N}{8}|E_1 - E_2|^2).$$

The above lemma requires to compute the average $\mathbb{E}_{X \leftarrow\$ \mathcal{X}}[\cos(X)]$ for the distributions of interest. For distributions over $\mathbb{Z}_q$, this is essentially the Fourier transform of the distribution at 1. We refer to [14] for a well-written analysis of such transforms on various distributions such as the (modular) discrete Gaussian, the rounded discrete Gaussian, the central binomial and the bounded uniform distribution. In this paper, we only require the following result which is well-known.

Lemma 5 ([14]). *Let $q \geqslant 1$ be an integer and $s > 0$. Then*

$$\begin{aligned} \mathbb{E}_{X \leftarrow\$ \mathcal{U}(\mathbb{Z}_q)}[\exp(2ik\pi X/q)] &= 0 && \text{for all } k \neq 0, \\ e^{-\pi s^2 k^2/q^2} \leqslant \mathbb{E}_{X \leftarrow\$ D_{\mathbb{Z}_q,s}}[\exp(2ik\pi X/q)] &\leqslant 2e^{-\pi s^2 k^2/q^2} && \text{for } k \in \{ -\lfloor q/2 \rfloor, \ldots, \lfloor q/2 \rfloor \}. \end{aligned}$$

2.5 Lattices

We denote by $\widehat{L}$ the dual of a lattice $L \subset \mathbb{R}^n$ defined by

$$\widehat{L} = \{ \mathbf{x} \in \operatorname{span}(L) : \forall \mathbf{y} \in L, \langle \mathbf{y}, \mathbf{x} \rangle \in \mathbb{Z} \}.$$

We denote by $L^* = L \setminus \{ \mathbf{0} \}$ the set of nonzero vectors of a lattice L. We denote by $\lambda_1(L)$ the length a shortest nonzero vector in L.

Let $n \in \mathbb{N}$, $1 \leqslant k \leqslant n$ and q be a prime power. We say that a lattice L is a n-dimensional *q-ary lattice* if $q\mathbb{Z}^n \subseteq L \subseteq \mathbb{Z}^n$. Given a matrix $\mathbf{A} \in \mathbb{Z}^{n \times k}$, we consider the following n-dimensional q-ary lattices:

$$\begin{aligned} L_q(\mathbf{A}) &= \{ \mathbf{x} \in \mathbb{Z}^n : \exists \mathbf{s} \in \mathbb{Z}^k, \mathbf{A}\mathbf{s} = \mathbf{x} \bmod q \}, \\ L_q^{\perp}(\mathbf{A}) &= \{ \mathbf{x} \in \mathbb{Z}^n : \mathbf{A}^T \mathbf{x} = \mathbf{0} \bmod q \}. \end{aligned}$$

We refer the reader to [20], [43, Section 2.5.1] or [34] for more details on those constructions. Note that, equivalently, we can write $L_q(\mathbf{A}) = \mathbf{A}\mathbb{Z}_q^k + q\mathbb{Z}^n$. It is well-know that for any q-ary lattice L, there exists $\mathbf{A}$ and $\mathbf{B}$ such that $L = L_q(\mathbf{A}) = L_q^{\perp}(\mathbf{B})$, and that $\widehat{L_q^{\perp}(\mathbf{A})} = \frac{1}{q} L_q(\mathbf{A})$. Furthermore $\det(L_q(\mathbf{A})) = q^{n - \operatorname{rk} \mathbf{A}} \geqslant q^{n-k}$ and therefore $\det(L_q^{\perp}(\mathbf{A})) = q^{\operatorname{rk} \mathbf{A}} \leqslant q^k$. Finally, since $\mathbb{Z}_q$ is a field, a random matrix $\mathbf{A}$ has full rank (equal to k) with high probability:

$$\Pr_{\mathbf{A} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{n \times k})}[\operatorname{rk}(\mathbf{A}) = k] = \prod_{i=0}^{k-1} (1 - q^{i-n}) \geqslant 1 - kq^{k-1-n}. \tag{1}$$

Given k and n, we will consider the distributions $\mathcal{L}_{n,k,q}$ and $\mathcal{L}_{n,k,q}^{\perp}$ of q-ary lattices defined over the set of integer lattices by

$$\begin{aligned} \mathcal{L}_{n,k,q}(L) &= \Pr_{\mathbf{A} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{n \times k})}[L = L_q(\mathbf{A})], \\ \mathcal{L}_{n,k,q}^{\perp}(L) &= \Pr_{\mathbf{A} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{n \times (n-k)})}\left[L = L_q^{\perp}(\mathbf{A})\right]. \end{aligned}$$

In other words, the distribution is obtained by taking a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$ with uniform and i.i.d entries, and looking at the q-ary lattice generated by $\mathbf{A}$; and similarly for the orthogonal version. Note that contrary to the Loeliger ensemble $\mathbb{L}_{n,k,q,1}$, we do not have the rescaling factor $q^{1-k/n}$, see e.g. [43, Definition 7.9.2]. It will be more convenient to use $\mathcal{L}^{\perp}_{n,k,q}$ for proofs, but we often want to apply them for $\mathcal{L}_{n,k,q}$. Whenever neither k nor $n-k$ are too small, those two distributions are very close. The following lemma was inspired by [17, Lemma 2] which does not contain any proof.

Lemma 6 (Appendix D.1). $\mathrm{d}_{\mathrm{TV}}(\mathcal{L}^{\perp}_{n,k,q}, \mathcal{L}_{n,k,q}) \leqslant \mathsf{poly}(n,k) q^{-\min(k,n-k)}$.

Those distribution satisfy good uniformity properties when q goes to infinity. In particular, the following theorem shows that we can computed statistical properties of lattices sampled according to $\mathcal{L}^{\perp}_{n,k,q}$. The first part of this theorem is close to [29, Theorem 1]. This result is in some sense the q-ary version of the result by Siegel on random (real) lattices and its generalization by Rogers and Macbeath [41,39,30].

Theorem 2 (Appendix D.2). *Let $1 \leqslant p \leqslant n$ and $f : (\mathbb{Z}_q^n)^p \to \mathbb{R}$, then*

$$\mathbb{E}_{L \leftarrow\$ \mathcal{L}^{\perp}_{n,k,q}}\left[\sum_{\mathbf{x}_1,\ldots,\mathbf{x}_p \in L} f(\mathbf{x}_1,\ldots,\mathbf{x}_p)\right] = \sum_{\mathbf{x}_1,\ldots,\mathbf{x}_p \in \mathbb{Z}^n} q^{(k-n)r(\mathbf{x}_1,\ldots,\mathbf{x}_p)} f(\mathbf{x}_1,\ldots,\mathbf{x}_p)$$

where $r(\mathbf{x_1},\ldots,\mathbf{x_p}) := \mathrm{rk}_{\mathbb{Z}_q^n}(\mathbf{x}_1,\ldots,\mathbf{x}_p)$ is the rank of the $\mathbf{x}_i \bmod q$ over $\mathbb{Z}_q^n$.

We can apply this theorem to bound the expected number of lattice points in a ball, and therefore obtain bounds on λ_1.

Theorem 3 (Appendix D.3). *For any $0 < r \leqslant q$,*

$$\mathbb{E}_{L \leftarrow\$ \mathcal{L}^{\perp}_{n,k,q}}[|L^* \cap B_n(r)|] = q^{k-n}\left(|B_n^{\mathbb{Z}}(r)| - 1\right),$$
$$\mathbb{V}_{L \leftarrow\$ \mathcal{L}^{\perp}_{n,k,q}}[|L^* \cap B_n(r)|] \leqslant q^{k-n}(q-1)(|B_n^{\mathbb{Z}}(r)| - 1).$$

In particular, if $|B_n^{\mathbb{Z}}(r)| \leqslant q^{n-k}$, then

$$\Pr_{L \leftarrow\$ \mathcal{L}^{\perp}_{n,k,q}}[\lambda_1(L) \leqslant r] \leqslant \left(q^{1+k-n}|B_n^{\mathbb{Z}}(r)|\right)^2.$$

Recall that the *Gaussian heuristic* says that for a "random" lattice L, $\lambda_1(L)$ is approximately

$$\mathrm{GH}(L) := \left(\frac{\mathrm{vol}(B_n)}{\det(L)}\right)^{-1/n} = \frac{\det(L)^{1/n}\Gamma(1+\frac{n}{2})^{1/n}}{\sqrt{\pi}} \approx \det(L)^{1/n}\sqrt{\frac{n}{2\pi e}}.$$

This heuristic is usually considered valid for "random" lattices and has been extensively tested. Up to our knowledge, the only formal analysis of λ_1 for random q-ary lattice is in [43, Lemma 7.9.2] which only analyzes the expected value and not the variance. The following corollary shows that this heuristic is indeed very sharp for random q-ary lattices.

Corollary 2 (Informal, Appendix D.4). *Under the assumption that* $|B_n^{\mathbb{Z}}(r)| \approx \mathrm{vol}(B_n(r))$*, which holds when* $r \gg \sqrt{n}$*, we have for any* $\alpha \leqslant 1$ *that*

$$\Pr_{L \leftarrow \$ \mathcal{L}_{n,k,q}}[\lambda_1(L) \leqslant \alpha GH(L)] \lessapprox (q\alpha^n)^2 .$$

Lemma 7 (The Pointwise Approximation Lemma [4, Lemma 1.3], modified). *Let* L *be an* n*-dimensional lattice, and let* h *be a function from* $\mathbb{R}^n$ *to* $\mathbb{R}$ *that is periodic over* L *and whose Fourier series* $\hat{h}$ *is a probability measure over the dual lattice* $\widehat{L}$*. Let* N *be an integer,* $\delta > 0$ *and* $X \subseteq \mathbb{R}^n$ *a finite set. Let* $W = (\mathbf{w}_1, \cdots, \mathbf{w}_N)$ *be a list of vectors in the dual lattice chosen randomly and independently from the distribution* $\hat{h}$*. Then with probability at least* $1 - |X|2^{-\Omega(N\delta^2)}$*,*

$$h_W(\mathbf{x}) \coloneqq \frac{1}{N} \sum_{i=1}^{N} \cos(2\pi \langle \mathbf{w}_i, \mathbf{x} \rangle) \tag{2}$$

satisfies that $|h_W(\mathbf{x}) - h(\mathbf{x})| \leqslant \delta$ *for all* $\mathbf{x} \in L + X$*.*

Proof. The proof is exactly the one in [4] with the following modifications. Let $\delta > 0$. For any $x \in \mathbb{R}^n$, the Chernoff-Hoeffding bounds guarantees that the mean of N samples is not within a window of δ of the correct expectation with probability at most $2^{-\Omega(N\delta^2)}$. Since f is periodic over the lattice L, it suffices to check that the inequality that we want holds for all $\mathbf{x} \in X$, and there are $|X|$ such points. Hence, by a union bound, the probability that the approximation is within a window δ of the correct expectation for all $\mathbf{x} \in X$ simultaneously is at least $1 - |X|2^{-\Omega(N\delta^2)}$. □

2.6 Short vector sampling

For the purpose of this paper, we will only need to know that there is a way to sample relatively short vectors (SV) in a lattice and we will treat such an algorithm as a black box. Since such an algorithm would typically be parametrized (see below), we introduce an integer parameter β to capture this fact.

Black box 1. *For any integers* $n \leqslant m$*,* β *and prime power* q*, there exists a deterministic algorithm* $\mathcal{B}$ *and two functions* T_{SV} *and* ℓ_{SV} *such that when* $\mathcal{B}$ *is given* $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$*, it returns a nonzero vector in* $L_q^{\perp}(\mathbf{A})$ *in time* $T_{\mathrm{SV}}(m, \beta, q^n)$ *and* $\mathbb{E}_{\mathbf{A} \leftarrow \$ \mathbb{Z}_q^{m \times n}}\left[\|\mathcal{B}(\mathbf{A})\|^2\right] \leqslant \ell_{\mathrm{SV}}(m, \beta, q^n)^2$*.*

One way to implement this black box is to use lattice reduction algorithms such as BKZ: they provide a very flexible way to take a basis of lattice and compute relatively short vectors in this lattice. Since the literature on this topic is quite extensive and there are many cost models associated to that task, we refer the reader to e.g. [23] for more details. For simplicity, we assume that the algorithm is deterministic but we could make it probabilistic by adding random coins to the input of the algorithm and take those into account in the expected value. In the case of BKZ, the parameter β is the block size.

3 Basic dual attack

In this section, we present a basic dual attack whose purpose is to introduce the reader to the ideas of dual attacks without overwhelming them with technical details. This dual attack is very naive and computes one short vector per LWE sample, in the spirit of [4]. We emphasize that this attack and Theorem 4 are not new but that our analysis is significantly simpler than in previous papers.

Fix $\mathbf{s} \in \mathbb{Z}_q^n$ an unknown secret and $(\mathbf{A}, \mathbf{b})$ a LWE sample. Recall that $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ for some unknown $\mathbf{e} \in \mathbb{Z}_q^m$. In its simplest form, a dual attack splits the secret $\mathbf{s}$ into two parts $\mathbf{s}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and $\mathbf{s}_{\text{dual}} \in \mathbb{Z}_q^{n_{\text{dual}}}$ where $n = n_{\text{guess}} + n_{\text{dual}}$. The matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is correspondingly split into two parts:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\text{guess}} & \mathbf{A}_{\text{dual}} \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} \mathbf{s}_{\text{guess}} \\ \mathbf{s}_{\text{dual}} \end{bmatrix}. \tag{3}$$

Therefore,

$$\mathbf{b} = \mathbf{A}_{\text{guess}}\mathbf{s}_{\text{guess}} + \mathbf{A}_{\text{dual}}\mathbf{s}_{\text{dual}} + \mathbf{e}.$$

The algorithm now makes a guess $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ on the value of $\mathbf{s}_{\text{guess}}$ and tries to check whether this guess is correct. Consider the lattice

$$L_q^{\perp}(\mathbf{A}_{\text{dual}}) = \{\, \mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T \mathbf{A}_{\text{dual}} = \mathbf{0} \bmod q \,\}. \tag{4}$$

By the inequalities of Section 2.5, we have that

$$\det(L_q^{\perp}(\mathbf{A}_{\text{dual}})) \leqslant q^{n_{\text{dual}}}. \tag{5}$$

Check that for any $\mathbf{x} \in L_q^{\perp}(\mathbf{A}_{\text{dual}})$,

$$\mathbf{x}^T\mathbf{b} = \mathbf{x}^T\mathbf{A}_{\text{guess}}\mathbf{s}_{\text{guess}} + \mathbf{x}^T\mathbf{A}_{\text{dual}}\mathbf{s}_{\text{dual}} + \mathbf{x}^T\mathbf{e} = \mathbf{x}^T\mathbf{A}_{\text{guess}}\mathbf{s}_{\text{guess}} + \mathbf{x}^T\mathbf{e} \pmod q.$$

Therefore,

$$\mathbf{x}^T(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}}) = \mathbf{x}^T\mathbf{A}_{\text{guess}}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}) + \mathbf{x}^T\mathbf{e} \pmod q.$$

The main observation is now that:

- if the guess is correct ($\tilde{\mathbf{s}}_{\text{guess}} = \mathbf{s}_{\text{guess}}$) then $\mathbf{x}^T(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}}) = \mathbf{x}^T\mathbf{e} \pmod q$ follows roughly a modular Gaussian distribution,
- if the guess is incorrect ($\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}$) then it follows a uniform distribution because $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{A}$ was chosen uniformly at random.

A crucial ingredient in the reasoning above is the length of $\mathbf{x}$. Indeed, the dot product $\mathbf{x}^T\mathbf{e}$ will follow a modular Gaussian whose deviation is proportional to $\|\mathbf{x}\|$. This is where the BKZ lattice reduction algorithm comes in: from a basis of $L_q^{\perp}(\mathbf{A}_{\text{dual}})$, we compute a short vector $\mathbf{x}$ using Black box 1.

The algorithm for this dual attack is described in Algorithm 1. In this attack, we compute one dual vector for each LWE sample. While this kind of attack is already known to be correct, we reprove it for several reasons. First, we are not satisfied with the informal treatement of the proof in the literature. Second, our proof does not use any assumption whereas most papers in the literature use the Central Limit Theorem or approximate sums of Gaussian as a Gaussian at some point (see Section 2.4 for more comments). Figure 1 gives a high level view of the variable involved and their dependencies.

Theorem 4 (Appendix C). *Let n, m, β be integers, q be a prime power, $n_{\text{guess}} + n_{\text{dual}} = n$, $\mathbf{s} \in \mathbb{Z}_q^n$, $\sigma_e > 0$ and $N \in \mathbb{N}$. Let $0 < \delta < \varepsilon := \exp\left(-\pi\sigma_e^2 \ell_{\text{SV}}(m, \beta, q^{n_{\text{dual}}})^2/q^2\right)$, where ℓ_{SV} comes from Black box 1. Let $(\mathbf{A}^{(1)}, \mathbf{b}^{(1)}), \ldots, (\mathbf{A}^{(N)}, \mathbf{b}^{(N)})$ be samples from $\mathrm{LWE}(m, \mathbf{s}, D_{\mathbb{Z}_q,\sigma_e})$, then Algorithm 1 on $(m, n_{\text{guess}}, n_{\text{dual}}, q, \delta, N, (\mathbf{A}^{(i)}, \mathbf{b}^{(i)})_i)$ runs in time $\mathsf{poly}(m, n) \cdot (N \cdot T_{\text{SV}}(m, \beta, q^n) + q^{n_{\text{guess}}})$ and returns $\mathbf{s}_{\text{guess}}$ with probability at least*

$$1 - \exp\left(-\frac{N(\varepsilon - \delta)^2}{2}\right) - (q^{n_{\text{guess}}} - 1)\exp\left(-\frac{N\delta^2}{2}\right)$$

over the choice of the $(\mathbf{A}^{(i)}, \mathbf{b}^{(i)})$.

Remark 1. As expected, we recover the well-known fact that for the attack to succeed with constant probability, we can take $\delta = \varepsilon/2$ and then we need at least $N = \frac{8n_{\text{guess}}\log(q) + \Omega(1)}{\varepsilon^2}$ samples. Furthermore, a careful look at the proof shows that Black box 1 can be weakened even further to only require an inequality on the moment-generating function of $\|\mathcal{B}(\mathbf{A})\|^2$.

Algorithm 1: Basic dual attack

Input: m, $n = n_{\text{guess}} + n_{\text{dual}}$ (see (3)), q prime power, $\delta > 0$ and $N \in \mathbb{N}$.
Input: list of N LWE samples $(\mathbf{A}^{(1)}, \mathbf{b}^{(1)}), \ldots, (\mathbf{A}^{(N)}, \mathbf{b}^{(N)})$.
Output: (Guess of) the first n_{guess} coordinates of the secret or $\perp$.

for j from 1 to N **do**
- Compute a basis of $L_q^{\perp}(\mathbf{A}^{(j)})$
- Compute a short vector $\mathbf{x}_j \in L_q^{\perp}(\mathbf{A}^{(j)})$ using Black box 1

for $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ **do**
- Compute the list $y_1, \ldots, y_N$ where $y_j = \mathbf{x}_j^T(\mathbf{b}^{(j)} - \mathbf{A}_{\text{guess}}^{(j)}\tilde{\mathbf{s}}_{\text{guess}})$
- $S \leftarrow \sum_{j=1}^{N} \cos(2\pi y_j/q)$
- **if** $S \geqslant N\delta$ **then**
  - **return** $\tilde{\mathbf{s}}_{\text{guess}}$

return $\perp$

---

4 Modern dual attack

The main limitation of the basic dual attack is the requirement to compute one short vector for each LWE sample. Looking at Figure 1, this is necessary to ensure the statistical independence of the variables that go into the distinguisher.

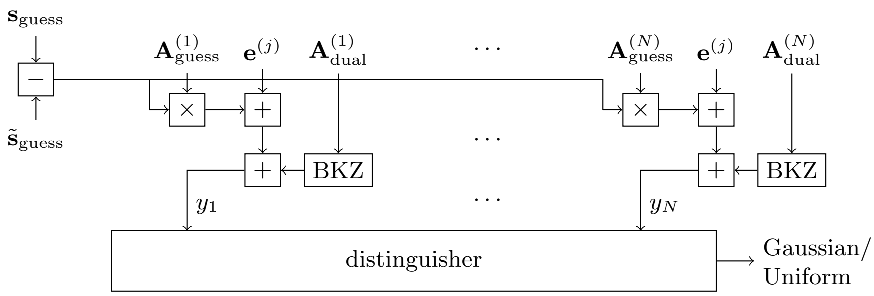

Fig. 1. Conceptual representation of the variables involved in Algorithm 1 and their dependencies.

Furthermore, the attack requires an exponential number of LWE samples, something which is not always possible.

As explained in the introduction, a series of work have progressively introduced the idea of generating *all short vectors* from a limited number, or even a single, LWE sample $(\mathbf{A}, \mathbf{b})$. This is the case in [6,23,21], and [31] where only a single LWE sample is used, and it dramatically reduces the complexity of the attack. Unfortunately, the statistical analysis of these attacks has been lacking in the literature: [6], [21][7] and [23] offer no real proof of correctness to speak of. Only [31] tries to provide a complete proof of correctness, which is very detailed, but has to rely on statistical assumptions. Those assumptions have been called into question [18], and more importantly are extremely difficult to verify. Stepping back, we believe that the reason for this situation is that they try to analyse their attack using a similar proof strategy to that of our basic dual attack (Section 3). The problem stems from the fact that the basic dual attack requires the independence of many variables to work. Since those variables become dependent in their attack, these papers inevitably have to assume that non-independent quantities are "independent enough".

In this section, we start completely from scratch: we design and analyze without any assumption a modern dual attack. Our proof scheme is completely different from the basic one and shows that those attacks do work. The main outcome of this proof is that we can finally understand the constraints on the various parameters that are necessary for the attack to work.

4.1 Intuition

Fix $\mathbf{s} \in \mathbb{Z}_q^n$ an unknown secret and $(\mathbf{A}, \mathbf{b})$ an LWE sample. Recall that $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e}$ for some unknown $\mathbf{e} \in \mathbb{Z}_q^m$. As in the basic dual attack, we split the secret $\mathbf{s}$ into

[7] Part of [21] presents an attack that is similar to our basic attack and analyzes this attack formally. This paragraph only applies to the part that relies on sieving to produce many short vectors from a single LWE sample.

two parts $\mathbf{s}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and $\mathbf{s}_{\text{dual}} \in \mathbb{Z}_q^{n_{\text{dual}}}$ where $n = n_{\text{guess}} + n_{\text{dual}}$. The matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is correspondingly split into two parts:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\text{guess}} \ \mathbf{A}_{\text{dual}} \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} \mathbf{s}_{\text{guess}} \\ \mathbf{s}_{\text{dual}} \end{bmatrix}. \tag{6}$$

The algorithm now makes a guess $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ on the value of $\mathbf{s}_{\text{guess}}$ and tries to check whether this guess is correct. Check that

$$\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}} = \mathbf{A}_{\text{guess}} \cdot (\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}) + \mathbf{A}_{\text{dual}} \cdot \mathbf{s}_{\text{dual}} + \mathbf{e}. \tag{7}$$

Consider the lattice

$$L_q^{\perp}(\mathbf{A}_{\text{dual}}) = \{ \mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T \cdot \mathbf{A}_{\text{dual}} = \mathbf{0} \bmod q \}. \tag{8}$$

Fix $N \in \mathbb{N}$ and $s > 0$, and let $W = (\mathbf{w}_1, \ldots, \mathbf{w}_N) \in L_q^{\perp}(\mathbf{A}_{\text{dual}})^N$ be sampled according to $D_{L_q^{\perp}(\mathbf{A}_{\text{dual}}),s}$. For any $\mathbf{x} \in \mathbb{R}^m$, define

$$g_W(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^{N} \cos(2\pi \langle \mathbf{x}, \mathbf{w}_j \rangle / q) \tag{9}$$

for all $\mathbf{x} \in \mathbb{R}^m$. We will evaluate g_W at $\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}$ for all $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and keep the highest value. We now explain the intuition for this. Let $L = L_q(\mathbf{A}_{\text{dual}})$ to simplify notations. Recall that in Section 2.3, we have defined the standard periodic Gaussian function

$$f_{L,1/s}(\mathbf{x}) = \frac{\rho_{1/s}(\mathbf{x} + L)}{\rho_{1/s}(L)}$$

for any $\mathbf{x} \in \mathbb{R}^m$ and $s > 0$. The important fact is that for large N, with high probability on the choice of the $\mathbf{w}_j$, the two functions above are close everywhere *for integer vectors* (Lemma 8). This fact essentially comes from [4].

Recall that we evaluate g_W at points $\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}$ which are all integers. Since g_W is very close to f_L over the integers, we have to analyse the behaviour of $f_{L,1/s}$. For this, we rely on standard Gaussian tailbounds (Lemma 9) to get that for any $s > 0$ and $\mathbf{x} \in \mathbb{R}^m$, we essentially have

$$f_{L,1/s}(\mathbf{x}) \approx \rho_{1/s}(\operatorname{dist}(\mathbf{x}, L)). \tag{10}$$

In other words, $f_{L,1/s}$ measure the *distance to the lattice* L.

We are now ready to see what makes the attack work. The intuition is that *for most choices* of $\mathbf{A}$ and $\mathbf{e}$, for all $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}} \setminus \{ \mathbf{s}_{\text{guess}} \}$,

$$\operatorname{dist}(\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \mathbf{s}_{\text{guess}}, L) \ll \operatorname{dist}(\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}, L) \tag{11}$$

and therefore

$$f_{L,1/s}(\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \mathbf{s}_{\text{guess}}) > f_{L,1/s}(\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}})$$

and the same will be true for g_W, which means that the algorithm will correctly output $\mathbf{s}_{\text{guess}}$. This is the main idea of our analysis but making it formal requires some care. In order to make that statement precise, we need a quantitative version of (11). The first step (Lemma 10) is to show that that essentially

$$\text{if } 2\|\mathbf{e}\| \leqslant \lambda_1(L_q(\mathbf{A})) \text{ then } f_{L,\frac{1}{s}}(\mathbf{e}) > f_{L,\frac{1}{s}}(\mathbf{e}+\mathbf{x}) \text{ for all } \mathbf{x} \in L_q(\mathbf{A}_{\text{guess}})\backslash L. \quad (12)$$

This requires some explanations. Going back to (11), we have that

$$\begin{aligned}
\operatorname{dist}(\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \mathbf{s}_{\text{guess}}, L) &= \operatorname{dist}(\mathbf{e} + \mathbf{A}_{\text{dual}} \cdot \mathbf{s}_{\text{dual}}, L) \\
&= \operatorname{dist}(\mathbf{e}, L) && \text{since } \mathbf{A}_{\text{dual}} \cdot \mathbf{s}_{\text{dual}} \in L \\
&= \|\mathbf{e}\| && \text{if } \|\mathbf{e}\| < \lambda_1(L)/2.
\end{aligned}$$

On the other hand, if $\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}$ then

$$\begin{aligned}
&\operatorname{dist}(\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}, L) \\
&= \operatorname{dist}(\mathbf{e} + \mathbf{A}_{\text{dual}} \cdot \mathbf{s}_{\text{dual}} + \mathbf{A}_{\text{guess}}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}), L) \\
&= \operatorname{dist}(\mathbf{e} + \mathbf{A}_{\text{guess}}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}), L) && \text{since } \mathbf{A}_{\text{dual}} \cdot \mathbf{s}_{\text{dual}} \in L \\
&= \operatorname{dist}(\mathbf{e} + \mathbf{x}, L)
\end{aligned}$$

where

$$\mathbf{x} = \mathbf{A}_{\text{guess}}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}) \in L_q(\mathbf{A}_{\text{guess}})$$

Assume for now that $\mathbf{x} \in L_q(\mathbf{A}_{\text{guess}}) \setminus L$ which we will see below is not always true but holds with probability exponentially close to 1 over the choice of $\mathbf{A}$. Then

$$\begin{aligned}
\operatorname{dist}(\mathbf{b} - \mathbf{A}_{\text{guess}} \cdot \tilde{\mathbf{s}}_{\text{guess}}, L) &= \operatorname{dist}(\mathbf{e} + \mathbf{x}, L) \\
&= \min\{\, \|\mathbf{e} + \mathbf{x} + \mathbf{z}\| : \mathbf{z} \in L \,\} \\
&\geqslant \min\{\, \|\mathbf{e} + \mathbf{y} + \mathbf{z}\| : \mathbf{z} \in L, \mathbf{y} \in L_q(\mathbf{A}_{\text{guess}}) \setminus L \,\} \\
&\geqslant \min\{\, \|\mathbf{y} + \mathbf{z}\| : \mathbf{z} \in L, \mathbf{y} \in L_q(\mathbf{A}_{\text{guess}}) \setminus L \,\} - \|\mathbf{e}\| \\
&\geqslant \lambda_1(L + L_q(\mathbf{A}_{\text{guess}})) - \|\mathbf{e}\|.
\end{aligned}$$

The last step holds because $\mathbf{y} + \mathbf{z} \neq \mathbf{0}$ for all $\mathbf{z} \in L$ and $\mathbf{y} \in L_q(\mathbf{A}_{\text{guess}}) \setminus L$. This is where our assumption that $\mathbf{x} \in L_q(\mathbf{A}_{\text{guess}}) \setminus L$ is crucial. The condition in (11) now becomes

$$\|\mathbf{e}\| \leqslant \lambda_1(L + L_q(\mathbf{A}_{\text{guess}})) - \|\mathbf{e}\|$$

and this gives us (12) because $L + L_q(\mathbf{A}_{\text{guess}}) = L_q(\mathbf{A}_{\text{dual}}) + L_q(\mathbf{A}_{\text{guess}}) = L_q(\mathbf{A})$.

Now that we have (12), the second step is to apply it to $\mathbf{A}$. Recall that we made a crucial assumption above: it only applies to $\mathbf{e} + \mathbf{x}$ for $\mathbf{x} \in L_q(\mathbf{A}_{\text{guess}}) \setminus L$ where $\mathbf{x} = \mathbf{A}_{\text{guess}}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}})$ and $\mathbf{s}_{\text{guess}} \neq \tilde{\mathbf{s}}_{\text{guess}}$. This condition is equivalent to $\mathbf{x} \notin \mathbf{A}_{\text{dual}}\mathbb{Z}_q^{n_{\text{dual}}} + q\mathbb{Z}^m$ since $L = L_q(\mathbf{A}_{\text{dual}})$. A sufficient condition for this to hold is that $\mathbf{A}$ has full rank over $\mathbb{Z}_q$ which happens with probability exponentially close to 1 over the choice of $\mathbf{A}$. This allows us to conclude (Theorem 5) that

Algorithm 2, which essentially performs the steps highlighted above, works for almost all $\mathbf{A}$ and $\mathbf{e}$ that satisfy roughly $2\|\mathbf{e}\| \leqslant \lambda_1(\mathbf{A})$. At this point, one can make two interesting observations:

- It tells us that if $2\|\mathbf{e}\| \leqslant \lambda_1(L_q(\mathbf{A}))$ then we can distinguish $\mathbf{e}$ from any $\mathbf{e}+\mathbf{x}$ by using $f_{L,1/s}$. This makes intuitive sense since this condition guarantees that $\mathbf{e}$ is the closest vector to $\mathbf{0}$ in $L_q(\mathbf{A})$ which is a *necessary condition* for the algorithm to work unconditionally.
- Even though we take short vectors in the dual lattice $L_q(\mathbf{A}_{\text{dual}})$, it looks like only the length of the shortest vectors in $\mathbf{A}$ matters for the analysis! This is just a result of the simplifications that we have made above to give the intuition. The length of the dual vectors does play a role in Lemma 10 and the subsequent lemmas.

4.2 Formal Analysis

This section gives a formal analysis of the intuitions from the previous section. We will reuse the notation defined there.

Lemma 8. *Let $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$, $s, \delta > 0$ and $N \in \mathbb{N}$. With probability at least $1 - q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of $W = (\mathbf{w}_1, \dots, \mathbf{w}_N)$ from $D^N_{L_q^\perp(\mathbf{B}),s}$, we have $|g_W(\mathbf{x}) - f_{L_q(\mathbf{B}),1/s}(\mathbf{x})| \leqslant \delta$ for all $\mathbf{x} \in \mathbb{Z}^m$, where g_W is defined in* (9) *and $f_{L_q(\mathbf{B})}$ is defined in Section 2.3.*

Proof. Let $L = L_q(\mathbf{B})$ and for any j, let $\mathbf{w}'_j = \frac{1}{q}\mathbf{w}_j$ and $W' = (\mathbf{w}'_j)_{j'}$. Since $\widehat{L} = \frac{1}{q}L_q^\perp(\mathbf{B})$, we indeed have that W' is sampled from from $D^N_{\widehat{L},s}$ which is a probability distribution over $\widehat{L}$. Let $h = f_{L,1/s}$ which is L-periodic, then $\widehat{h} = D_{\widehat{L},s}$ by Lemma 3. For any $\mathbf{x} \in \mathbb{R}^m$,

$$g_W(\mathbf{x}) = \frac{1}{N}\sum_{j=1}^{N}\cos(2\pi\langle\mathbf{x},\mathbf{w}_j\rangle/q) = \frac{1}{N}\sum_{j=1}^{N}\cos(2\pi\langle\mathbf{x},\mathbf{w}'_j\rangle) = h_{W'}(\mathbf{x}).$$

Apply Lemma 7 to h with $X = \{0,\dots,q-1\}^m$ to get that with probability at least $1 - |X|2^{-\Omega(N\delta^2)}$ over the choice of W', we have $|h(\mathbf{x}) - h_{W'}(\mathbf{x})| \leqslant \delta$ for all $\mathbf{x} \in L + X$. But $L = L_q(\mathbf{B})$ is a q-ary lattice, *i.e.* $q\mathbb{Z}^m \subset L$ so $L + X \supset q\mathbb{Z}^m + \{0,\dots,q-1\}^m = \mathbb{Z}^m$ which concludes the proof. ☐

Lemma 9. *Let $L \subset \mathbb{R}^m$ and $s > 0$, then for any $\mathbf{x} \in \mathbb{R}^m$:*

- $f_{L,1/s}(\mathbf{x}) \geqslant \rho_{1/s}(\operatorname{dist}(\mathbf{x}, L))$,
- *if* $\operatorname{dist}(\mathbf{x}, L) \geqslant \tau := \frac{1}{s}\sqrt{m/2\pi}$ *then* $f_{L,1/s}(\mathbf{x}) \leqslant \rho_{1/s}(\operatorname{dist}(\mathbf{x}, L) - \tau)$.

Proof. The first fact is a direct consequence of Lemma 1. Indeed, write $\mathbf{x} = \mathbf{z}+\mathbf{t}$ where $\mathbf{z} \in L$ and $\mathbf{t} \in \mathbb{R}^m$ are such that $\operatorname{dist}(\mathbf{x}, L) = \|\mathbf{t}\|$. Since $f_{L,1/s}$ is L-periodic and $\mathbf{z} \in L$,

$$f_{L,1/s}(\mathbf{x}) = f_{L,1/s}(\mathbf{x}-\mathbf{z}) = f_{L,1/s}(\mathbf{t}) \geqslant \rho_{1/s}(\mathbf{t}) = \rho_{1/s}(\|\mathbf{t}\|).$$

For the second fact, let $\ell = \operatorname{dist}(\mathbf{x}, L)$ and observe that by definition $(L - \mathbf{x}) \setminus B_m(\ell) = L - \mathbf{x}$. By assumption, $\ell \geqslant \tau := \frac{1}{s}\sqrt{m/2\pi}$, so we can apply Corollary 1 to get that

$$\rho_{1/s}((L - \mathbf{x}) \setminus B_m(\ell)) \leqslant \rho_{1/s}(\ell - \tau)\rho_{1/s}(L)$$

and therefore

$$f_{L,1/s}(\mathbf{x}) = \frac{\rho_{1/s}(L - \mathbf{x})}{\rho_{1/s}(L)} = \frac{\rho_{1/s}((L - \mathbf{x}) \setminus B_m(\ell))}{\rho_{1/s}(L)} \leqslant \rho_{1/s}(\ell - \tau).$$

□

Lemma 10. *Let $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$, $L \subset \mathbb{Z}^m$ a lattice, $\mathbf{e} \in \mathbb{Z}^m$, $s, \delta > 0$ and $N \in \mathbb{N}$. Let $\tau = \frac{1}{s}\sqrt{m/2\pi}$ and $\eta > 0$ and assume that $\lambda_1(L + L_q(\mathbf{B})) \geqslant \tau + \|\mathbf{e}\|$ and*

$$\rho_{1/s}(\mathbf{e}) - \rho_{1/s}(\lambda_1(L + L_q(\mathbf{B})) - \|\mathbf{e}\| - \tau) > 2\delta + \eta.$$

Then, with probability at least $1-q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of $W = (\mathbf{w}_1, \ldots, \mathbf{w}_N)$ from $D^N_{L_q^\perp(\mathbf{B}),s}$, we have

$$g_W(\mathbf{e}) \geqslant \rho_{1/s}(\mathbf{e}) - \delta > \rho_{1/s}(\lambda_1(L + L_q(\mathbf{B})) - \|\mathbf{e}\| - \tau) + \delta + \eta \geqslant g_W(\mathbf{e} + \mathbf{x}) + \eta$$

for all $\mathbf{x} \in L \setminus L_q(\mathbf{B})$, where g_W is defined in (9).

Proof. Apply Lemma 8 to get that with probability at least $1-q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of $\mathbf{w}_1, \ldots, \mathbf{w}_N$ i.i.d. from $D_{L_q^\perp(\mathbf{B}),s}$, we have $|g_W(\mathbf{y}) - f_{L_q(\mathbf{B}),1/s}(\mathbf{y})| \leqslant \delta$ for all $\mathbf{y} \in \mathbb{Z}^m$. By Lemma 9, we have

$$g_W(\mathbf{e}) \geqslant f_{L_q(\mathbf{B}),1/s}(\mathbf{e}) - \delta \geqslant \rho_{1/s}(\mathbf{e}) - \delta.$$

Let $\mathbf{x} \in L \setminus L_q(\mathbf{B})$, then $\mathbf{z} - \mathbf{x} \in L + L_q(\mathbf{B})$ and $\mathbf{z} - \mathbf{x} \neq \mathbf{0}$ for any $z \in L_q(\mathbf{B})$. As a result, $L_q(\mathbf{B}) - \mathbf{x} \subseteq (L + L_q(\mathbf{B})) \setminus \{\,\mathbf{0}\,\}$. Hence,

$$\operatorname{dist}(\mathbf{x}, L_q(\mathbf{B})) = \min_{\mathbf{z}\in L_q(\mathbf{B})} \|\mathbf{x} + \mathbf{z}\| \geqslant \min_{\mathbf{y}\in(L+L_q(\mathbf{B}))\setminus\{\,\mathbf{0}\,\}} \|\mathbf{y}\| = \lambda_1(L+L_q(\mathbf{B})) \geqslant \tau+\|\mathbf{e}\|. \tag{13}$$

But then

$$\operatorname{dist}(\mathbf{e} + \mathbf{x}, L_q(\mathbf{B})) \geqslant \operatorname{dist}(\mathbf{x}, L_q(\mathbf{B})) - \|\mathbf{e}\| \geqslant \tau. \tag{14}$$

We can therefore apply Lemma 9 to get that for any $\mathbf{x} \in L \setminus \{\,\mathbf{0}\,\}$,

$$g_W(\mathbf{e} + \mathbf{x}) \leqslant f_{L_q(\mathbf{B}),1/s}(\mathbf{e} + \mathbf{x}) + \delta \leqslant \rho_{1/s}(\operatorname{dist}(\mathbf{e} + \mathbf{x}, L_q(\mathbf{B})) - \tau) + \delta.$$

Since $\rho_{1/s} : [0, \infty) \to \mathbb{R}$ is decreasing, and reusing (13) and (14) we further have

$$\begin{aligned}\rho_{1/s}(\operatorname{dist}(\mathbf{e} + \mathbf{x}, L_q(\mathbf{B})) - \tau) &\leqslant \rho_{1/s}(\operatorname{dist}(\mathbf{x}, L_q(\mathbf{B})) - \|\mathbf{e}\| - \tau)\\ &\leqslant \rho_{1/s}(\lambda_1(L + L_q(\mathbf{B})) - \|\mathbf{e}\| - \tau).\end{aligned}$$

Putting everything together, we have

$$g_W(\mathbf{e}) - g_W(\mathbf{e} + \mathbf{x}) \geqslant \rho_{1/s}(\mathbf{e}) - \rho_{1/s}(\lambda_1(L + L_q(\mathbf{B})) - \|\mathbf{e}\| - \tau) - 2\delta > \eta$$

by our assumption. □

Algorithm 2: Modern dual attack

Input: m, $n = n_{\text{guess}} + n_{\text{dual}}$ (see (3)), q prime power, $N \in \mathbb{N}$
Input: LWE sample $(\mathbf{A}, \mathbf{b})$, list $W = (\mathbf{w}_1, \dots, \mathbf{w}_N)$ of vectors in $L_q^{\perp}(\mathbf{A}_{\text{dual}})$.
Output: (Guess of) the first n_{guess} coordinates of the secret, or $\perp$.

```
s_guess ← ⊥
S_max ← 0
for s̃_guess ∈ Z_q^{n_guess} do
    Compute the list y_1, ..., y_N where y_j = w_j^T (b − A_guess s̃_guess)
    S ← Σ_{j=1}^{N} cos(2π y_j / q)
    if S ⩾ S_max then
        S_max ← S
        s_guess ← s̃_guess
return s_guess
```

We can now finally start our main result by putting everything together. It will be useful to note that $L_q(\mathbf{A}_{\text{guess}}) + L_q(\mathbf{A}_{\text{dual}}) = L_q(\mathbf{A})$ which is readily verified.

Theorem 5. *Let $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{e} \in \mathbb{Z}^m$, $\mathbf{s} \in \mathbb{Z}_q^n$, $s, \delta > 0$ and $N \in \mathbb{N}$. Let $\tau = \frac{1}{s}\sqrt{m/2\pi}$. Assume that $m \geqslant n$, $\mathbf{A}$ has full rank, $\lambda_1(L_q(\mathbf{A})) \geqslant \tau + \|\mathbf{e}\|$, and*

$$\rho_{1/s}(\mathbf{e}) - \rho_{1/s}(\lambda_1(L_q(\mathbf{A})) - \|\mathbf{e}\| - \tau) > 2\delta.$$

Let $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$. Let $W = (\mathbf{w}_1, \dots, \mathbf{w}_N)$ be samples from $D^N_{L_q^{\perp}(\mathbf{A}_{\text{dual}}),s}$, then Algorithm 2 on $(m, n_{\text{guess}}, n_{\text{dual}}, q, N, (\mathbf{A}, \mathbf{b}), W)$ runs in time $\mathsf{poly}(m, n) \cdot (N + q^{n_{\text{guess}}})$ and returns $\mathbf{s}_{\text{guess}}$ with probability at least $1 - q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of W.

Proof. Let $\mathbf{B} = \mathbf{A}_{\text{dual}}$ and $L = L_q(\mathbf{A}_{\text{guess}})$. Then $L + L_q(\mathbf{B}) = L_q(\mathbf{A})$. Our assumptions are therefore exactly that of Lemma 10 for $\eta = 0$ which we can apply to get that with probability at least $1 - q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of $W = (\mathbf{w}_1, \dots, \mathbf{w}_N)$ from $D^N_{L_q^{\perp}(\mathbf{B}),s} = D^N_{L_q^{\perp}(\mathbf{A}_{\text{dual}}),s}$, we have

$$g_W(\mathbf{e}) > g_W(\mathbf{e} + \mathbf{x}) \tag{15}$$

for all $\mathbf{x} \in L \setminus L_q(\mathbf{A}_{\text{dual}})$, where g_W is defined in (9). Furthermore, $\mathbf{A}$ has full rank and $m \geqslant n$ so its columns are linearly independent over $\mathbb{Z}_q$ and

$$L \setminus L_q(\mathbf{A}_{\text{dual}}) = L_q(\mathbf{A}_{\text{guess}}) \setminus L_q(\mathbf{A}_{\text{dual}}) = L_q(\mathbf{A}_{\text{guess}}) \setminus q\mathbb{Z}^m. \tag{16}$$

Assume that we are in the case where W satisfies the above inequalities and consider the run of Algorithm 2 on $(m, n_{\text{guess}}, n_{\text{dual}}, q, N, (\mathbf{A}, \mathbf{b}), W)$. The algorithm tests all possible values of $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and returns the one that maximizes S. Let $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and $\Delta\tilde{\mathbf{s}}_{\text{guess}} = \mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}$. First note that

$$\begin{aligned}\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}} &= (\mathbf{A}\mathbf{s} + \mathbf{e} \bmod q) - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}} \\ &= \mathbf{A}_{\text{dual}}\mathbf{s}_{\text{dual}} + \mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e} \bmod q.\end{aligned}$$

For any j, let $y_j(\tilde{\mathbf{s}}_{\text{guess}})$ be the value computed at Line 4. Note that

$$\begin{aligned} y_j(\tilde{\mathbf{s}}_{\text{guess}}) &= \mathbf{w}_j^T(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}}) \\ &= \mathbf{w}_j^T\mathbf{A}_{\text{dual}}\mathbf{s}_{\text{dual}} + \mathbf{w}_j^T(\mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e}) \bmod q \end{aligned}$$

but $\mathbf{w}_j \in L_q^\perp(\mathbf{A}_{\text{dual}})$ so $\mathbf{w}_j^T\mathbf{A}_{\text{dual}} = \mathbf{0} \bmod q$, hence

$$= \mathbf{w}_j^T(\mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e}) \bmod \ q.$$

Let $S(\tilde{\mathbf{s}}_{\text{guess}})$ be the value computed at Line 5 and check that

$$\begin{aligned} S(\tilde{\mathbf{s}}_{\text{guess}}) &= \sum\nolimits_{j=1}^{N} \cos(2\pi y_j(\tilde{\mathbf{s}}_{\text{guess}})/q) \\ &= \sum\nolimits_{j=1}^{N} \cos(2\pi \mathbf{w}_j^T(\mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e})/q) \qquad \text{by periodicity of } \cos \\ &= N g_W(\mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e}). \end{aligned}$$

There are two cases to distinguish:

- If $\tilde{\mathbf{s}}_{\text{guess}} = \mathbf{s}_{\text{guess}}$ then $S(\tilde{\mathbf{s}}_{\text{guess}}) = N g_W(\mathbf{e})$.
- If $\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}$ then $S(\tilde{\mathbf{s}}_{\text{guess}}) = N g_W(\mathbf{e} + \mathbf{x})$ where $\mathbf{x} = \mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} \in L_q(\mathbf{A}_{\text{guess}}) = L$. But $\mathbf{A}$ (and hence $\mathbf{A}_{\text{guess}}$) has full rank by assumption and $\Delta\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{0}$ so $\mathbf{x} \neq \mathbf{0} \bmod q$. It follows by (16), $\mathbf{x} \in L_q(\mathbf{A}_{\text{dual}}) \setminus q\mathbb{Z}^m = L \setminus L_q(\mathbf{A}_{\text{dual}})$. Hence, by (15), $S(\tilde{\mathbf{s}}_{\text{guess}}) < N g_W(\mathbf{e}) = S(\mathbf{s}_{\text{guess}})$.

This shows that $S(\mathbf{s}_{\text{guess}}) > S(\tilde{\mathbf{s}}_{\text{guess}})$ for all $\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}$. Therefore, Algorithm 2 correctly returns $\mathbf{s}_{\text{guess}}$. Note that the entire argument was under the assumption that (15) holds for W, which we already argued holds with probability at least $1 - q^m \cdot 2^{-\Omega(N\delta^2)}$.

The naive analysis of the complexity is straightforward and gives

$$q^{n_{\text{guess}}} \cdot \mathsf{poly}(m, n) \cdot N.$$

By using the DFT trick as we did in the proof of Theorem 4 we can improve the running time to

$$\mathsf{poly}(m, n) \cdot (N + q^{n_{\text{guess}}}).$$

4.3 Informal Application

Choosing the parameters in order to apply Theorem 5 is not immediately obvious. In this section, we explain how to do so in a concrete case of interest. In order to simplify things, we will neglect some factors and point out the various lemmas that can be used to make this reasoning completely formal.

Fix n, m and let q be a prime power. Let $\mathbf{s} \in \mathbb{Z}_q^n$ be a secret and $\sigma_e > 0$. Let $(\mathbf{A}, \mathbf{b})$ be sample from $\mathrm{LWE}(m, \mathbf{s}, D_{\mathbb{Z}_q,\sigma_e})$, and $\mathbf{e}$ so that $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$. By Corollary 1, we have

$$\|\mathbf{e}\| \lessapprox \sigma_e\sqrt{m/2\pi}$$

with high probability. Let $s > 0$ to be defined later. We choose δ to be quite smaller than the smallest possible value $\rho_{1/s}(\|\mathbf{e}\|)$, for example

$$\delta = \tfrac{1}{10}\rho_{1/s}(\sigma_e\sqrt{m/2\pi}) = \tfrac{1}{10}e^{-ms^2\sigma_e^2/2}.$$

We choose N accordingly so that the success probability is very high, *i.e.*

$$N = \frac{\mathsf{poly}(m) + n\log_2(q)}{\delta^2}.$$

By (1), $\mathbf{A}$ has full rank with high probability and therefore $\det(L_q(\mathbf{A})) = q^{m-n}$. By Theorem 3, and the informal Corollary 2, we have

$$\lambda_1(L_q(\mathbf{A})) \gtrapprox GH(L_q(\mathbf{A})) = \mathrm{vol}(B_m)^{-1/n}q^{1-m/n} \approx \sqrt{\frac{m}{2\pi e}}q^{1-n/m}.$$

Let $\tau = \frac{1}{s}\sqrt{m/2\pi}$. In order to apply Theorem 5, we need to satisfy the conditions

$$\lambda_1(L_q(\mathbf{A})) \geqslant \tau + \|\mathbf{e}\| \quad \text{and} \quad \rho_{1/s}(\mathbf{e}) - \rho_{1/s}(\lambda_1(L_q(\mathbf{A})) - \|\mathbf{e}\| - \tau) > 2\delta.$$

Since we have chosen δ to be very small compared to $\rho_{1/s}(\mathbf{e})$, those inequalities are almost equivalent to

$$\lambda_1(L_q(\mathbf{A})) \geqslant \tau + 2\|\mathbf{e}\|.$$

This condition will be satisfied roughly when

$$\sqrt{\frac{m}{2\pi e}}q^{1-n/m} \geqslant \tfrac{1}{s}\sqrt{m/2\pi} + 2\sigma_e\sqrt{m/2\pi}$$

that is

$$q^{1-n/m} \geqslant (\tfrac{1}{s} + 2\sigma_e)\sqrt{e}.$$

In other words, we have a lower bound on s. We observe that there is a trade-off between the cost of sampling from $D_{L_q^\perp(\mathbf{A}_{\mathrm{dual}}),s}$ and the cost of running Algorithm 2 since a large value of s:

- makes it easy to sample from $D_{L_q^\perp(\mathbf{A}_{\mathrm{dual}}),s}$,
- but makes $\delta = \frac{1}{10}\rho_{1/s}(\sigma_e\sqrt{m/2\pi})$ small and therefore $N = \Omega(\delta^{-2})$, and the complexity, gigantic.

We note that the total complexity of the attack, including the cost of generating the small dual vectors, is a highly nontrivial function of the parameters. Consequently, it is not at all clear that the optimal choice of s is the lower bound identified above. In previous papers on the topic, the optimal choice of the parameters has usually been obtained by running an optimisation procedure for concrete LWE instances. Furthermore, the majority of such papers, including ours, only provide a complexity up to polynomial factors which makes any comparison very delicate at best. Since this paper is mostly concerned with the correctness of the algorithm, we do not attempt to give a closed form formula for the overall complexity or give any numbers for concrete LWE schemes. We simply state that the overall cost of the attack is[8], up to polynomial factors,

$$T_{\text{short-vec}}(s, N(s)) + N(s) + q^{n_{\text{guess}}}$$

where $T_{\text{short-vec}}(N)$ is the cost of generating N independent dual vectors according to $D_{L_q^\perp(\mathbf{A}_{\text{dual}}),s}$ and we write $N(s) := N$ to emphasize that its value depends on the choice of s. As an example, one could implement this by first applying BKZ with block size β to a basis of $L_q^\perp(\mathbf{A}_{\text{dual}})$ and then use the resulting matrix in a Klein's sampler [22]. If done this way, the value of s depends on the quality of reduced basis and therefore of β and would give $T_{\text{short-vec}}(s, N) = T_{\text{BKZ}}(\beta) + N\mathsf{poly}(m)$.

5 Quantum dual attack

In this section, we present a quantum version of Algorithm 2 and show that we can obtain a speed-up on the complexity. The technique is inspired by [7] which was never published and is a quantum variant of [31]. The authors of [7] have confirmed to us that they have no objection to our adaptation of this idea.

5.1 Algorithm and analysis

We will need a quantum algorithm which estimates the mean value of $\cos(2\pi(\langle \mathbf{w}_i, \mathbf{b}\rangle)/q)$ where the $\mathbf{w}_i$ are vectors accessible via a quantum oracle. This mean value can be used to compute the DFT sums in the algorithm much faster than with a classical computer. The idea is inspired by [1, Theorem 47] and can be seen as a special case of quantum speedup of Monte Carlo methods [36]. For more background on quantum algorithms, we refer the reader [7, Sections 2.4 and 4].

Theorem 6 ([7, Theorem 5]). *Let N be a positive integer and W be a list of N vectors in $\mathbb{Z}^n$: $\mathbf{w}_0, \dots, \mathbf{w}_{N-1}$. Let $f_W(\mathbf{b}) = \frac{1}{N}\sum_{i=0}^{N-1}\cos(2\pi(\langle \mathbf{w}_i, \mathbf{b}\rangle)/q)$, where $\mathbf{b} \in \mathbb{Z}_q^n$. Let $\mathcal{O}_W$ be defined by*

$$\mathcal{O}_W : |j\rangle\,|0\rangle \mapsto |j\rangle\,|\mathbf{w}_j\rangle\,. \tag{17}$$

For any $\epsilon, \delta > 0$, there exists a quantum algorithm $\mathcal{A}$ that given $\mathbf{b} \in \mathbb{Z}_q^n$ and oracle access to $\mathcal{O}_W$ outputs $\mathcal{A}^{\mathcal{O}_W}(\mathbf{b})$ which satisfies $|\mathcal{A}^{\mathcal{O}_W}(\mathbf{b}) - f_W(\mathbf{b})| \le \epsilon$ with probability $1-\delta$. The algorithm makes $\mathcal{O}(\epsilon^{-1}\cdot\log\frac{1}{\delta})$ queries to $\mathcal{O}_W$, and requires $O(\log(\frac{1}{\epsilon}) + \mathsf{poly}(\log(n)))$ qubits.

We will have to search for a minimum element in a collection but the oracle that computes the value of each element is probabilistic and may return a wrong result with small probability. We say that a (probabilistic) real function f has

[8] Recall that this complexity already takes into account the DFT trick to compute the sum in time $N + q^{n_{\text{guess}}}$ instead of $N \cdot q^{n_{\text{guess}}}$.

bounded error if there exists $x \in \mathbb{R}$ such that $f()$ returns x with probability at least 9/10. The problem of finding the minimum in a collection (without errors) has been studied in [19, Theorem 1]. On the other hand, the problem of searching for a marked element in a collection with bounded-error oracle has been studied in [24]. This idea can easily be used to adapt the algorithm of [19] to bounded-error oracles. Indeed, the algorithm in [19] simply performs a constant number of Grover searches by marking nodes that are bigger than the current value. Therefore it suffices to replace this Grover search by the algorithm of [24].

Theorem 7 ([24]+[19]). *Given n algorithms, quantum or classical, each computing some real value with bounded error probability there is a quantum algorithm that makes an expected $O(\sqrt{n})$ queries and with probability at least 9/10 returns the index of the minimum among the n values. This algorithm uses $\mathsf{poly}(\log(n))$ qubits.*

Algorithm 3: Quantum modern dual attack

Input: m, $n = n_{\text{guess}} + n_{\text{dual}}$ (see (3)), q prime power, $N \in \mathbb{N}$, $\eta > 0$.
Input: LWE sample $(\mathbf{A}, \mathbf{b})$.
Input: Oracle $\mathcal{O}_W$ for a list $W = (\mathbf{w}_1, \dots, \mathbf{w}_N)$ of vectors in $L_q^{\perp}(\mathbf{A}_{\text{dual}})$.
Output: (Guess of) the first n_{guess} coordinates of the secret, or $\perp$.

Use Theorem 6 to create an algorithm $\mathcal{A}$ with $\delta = \frac{1}{10}$, $\varepsilon = \eta$ and q
create oracle $\hat{\mathcal{O}}(\tilde{\mathbf{s}}_{\text{guess}})$**:**
  return $\mathcal{A}^{\mathcal{O}_W}(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}})$
Use Theorem 7 to find $\tilde{s}_{\text{guess}}$ such that $\hat{\mathcal{O}}(\tilde{\mathbf{s}}_{\text{guess}})$ is maximum
return $\tilde{\mathbf{s}}_{\text{guess}}$

Theorem 8 (Appendix E.1). *Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, $\mathbf{e} \in \mathbb{Z}^m$, $\mathbf{s} \in \mathbb{Z}_q^n$, $s, \delta > 0$ and $N \in \mathbb{N}$. Let $\tau = \frac{1}{s}\sqrt{m/2\pi}$ and $\eta > 0$. Assume that $m \geqslant n$, $\mathbf{A}$ has full rank, $\lambda_1(L_q(\mathbf{A})) \geqslant \tau + \|\mathbf{e}\|$, and*

$$\rho_{1/s}(\mathbf{e}) - \rho_{1/s}(\lambda_1(L_q(\mathbf{A})) - \|\mathbf{e}\| - \tau) > 2\delta + \eta.$$

Let $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q$. Let $W = (\mathbf{w}_1, \dots, \mathbf{w}_N)$ be samples from $D^N_{L_q^{\perp}(\mathbf{A}_{\text{dual}}),s}$ and $\mathcal{O}_W$ an oracle for W in the sense of (17). Then Algorithm 3 on $m, n_{\text{guess}}, n_{\text{dual}}, q, N, \eta/2, (\mathbf{A}, \mathbf{b}), \mathcal{O}_W$ makes an expected $O\left(\eta^{-1} \cdot q^{n_{\text{guess}}/2}\right)$ calls to $\mathcal{O}_W$ and returns $\mathbf{s}_{\text{guess}}$ with probability at least $1 - q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of W. The algorithm uses $O(\log(\eta^{-1}) + \mathsf{poly}(\log(N))$ qubits.

In terms of proofs, the correctness of the quantum algorithm is very similar to the classical one. The main difference is that we use Theorem 6 to compute g_W which only returns an approximation. This adds an additional error term that we can take into account in Lemma 10 using η.

5.2 Applications

In order to apply Theorem 8, one needs to provide an oracle $\mathcal{O}_W$ to access the samples. The implementation of this oracle has a significant impact on the complexity since it is queried an exponential number of times by the algorithm. We outline two possible implementations. Before that, note that in practice we will usually choose η to be a small value compared to δ in Theorem 8, say $\eta = \delta/100$. This way, η has almost no influence on the maximum length of the errors $\mathbf{e}$ that we can handle.

BKZ preprocessing with a quantum Klein sampler. For a value of s that is not too small, one can first compute a basis $L_q^{\perp}(\mathbf{A}_{\text{dual}})$ of the dual lattice and reduce it using BKZ with block size β to obtain a new basis $\mathbf{M}$. One then creates a quantum circuit that implements the Klein sampler with $\mathbf{M}$ hard-coded in the circuit. This circuit will be the oracle $\mathcal{O}_W$. In the details, the Klein sampler is a probabilistic algorithm so we can view it as a deterministic algorithm that takes random coins (and $\mathbf{M}$) as input. We can see the input j of the oracle as the value of the random coins so that the outputs $\mathbf{w}_1, \dots, \mathbf{w}_N$ that correspond to inputs $1, \dots, N$ are distributed according to the Gaussian distribution. Since the Klein sampler [22] runs in polynomial time, each call to $\mathcal{O}_W$ takes polynomial time. The BKZ preprocessing is purely classical and done only once before the quantum algorithm runs. This means that the total runtime will be[9], per Theorem 8

$$T_{\text{BKZ}}(\beta) + \sqrt{N} \cdot q^{n_{\text{guess}}/2} \cdot \mathsf{poly}(\log(m)).$$

This is always better than the classical complexity since $\sqrt{N \cdot q^{n_{\text{guess}}}} \leqslant N + q^{n_{\text{guess}}}$. Note that when using a Klein sampler, the value of s is a function of the quality of the basis $\mathbf{M}$ and therefore depends on β. Furthermore, it is impossible for s to be smaller than the smoothing parameter of the lattice this way.

Classical sampler with a quantum memory. A feature of the Klein sampler is that it can output only one sampler and the running time is proportional to the number of samples. This is not the case of all samplers. For example, [3] describes Gaussian samplers that works for smaller values of s than the smoothing parameter and produces $2^{n/2}$ samples but runs in time 2^n, even if we only require one sample. [2] contains another such algorithm with a time-space trade-off. Using such samplers with our quantum algorithm is problematic because the samples are produced and stored in a classical memory, but the algorithm requires quantum oracle access to those samples. We have two options:

- We can assume that we have access to a QRACM (classical memory with quantum random access) [27]. A QRACM of size N is a special quantum memory holding N classical values but providing $O(\log(N))$-time quantum access to those values. Such a QRACM directly implements the oracle $\mathcal{O}_W$ so the total execution time becomes

$$T_{\text{sampler}} + \sqrt{N} \cdot q^{n_{\text{guess}}/2} \cdot \log(N) \cdot \mathsf{poly}(\log(m)).$$

[9] We chose $\eta = \delta/100$ and we explained in Section 4.3 that we need to choose $N = 1/\delta^2$ up to polynomial factors so $\eta^{-1} = \mathsf{poly}(\log(m)) \cdot \sqrt{N}$.

We note however that practical realizability of QRACM is debated and is potentially a strong assumption. We refer the reader to [25] for more details.
– We can replace $\mathcal{A}$ in the algorithm by a very large circuit containing all N hard-coded samples that computes the sum g_W in a naive way (without Theorem 6). This circuit will take time $N\mathsf{poly}(\log(m))$ to evaluate, therefore the total complexity will be

$$T_{\text{sampler}} + N \cdot q^{n_{\text{guess}}/2} \cdot \mathsf{poly}(\log(m)).$$

Note that this might be worse than the classical algorithm if the value of N is larger than $q^{n_{\text{guess}}/2}$.

Finally, we note that presently samplers such as [2] are still too expensive to be useful in dual attacks but future samplers might get more efficient.

6 Comparison with [18]'s Contradictory Regime

In [18], the authors claim that [31] falls into what they call the "contradictory regime" and conclude that the result is most likely incorrect. They similarly conclude the recent derivative works [7,13], as well as [23] are flawed. They do so by reconstructing the key heuristic claim of [31] and showing, both by theoretical arguments and experiments, that this heuristic is incorrect. We copy this heuristic below, slightly adjusted to our notations.

Heuristic 1 ([18, Heuristic Claim 3]). *Let $\Lambda \subseteq \mathbb{R}^n$ be a random lattice of determinant 1, $\mathcal{W} \subseteq \widehat{\Lambda}$ be the set consisting of the $N = (4/3)^{n/2}$ shortest vectors of $\widehat{\Lambda}$. For some $\sigma > 0$ and $T \geqslant 1$, consider $\mathbf{t}_{BDD} \leftarrow\!\!\$\ \mathcal{N}(0,\sigma^2)^n$ and i.i.d $\mathbf{t}^{(i)}_{unif} \leftarrow\!\!\$\ \mathcal{U}(\mathbb{R}^n/\Lambda)$ where $i \in \{1,\dots,T\}$. Let[10] $\ell = \sqrt{4/3}\cdot \mathrm{GH}(n)$, $\varepsilon = \exp(-2\pi^2\sigma^2\ell^2)$. If $\ln T \leqslant N\varepsilon^2$,*

$$\Pr\Big[f_{\mathcal{W}}(\mathbf{t}_{BDD}) > f_{\mathcal{W}}(\mathbf{t}^{(i)}_{unif}) \text{ for all } i \in \{1,\dots,T\}\Big] \geqslant 1 - O\left(\frac{1}{\sqrt{\ln T}}\right)$$

where $\mathcal{N}(0,\sigma^2)$ denotes the normal distribution.

In this heuristic, the function $f_{\mathcal{W}}$ is the same as $h_{\mathcal{W}}$ in Lemma 7, which is the same as $g_{\mathcal{W}}$ defined in (9) up to a factor $1/q$ in the cosine. There are several obvious (minor) problems about this heuristic since [31] works with integer lattices and discrete Gaussians. As a first step, we rewrite this heuristic in a way that is closer to [31] and we also change the notations to ours (see Appendix F.1 for details about the rewrite).

Heuristic 2 ([18, Heuristic Claim 3] adapted). *Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ with i.i.d. coefficients. Let $L = L_q(\mathbf{A}) \subseteq \mathbb{Z}^m$ and $W \subseteq L_q^{\perp}(\mathbf{A})$ be the set consisting of*

[10] We overload the notation GH: in [18], $\mathrm{GH}(m)$ corresponds to our $\mathrm{GH}(L)$ for L of volume 1, that is $\mathrm{vol}(B_m)^{-1/m}$.

the $N = (4/3)^{d/2}$ shortest vectors of $L_q^{\perp}(\mathbf{A})$. For some $\sigma_e > 0$ and $T \geqslant 1$, consider $\mathbf{e} \hookleftarrow\$ D^n_{\mathbb{Z}_q,\sigma_e}$ and i.i.d $\mathbf{t}^{(i)}_{unif} \hookleftarrow\$ \mathcal{U}(\mathbb{Z}^m/L)$ where $i \in \{1,\dots,T\}$. Let $\ell = \sqrt{4/3}\cdot \mathrm{GH}(L)$, $\varepsilon = \exp(-\pi\sigma_e^2\ell^2)$. If $\ln T \leqslant N\varepsilon^2$,

$$\Pr\Big[g_W(\mathbf{e}) > g_W(\mathbf{t}^{(i)}_{unif}) \text{ for all } i \in \{1,\dots,T\}\Big] \geqslant 1 - O\left(\frac{1}{\sqrt{\ln T}}\right).$$

In [18, Section 4.2 and 4.3], the authors argue by theoretical arguments that Heuristic 1 does not hold. Although [18] did not define what they mean by "random lattice" in the heuristic, they in fact use random q-ary lattices in their experiments and also the theoretical properties of "random lattices" that they use hold for q-ary lattices. Therefore, their analysis hold also for Heuristic 2.

Their reasoning is as follows: assume that we have a large number of random candidates (the $\mathbf{t}^{(i)}_{unif}$) and one point close to the lattice L (the point $\mathbf{e}$), then Heuristic 2 says that we can always distinguish $\mathbf{e}$ from the candidates (since it has maximum value of g_W). The contradiction comes from the fact that in reality, for T large enough, many of candidates will be closer to L than $\mathbf{e}$ and therefore no algorithm can distinguish them [16]. This gives rise to what [18] calls the "contradictory regime" where an algorithm would somehow be able to distinguish indistinguishable distributions.

We first compare this regime to that of our algorithm and we then discuss the statistical model chosen by [18] in Heuristic 1.

6.1 Almost complementary regimes

In Section 4.3, we have applied our main theorem to a concrete instance and derived that[11] for a typical LWE problem where the ratio m/n is fixed (and not too close to 0 or 1), q is large and the error follows a discrete Gaussian of parameter σ_e, our algorithm works as soon as

$$q^{1-n/m} \geqslant (\tfrac{1}{s} + 2\sigma_e)\sqrt{e} \tag{18}$$

where

$$N = \frac{\mathsf{poly}(m) + n\log_2(q)}{\delta^2}, \qquad \delta = \tfrac{1}{10}e^{-ms^2\sigma_e^2/2}.$$

In our attack, T is the number of guesses that the algorithm makes, that is $T = q^{n_{\text{guess}}}$. In order to match [18, page 21], we will choose s so that $\ln T = N\varepsilon^2$:

$$\begin{aligned}
\ln T = N\varepsilon^2 \quad &\Leftrightarrow \quad n_{\text{guess}}\ln(q) = \frac{\mathsf{poly}(m) + n\log_2(q)}{\delta^2}\varepsilon^2\\
&\Leftrightarrow \quad n_{\text{guess}}\ln(q) = (\mathsf{poly}(m) + n\log_2(q))100e^{2ms^2\sigma_e^2/2}e^{-2\pi\sigma_e^2\ell^2}\\
&\Leftrightarrow \quad \frac{n_{\text{guess}}\ln(q)}{100(\mathsf{poly}(m) + n\log_2(q))} = e^{(ms^2-2\pi\ell^2)\sigma_e^2}.
\end{aligned}$$

[11] Under some mild technical simplification to make the computation easier.

Note that $n_{\text{guess}} < n < m$ so for large enough value of q and m, the left-hand side of this expression is smaller than 1 (recall that $\mathsf{poly}(m)$ comes from the choice of N so we can always make it slightly bigger to artificially increase the denominator if we want). It follows that we can always choose such that $\ln T = N\varepsilon^2$ in such a way that (18) holds (see Appendix F.2) and therefore Theorem 5 ensures that our algorithm works in this regime.

We will now compare this with [18]'s contradictory regime. This regime, defined in [18, page 21] is when[12]

$$r\,\mathrm{GH}(L_q(\mathbf{A}_{\text{dual}})) < \sqrt{\frac{m}{2\pi}}\sigma_e, \qquad \text{where } r = T^{-1/m}. \tag{19}$$

Note here that the lattice is $\mathbf{A}_{\text{dual}}$ because [18] modularizes the algorithm by separating the lattice in which dual-distinguishing is done, with the part of the lattice that is enumerated over (see Section 6.2). Indeed, this regime comes from Heuristic 1 and the lattice in question is the one where dual vectors are generated.

Recall that for the algorithm to work, $\mathbf{A}$ and therefore $\mathbf{A}_{\text{dual}}$ must have full rank, so $\det(L_q(\mathbf{A}_{\text{dual}})) = q^{m-n_{\text{dual}}}$. Now observe that

$$\frac{r\,\mathrm{GH}(L_q(\mathbf{A}_{\text{dual}}))}{\sqrt{\frac{m}{2\pi}}\sigma_e} = \frac{T^{-1/m}\sqrt{\frac{m}{2\pi e}}q^{1-n_{\text{dual}}/m}}{\sqrt{\frac{m}{2\pi}}\sigma_e} = \frac{q^{-n_{\text{guess}}/m}q^{1-n_{\text{dual}}/m}}{\sqrt{e}\sigma_e}.$$

Recall that $n = n_{\text{dual}} + n_{\text{guess}}$ so the contradictory regimes corresponds to

$$q^{1-n/m} < \sigma_e\sqrt{e}. \tag{20}$$

Comparing between the working regime (18) and the contradictory one (20), and recalling that we can choose s as large as we want, we observe that they do not overlap and the bounds only differ by a factor of two. This suggest that, for our algorithm, the "theoretically working" regime and the contradictory regime almost characterize whether the dual attack will work or not. However, the next section will explain that those regimes are based on different distributions of targets.

6.2 On the distribution of targets

The authors of [18] decided to modularize the algorithm by separating the lattice in which dual-distinguishing is done ($L_q(\mathbf{A}_{\text{dual}})$) from the part of the lattice that is enumerated over ($L_q(\mathbf{A}_{\text{guess}})$). In fact, Heuristic 1 only mentions the dual-distinguishing and not the enumeration. This however, poses a difficulty because it is clear that the "targets" ($\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}}$) in our terminology, $\mathbf{t}_{unif}^{(i)}$ in Heuristic 1) are not arbitrary but have some structure.

The authors of [18] decided to model the statistics of the targets in a way that is independent of the actual choice of $\mathbf{A}_{\text{guess}}$: they chose the uniform distribution

[12] Recall that because of the difference between the normal distribution and the dicrete Gaussian, we have $\sigma = \sigma_e/\sqrt{2\pi}$ in our analysis, see Appendix F.1.

over the fundamental domain of $L_q(\mathbf{A}_{\text{dual}})$. In the case of [31] and our algorithm, the algorithm exclusively works over integers which is why we propose Heuristic 2 as an integer-version of Heuristic 1. This means that we now have two different settings:

- In Heuristic 2, $\mathbf{t}_{unif}^{(i)}$ is sampled uniformly in $\mathbb{Z}^m/L$.
- In reality, $\mathbf{t}_{unif}^{(i)} = \mathbf{e}+\mathbf{x}^{(i)}$ where $\mathbf{x}^{(i)}$ can be any vector in $L' \setminus q\mathbb{Z}^m$ where L' is another random q-ary lattice, chosen independently of L. In our algorithm, $L = L_q(\mathbf{A}_{\text{dual}})$ and $L' = L_q(\mathbf{A}_{\text{guess}})$.

Indeed, a key point in the proof of Theorem 5 is to show that points of the form $\mathbf{e} + \mathbf{x}^{(i)}$ as described are always far away from L, a fact that does not hold for completely uniform targets. As a result, with high probability over the choice of $\mathbf{A}$, the targets (except for the correct guess) are *all bounded away* from 0 in the dual lattice. For uniform targets, the argument of [18] is statistical in nature: while there can be very short vectors, they are unlikely and the contradiction comes from the fact that if we try too many targets, we will eventually find a short one and get a false-positive. On the other hand, our algorithm and analysis is not statistical: for the vast majority of choices of $\mathbf{A}$, all targets satisfy the bound unconditionally and we can safely look at all targets without the risk of any false-positive.

In conclusion of this section, it seems that the contradictory regime of [18] nicely complements the working regime of our algorithm. On the other hand, the statistical model that underlines this contradictory regime and what happens in our algorithm are different. We leave it as an open question to explain exactly why the two regimes seem to align perfectly.

7 Open questions

We have analyse formally a dual attack in the spirit of [31]. However, as noted in [18], the algorithm used by [31] produces many short dual vectors *in a sublattice* L'' *of* $L_q^\perp(\mathbf{A}_{\text{dual}})$ (instead of the entire $L_q^\perp(\mathbf{A}_{\text{dual}})$). In other words, W is roughly the set of vectors of L'' in a ball and therefore g_W does not exactly measure the distance to L but rather to a more complicated lattice. This fact makes the analysis of g_W considerably more challenging and we believe that more research is needed to understand how this affects the choice of the parameters.

Another issue that we have avoided is that of modulus switching. Indeed, while [31] claims that this techniques bring significant improvements in the complexity, [18] claims that geometric arguments contradicts this statement. We leave as an open problem the study of a modification of our algorithm that would include modulus switching. We believe that a formal analysis would be the best way to resolve this issue. A priori, we do not see any major reason why this could not be analysed formally but it may prove to be a nontrivial technical challenge due to the effects of rounding modulo p on the uniform distribution modulo q. We note in this direction that the approach of [13] of using lattice codes instead of modulus switching might be a better fit for a formal analysis.

Acknowledgments

We thank the anonymous TCC reviewers for pointing out an error in a previous version of this paper where we misunderstood the contradictory regime of [18]. We thank Martin Albrecht for his helpful comments on a previous version of this paper. We thank Léo Ducas for helpful discussions on the statistical model of [18]. Y.S. is supported by EPSRC grant EP/W02778X/1.

References

1. Aggarwal, D., Chen, Y., Kumar, R., Shen, Y.: Improved classical and quantum algorithms for the shortest vector problem via bounded distance decoding (2020), https://arxiv.org/abs/2002.07955, arXiv:2002.07955
2. Aggarwal, D., Chen, Y., Kumar, R., Shen, Y.: Improved (provable) algorithms for the shortest vector problem via bounded distance decoding. In: Bläser, M., Monmege, B. (eds.) 38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference). LIPIcs, vol. 187, pp. 4:1–4:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.STACS.2021.4, https://doi.org/10.4230/LIPIcs.STACS.2021.4
3. Aggarwal, D., Dadush, D., Regev, O., Stephens-Davidowitz, N.: Solving the shortest vector problem in 2^n time using discrete gaussian sampling: Extended abstract. In: Servedio, R.A., Rubinfeld, R. (eds.) Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015. pp. 733–742. ACM (2015). https://doi.org/10.1145/2746539.2746606, https://doi.org/10.1145/2746539.2746606
4. Aharonov, D., Regev, O.: Lattice problems in np $\cap$ conp. J. ACM **52**(5), 749–765 (sep 2005). https://doi.org/10.1145/1089023.1089025, https://doi.org/10.1145/1089023.1089025
5. Ajtai, M.: The shortest vector problem in l2 is np-hard for randomized reductions (extended abstract). In: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing. p. 10–19. STOC '98, Association for Computing Machinery, New York, NY, USA (1998). https://doi.org/10.1145/276698.276705, https://doi.org/10.1145/276698.276705
6. Albrecht, M.R.: On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. In: EUROCRYPT (2). pp. 103–129. Springer (2017). https://doi.org/10.1007/978-3-319-56614-6_4
7. Albrecht, M.R., Shen, Y.: Quantum augmented dual attack. Cryptology ePrint Archive, Paper 2022/656 (2022), https://eprint.iacr.org/2022/656, https://eprint.iacr.org/2022/656
8. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange: A new hope. In: Proceedings of the 25th USENIX Conference on Security Symposium. p. 327–343. SEC'16, USENIX Association, USA (2016)
9. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen **296**, 625–635 (1993)
10. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Krauthgamer, R. (ed.) Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016. pp. 10–24. SIAM

(2016). https://doi.org/10.1137/1.9781611974331.ch2, https://doi.org/10.1137/1.9781611974331.ch2
11. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. p. 575–584. STOC '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2488608.2488680, https://doi.org/10.1145/2488608.2488680
12. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-lwe and security for key dependent messages. In: Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference. Lecture Notes in Computer Science, vol. 6841, p. 501. Springer (2011). https://doi.org/10.1007/978-3-642-22792-9_29, https://www.iacr.org/archive/crypto2011/68410501/68410501.pdf
13. Carrier, K., Shen, Y., Tillich, J.P.: Faster dual lattice attacks by using coding theory (2022), https://eprint.iacr.org/2022/1750, https://eprint.iacr.org/2022/1750
14. Chunhuan, Z., Zhongxiang, Z., Xiaoyun, W., Guangwu, X.: Distinguishing lwe instances using fourier transform: A refined framework and its applications. Cryptology ePrint Archive, Paper 2019/1231 (2019), https://eprint.iacr.org/2019/1231, https://eprint.iacr.org/2019/1231
15. Dadush, D., Regev, O., Stephens-Davidowitz, N.: On the closest vector problem with a distance guarantee. In: 2014 IEEE 29th Conference on Computational Complexity (CCC). pp. 98–109 (2014). https://doi.org/10.1109/CCC.2014.18
16. Debris, T., Ducas, L., Resch, N., Tillich, J.P.: Smoothing codes and lattices: Systematic study and new bounds. Cryptology ePrint Archive, Paper 2022/615 (2022), https://eprint.iacr.org/2022/615, https://eprint.iacr.org/2022/615
17. Debris-Alazard, T., Remaud, M., Tillich, J.P.: Quantum reduction of finding short code vectors to the decoding problem. Cryptology ePrint Archive, Paper 2021/752 (2021), https://eprint.iacr.org/2021/752, https://eprint.iacr.org/2021/752
18. Ducas, L., Pulles, L.N.: Does the dual-sieve attack on learning with errors even work? IACR Cryptol. ePrint Arch. p. 302 (2023), https://eprint.iacr.org/2023/302
19. Dürr, C., Høyer, P.: A quantum algorithm for finding the minimum. CoRR **quant-ph/9607014** (1996), http://arxiv.org/abs/quant-ph/9607014
20. Erez, U., Litsyn, S., Zamir, R.: Lattices which are good for (almost) everything. IEEE Transactions on Information Theory **51**(10), 3401–3416 (2005). https://doi.org/10.1109/TIT.2005.855591
21. Espitau, T., Joux, A., Kharchenko, N.: On a Dual/Hybrid Approach to Small Secret LWE: A Dual/Enumeration Technique for Learning with Errors and Application to Security Estimates of FHE Schemes. In: Bhargavan, K., Oswald, E., Prabhakaran, M. (eds.) Progress in Cryptology – INDOCRYPT 2020, vol. 12578, p. 440–462. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-65277-7_20, http://link.springer.com/10.1007/978-3-030-65277-7_20, series Title: Lecture Notes in Computer Science
22. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing. p. 197–206. STOC '08, Association for Computing Machinery, New York, NY, USA (2008). https://doi.org/10.1145/1374376.1374407, https://doi.org/10.1145/1374376.1374407

23. Guo, Q., Johansson, T.: Faster dual lattice attacks for solving lwe – with applications to crystals. Springer-Verlag (2021)
24. Høyer, P., Mosca, M., de Wolf, R.: Quantum search on bounded-error inputs. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) Automata, Languages and Programming. pp. 291–299. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
25. Jaques, S., Rattew, A.G.: Qram: A survey and critique (2023)
26. Klein, P.: Finding the closest lattice vector when it's unusually close. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms. p. 937–941. SODA '00, Society for Industrial and Applied Mathematics, USA (2000)
27. Kuperberg, G.: A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. SIAM Journal on Computing **35**(1), 170–188 (2005). https://doi.org/10.1137/S0097539703436345, https://doi.org/10.1137/S0097539703436345
28. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011. p. 319–339. CT-RSA'11, Springer-Verlag, Berlin, Heidelberg (2011)
29. Loeliger, H.A.: Averaging bounds for lattices and linear codes. IEEE Transactions on Information Theory **43**(6), 1767–1773 (1997). https://doi.org/10.1109/18.641543
30. Macbeath, A.M., Rogers, C.A.: Siegel's mean value theorem in the geometry of numbers. Mathematical Proceedings of the Cambridge Philosophical Society **54**(2), 139–151 (1958). https://doi.org/10.1017/S0305004100033302
31. MATZOV: Report on the Security of LWE: Improved Dual Lattice Attack. Available at https://doi.org/10.5281/zenodo.6412487 (Apr 2022). https://doi.org/10.5281/zenodo.6412487, https://doi.org/10.5281/zenodo.6412487
32. Meyer-Hilfiger, C., Tillich, J.P.: Rigorous foundations for dual attacks in coding theory. Cryptology ePrint Archive, Paper 2023/1460, Accepted to TCC (2023), https://eprint.iacr.org/2023/1460, https://eprint.iacr.org/2023/1460
33. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. In: 45th Annual IEEE Symposium on Foundations of Computer Science. pp. 372–381 (2004). https://doi.org/10.1109/FOCS.2004.72
34. Micciancio, D., Regev, O.: Lattice-based Cryptography, pp. 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_5, https://doi.org/10.1007/978-3-540-88702-7_5
35. Micciancio, D., Voulgaris, P.: Faster exponential time algorithms for the shortest vector problem. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms. p. 1468–1480. SODA '10, Society for Industrial and Applied Mathematics, USA (2010)
36. Montanaro, A.: Quantum speedup of monte carlo methods. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **471**(2181), 20150301 (sep 2015). https://doi.org/10.1098/rspa.2015.0301, https://doi.org/10.1098%2Frspa.2015.0301
37. Nguyen, P.Q., Vidick, T.: Sieve algorithms for the shortest vector problem are practical. Journal of Mathematical Cryptology **2**(2), 181–207 (2008). https://doi.org/doi:10.1515/JMC.2008.009, https://doi.org/10.1515/JMC.2008.009
38. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing. p. 84–93. STOC '05, Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1060590.1060603, https://doi.org/10.1145/1060590.1060603

39. Rogers, C.A.: Mean values over the space of lattices. Acta Mathematica **94**(0), 249–287 (1955). https://doi.org/10.1007/BF02392493, http://projecteuclid.org/euclid.acta/1485892152
40. Schnorr, C.: A hierarchy of polynomial time lattice basis reduction algorithms. Theor. Comput. Sci. **53**(2), 201–224 (jun 1987)
41. Siegel, C.L.: A Mean Value Theorem in Geometry of Numbers. The Annals of Mathematics **46**(2), 340 (Apr 1945). https://doi.org/10.2307/1969027, https://www.jstor.org/stable/1969027?origin=crossref
42. Stephens-Davidowitz, N.: On the Gaussian measure over lattices. Phd thesis, New York University (2017)
43. Zamir, R., Nazer, B., Kochman, Y., Bistritz, I.: Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation and Multiuser Information Theory. Cambridge University Press (2014). https://doi.org/10.1017/CBO9781139045520

Supplementary material

A Details on the comparison with [32]

In this section, we try to compare our approach (and more generally lattice dual attacks) with the approach of [32]. As much as possible, we will try to express things with our notations.

The starting point of the "coding dual attack" is to consider a code $\mathcal{C}$ of dimension k and length n with parity matrix $\mathbf{H}$. Following our notation, we will instead use n to denote the dimension and m the ambient dimension, *i.e.* $k \to n$ and $n \to m$. In our language, this corresponds to a lattice $L = L_q^\perp(\mathbf{H}) \subseteq \mathbb{Z}^m$ for some $\mathbf{H} \in \mathbb{Z}_q^{m \times (m-n)}$. We get a vector $\mathbf{b} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in L$ and $\mathbf{e}$ sampled according to some distribution and the goal is to recover $\mathbf{c}$. Let $\mathbf{h}$ be a parity-check vector of L, *i.e.*[13] $\mathbf{h} \in q\widehat{L}$. Note that since $L = L_q^\perp(\mathbf{H})$ then there exists a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ such that $q\widehat{L} = L_q(\mathbf{A})$. Therefore $\mathbf{h} \in L_q(\mathbf{A})$.

The idea of the attack is to partition the coordinates $\{1, \ldots, m\}$ into two sets $\mathcal{P}$ and $\mathcal{N}$. Then observe that

$$\langle \mathbf{b}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathcal{P}}, \mathbf{h}_{\mathcal{P}} \rangle + \langle \mathbf{e}_{\mathcal{N}}, \mathbf{h}_{\mathcal{N}} \rangle.$$

Therefore, $\langle \mathbf{b}, \mathbf{h} \rangle$ can be seen an "LWE" (LPN) sample given by

$$(\mathbf{a}, \langle \mathbf{a}, \mathbf{u} \rangle + e) \text{ where } \begin{cases} \mathbf{a} = \mathbf{h}_{\mathcal{P}}, \\ \mathbf{u} = \mathbf{e}_{\mathcal{P}}, \\ e = \langle \mathbf{e}_{\mathcal{N}}, \mathbf{h}_{\mathcal{N}} \rangle. \end{cases}$$

The idea is that by generating many parity-check vectors $\mathbf{h}$, we can obtain many "LWE" samples for the same secret $\mathbf{u}$. The attack then continues using the so-called RLPN (or ISD) algorithm to recover the secret $\mathbf{u} = \mathbf{e}_{\mathcal{P}}$ and then recursively solve the rest of the problem in a much lower dimension.

Superficially, it may look like the partition between $\mathcal{P}$ and $\mathcal{N}$ is similar to our splitting between $\mathbf{s}_{\text{dual}}$ and $\mathbf{s}_{\text{guess}}$, and the corresponding split of the matrix $\mathbf{A}$. In the details, however, this is quite different since $\mathcal{P}$ and $\mathcal{N}$ correspond to a "horizontal/row" split of the matrix $\mathbf{A}$ whereas lattice dual attack perform a "vertical/column" split of $\mathbf{A}$. Furthermore, the distribution of the "LWE" samples is completely non-standard. Since $\mathbf{h}_{\mathcal{P}}$ and $\mathbf{h}_{\mathcal{N}}$ are components of a single parity check vector $\mathbf{h} \in L_q(\mathbf{A})$:

- the components of $\mathbf{a}$ are not independent and not uniform,
- $\mathbf{a}$ and e are not independent.

In particular, it looks like [32, Proposition 1] is needed to fix the dependency introduced by this particular approach. By comparison, such a problem does not appear in our approach and therefore we do not need a similar proposition.

[13] The factor q is necessary to keep $\mathbf{h}$ an integer vector.

In conclusion, it looks like [32], if applied to lattices, might be a completely different kind of "dual attack" on LWE, probably quite different from the existing ones. In particular, we are not sure where [32, Proposition 1] would fit in existing attacks.

B Distinguisher

B.1 Proof of Lemma 4

Proof. Fix $j \in \{1, 2\}$. Let $X_1, \dots, X_N \leftarrow\!\!\$\; \mathcal{X}_j$. Let

$$S_N \coloneqq \sum_{i=1}^{N} Y_i, \qquad Y_i \coloneqq \cos(X_i).$$

By linearity of the X_i,

$$\mathbb{E}_{X_i \leftarrow\$\mathcal{X}_j}[S_N] = N\,\mathbb{E}_{X \leftarrow\$\mathcal{X}_j}[\cos(X)] = NE_j.$$

Furthermore, it is clear that $-1 \leqslant Y_i \leqslant 1$. Therefore, by Theorem 1, for all $t > 0$,

$$\Pr_{X_i \leftarrow\$\mathcal{X}_j}[|S_N - NE_j| \geqslant t] \leqslant 2\exp\left(\frac{-t^2}{2N}\right).$$

Set $t = \frac{N}{2}|E_1 - E_2| > 0$, then $|S_N - NE_1| < t$ implies that $|S_N - NE_2| \geqslant t$ and vice versa. Therefore the algorithm can simply compute $|S_N - NE_1|$ and check whether this quantity is strictly less than t. This algorithm is correct with probability at least $1 - 2\exp(-\frac{N}{8}|E_1 - E_2|^2)$.

C Basic dual attack: proof of Theorem 4

Lemma 11. *Let q be a prime power, $k \in \mathbb{N}$ and $f : \mathbb{Z}_q^k \to \mathbb{Z}_q$ be linear map. If f is not the zero map then $f(\mathcal{U}(\mathbb{Z}_q)) = \mathcal{U}(\mathbb{Z}_q)$.*

Proof. Since q be a prime power, $\mathbb{Z}_q$ is a field. Since f is not the zero map, $f(\mathbb{Z}_q)$ has at least dimension one, therefore $f(\mathbb{Z}_q) = \mathbb{Z}_q$. But then for any $u \in \mathbb{Z}_q$, $f^{-1}(\{u\})$ has size exactly $|\ker f|$. It follows that for any $u \in \mathbb{Z}_q$,

$$\Pr_{\mathbf{x} \leftarrow\$\mathcal{U}(\mathbb{Z}_q^k)}[f(\mathbf{x}) = u] = \frac{|f^{-1}(\{u\})|}{|\mathbb{Z}_q^k|} = \frac{|\ker f|}{q^k}$$

is independent of u which proves the result. □

Proof (Proof of Theorem 4). Denote by $\mathcal{B}$ the algorithm from Black box 1, which we assume to be deterministic (otherwise we would need to add the random bits to the probability below which can be done easily). By the algorithm and Equation (5), $\mathbf{x}_j = \mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)}) \in L_q^{\perp}(\mathbf{A}_{\text{dual}}^{(j)})$. Furthermore, by Black box 1,

$$\mathbb{E}_{\mathbf{A}_{\text{dual}} \leftarrow\$\mathbb{Z}_q^{m \times n_{\text{dual}}}}\left[\|\mathcal{B}(\mathbf{A}_{\text{dual}})\|^2\right] \leqslant \ell_{\text{SV}}(m, \beta, q^{n_{\text{dual}}})^2 \tag{21}$$

Furthermore, note that since $q\mathbb{Z}^m \subseteq L_q^\perp(\mathbf{A}_{\text{dual}}^{(j)})$, we can always make sure that the coordinates of $\mathbf{x}_j$ are in[14] $\{-\lfloor q/2 \rfloor, \ldots, \lfloor q/2 \rfloor\}$.

Denote by $\mathcal{A}$ the algorithm from Algorithm 1. Again we assume that $\mathcal{A}$ is deterministic for simplicity, *i.e.* the loop looks at the elements in deterministic order. For any $\tilde{\mathbf{s}}_{\text{guess}}$, let

$$y_j(\tilde{\mathbf{s}}_{\text{guess}}) = \mathbf{x}_j^T(\mathbf{b}^{(j)} - \mathbf{A}_{\text{guess}}^{(j)}\tilde{\mathbf{s}}_{\text{guess}}), \qquad S(\tilde{\mathbf{s}}_{\text{guess}}) = \sum_{j=1}^{N} \cos(2\pi y_j(\tilde{\mathbf{s}}_{\text{guess}})/q)$$

be the values computed at Lines 5 and 6.

Let $\tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_{q^{n_{\text{guess}}}}$ be the order in which the algorithm tries the values of $\tilde{\mathbf{s}}_{\text{guess}}$. Let M be the index such that $\tilde{\mathbf{s}}_M = \mathbf{s}_{\text{guess}}$. Therefore, the algorithm returns the correct value if the test at Line 7 fails during the first $M-1$ iterations of the loop, and succeeds at the M^{th} iteration. In other word, the success probability of the algorithm is

$$\begin{aligned}
p &\coloneqq \Pr_{(\mathbf{A}^{(j)},\mathbf{b}_j) \leftarrow\$ \mathrm{LWE}(m,\mathbf{s},D_{\mathbb{Z}_q,\sigma_e})}\left[\bigwedge_{j=1}^{M-1} S(\tilde{\mathbf{s}}_j) < N\delta \bigwedge S(\mathbf{s}_{\text{guess}}) \geqslant N\delta\right] \\
&= 1 - \Pr_{\mathbf{A}^{(j)},\mathbf{b}_j}\left[\bigvee_{j=1}^{M-1} S(\tilde{\mathbf{s}}_j) \geqslant N\delta \bigvee S(\mathbf{s}_{\text{guess}}) < N\delta\right] \\
&\geqslant 1 - \sum_{j=1}^{M-1} \Pr_{\mathbf{A}^{(j)},\mathbf{b}_j}[S(\tilde{\mathbf{s}}_j) \geqslant N\delta] - \Pr_{(\mathbf{A}^{(j)},\mathbf{b}_j)}[S(\mathbf{s}_{\text{guess}}) < N\delta] \\
&= q(\mathbf{s}_{\text{guess}}) - \sum_{j=1}^{M-1} q(\tilde{\mathbf{s}}_j)
\end{aligned}$$

where for any $\tilde{\mathbf{s}} \in \mathbb{Z}_q^{n_{\text{guess}}}$,

$$q(\tilde{\mathbf{s}}_{\text{guess}}) \coloneqq \Pr_{(\mathbf{A}^{(j)},\mathbf{b}_j) \leftarrow\$ \mathrm{LWE}(m,\mathbf{s},D_{\mathbb{Z}_q,\sigma_e}}[S(\tilde{\mathbf{s}}_{\text{guess}}) \geqslant N\delta].$$

Recall that by definition of the LWE distribution, $\mathbf{b}^{(j)} = \mathbf{A}^{(j)}\mathbf{s} + \mathbf{e}^{(j)}$ where $\mathbf{e}^{(j)} \leftarrow\$ D_{\mathbb{Z}_q,\sigma_e}$, and $\mathbf{A}^{(j)} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{m\times n})$. Therefore,

$$\begin{aligned}
y_j(\tilde{\mathbf{s}}_{\text{guess}}) &= \mathbf{x}_j^T(\mathbf{b}^{(j)} - \mathbf{A}_{\text{guess}}^{(j)}\tilde{\mathbf{s}}_{\text{guess}}) \\
&= \mathbf{x}_j^T(\mathbf{A}^{(j)}\mathbf{s} + \mathbf{e}^{(j)} - \mathbf{A}_{\text{guess}}^{(j)}\tilde{\mathbf{s}}) \\
&= \mathbf{x}_j^T(\mathbf{A}_{\text{dual}}^{(j)}\mathbf{s}_{\text{dual}} + \mathbf{e}^{(j)} + \mathbf{A}_{\text{guess}}^{(j)}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}})) \\
&= \mathbf{x}_j^T(\mathbf{e}^{(j)} + \mathbf{A}_{\text{guess}}^{(j)}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}))
\end{aligned}$$

since $\mathbf{x}_j^T\mathbf{A}_{\text{dual}}^{(j)} = 0$ by definition of $\mathbf{x}_j$. Recall, as this is crucial for the proof, that $\mathbf{x}_j = \mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})$ only depends on $\mathbf{A}_{\text{dual}}^{(j)}$ which is independent of $\mathbf{A}_{\text{guess}}^{(j)}$. We now analyse two cases. To emphasize the various (in)dependence between the variables, the coming calculations will be extremely detailed.

[14] This is a technical condition, we obviously expect the coordinates to be much smaller than that.

• **If** $\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}$ then for every j,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}[\cos(2\pi y_j(\tilde{\mathbf{s}}_{\text{guess}})/q)] \\
&= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}[\exp(2i\pi y_j(\tilde{\mathbf{s}}_{\text{guess}})/q)]\right) \\
&= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}\left[\exp(2i\pi \mathbf{x}_j^T(\mathbf{e}^{(j)} + \mathbf{A}_{\text{guess}}^{(j)}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}))/q)\right]\right) \\
&= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}\left[\exp(2i\pi \mathbf{x}_j^T\mathbf{e}^{(j)})/q)\exp(2i\pi \mathbf{x}_j^T\mathbf{A}_{\text{guess}}^{(j)}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}})/q)\right]\right)
\end{aligned}$$

but $\mathbf{A}_{\text{dual}}^{(j)}$ and $\mathbf{A}_{\text{guess}}^{(j)}$ are independent so

$$= \Re\left(\mathbb{E}_{\mathbf{A}_{\text{dual}}^{(j)}}\left[\mathbb{E}_{\mathbf{A}_{\text{guess}}^{(j)},\mathbf{e}^{(j)}}\left[\begin{array}{l}\exp(2i\pi\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T\mathbf{e}^{(j)})/q) \\ \times\exp(2i\pi\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T\mathbf{A}_{\text{guess}}^{(j)}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}})/q)\end{array}\right]\right]\right)$$

but $\mathbf{A}_{\text{guess}}^{(j)}$ and $\mathbf{e}^{(j)}$ are independent so

$$\begin{aligned}
&= \Re\left(\mathbb{E}_{\mathbf{A}_{\text{dual}}^{(j)}}\left[\begin{array}{l}\mathbb{E}_{\mathbf{e}^{(j)}}\left[\exp(2i\pi\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T\mathbf{e}^{(j)})/q)\right] \\ \times\mathbb{E}_{\mathbf{A}_{\text{guess}}^{(j)}}\left[\exp(2i\pi\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T\mathbf{A}_{\text{guess}}^{(j)}(\mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}})/q)\right]\end{array}\right]\right) \\
&= \Re\left(\mathbb{E}_{\mathbf{A}_{\text{dual}}^{(j)}}\left[\begin{array}{l}\mathbb{E}_{\mathbf{e}^{(j)}}\left[\exp(2i\pi\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T\mathbf{e}^{(j)})/q)\right] \\ \times\mathbb{E}_{\mathbf{A}_{\text{guess}}^{(j)}}\left[\exp(2i\pi f_{\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})}(\mathbf{A}_{\text{guess}}^{(j)})/q)\right]\end{array}\right]\right)
\end{aligned}$$

where for any nonzero $\mathbf{x} \in \mathbb{Z}_q^n$, we let $f_{\mathbf{x}} : \mathbf{A}_{\text{guess}} \mapsto \mathbf{x}^T\mathbf{A}_{\text{guess}}(\tilde{\mathbf{s}} - \mathbf{s}_{\text{guess}})$ which is a linear map and not the zero-map since $\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}$; but then for any fixed $\mathbf{A}_{\text{dual}}^{(j)}$, $f_{\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})}(\mathcal{U}(\mathbb{Z}_q^{m\times n_{\text{guess}}})) = \mathcal{U}(\mathbb{Z}_q)$ by Lemma 11 and since $\mathbf{A}_{\text{guess}}^{(j)} \hookleftarrow\!\!\$\ \mathcal{U}(\mathbb{Z}_q^{m\times n_{\text{guess}}})$ is independent of $\mathbf{A}_{\text{dual}}^{(j)}$,

$$\begin{aligned}
&= \Re\left(\mathbb{E}_{\mathbf{A}_{\text{dual}}^{(j)}}\left[\mathbb{E}_{\mathbf{e}^{(j)}}\left[\exp(2i\pi\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T\mathbf{e}^{(j)})/q)\right]\mathbb{E}_{\alpha\hookleftarrow\$\,\mathcal{U}(\mathbb{Z}_q)}[\exp(2i\pi\alpha/q)]\right]\right) \\
&= 0
\end{aligned}$$

since $\mathbb{E}_{\alpha\hookleftarrow\$\,\mathcal{U}(\mathbb{Z}_q)}[\exp(2i\pi\alpha/q)] = 0$ by Lemma 5. It follows that

$$\mathbb{E}[S(\tilde{\mathbf{s}}_{\text{guess}})] = \sum_{j=1}^{N}\mathbb{E}[\cos(2\pi y_j(\tilde{\mathbf{s}}_{\text{guess}})/q)] = \sum_{j=1}^{N}\mathbb{E}_{y\hookleftarrow\$\,\mathcal{U}(\mathbb{Z}_q)}[\cos(2\pi y/q)] = 0.$$

On the other hand, note that $-1 \leqslant \cos(2\pi y_j(\tilde{\mathbf{s}}_{\text{guess}})/q) \leqslant 1$ for all j. Furthermore, note that the $y_j(\tilde{\mathbf{s}}_{\text{guess}})$ are independent since the $(\mathbf{A}^{(j)}, \mathbf{e}^{(j)})$ are independent. Therefore, we can apply Theorem 1 to get for any $\delta > 0$ that

$$\begin{aligned}
q(\tilde{\mathbf{s}}_{\text{guess}}) = \Pr_{(\mathbf{A}^{(j)},\mathbf{b}^{(j)})}[S(\tilde{\mathbf{s}}_{\text{guess}}) \geqslant N\delta] &\leqslant \exp\left(-\frac{2(N\delta)^2}{N\cdot(1-(-1))^2}\right) \\
&= \exp\left(-\frac{N\delta^2}{2}\right) \qquad \text{when } \tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}.
\end{aligned}$$

• **If** $\tilde{s} = s_{\text{guess}}$ then for every j,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}[\cos(2\pi y_j(\mathbf{s}_{\text{guess}})/q)] \\
&= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}[\exp(2i\pi y_j(\mathbf{s}_{\text{guess}})/q)]\right) \\
&= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}\left[\exp(2i\pi \mathbf{x}_j^T \mathbf{e}^{(j)}/q)\right]\right) \\
&= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)},\mathbf{e}^{(j)}}\left[\exp(2i\pi \mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T \mathbf{e}^{(j)}/q)\right]\right) \\
&= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)}}\left[\mathbb{E}_{\mathbf{e}^{(j)}}\left[\exp(2i\pi \mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})^T \mathbf{e}^{(j)}/q)\right]\right]\right)
\end{aligned}
$$

but the coordinates of $\mathbf{e}^{(j)}$ are independently sampled from $D_{\mathbb{Z}_q,\sigma_e}$,

$$
= \Re\left(\mathbb{E}_{\mathbf{A}^{(j)}}\left[\prod_{p=1}^{m} \mathbb{E}_{e \leftarrow\$ D_{\mathbb{Z}_q,\sigma_e}}\left[\exp(2i\pi \mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})_p e/q)\right]\right]\right)
$$

but note that each expected value in the product is real and lower-bounded by Lemma 5,

$$
\begin{aligned}
&\geqslant \mathbb{E}_{\mathbf{A}^{(j)}}\left[\prod_{p=1}^{m} \exp\left(-\pi\sigma_e^2 \mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})_p^2/q^2\right)\right] \\
&\geqslant \mathbb{E}_{\mathbf{A}^{(j)}}\left[\exp\left(-\pi\sigma_e^2 \left\|\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})\right\|^2/q^2\right)\right]
\end{aligned}
$$

but the function $X \mapsto \exp\left(-\pi\sigma_e^2 X/q^2\right)$ is convex so by Jensen's inequality,

$$
\geqslant \exp\left(-\pi\sigma_e^2\, \mathbb{E}_{\mathbf{A}^{(j)}}\left[\left\|\mathcal{B}(\mathbf{A}_{\text{dual}}^{(j)})\right\|^2\right]/q^2\right)
$$

but $\mathbf{A}_{\text{dual}}^{(j)}$ and $\mathbf{A}_{\text{guess}}^{(j)}$ are independent so by (21)

$$
\begin{aligned}
&\geqslant \exp\left(-\pi\sigma_e^2 \ell_{\text{SV}}(m,\beta,q^{n_{\text{dual}}})^2/q^2\right) \\
&= \varepsilon
\end{aligned}
$$

where ε is defined in the statement of Theorem 4. It follows that

$$
\mathbb{E}[S(\mathbf{s}_{\text{guess}})] = \sum_{j=1}^{N} \mathbb{E}[\cos(2\pi y_j(\mathbf{s}_{\text{guess}})/q)] \geqslant N\varepsilon.
$$

On the other hand, note that $-1 \leqslant \cos(2\pi y_j(\mathbf{s}_{\text{guess}})/q) \leqslant 1$ for all j. Furthermore, note that the $y_j(\mathbf{s}_{\text{guess}})$ are independent since the $(\mathbf{A}^{(j)}, \mathbf{e}^{(j)})$ are independent. Therefore,

$$\begin{aligned}
q(\mathbf{s}_{\text{guess}}) &= \Pr_{(\mathbf{A}^{(j)},\mathbf{b}^{(j)})}[S(\mathbf{s}_{\text{guess}}) \geqslant N\delta] \\
&= 1 - \Pr_{(\mathbf{A}^{(j)},\mathbf{b}^{(j)})}[S(\mathbf{s}_{\text{guess}}) < N\delta] \\
&= 1 - \Pr_{(\mathbf{A}^{(j)},\mathbf{b}^{(j)})}[N\varepsilon - S(\mathbf{s}_{\text{guess}}) > N(\varepsilon - \delta)] \\
&= 1 - \Pr_{(\mathbf{A}^{(j)},\mathbf{b}^{(j)})}[\mathbb{E}[S(\mathbf{s}_{\text{guess}})] - S(\mathbf{s}_{\text{guess}}) > N(\varepsilon - \delta)] \\
&\geqslant 1 - \exp\left(-\frac{2(N(\varepsilon-\delta))^2}{N \cdot (1-(-1))^2}\right) \qquad \begin{array}{l}\text{by Theorem 1 applied} \\ \text{to } -S(\mathbf{s}_{\text{guess}}) \text{ when } \varepsilon > \delta\end{array} \\
&= 1 - \exp\left(-\frac{N(\varepsilon-\delta)^2}{2}\right).
\end{aligned}$$

Putting everything together, we get that

$$p \geqslant 1 - \exp\left(-\frac{N(\varepsilon-\delta)^2}{2}\right) - (M-1)\exp\left(-\frac{N\delta^2}{2}\right)$$

and we conclude by nothing that $M \leqslant q^{n_{\text{guess}}}$.

The naive analysis of the complexity is straightforward:

- the first loop has complexity $N \cdot (\mathsf{poly}(m,n) + T_{\text{SV}}(m,\beta,q^n)$,
- the second loop has complexity $q^{n_{\text{guess}}} \cdot N \cdot \mathsf{poly}(m,n)$,

hence a total of

$$N \cdot \mathsf{poly}(m,n) \cdot (T_{\text{SV}}(m,\beta,q^n) + q^{n_{\text{guess}}}).$$

By using the DFT trick of [23], we can compute all scores S using a DFT for a total cost of $N + q^{n_{\text{guess}}}$ instead of $N \cdot q^{n_{\text{guess}}}$ so the final cost is

$$\mathsf{poly}(m,n) \cdot (N \cdot T_{\text{SV}}(m,\beta,q^n) + q^{n_{\text{guess}}}).$$

□

D Random q-ary lattices

As usual, an (n,k,q) (linear) code is a k-dimensional subspace of $\mathbb{Z}_q^n$. For any subset E of $\mathbb{Z}_q^n$, we denote by E^* the set $\{ e \in E : e \neq 0 \}$ of nonzero elements. For any matrix $A \in \mathbb{Z}_q^{n \times k}$, we let $C_A = A\mathbb{Z}_q^n = \{ x \in \mathbb{Z}_q^n : \exists s \in \mathbb{Z}_q^k : x = As \}$. For any matrix $B \in \mathbb{Z}_q^{n \times (n-k)}$, we let $C_B^\perp = \{ x \in \mathbb{Z}_q^n : B^T x = 0 \}$. It follows almost from the definition that such codes satisfy the following averaging lemma.

Lemma 12. *Let $f : \mathbb{Z}_q^n \to \mathbb{R}$, then*

$$\mathbb{E}_{B \sim U(\mathbb{Z}_q^{n \times (n-k)})}\left[\sum_{v \in (C_B^\perp)^*} f(v)\right] = q^{k-n} \sum_{v \in (\mathbb{Z}_q^n)^*} f(v).$$

More generally, let $1 \leqslant p \leqslant n$ *and* $f : (\mathbb{Z}_q^n)^p \to \mathbb{R}$*, then*

$$\begin{aligned}&\mathbb{E}_{B\sim U(\mathbb{Z}_q^{n\times(n-k)})}\left[\sum \begin{bmatrix} v_1,\dots,v_p \in C_B^\perp \\ \textit{lin indep}\end{bmatrix} f(v_1,\dots,v_p)\right]\\ &= q^{(k-n)p}\sum \begin{bmatrix} v_1,\dots,v_p \in \mathbb{Z}_q^n \\ \textit{lin indep}\end{bmatrix} f(v_1,\dots,v_p).\end{aligned}$$

Proof. For any $v \in \mathbb{Z}_q^n$, let $N_v = |\{\, B \in \mathbb{Z}_q^{n\times(n-k)} : v \in C_B^\perp \,\}$. We observe that if $v \neq 0$ then $N_v = q^{n(n-k)-n+k}$. Indeed, let $\phi : \mathbb{Z}_q^{n\times(n-k)} \to \mathbb{Z}_q^{n-k}$ defined by $\phi(B) = B^T v$. If $v \neq 0$ then $\operatorname{rk}\phi = n-k$ and therefore $\dim\ker\phi = n(n-k) - \operatorname{rk}\phi$. The result follows immediately by writing

$$\mathbb{E}_{B\sim U(\mathbb{Z}_q^{n\times(n-k)})}\left[\sum_{v\in(C_B^\perp)^*} f(v)\right] = \frac{1}{|\mathbb{Z}_q^{n\times(n-k)}|}\sum_{v\in(\mathbb{Z}_q^n)^*} N_v f(v).$$

The more general result follows the same proof by considering the map $\phi : \mathbb{Z}_q^{n\times(n-k)} \to (\mathbb{Z}_q^{n-k})^p$ defined by

$$\phi(B) = \bigl(B^T v_1,\dots,B^T v_p\bigr)$$

which has rank $(n-k)p$ for any $v_1,\dots,v_p$ are linearly independent. $\square$

Theorem 9. *Let* $1 \leqslant p \leqslant n$ *and* $f : (\mathbb{Z}_q^n)^p \to \mathbb{R}$*, then*

$$\begin{aligned}&\mathbb{E}_{B\sim U(\mathbb{Z}_q^{n\times(n-k)})}\left[\sum_{c_1\in C_B^\perp}\cdots\sum_{c_p\in C_B^\perp} f(c_1,\dots,c_p)\right]\\ &= \sum_{r=0}^{p} q^{(k-n)r}\sum \begin{bmatrix} c_1,\dots,c_p \in \mathbb{Z}_q^n \\ \operatorname{rk}(c_1,\dots,c_p) = r\end{bmatrix} f(c_1,\dots,c_p).\end{aligned}$$

Proof. Let $0 \leqslant r \leqslant p$ and let

$$\sigma_r = \mathbb{E}_B\left[\sum \begin{bmatrix} c_1,\dots,c_p \in C_B^\perp \\ \operatorname{rk}(c_1,\dots,c_p) = r\end{bmatrix} f(c_1,\dots,c_p)\right].$$

We can write

$$\mathbb{E}_B\left[\sum_{c_1\in C_B^\perp}\cdots\sum_{c_p\in C_B^\perp} f(c_1,\dots,c_p)\right] = \sum_{r=0}^{p}\sigma_r$$

so it suffices to compute each σ_r. Note that if $c_1,\dots,c_p \in C_B^\perp$ are such that $\operatorname{rk}(c_1,\dots,c_p) = r$ then r among them must be linearly independent and the others are in the span of those. Let G be the set of permutations on $\{\, 1,\dots,p \,\}$.

Then by Lemma 12,

$$
\begin{aligned}
\sigma_r &= \mathbb{E}_B\left[\sum \begin{bmatrix} c_1,\dots,c_r \in C_B^{\perp} \\ \text{lin indep} \end{bmatrix} \sum \left[\, c_{r+1},\dots,c_p \in \operatorname{span}(c_1,\dots,c_r) \,\right] \sum_{\sigma\in G} f(c_{\sigma(1)},\dots,c_{\sigma(p)})\right] \\
&= q^{(k-n)p} \sum \begin{bmatrix} c_1,\dots,c_r \in \mathbb{Z}_q^n \\ \text{lin indep} \end{bmatrix} \sum \left[\, c_{r+1},\dots,c_p \in \operatorname{span}(c_1,\dots,c_r) \,\right] \sum_{\sigma\in G} f(c_{\sigma(1)},\dots,c_{\sigma(p)}) \\
&= q^{(k-n)r} \sum \begin{bmatrix} c_1,\dots,c_p \in \mathbb{Z}_q^n \\ \operatorname{rk}(c_1,\dots,c_p) = r \end{bmatrix} f(c_1,\dots,c_p).
\end{aligned}
$$

□

D.1 Proof of Lemma 6

Proof. Let $n, k \in \mathbb{N}$ and q be a prime power. Let L be a q-ary lattice and let

$$\mathcal{A}(L) = \{\, \mathbf{A} \in \mathbb{Z}_q^{n\times k} : L = L_q(\mathbf{A}) \,\}, \quad \mathcal{A}^{\perp}(L) = \{\, \mathbf{B} \in \mathbb{Z}_q^{n\times(n-k)} : L = L_q^{\perp}(\mathbf{B}) \,\}.$$

By definition of L, $L \bmod q \coloneqq \{\, \mathbf{x} \bmod q : \mathbf{x} \in L \,\}$ is a linear subspace of $\mathbb{Z}_q^n$ of dimension $1 \leqslant r \leqslant n$. We denote this dimension by $\operatorname{rk}_{\mathbb{Z}_q}(L) \coloneqq r$. Now observe that by definition, if $L = L_q(\mathbf{A})$ then $L_q(\mathbf{A}) \bmod q = \mathbf{A}\mathbb{Z}_q^n$ and therefore $\operatorname{rk}(\mathbf{A}) = \operatorname{rk}_{\mathbb{Z}_q}(L)$. Similarly, if $L = L_q^{\perp}(\mathbf{B})$ then $L \bmod q = \ker(\mathbf{B}^T)$ and therefore $\operatorname{rk}(\mathbf{B}) = \operatorname{rk}(\mathbf{B}^T) = \dim(\operatorname{Im}(\mathbf{B}^T)) = n - \operatorname{rk}_{\mathbb{Z}_q}(L)$. Now let

$$\mathcal{F} = \{\, L : q\mathbb{Z}^n \subseteq L \subseteq \mathbb{Z}^n, \operatorname{rk}_{\mathbb{Z}_q} = k \,\}.$$

Then

$$
\begin{aligned}
\Pr_{L \leftarrow\$ \mathcal{L}_{n,k,q}}[L \in \mathcal{F}] &= \Pr_{L \leftarrow\$ \mathcal{L}_{n,k,q}}\left[\operatorname{rk}_{\mathbb{Z}_q}(L) = k\right] \\
&= \Pr_{\mathbf{A} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{n\times k})}\left[\operatorname{rk}_{\mathbb{Z}_q}(L_q(\mathbf{A})) = k\right] \quad \text{by definition} \\
&= \Pr_{\mathbf{A} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{n\times k})}[\operatorname{rk}(\mathbf{A}) = k] \quad \text{by the analysis above} \\
&\geqslant 1 - kq^{k-1-n} \quad \text{by (1).}
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\Pr_{L \leftarrow\$ \mathcal{L}^{\perp}_{n,k,q}}[L \in \mathcal{F}] &= \Pr_{L \leftarrow\$ \mathcal{L}^{\perp}_{n,k,q}}\left[\operatorname{rk}_{\mathbb{Z}_q}(L) = k\right] \\
&= \Pr_{\mathbf{B} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{n\times(n-k)})}\left[\operatorname{rk}_{\mathbb{Z}_q}(L_q^{\perp}(\mathbf{B})) = k\right] \quad \text{by definition} \\
&= \Pr_{\mathbf{B} \leftarrow\$ \mathcal{U}(\mathbb{Z}_q^{n\times(n-k)})}[\operatorname{rk}(\mathbf{B}) = n-k] \quad \text{by the analysis above} \\
&\geqslant 1 - (n-k)q^{k-1} \quad \text{by (1).}
\end{aligned}
$$

Now observe that

$$
\begin{aligned}
&\mathrm{d}_{\mathrm{TV}}(\mathcal{L}^{\perp}_{n,k,q}, \mathcal{L}_{n,k,q}) \\
&= \frac{1}{2}\sum_{L} \left|\mathcal{L}^{\perp}_{n,k,q}(L) - \mathcal{L}_{n,k,q}(L)\right| \\
&\leqslant \frac{1}{2}\sum_{L\in\mathcal{F}} \left|\mathcal{L}^{\perp}_{n,k,q}(L) - \mathcal{L}_{n,k,q}(L)\right| + \frac{1}{2}\sum_{L\notin\mathcal{F}} (\mathcal{L}^{\perp}_{n,k,q}(L) + \mathcal{L}_{n,k,q}(L)) \\
&\leqslant \frac{1}{2}\sum_{L\in\mathcal{F}} \left|\mathcal{L}^{\perp}_{n,k,q}(L) - \mathcal{L}_{n,k,q}(L)\right| + \frac{1}{2}\Pr_{L\leftarrow\$\mathcal{L}_{n,k,q}}[L\notin\mathcal{F}] \\
&\quad + \frac{1}{2}\Pr_{L\leftarrow\$\mathcal{L}^{\perp}_{n,k,q}}[L\notin\mathcal{F}] \\
&\leqslant \frac{1}{2}\sum_{L\in\mathcal{F}} \left|\mathcal{L}^{\perp}_{n,k,q}(L) - \mathcal{L}_{n,k,q}(L)\right| + kq^{k-1-n} + (n-k)q^{k-1}.
\end{aligned} \tag{22}
$$

Furthermore, we observe that

$$
\sum_{L\in\mathcal{F}} \left|\mathcal{L}^{\perp}_{n,k,q}(L) - \mathcal{L}_{n,k,q}(L)\right| = \sum_{L\in\mathcal{F}} \left|\frac{|\mathcal{A}(L)|}{q^{nk}} - \frac{|\mathcal{A}^{\perp}(L)|}{q^{n(n-k)}}\right|.
$$

We now claim that for $L\in\mathcal{F}$, $\frac{|\mathcal{A}(L)|}{q^{nk}} - \frac{|\mathcal{A}^{\perp}(L)|}{q^{n(n-k)}}$ is a constant independent of L. To see that, first observe that if $L = L_q(\mathbf{A})$ for some $\mathbf{A}\in\mathbb{Z}_q^{n\times k}$ of full rank (*i.e.* $\mathrm{rk}(\mathbf{A}) = k$) then the columns of $\mathbf{A}$ form a basis of $L \bmod q$. Therefore, $\mathcal{A}(L)$ is simply is the set of all basis of $L \bmod q$, a k-dimensional subspace, which is well-known to have cardinal

$$
|\mathcal{A}(L)| = (q^k-1)(q^k-q)\cdots(q^k-q^{k-1}).
$$

Similarly, if $L = L_q^{\perp}(\mathbf{B})$ for some $\mathbf{B}\in\mathbb{Z}_q^{n\times(n-k)}$ of full rank (*i.e.* $\mathrm{rk}(\mathbf{B}) = n-k$) then the columns of $\mathbf{B}$ form a basis of the orthogonal subspace of $L \bmod q$ which has dimension $n-k$. Therefore, $\mathcal{A}^{\perp}(L)$ is simply is the set of all basis of the orthogonal subspace of $L \bmod q$, a $n-k$-dimensional subspace, which is well-known to have cardinal

$$
|\mathcal{A}^{\perp}(L)| = (q^{n-k}-1)(q^{n-k}-q)\cdots(q^{n-k}-q^{n-k-1}).
$$

We can then notice that $\mathcal{F}$ is in fact isomorphic to the set of all k-dimensional subspaces of $\mathbb{Z}_q^n$ since a lattice L of $\mathcal{F}$ is uniquely identified by $L \bmod q$ which is a k-dimensional subspace, and every such subspace can be trivially realized by $L_q(\mathbf{A})$ where $\mathbf{A}$ is a basis. Therefore

$$
|\mathcal{F}| = \frac{(q^n-1)(q^n-q)\cdots(q^n-q^{k-1})}{(q^k-1)(q^n-q)\cdots(q^k-q^{k-1})}.
$$

It follows that

$$
\begin{aligned}
&\sum_{L\in\mathcal{F}} \left|\mathcal{L}^{\perp}_{n,k,q}(L)-\mathcal{L}_{n,k,q}(L)\right| \\
&= |\mathcal{F}| \left| \frac{1}{q^{nk}} \prod_{i=0}^{k-1}(q^k-q^i) - \frac{1}{q^{n(n-k)}} \prod_{i=0}^{n-k-1}(q^{n-k}-q^i)\right| \\
&= \frac{(q^n-1)(q^n-q)\cdots(q^n-q^{k-1})}{q^{nk}} \cdot \left|1-q^{-n(n-2k)}\frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=0}^{k-1}(q^k-q^i)}\right| \\
&\leqslant \left|1-q^{-n(n-2k)}\frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=0}^{k-1}(q^k-q^i)}\right|.
\end{aligned}
$$

There are two cases to consider.
•If $k \leqslant n-k$, then $n-2k \geqslant 0$ and

$$
\begin{aligned}
\frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=0}^{k-1}(q^k-q^i)} &= \frac{\prod_{i=n-2k}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=0}^{k-1}(q^k-q^i)} \prod_{i=0}^{n-2k-1}(q^{n-k}-q^i) \\
&= \frac{\prod_{j=0}^{k-1}(q^{n-k}-q^{n-2k+j})}{\prod_{i=0}^{k-1}(q^k-q^i)} \prod_{i=0}^{n-2k-1}(q^{n-k}-q^i) \\
&= \frac{\prod_{j=0}^{k-1}(q^k-q^j)}{\prod_{i=0}^{k-1}(q^k-q^i)} q^{(n-2k)k} \prod_{i=0}^{n-2k-1}(q^{n-k}-q^i) \\
&= q^{(n-2k)k} \prod_{i=0}^{n-2k-1}(q^{n-k}-q^i).
\end{aligned}
$$

And therefore

$$
\begin{aligned}
&\left|1-q^{-n(n-2k)}\frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=0}^{k-1}(q^k-q^i)}\right| \\
&= \left|1-q^{(n-2k)(k-n)}\prod_{i=0}^{n-2k-1}(q^{n-k}-q^i)\right| \\
&= \left|1-\prod_{i=0}^{n-2k-1}(1-q^{i-n+k})\right| \\
&= 1-\prod_{i=0}^{n-2k-1}(1-q^{i-n+k}) \\
&\leqslant \sum_{i=0}^{n-2k-1} q^{i-n+k} \quad \text{by Weierstrass product inequality} \\
&\leqslant q^{(n-2k)-n+k} \\
&= q^{-k}. \qquad (23)
\end{aligned}
$$

•If $k \geqslant n-k$, then $2k-n \geqslant 0$ so exactly the same argument yields

$$\begin{aligned}
\frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=0}^{k-1}(q^k-q^i)} &= \frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=2k-n}^{k-1}(q^k-q^i)\prod_{i=0}^{2k-n-1}(q^k-q^i)} \\
&= \frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{j=0}^{n-k-1}(q^k-q^{2k-n+j})\prod_{i=0}^{2k-n-1}(q^k-q^i)} \\
&= \frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{q^{(2k-n)(n-k)}\prod_{j=0}^{n-k-1}(q^{n-k}-q^j)\prod_{i=0}^{2k-n-1}(q^k-q^i)} \\
&= \frac{q^{(2k-n)(k-n)}}{\prod_{i=0}^{2k-n-1}(q^k-q^i)}.
\end{aligned}$$

And therefore

$$\begin{aligned}
&\left|1-q^{-n(n-2k)}\frac{\prod_{i=0}^{n-k-1}(q^{n-k}-q^i)}{\prod_{i=0}^{k-1}(q^k-q^i)}\right| \\
&= \left|1-\frac{q^{k(2k-n)}}{\prod_{i=0}^{2k-n-1}(q^k-q^i)}\right| \\
&= \left|1-\frac{1}{\prod_{i=0}^{2k-n-1}(1-q^{i-k})}\right| \\
&= \frac{1}{\prod_{i=0}^{2k-n-1}(1-q^{i-k})}-1 \\
&\leqslant \frac{1}{1-\sum_{i=0}^{2k-n-1}q^{i-k}}-1 \quad \text{by Weierstrass product inequality} \\
&= \frac{q^{-k}(q^{2k-n}-1)}{q+q^{-k}-1-q^{k-n}} \quad \text{after calculation} \\
&\leqslant \frac{q^{k-n}}{q+q^{-k}-1-q^{k-n}}. \qquad (24)
\end{aligned}$$

Recall that $k \leqslant n-k$ so if $q \geqslant 3$ then the denominator is at least 1 so the above quantity is less than q^{k-n}. If $q=2$ then the denominator becomes $1+2^{-k}-2^{k-n}$ which can be seen to be at least 1/2. Therefore in all cases, this quantity is bounded by $2q^{k-n}$.

Putting everything together, it follows from (22), (23) and (24) that

$$\mathrm{d}_{\mathrm{TV}}(\mathcal{L}_{n,k,q}^{\perp},\mathcal{L}_{n,k,q}) \leqslant \tfrac{1}{2}\max(q^{-k},2q^{k-n})+kq^{k-1-n}+(n-k)q^{k-1}$$

□

D.2 Proof of Theorem 2

Proof. Recall that to each matrix $B \in \mathbb{Z}_q^{n\times(n-k)}$, we associated the code $C_B^{\perp} = \{\, x \in \mathbb{Z}_q^n : B^T x = 0 \,\}$. We now let $\Lambda_B^{\perp} = \{\, x \in \mathbb{Z}^n : (x \mod q) \in C_B^{\perp} \,\}$: this is clearly a lattice and in fact, $L_q^\perp(B) = \Lambda_B^\perp$. In what follows, for any set X we write $f(X)$ for $\sum_{x\in X} f(x)$. We can therefore write

$$\begin{aligned}\mathbb{E}_{L\sim\mathcal{L}_{n,k,q}^\perp}\left[\sum_{x\in L^*} f(x)\right] &= \mathbb{E}_{B\sim U(\mathbb{Z}_q^{n\times(n-k)})}\left[\sum_{v\in(\Lambda_B^\perp)^*} f(v)\right]\\ &= \mathbb{E}_B\left[f((q\mathbb{Z}^n)^*) + \sum_{v\in(C_B^\perp)^*} f(v+q\mathbb{Z}^n)\right]\end{aligned}$$

where the last step follows because $\Lambda_B^\perp = q\mathbb{Z}^n + C_B^\perp$. Using Lemma 12, we get

$$\begin{aligned}\mathbb{E}_L\left[\sum_{x\in L^*} f(x)\right] &= f((q\mathbb{Z}^n)^*) + q^{k-n}\sum_{v\in(\mathbb{Z}_q^n)^*} f(v+q\mathbb{Z}^n)\\ &= (1-q^{k-n})f((q\mathbb{Z}^n)^*) + q^{k-n}\sum_{v\in(\mathbb{Z}^n)^*} f(v).\end{aligned}$$

□

D.3 Proof of Theorem 3

Proof. Let $0 < r \leqslant q$ and let f_r be the indicator function of $B_n(r)$. Note that if $\mathbf{x} \in q(\mathbb{Z}^n)^*$ then $\|\mathbf{x}\| \geqslant q$ so $\mathbf{x} \notin B_n(r)$ and $f_r(\mathbf{x}) = 0$. By Theorem 2,

$$\begin{aligned}\mu &:= \mathbb{E}_{L\leftarrow\$\mathcal{L}_{n,k,q}^\perp}\left[|L^*\cap B_n(r)|\right]\\ &= \mathbb{E}_{L\leftarrow\$\mathcal{L}_{n,k,q}^\perp}\left[\sum_{\mathbf{x}\in L^*} f_r(\mathbf{x})\right]\\ &= \left(1-q^{k-n}\right)\sum_{\mathbf{v}\in q(\mathbb{Z}^n)^*} f_r(\mathbf{v}) + q^{k-n}\sum_{\mathbf{x}\in(\mathbb{Z}^n)^*} f_r(\mathbf{x})\\ &= q^{k-n}\left(|B_n^{\mathbb{Z}}(r)| - 1\right).\end{aligned}$$

Similarly,

$$\begin{aligned}\mu_2 &:= \mathbb{E}_{L\leftarrow\$\mathcal{L}_{n,k,q}^\perp}\left[|L^*\cap B_n(r)|^2\right]\\ &= \mathbb{E}_{L\leftarrow\$\mathcal{L}_{n,k,q}^\perp}\left[\left(\sum_{\mathbf{x}\in L^*} f_r(\mathbf{x})\right)^2\right]\\ &= \mathbb{E}_{L\leftarrow\$\mathcal{L}_{n,k,q}^\perp}\left[\sum_{\mathbf{x},\mathbf{y}\in L^*} f_r(\mathbf{x})f_r(\mathbf{y})\right]\\ &= \sum_{\mathbf{x},\mathbf{y}\in(\mathbb{Z}^n)^*} q^{(k-n)\mathrm{rk}(\mathbf{x},\mathbf{y})} f_r(\mathbf{x})f_r(\mathbf{y})\end{aligned}$$

where $\mathrm{rk} \coloneqq \mathrm{rk}_{\mathbb{Z}_q^n}$. We now consider the various cases that can occur. If $\mathbf{x} \in q(\mathbb{Z}^n)^*$ then $f_r(\mathbf{x}) = 0$ so this term is zero, and similarly if $\mathbf{y} \in q(\mathbb{Z}^n)^*$. Therefore all nonzero terms must have $\mathrm{rk}(\mathbf{x}) = \mathrm{rk}(\mathbf{y}) = 1$ and $\mathrm{rk}(\mathbf{x}, \mathbf{y}) \in \{1, 2\}$. For $i \in \{1, 2\}$, let

$$S_i = \{ (\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \notin q\mathbb{Z}^n \text{ and } \mathrm{rk}(\mathbf{x}, \mathbf{y}) = i \}.$$

Let $(\mathbf{x}, \mathbf{y}) \in S_1$, then there exists $u \in \mathbb{Z}_q$ such that $\mathbf{y} = u\mathbf{x} \bmod q$, and $u \neq 0$ since a shifted sphere of radius $r \leqslant q$ cannot only contain at most point of $q\mathbb{Z}^n$. In other words,

$$S_1 = \{ (\mathbf{x}, u\mathbf{x} + q\mathbf{z}) : \mathbf{x} \notin q\mathbb{Z}^n, u \in \mathbb{Z}_q^*, \mathbf{z} \in \mathbb{Z}^n \}.$$

Assume that $f_r(\mathbf{x}) \neq 0$, then $\|\mathbf{x}\| \leqslant r$. Let $u \in \mathbb{Z}_q^*$, then $|\{ u\mathbf{x} + q\mathbf{z} : \mathbf{z} \in \mathbb{Z}^n \} \cap B_n(r)| \leqslant 1$ since $r \leqslant q$. It follows that $|S_1| \leqslant (q-1)(|B_n^{\mathbb{Z}}(r)| - 1)$. On the other hand, $|S_2| \leqslant (|B_n^{\mathbb{Z}}(r)| - 1)^2$. It follows that

$$\begin{aligned}
\mu_2 &= |S_1| q^{k-n} + |S_2| q^{2(k-n)} \\
&\leqslant q^{k-n}(q-1)(|B_n^{\mathbb{Z}}(r)| - 1) + q^{2(k-n)}(|B_n^{\mathbb{Z}}(r)| - 1)^2 \\
&= q^{k-n}(q-1)(|B_n^{\mathbb{Z}}(r)| - 1) + \mu^2.
\end{aligned}$$

It follows that

$$\sigma^2 \coloneqq \mathbb{V}_{L \leftarrow\$ \mathcal{L}_{n,k,q}^{\perp}}[|L^* \cap B_n(r)|] = \mu_2 - \mu^2 \leqslant q^{k-n}(q-1)(|B_n^{\mathbb{Z}}(r)| - 1).$$

We can now apply the above results to study λ_1 of a random lattice. For any lattice L, let $X_r(L) \coloneqq |L^* \cap B_n(r)|$. First note that for any radius $r > 0$, $\lambda_1(L) \leqslant r$ if and only if $X_r(L) \geqslant 2$ since as soon as there is one point in the ball, there are at least two (a vector and its opposite). Therefore we need to study the probability that $X_r(L) \geqslant 2$. Choose r such that $\mu \leqslant 1$, which is equivalent to

$$|B_n^{\mathbb{Z}}(r)| - 1 \leqslant q^{n-k}, \qquad \text{and therefore implied by} \qquad |B_n^{\mathbb{Z}}(r)| \leqslant q^{n-k}.$$

Then observe that

$$\begin{aligned}
\Pr_{L \leftarrow\$ \mathcal{L}_{n,k,q}^{\perp}}[X_r(L) \geqslant 2] &= \Pr_{L \leftarrow\$ \mathcal{L}_{n,k,q}^{\perp}}[X_r(L) - \mu \geqslant 2 - \mu] \\
&\leqslant \Pr_{L \leftarrow\$ \mathcal{L}_{n,k,q}^{\perp}}[|X_r(L) - \mu| \geqslant 2 - \mu] \\
&\leqslant \left(\frac{\sigma^2}{2 - \mu} \right)^2 \qquad \text{by Chebyshev's inequality} \\
&\leqslant \sigma^4 \\
&\leqslant q^{2(k-n)}(q-1)^2(|B_n^{\mathbb{Z}}(r)| - 1)^2.
\end{aligned}$$

□

D.4 Proof of Corollary 2

Proof. Note that by Section 2.5, if L is sampled according to $\mathcal{L}^{\perp}_{n,k,q}$ then $L = L_q(\mathbf{A})$ for some $\mathbf{A} \in \mathbb{Z}_q^{n\times(n-k)}$ and therefore $\det(L) \leqslant q^{n-k}$. As a result,

$$\mathrm{GH}(L) = \det(L)^{1/n} \operatorname{vol}(B_n)^{-1/n} \leqslant r := q^{1-k/n} \operatorname{vol}(B_n)^{-1/n}.$$

Note that since $\alpha \leqslant 1$, αr is such that

$$|B_n^{\mathbb{Z}}(\alpha r)| \approx \operatorname{vol}(B_n(\alpha r)) = \alpha^n r^n \operatorname{vol}(B_n) = \alpha^n \det(L) \leqslant \alpha^n q^{n-k} \leqslant q^{n-k}.$$

We can therefore apply Theorem 3 to αr to get that

$$\Pr_{L \leftarrow\$ \mathcal{L}^{\perp}_{n,k,q}}[\lambda_1(L) \leqslant \alpha r] \leqslant \left(q^{1+k-n}|B_n^{\mathbb{Z}}(\alpha r)|\right)^2 = (q\alpha^n)^2 .$$

By Lemma 6, the distributions $\mathcal{L}^{\perp}_{n,k,q}$ and $\mathcal{L}_{n,k,q}$ are very close and therefore the same holds for $\mathcal{L}_{n,k,q}$. □

E Quantum dual attack

E.1 Proof of Theorem 8

Proof. Let $\mathbf{B} = \mathbf{A}_{\text{dual}}$ and $L = L_q(\mathbf{A}_{\text{guess}})$. Then $L + L_q(\mathbf{B}) = L_q(\mathbf{A})$. Our assumptions are therefore exactly that of Lemma 10 which we can apply to get that with probability at least $1 - q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of $W = (\mathbf{w}_1, \dots, \mathbf{w}_N)$ from $D^N_{L_q^{\perp}(\mathbf{B}),s} = D^N_{L_q^{\perp}(\mathbf{A}_{\text{dual}}),s}$, we have

$$g_W(\mathbf{e}) > g_W(\mathbf{e} + \mathbf{x}) + \eta \tag{25}$$

for all $\mathbf{x} \in L \setminus L_q(\mathbf{A}_{\text{dual}})$, where g_W is defined in (9). Furthermore, $\mathbf{A}$ has full rank and $m \geqslant n$ so its columns are linearly independent over $\mathbb{Z}_q$ and

$$L \setminus L_q(\mathbf{A}_{\text{dual}}) = L_q(\mathbf{A}_{\text{guess}}) \setminus L_q(\mathbf{A}_{\text{dual}}) = L_q(\mathbf{A}_{\text{guess}}) \setminus q\mathbb{Z}^m. \tag{26}$$

Let $\eta' = \eta/2$. Assume that we are in the case where W satisfies the above inequalities and consider the run of Algorithm 3 on $m, n_{\text{guess}}, n_{\text{dual}}, q, N, \eta'$, $(\mathbf{A}, \mathbf{b})$, $\mathcal{O}_W$. Let $\tilde{\mathbf{s}}_{\text{guess}} \in \mathbb{Z}_q^{n_{\text{guess}}}$ and $\Delta\tilde{\mathbf{s}}_{\text{guess}} = \mathbf{s}_{\text{guess}} - \tilde{\mathbf{s}}_{\text{guess}}$. Recall that for $\mathbf{x} \in \mathbb{Z}_q^m$,

$$g_W(\mathbf{x}) = \frac{1}{N}\sum\nolimits_{j=1}^{N} \cos(2\pi \mathbf{w}_j^T \mathbf{x}/q)$$

and that since $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$, we have

$$\begin{aligned} g_W(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}}) &= \frac{1}{N}\sum\nolimits_{j=1}^{N} \cos(2\pi \mathbf{w}_j^T (\mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e})/q) \\ &= g_W(\mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e}). \end{aligned}$$

We claim that for all $\tilde{\mathbf{s}}_{\text{guess}}$, with probability at least 9/10, we have

$$|\mathcal{O}'(\tilde{\mathbf{s}}_{\text{guess}}) - g_W(\mathbf{A}_{\text{guess}}\Delta\tilde{\mathbf{s}}_{\text{guess}} + \mathbf{e})| \leqslant \eta'.$$

Indeed, $\mathcal{O}'(\tilde{\mathbf{s}}_{\text{guess}})$ returns $\mathcal{A}^{\mathcal{O}_W}(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}})$ where $\mathcal{A}$ was built by Theorem 6 with $''\delta'' = \frac{1}{10}$, $\varepsilon = \eta'$ and q. Therefore, with probability at least 9/10, we have

$$|\mathcal{A}^{\mathcal{O}_W}(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}}) - g_W(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}})| \leqslant \eta' = \eta/2$$

and the equations above conclude.

The reasoning above shows that $\mathcal{O}$ is a (real) bounded-error oracle for $g_W \pm \eta'$. Therefore by Theorem 7, the algorithm will correctly return, with probability at least 9/10, a value $\tilde{\mathbf{s}}_{\text{guess}}$ that maximizes $\hat{\mathcal{O}}$. We claim that when this happens, $\tilde{\mathbf{s}}_{\text{guess}} = \mathbf{s}_{\text{guess}}$. Assume the contrary, for sake of contradiction. We have

$$|\hat{\mathcal{O}}(\tilde{\mathbf{s}}_{\text{guess}}) - g_W(\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}})| < \eta'$$

and

$$|\hat{\mathcal{O}}(\mathbf{s}_{\text{guess}}) - g_W(\mathbf{b} - \mathbf{A}_{\text{guess}}\mathbf{s}_{\text{guess}})| < \eta'.$$

Recall that by (25),

$$g_W(\mathbf{b} - \mathbf{A}_{\text{guess}}\mathbf{s}_{\text{guess}}) = g_W(\mathbf{e}) > g_W(\mathbf{e} + \mathbf{x}) + \eta$$

for all $\mathbf{x} \neq \mathbf{0}$. But since $\tilde{\mathbf{s}}_{\text{guess}} \neq \mathbf{s}_{\text{guess}}$ by our assumption, we have that $\mathbf{b} - \mathbf{A}_{\text{guess}}\tilde{\mathbf{s}}_{\text{guess}} = \mathbf{e} + \mathbf{x}$ for some $\mathbf{x} \neq \mathbf{0}$. Hence,

$$\begin{aligned}
\hat{\mathcal{O}}(\tilde{\mathbf{s}}_{\text{guess}}) &< g_W(\mathbf{e} + \mathbf{x}) + \eta' \\
&\leqslant g_W(\mathbf{e}) - \eta + \eta' \\
&< \hat{\mathcal{O}}(\mathbf{s}_{\text{guess}}) + \eta' - \eta + \eta' \\
&= \hat{\mathcal{O}}(\mathbf{s}_{\text{guess}})
\end{aligned}$$

since $2\eta = \eta'$. But this contradicts the fact that $\tilde{\mathbf{s}}_{\text{guess}}$ was maximizing $\hat{\mathcal{O}}$. Therefore we must have $\tilde{\mathbf{s}}_{\text{guess}} = \mathbf{s}_{\text{guess}}$ and the algorithm is correct. Note that the entire argument was under the assumption that (25) holds for W, which we already argued holds with probability at least $1 - q^m \cdot 2^{-\Omega(N\delta^2)}$ over the choice of W.

Regarding the complexity, the algorithm in Theorem 7 makes an expected $O(\sqrt{q^{n_{\text{guess}}}})$ queries to $\hat{\mathcal{O}}$ and uses $\mathsf{poly}(\log(n))$ qubits. By Theorem 6, each call to $\hat{\mathcal{O}}$ makes

$$O\left(\eta'^{-1} \cdot \log(10)\right) = O(\eta^{-1})$$

queries to $\mathcal{O}_W$ and requires $O(\log(\eta^{-1}) + \mathsf{poly}(\log(N)))$ qubits.

F Comparison with [18]

F.1 Rewrite of Heuristic 1 into Heuristic 2

Similarly to our algorithm Algorithm 2, and as was made clear by the analysis of Section 4.2, [31] only works with q-ary lattices. Therefore we can specialize the heuristic to $L = L_q(\mathbf{A})$ where L is random in the sense that $\mathbf{A}$ is sampled from $\mathcal{U}(\mathbb{Z}_q^{m\times n})$. The dual of L is $\frac{1}{q}L_q^{\perp}(\mathbf{A})$ by Section 2.5. In a similar way to what we did in the proof of Lemma 8, we can "integrate" the $1/q$ factor directly in the score function which because g_W defined in (9) and therefore the set W becomes a subset of $L_q^{\perp}(\mathbf{A})$. The target $\mathbf{t}_{BDD}$ corresponds to the error $\mathbf{e}$ of our algorithm and is sampled according to a discrete Gaussian $D_{\mathbb{Z}_q,\sigma_e}$ instead of a normal distribution. Care must be taken at this point since the normalization factor in the exponent of the normal distribution is $1/2$ whereas it is π for the discrete Gaussian. This means that $D_{\mathbb{Z}_q,\sigma_e}$ corresponds to $\mathcal{N}(0,\sigma^2)$ where $\sigma = \sigma_e/\sqrt{2\pi}$. We reflect this change in value of ε. Finally, note that Heuristic 1 assumes that L has determinant 1 which is clearly not the case of $L_q(\mathbf{A})$. Instead of rescaling L, we observe that the only place where this makes a difference is the Gaussian Heuristic: [18] defines $\mathrm{GH}(m)$ as the typical value for a m-dimensional lattice of volume 1. In contrast, we define $\mathrm{GH}(L)$ as the typical value for a m-dimensional lattice of volume $\det(L)$.

F.2 Analysis of the value of s

Recall that we choose s such that

$$\frac{n_{\text{guess}}\ln(q)}{100(\mathsf{poly}(m)+n\log_2(q))} = e^{(ms^2-2\pi\ell^2)\sigma_e^2}$$

Note that the left-hand side is less than 1 and so its value is $1/P$ where P is a polynomial in m,n and $\ln(q)$. Hence, we need to take

$$\begin{aligned} s^2 = 2\frac{\pi\ell^2}{m} + \frac{-\ln(P)}{m\sigma_e^2} &= \frac{2\pi}{m}\cdot\left(\sqrt{\frac{4}{3}}\sqrt{\frac{m}{2\pi e}}q^{1-n/m}\right)^2 + \frac{-\ln(P)}{m\sigma_e^2} \\ &= \frac{4}{3e}q^{2(1-n/m)} + \frac{-\ln(P)}{m\sigma_e^2} \end{aligned}$$

Since P is polynomial, the second term of the right-hand side is negligible compared to the first one and therefore

$$s \approx \sqrt{\frac{4}{3e}q^{1-n/m}}.$$

We want to verify that it satisfies (18), that is

$$q^{1-n/m} \geqslant (\tfrac{1}{s}+2\sigma_e)\sqrt{e}.$$

In our choice of parameters, $n = 2m/3$ so $1 - n/m = 1/3$ and q is large so $q^{1-n/m}$ is much larger that $2\sigma_e\sqrt{e}$. Hence to satisfy the condition it is essentially enough to satisfy

$$s \geqslant \sqrt{e}q^{n/m-1}$$

which is trivially satisfied since s is very large (multiple of $q^{1-n/m}$) and the right-hand side if very small.