Cryptographic Hardness of Learning Halfspaces with Massart Noise

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of PAC learning halfspaces in the presence of Massart noise. In this model, we are given i.i.d. labeled examples $(\mathbf{x}, y) \in \mathbb{R}^N \times \{\pm 1\}$, where the distribution of $\mathbf{x}$ is arbitrary and the label y is a Massart corruption of $f(\mathbf{x})$, for an unknown halfspace $f : \mathbb{R}^N \to \{\pm 1\}$, with flipping probability $\eta(\mathbf{x}) \leq \eta < 1/2$. The goal is to compute a hypothesis with small 0-1 error. We give a reduction of the Learning with Errors (LWE) problem to the Massart halfspace learning problem. Assuming the (widely believed) subexponential-time hardness of the LWE problem, we show that no polynomial-time learner for Massart halfspaces can achieve error better than $\Omega(\eta)$, even if $\mathrm{OPT} = \mathbf{E}_x[\eta(\mathbf{x})] = 2^{-\log^c(N)}$, for any universal constant $c \in (0, 1)$. Prior work provided qualitatively similar evidence of hardness in the Statistical Query model. Our hardness result shows that known learning algorithms for Massart halfspaces are essentially best possible.

1 Introduction

A halfspace or linear threshold function (LTF) is any function $h_{\mathbf{w},t} : \mathbb{R}^N \to \{\pm 1\}$ of the form $h_{\mathbf{w},t}(\mathbf{x}) := \mathrm{sign}(\langle \mathbf{w}, \mathbf{x} \rangle - t)$, where the vector $\mathbf{w} \in \mathbb{R}^N$ is called the weight vector, $t \in \mathbb{R}$ is called the threshold, and $\mathrm{sign} : \mathbb{R} \to \{\pm 1\}$ is defined by $\mathrm{sign}(t) = 1$ if $t \geq 0$ and $\mathrm{sign}(t) = -1$ otherwise. We will use LTF_N for the class of halfspaces on $\mathbb{R}^N$; when N is clear from the context, we may discard it and simply write LTF. Halfspaces are a central concept class in machine learning, extensively investigated since the 1950s [Ros58, Nov62, MP68]. Here we study the complexity of learning halfspaces in Valiant's (distribution independent) PAC model [Val84] when the labels have been corrupted by *Massart noise* [MN06]. We define the Massart noise model below.

Definition 1.1 (Massart Noise)**.** We say that a joint distribution D of labeled examples $(\mathbf{x}, y)$, supported on $\mathbb{R}^N \times \{\pm 1\}$, satisfies the Massart noise condition with noise parameter $\eta \in [0, 1/2]$ with respect to a concept class C of Boolean-valued functions on $\mathbb{R}^N$ if there is a concept $c \in C$ such that for all $\mathbf{x}_0 \in \mathbb{R}^N$ we have that $\mathbf{Pr}_{(\mathbf{x},y)\sim D}[c(\mathbf{x}) \neq y \mid \mathbf{x} = \mathbf{x}_0] \leq \eta$.

Given i.i.d. samples from a Massart distribution D, the goal is to output a hypothesis with small 0-1 error. We use $R_C(D) \stackrel{\mathrm{def}}{=} \min_{h \in C} \mathbf{E}_D[\mathbf{Pr}(h(\mathbf{x}) \neq y)]$ to denote the error of the best classifier in C with respect to D, and $R_{\mathrm{OPT}}(D) \stackrel{\mathrm{def}}{=} \min_{h: X \to \{\pm 1\}} \mathbf{E}_D[\mathbf{Pr}(h(\mathbf{x}) \neq y)]$ for the error of the Bayes optimal classifier. The computational task of PAC learning Massart halfspaces is a classical problem in learning theory that has been posed by several authors since the 1980s [Slo88, Coh97, Blu03]. [DGT19] made the first algorithmic progress on this problem: they gave a $\mathrm{poly}(N, 1/\epsilon)$-time learning algorithm with error guarantee of $\eta + \epsilon$. Subsequent work improved the sample complexity of this algorithm [DKT21] and developed an efficient proper learner with the same error guarantee [CKMY20]. More recent work [DIK+21] gave an efficient boosting algorithm, achieving error $\eta + \epsilon$ for any concept class, assuming the existence of a weak learner for the class.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

The error bound of $\eta+\epsilon$ can be very far from the information-theoretically optimum error of $\mathrm{OPT}+\epsilon$, where $\mathrm{OPT} = R_{\mathrm{LTF}}(D) \leq \eta$. Indeed, it could well be the case that $\mathrm{OPT} \ll \eta$; known algorithms only guarantee error $\eta+\epsilon$ even if OPT is very small. This discussion prompts the following question:

Question 1.1. *Is there an efficient learning algorithm for Massart halfspaces with a* relative *error guarantee? Specifically, if* $\mathrm{OPT} \ll \eta$ *is it possible to achieve error significantly better than* η*?*

1.1 Our Results

Our main result answers this question in the negative, assuming the sub-exponential time hardness for the classical Learning with Errors (LWE) problem. In other words, *we essentially resolve the efficient PAC learnability of Massart halfspaces*, under a widely-believed cryptographic assumption.

To state our result in more detail, let us recall that in the classical LWE problem, we are given samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ and the goal is to distinguish between the following two cases:

- Each $\mathbf{x}_i$ is drawn uniformly at random (u.a.r.) from $\mathbb{Z}_q^n$ and there is a hidden secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ such that $y_i = \langle \mathbf{x}_i, \mathbf{s} \rangle + z_i$, where $z_i \in \mathbb{Z}_q$ is a discrete Gaussian noise independent of $\mathbf{x}_i$.
- Each $\mathbf{x}_i$ and each y_i are independent and are sampled u.a.r. from $\mathbb{Z}_q^n$ and $\mathbb{Z}_q$ respectively.

Formal definitions of LWE (Definition 2.2) and related distributions together with the exact hardness assumption (Assumption 2.3) are given in Section 2. Our main result can now be stated as follows:

Theorem 1.2 (Informal Main Theorem)**.** *Assume that LWE cannot be solved in* $2^{n^{1-\Omega(1)}}$ *time. Then, for any* $\zeta > 0$*, there is no polynomial-time learning algorithm for Massart halfspaces on* $\mathbb{R}^N$ *that can output a hypothesis with 0-1 error smaller than* $\Omega(\eta)$*, even when* $\mathrm{OPT} \leq 2^{\log^{1-\zeta} N}$ *and the Massart noise parameter* η *is a small positive constant.*

We note that Theorem 1.2 rules out *improper learning*, where the learner is allowed to output any polynomially evaluatable hypothesis. As a corollary, it follows that the algorithm given in [DGT19] is essentially the best possible, even when assuming that OPT is almost inverse polynomially small. We also remark that this latter assumption is also nearly the best possible: if OPT is $o(\epsilon/N)$, then we can just draw $\Omega(N/\epsilon)$ samples and output any halfspace that agrees with these samples.

No computational hardness result was previously known for this problem. Recent work established (unconditional) hardness in the Statistical Query (SQ) model [Kea98] for exact [CKMY20] and, subsequently, approximate learning [DK21, NT22]. While the SQ model is quite broad, it is also restricted. That is, these prior results do not have any implications for the class of all polynomial-time algorithms. Interestingly, as we will explain in more detail below, our computational hardness reduction is inspired by the SQ-hard instances developed in [DK21].

1.2 Brief Technical Overview

Here we give a high-level overview of our approach. We will sometimes be informal; all details will be formalized later in the paper. Our reduction proceeds in two steps. The first is to reduce the standard LWE problem (as described above) to a different "continuous" LWE problem more suitable for our purposes. In particular, we consider the problem where the $\mathbf{x}$ samples are taken uniformly from $\mathbb{R}^n/\mathbb{Z}^n$, y is either taken to be an independent random element of $\mathbb{R}/\mathbb{Z}$ or is taken to be $\langle \mathbf{x}, \mathbf{s} \rangle$ mod 1 plus a small amount of (continuous) Gaussian noise, where $\mathbf{s}$ is some unknown vector in $\{\pm 1\}^n$. The reduction between these problems follows from existing techniques [Mic18, GVV22].

The second step — which is the main technical contribution of our work —is reducing this continuous LWE problem to that of learning halfspaces with Massart noise. The basic idea is to perform a rejection sampling procedure that allows us to take LWE samples $(\mathbf{x}, y)$ and produce some new samples $(\tilde{\mathbf{x}}, \tilde{y})$. We will do this so that if y is independent of $\mathbf{x}$, then $\tilde{y}$ is (nearly) independent of $\tilde{\mathbf{x}}$; but if $y = \langle \mathbf{x}, \mathbf{s} \rangle +$ noise, then $\tilde{y}$ is a halfspace of $\tilde{\mathbf{x}}'$ with a small amount of Massart noise. An algorithm capable of learning halfspaces with Massart noise (with appropriate parameters) would be able to distinguish these cases by learning a hypothesis h and then looking at the probability that $h(\tilde{\mathbf{x}}) \neq \tilde{y}$. In the case where $\tilde{y}$ was a halfspace with noise, this would necessarily be small; but in the case where $\tilde{\mathbf{x}}$ and $\tilde{y}$ were independent, it could not be.

In order to manage this reduction, we will attempt to produce a distribution $(\tilde{\mathbf{x}}, \tilde{y})$ similar to the SQ-hard instances of Massart halfspaces from [DK21]. These instances can best be thought of as

instances of a random variable $(\mathbf{x}', y')$ in $\mathbb{R}^n \times \{\pm 1\}$, where y' is given by a low-degree polynomial threshold function (PTF) of $\mathbf{x}'$ with a small amount of Massart noise. Then, letting $\tilde{\mathbf{x}}$ be the Veronese map applied to $\mathbf{x}'$, we see that any low-degree polynomial in $\mathbf{x}'$ is a linear function of $\tilde{\mathbf{x}}$, and so $\tilde{y} = y'$ is an LTF of $\tilde{\mathbf{x}}$ plus a small amount of Massart noise.

As for how the distribution over $(\mathbf{x}', y')$ is constructed in [DK21], essentially the conditional distribution of $\mathbf{x}'$ on $y' = 1$ and on $y' = -1$ are carefully chosen mixtures of discrete Gaussians in the $\mathbf{v}$-direction (for some randomly chosen unit vector $\mathbf{v}$), and independent standard Gaussians in the orthogonal directions. Our goal will be to find a way to perform rejection sampling on the distribution $(\mathbf{x}, y)$ to produce a distribution of this form.

In pursuit of this, for some small real number b and some $a \in [0, b)$, we let $\mathbf{x}'$ be a random Gaussian subject to $\mathbf{x}' \equiv b\mathbf{x} \pmod{b}$ (in the coordinate-wise sense) conditioned on $by \equiv a \pmod{b}$. We note that if we ignore the noise in the definition of y, that this implies that $\langle \mathbf{x}', \mathbf{s} \rangle \equiv \langle b\mathbf{x}, \mathbf{s} \rangle \equiv b \langle \mathbf{x}, \mathbf{s} \rangle \equiv by \equiv a \pmod{b}$ (recall $\mathbf{s} \in \{\pm 1\}^n$). In fact, it is not hard to see that the resulting distribution on $\mathbf{x}'$ is close to a standard Gaussian conditioned on $\langle \mathbf{x}', \mathbf{s} \rangle \equiv a \pmod{b}$. In other words, $\mathbf{x}'$ is close to a discrete Gaussian with spacing $b/\|\mathbf{s}\|_2$ and offset $a/\|\mathbf{s}\|_2$ in the $\mathbf{s}$-direction, and an independent standard Gaussian in orthogonal directions. Furthermore, this $\mathbf{x}'$ can be obtained from $(\mathbf{x}, y)$ samples by rejection sampling; taking many samples until one is found with $by \equiv a \pmod{b}$, and then returning a random $\mathbf{x}'$ with $\mathbf{x}' \equiv b\mathbf{x} \pmod{b}$. By taking an appropriate mixture of these distributions, we can manufacture a distribution close to the hard instances in [DK21].

Unfortunately, applying this construction directly does not quite work. This is because the small noise (in the definition of y) leads to a small amount of noise in the final values of $\langle \mathbf{x}', \mathbf{s} \rangle$. This gives us distributions that are fairly similar to the hard instances of [DK21], but leads to small regions of values for $\mathbf{u}$, where $\mathbf{Pr}(y' = +1 | \mathbf{x}' = \mathbf{u}) = \mathbf{Pr}(y' = -1 | \mathbf{x}' = \mathbf{u})$, which cannot happen if y' is a function of $\mathbf{x}'$ with Massart noise. In order to fix this, we need to modify the construction slightly by "carving out" of the support of the conditional distribution of $\mathbf{x}'$ on $y' = -1$, in order to eliminate these mixed regions. This procedure is discussed in more detail in Section 3.3.2.

2 Preliminaries

For $\mathbf{x}, \mathbf{s} \in \mathbb{R}^n$ with $\mathbf{s} \neq \mathbf{0}$, let $\mathbf{x}^{\mathbf{s}} \stackrel{\text{def}}{=} \langle \mathbf{x}, \mathbf{s} \rangle / \|\mathbf{s}\|_2$ be the length of the projection of $\mathbf{x}$ in the $\mathbf{s}$ direction, and $\mathbf{x}^{\perp \mathbf{s}} \in \mathbb{R}^{n-1}$ be the projection[1] of $\mathbf{x}$ on the orthogonal complement of $\mathbf{s}$.

$X \sim D$ denotes a random variable X with distribution D. We use P_D or P_X for the corresponding probability mass function (pmf) or density function (pdf). Let $U(S)$ denote the uniform distribution on S. For a distribution D on $\mathbb{R}^n$ and $S \subseteq \mathbb{R}^n$, we denote by $D \mid S$ the conditional distribution of $X \sim D$ given $X \in S$. Let $D^{\mathbf{s}}$ (resp. $D^{\perp \mathbf{s}}$) be the distribution of $\mathbf{x}^{\mathbf{s}}$ (resp. $\mathbf{x}^{\perp \mathbf{s}}$) where $\mathbf{x} \sim D$. For distributions D_1, D_2, we use $D_1 + D_2$ to denote the pseudo distribution with the measure function $\mathbf{Pr}_{D_1 + D_2}(A) = \mathbf{Pr}_{D_1}(A) + \mathbf{Pr}_{D_2}(A)$. Similarly, for $a \in \mathbb{R}$, we use $a\,D$ to denote the pseudo distribution with measure function $a\mathbf{Pr}_D$. On the other hand, we use $a \circ D$ to denote the distribution of aX where $X \sim D$. We use $D_1 \star D_2$ to denote the convolution of distributions D_1, D_2.

We will also require the notion of a degree-d PTF. The class $\mathrm{PTF}_{N,d}$ consists of all functions $h_p : \mathbb{R}^N \to \{\pm 1\}$ of the form $h_{\mathbf{w}}(\mathbf{x}) := \mathrm{sign}(p(\mathbf{x}))$, where $p : \mathbb{R}^N \to \mathbb{R}$ is a degree-d polynomial. For $q \in \mathbb{N}$, we use $\mathbb{Z}_q \stackrel{\text{def}}{=} \{0, 1, \cdots, q-1\}$ and $\mathbb{R}_q \stackrel{\text{def}}{=} [0, q)$. We use $\mathrm{mod}_q : \mathbb{R}^n \mapsto \mathbb{R}_q^n$ to denote the function that applies $\mathrm{mod}_q(\mathbf{x})$ on each coordinate of $\mathbf{x}$.

We use $D^{\mathcal{N}}_{\mathbb{R}^n, \sigma}$ to denote the n-dimensional Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\sigma^2/(2\pi) \cdot \mathbf{I}_n$ and use $D^{\mathcal{N}}_{\sigma}$ as a short hand for $D^{\mathcal{N}}_{\mathbb{R}, \sigma}$.

Definition 2.1 (Partially Supported Gaussian Distribution)**.** For $\sigma \in \mathbb{R}_+$ and $\mathbf{x} \in \mathbb{R}^n$, let $\rho_\sigma(\mathbf{x}) \stackrel{\text{def}}{=} \sigma^{-n} \exp\left(-\pi(\|\mathbf{x}\|_2/\sigma)^2\right)$. For any countable set[2] $S \subseteq \mathbb{R}^n$, we let $\rho_\sigma(S) \stackrel{\text{def}}{=} \sum_{\mathbf{x} \in S} \rho_\sigma(\mathbf{x})$, and let $D^{\mathcal{N}}_{S,\sigma}$ be the distribution supported on S with pmf $P_{D^{\mathcal{N}}_{S,\sigma}}(\mathbf{x}) = \rho_\sigma(\mathbf{x})/\rho_\sigma(S)$.

LWE. We use the following definition of LWE, which allows for flexible distributions of samples, secrets, and noises. Here m is the number of samples, n is the dimension, and q is the ring size.

[1] More precisely, let $\mathbf{B}_{\perp \mathbf{s}} \in \mathbb{R}^{n \times (n-1)}$ for the matrix whose columns form an (arbitrary) orthonormal basis for the orthogonal complement of $\mathbf{s}$ and let $\mathbf{x}^{\perp \mathbf{s}} \stackrel{\text{def}}{=} (\mathbf{B}_{\perp \mathbf{s}})^T$.

[2] We will take S to be shifts of lattices, guaranteeing that $\rho_{\sigma^2}(S)$ is finite and the distribution is well-defined.

Definition 2.2 (Generic LWE)**.** Let $m, n, q \in \mathbb{N}$, and let $D_{\text{sample}}, D_{\text{secret}}, D_{\text{noise}}$ be distributions on $\mathbb{R}^n, \mathbb{R}^n, \mathbb{R}$ respectively. In the $\text{LWE}(m, D_{\text{sample}}, D_{\text{secret}}, D_{\text{noise}}, \text{mod}_q)$ problem, we are given m independent samples $(\mathbf{x}, y)$ and want to distinguish between the following two cases:

(i) **Alternative hypothesis**: $\mathbf{s}$ is drawn from D_{secret}. Then, each sample is generated by taking $\mathbf{x} \sim D_{\text{sample}}, z \sim D_{\text{noise}}$ and letting $y = \text{mod}_q(\langle \mathbf{x}, \mathbf{s} \rangle + z)$.

(ii) **Null hypothesis**: $\mathbf{x}, y$ are independent and each has the same marginal distribution above.

When a distribution in LWE is uniform over some set S, we may abbreviate $U(S)$ merely as S. Note that $\text{LWE}(m, \mathbb{Z}_q^n, \mathbb{Z}_q^n, \text{mod}_q(D^{\mathcal{N}}_{\mathbb{Z},\sigma}), \text{mod}_q)$ is equivalent to the classical LWE problem.

Assumption on Hardness of LWE. As alluded to earlier, the starting assumption for our hardness result is the hardness of the (classic) LWE problem, with the parameter as stated below.

Assumption 2.3 (Standard LWE Assumption (see, e.g., [LP11])**.** *Let $c > 0$ be a sufficiently large constant. For any $\beta \in (0, 1)$, $\kappa \in \mathbb{N}$, $\text{LWE}(2^{O(n^\beta)}, \mathbb{Z}_q^n, \mathbb{Z}_q^n, D^{\mathcal{N}}_{\mathbb{Z},\sigma}, \text{mod}_q)$ with $q \leq n^\kappa$ and $\sigma = c\sqrt{n}$ cannot be solved in $2^{O(n^\beta)}$ time with $2^{-O(n^\beta)}$ advantage.*

[Reg09, Pei09] gave a polynomial-time *quantum* reduction from approximating the Shortest Vector Problem (GapSVP) to LWE (with similar n, q, σ parameters). While our hardness assumption requires *sub-exponential* hardness of LWE, this is widely believed; the fastest known algorithm for GapSVP takes $2^{O(n)}$ time [ALNS20]. Thus, refuting the conjecture would be a major breakthrough. A similar assumption was also used in [GVV22] to establish hardness of learning Gaussian mixtures. Our use of a sub-exponential hardness of LWE is not a coincidence; see Section 4.

As mentioned earlier, we need a different version of LWE where the sample is from $\mathbb{R}_1^n$, the secret is from $\{\pm 1\}^n$ and the noise is drawn from a continuous Gaussian distribution. Its hardness is stated below; the proof, which follows mostly from [Mic18, GVV22], is deferred to Appendix B.

Lemma 2.4. *Under Assumption 2.3, for any $\beta \in (0, 1)$ and $\gamma \in \mathbb{R}_+$, there is no $2^{O(n^\beta)}$ time algorithm to solve* $\text{LWE}\left(2^{O(n^\beta)}, \mathbb{R}_1^n, \{\pm 1\}^n, D^{\mathcal{N}}_{O(n^{-\gamma})}, \text{mod}_1\right)$ *with $2^{-O(n^\beta)}$ advantage.*

Decision Version of Massart Problem. We will prove hardness for the following decision version of learning Massart halfspaces.

Definition 2.5 (Testing LTFs with Massart Noise)**.** For $n, N \in \mathbb{Z}^+$, $\eta, \text{OPT} \in (0, 1/2)$, let $\text{Massart}(m, N, \eta, \text{OPT})$ denote the problem of distinguishing, given m i.i.d. samples from D on $\mathbb{R}^N \times \{\pm 1\}$, between the cases (i) D satisfies the Massart halfspace condition with noise parameter $\Theta(\eta)$ and $R_{\text{LTF}}(D) \leq \text{OPT}$, and (ii) the Bayes optimal classifier has $\Omega(\eta)$ error.

3 Reduction from LWE to Learning Massart Halfspaces

By Lemma 2.4, Assumption 2.3 implies the hardness of $\text{LWE}(m, \mathbb{R}_1^n, \{\pm 1\}^n, D^{\mathcal{N}}_\sigma, \text{mod}_1)$. We will reduce this problem to the Massart halfspace learning problem. The following condition encapsulates the relation between the underlying parameters appearing in the reduction.

Condition 3.1. *Let $n, m, m' \in \mathbb{N}$, $t, \epsilon, \sigma \in \mathbb{R}_+$, $\delta \in (0, 1)$, satisfy: (i) t/ϵ is a sufficiently large even integer, (ii) $\sigma \leq \sqrt{n}$, (iii) $\frac{1}{t\sqrt{n}} \geq \sqrt{c \log(n/\delta)}$, where c is sufficiently large, (iv) $(\frac{c'\epsilon}{c''t\sigma})^2 \geq \log(m'/\delta)$, where $c' > 0$ is a sufficiently small and $c'' > 0$ is sufficiently large.*

The main theorem of this work is stated below.

Theorem 3.2. *Let $n, m, m' \in \mathbb{N}$, $t, \epsilon, \sigma \in \mathbb{R}_+$, $\delta \in (0, 1)$ satisfy Condition 3.1 and $\eta > 0$ be sufficiently small. Moreover, assume that $m' = c(\epsilon/t)m$, where $c > 0$ is sufficiently small and $m(\epsilon/t)^2$ is sufficiently large, and $N = (n+1)^d$, where $d/(t/\epsilon)$ is sufficiently large. Suppose there is no $T + \text{poly}(m, N, \log(1/\delta))$ time algorithm for solving $\text{LWE}(m, \mathbb{R}_1^n, \{\pm 1\}^n, D^{\mathcal{N}}_\sigma, \text{mod}_1)$ with $\epsilon - O(\delta)$ advantage. Then there is no T time algorithm for solving $\text{Massart}(m', N, \eta, \text{OPT})$ with 2ϵ advantage, where $\text{OPT} = \exp(-\Omega(t^4/\epsilon^2))$.*

Before we continue, let us note that the theorem above, together with Lemma 2.4, can be easily used to prove our main hardness result (e.g., by plugging in $t = n^{-0.5-\Theta(\zeta)}$ and $\epsilon = \Theta(n^{-1.5})$). As such, we devote the remainder of the paper to giving an overview to the proof of Theorem 3.2.

The starting point of our computational hardness reduction is the family of SQ-hard instances developed in [DK21]. At a high-level, these instances are constructed using mixtures of "hidden direction" discrete Gaussian distributions, i.e., distributions that are discrete Gaussians in a hidden direction and continuous Gaussians on the orthogonal directions (see below for an overview).

In Section 3.1, we note that by using an appropriate rejection sampling procedure on the LWE samples (drawn from the alternative hypothesis), we obtain a distribution very similar to the "hidden direction discrete Gaussian". A crucial difference in our setting is the existence of a small amount of additional "noise". A natural attempt is to replace the discrete Gaussians in [DK21] with the noisy ones obtained from our rejection sampling procedure. This produces problems similar to the hard instances from [DK21]. Unfortunately, the extra noise in our construction means that the naive version of this construction will not work; even with small amounts of noise, the resulting distributions will not satisfy the assumptions of a PTF with Massart noise. In Section 3.2, we elaborate on this issue and the modifications we need to make to our construction in order to overcome it. In Section 3.3, we provide the complete construction of our Massart PTF hard instance.

Overview of the [DK21] SQ-hard Construction. [DK21] showed SQ-hardness for the following hypothesis testing version of the problem (which implies hardness for the learning problem): For an input distribution D on $\mathbb{R}^n \times \{\pm 1\}$, distinguish between the cases where D is a specific distribution D_{null} in which $\mathbf{x}$ and y are independent or where D belongs to a class of alternative hypothesis distributions $\mathcal{D}_{\mathrm{alternative}}$. In particular, for $D \in \mathcal{D}_{\mathrm{alternative}}$, y will be given by a low-degree PTF in $\mathbf{x}$ with a small amount of Massart noise. As we will be trying to reproduce it, it is important for us to understand this alternative hypothesis distribution. Each distribution in $\mathcal{D}_{\mathrm{alternative}}$ is parameterized by a hidden direction $\mathbf{s} \in S^{n-1}$. We will denote the corresponding distribution by $D_{\mathbf{s}}$. $D_{\mathbf{s}}$ is constructed so that $\mathbf{x}^{\perp \mathbf{s}} \sim D^{\mathcal{N}}_{\mathbb{R}^{n-1},1}$ is independent of $\mathbf{x}^{\mathbf{s}}$ and y. This means that we can specify $D_{\mathbf{s}}$ by describing the simpler distribution of $(\mathbf{x}^{\mathbf{s}}, y) \in \mathbb{R} \times \{\pm 1\}$. For $(\mathbf{x}^{\mathbf{s}}, y)$, we have that $y = +1$ with probability $1 - \eta$. The distributions of $\mathbf{x}^{\mathbf{s}}$ conditioned on $y = \pm 1$ are defined to be mixtures of discrete Gaussians as follows:

$$D_{\mathbf{x}^{\mathbf{s}}|(y=+1)} = \frac{1}{\epsilon} \int_0^{\epsilon} D^{\mathcal{N}}_{u+(t+u)\mathbb{Z},1} du \quad \text{and} \quad D_{\mathbf{x}^{\mathbf{s}}|(y=-1)} = \frac{1}{\epsilon} \int_{t/2}^{t/2+\epsilon} D^{\mathcal{N}}_{u+(t+u-t/2)\mathbb{Z},1} du \,. \tag{1}$$

As we noted, both $\mathbf{x}^{\mathbf{s}} \mid (y = +1)$ and $\mathbf{x}^{\mathbf{s}} \mid (y = -1)$ are mixtures of discrete Gaussians. Combining this with the fact that $\mathbf{x}^{\perp \mathbf{s}} \sim \mathcal{N}(n, \mathbf{I}_{n-1})$, this indicates that $\mathbf{x} \mid (y = +1)$ and $\mathbf{x} \mid (y = -1)$ are mixtures of "hidden direction discrete Gaussians" — with different period and shift for their support on the hidden direction. These conditional distributions were carefully selected to ensure that y is a Massart PTF of $\mathbf{x}$ with small error. To see why this is, first, notice that the support of $\mathbf{x}^{\mathbf{s}} \mid (y = +1)$ is $\bigcup_{i \in \mathbb{Z}} [it, it + (i+1)\epsilon]$, while the support of $\mathbf{x}^{\mathbf{s}} \mid (y = -1)$ is $\bigcup_{i \in \mathbb{Z}} [it + t/2, it + t/2 + (i+1)\epsilon]$; both supports are unions of intervals (indexed by i). Consider the implications of this for three different ranges of $\mathbf{x}^{\mathbf{s}}$:

1. For $\mathbf{x}^{\mathbf{s}} \in [-t/(2\epsilon), t/(2\epsilon)]$, the intervals have lengths in $[0, t/2]$; thus, the $+1$ intervals and the -1 intervals do not overlap at all.
2. For $\mathbf{x}^{\mathbf{s}} \in [-t/\epsilon, -t/(2\epsilon)) \cup (t/(2\epsilon), t/\epsilon]$, the intervals have lengths in $[t/2, t]$; thus, the $+1$ intervals and the -1 intervals overlap, so that their union covers the space. We note that in this case there are gaps between the $+1$ intervals; specifically, there are at most $O(t/\epsilon)$ such gaps.
3. For $\mathbf{x}^{\mathbf{s}} \in (-\infty, -t/\epsilon) \cup (t/\epsilon, \infty)$, the intervals have lengths in $[t, \infty)$, so the $+1$ intervals cover the space by themselves.

Consider the degree-$O(t/\epsilon)$ PTF $\mathrm{sign}(p(\mathbf{x}))$ such that $\mathrm{sign}(p(\mathbf{x})) = +1$ iff $\mathbf{x}^{\mathbf{s}} \in \bigcup_{i \in \mathbb{Z}} [it, it + (i+1)\epsilon]$. In particular, $\mathrm{sign}(p(\mathbf{x})) = 1$ for $\mathbf{x}$ in the support of the conditional distribution on $y = 1$. Note that the PTF $\mathrm{sign}(p(\mathbf{x}))$ has zero error in the first case; thus, its total 0-1 error is at most $\exp(-\Omega(t^2/\epsilon)^2)$. Moreover, since the probability of $y = 1$ is substantially larger than the probability of $y = -1$, it is not hard to see that for any x with $\mathrm{sign}(p(x)) = 1$ that $\mathbf{Pr}(y = 1 \mid \mathbf{x} = x) > 1 - O(\eta)$. This implies that y is given by $\mathrm{sign}(p(x))$ with Massart noise $O(\eta)$.

In Section 3.1, we show that by performing an appropriate rejection sampling procedure on the LWE samples from the alternative hypothesis, we obtain a distribution similar to the "hidden direction discrete Gaussian". This modified distribution has the property that on $\perp \mathbf{s}$, it is close to $D^{\mathcal{N}}_{\mathbb{R}^{n-1},1}$, and on the hidden direction $\mathbf{s}$, it is a discrete Gaussian plus a small amount of extra noise.

3.1 Basic Rejection Sampling Procedure

For the sake of intuition, we start with the following simple lemma:

Lemma 3.3. *Let* $(\mathbf{x}, y)$ *be a sample of the* $\mathrm{LWE}(m, \mathbb{R}_1^n, \{\pm 1\}^n, D_\sigma^{\mathcal{N}}, \mathrm{mod}_1)$ *from the alternative hypothesis case, let* y' *be any constant in* $[0, 1)$*, and let* $\mathbf{x}' \sim (1/\sigma_{\mathrm{scale}}) \circ D^{\mathcal{N}}_{x+\mathbb{Z}^n, \sigma_{\mathrm{scale}}} \mid (y = y')$ *. Then we have the following:*

(i) There is a "clean" distribution D' *on* $\mathbb{R}$ *such that for any* $u \in \mathbb{R}$*, the pmf of* D' *satisfies* $P_{D'}(u) = (1 \pm O(\delta)) P_{D^{\mathcal{N}}_{T(y'+\mathbb{Z}), \sigma_{\mathrm{signal}}}}(u)$*, and the distribution of* $\mathbf{x'^s}$ *is* $D_{\mathbf{x'^s}} = D' \star D^{\mathcal{N}}_{\sigma_{\mathrm{noise}}}$*, where* $T = \mathrm{SR}/(n^{1/2}\sigma_{\mathrm{scale}})$*,* $\sigma_{\mathrm{signal}} = \sqrt{\mathrm{SR}}$*,* $\sigma_{\mathrm{noise}} = \sqrt{1 - \mathrm{SR}}$*, and* $\mathrm{SR} = \frac{\sigma^2_{\mathrm{scale}}}{\sigma^2_{\mathrm{scale}} + \sigma^2/n}$*.*

(ii) $\mathbf{x}'^{\perp \mathbf{s}}$ *is "nearly independent" of* $\mathbf{x'^s}$*, namely for any* $l \in \mathbb{R}$ *and* $\mathbf{u} \in \mathbb{R}^{n-1}$ *we have that* $P_{\mathbf{x}^{\perp \mathbf{s}} | \mathbf{x}^{\mathbf{s}} = l}(\mathbf{u}) = (1 \pm O(\delta)) P_{D^{\mathcal{N}}_{\mathbb{R}^{n-1}, 1}}(\mathbf{u})$*.*

Lemma 3.3 is a special case of Lemma 3.5, which is the main lemma required for our proof. Note that the distribution of $\mathbf{x}'$ obtained from the above rejection sampling is very similar to the "hidden direction discrete Gaussian" used in [DK21]. The key differences are as follows: (i) on the hidden direction, $\mathbf{x'^s}$ is close to a discrete Gaussian plus extra Gaussian noise (instead of simply being a discrete Gaussian); and (ii) $\mathbf{x}'^{\perp \mathbf{s}}$ and $\mathbf{x'^s}$ are not perfectly independent. More importantly, by taking different values for y' and σ_{scale}, we can obtain distributions with the same hidden direction, but their discrete Gaussian on the hidden direction has different period (T) and shift (y').

Our goal will be to simulate the instances from [DK21] by replacing the hidden direction discrete Gaussians by the noisy versions that we obtain from this rejection sampling. We next discuss this procedure and see why a naive implementation of it does not produce a PTF with Massart noise.

3.2 Intuition for the Hard Instance

The natural thing to try is to simulate the conditional distributions from [DK21] by replacing the hidden direction discrete Gaussian terms in (1) with similar distributions obtained from rejection sampling. In particular, Lemma 3.3 says that we can obtain a distribution which is close to this hidden direction Gaussian plus a small amount of Gaussian noise. Unfortunately, this extra noise will cause problems for our construction.

Recall that the support of $\mathbf{x}^{\mathbf{s}} \mid (y = +1)$ was $\bigcup_{i \in \mathbb{Z}} [it, it + (i+1)\epsilon]$, and the support of $\mathbf{x}^{\mathbf{s}} \mid (y = -1)$ was $\bigcup_{i \in \mathbb{Z}} [it + t/2, it + t/2 + (i+1)\epsilon]$. With the extra noise, there is a decaying density tail in both sides of each $[it, it + (i+1)\epsilon]$ interval of the support of $\mathbf{x}^{\mathbf{s}} \mid (y = +1)$. The same holds for each interval in the support of $\mathbf{x}^{\mathbf{s}} \mid (y = -1)$. Recalling the three cases of these intervals discussed earlier, this leads to the following issue. In the second case, the intervals have length within $[t/2, t]$; thus, the intervals $[it, it + (i+1)\epsilon]$ and $[it + t/2, it + t/2 + (i+1)\epsilon]$ overlap, i.e., $it + (i+1)\epsilon \geq it + t/2$. On the right side of $[it, it + (i+1)\epsilon]$, in the support of $\mathbf{x}^{\mathbf{s}} \mid (y = +1)$, there is a small sub-interval where the corresponding decaying density tail matches the scale of the density of $[it + t/2, it + t/2 + (i+1)\epsilon]$, which is in the support of $\mathbf{x}^{\mathbf{s}} \mid (y = -1)$. This causes the labels $y = +1$ and $y = -1$ to be equally likely over that sub-interval, violating the Massart condition.

We can address this issue by carving out empty spaces in the $[it + t/2, it + t/2 + (i+1)\epsilon]$ intervals for $\mathbf{x}^{\mathbf{s}} \mid (y = -1)$, so that these decaying parts can fit into. Since this only needs to be done for intervals of Case 2, at most $O(t/\epsilon)$ many such slots are needed. Note that no finite slot will totally prevent this from occurring; but we only need the slot to be wide enough that the decay of the error means that there is only a negligible amount of mass (which can be treated as an error) in the overlap.

We also need to discuss another technical detail. In the last section, we defined the rejection sampling process as taking $(1/\sigma_{\mathrm{scale}}) \circ D^{\mathcal{N}}_{\mathbf{x}+\mathbb{Z}^n, \sigma_{\mathrm{scale}}} \mid (y = y')$, where we can control the shift by y' and period by σ_{scale}. However, the extra noise has scale $\sigma^2_{\mathrm{noise}} = 1 - \mathrm{SR}$, which has dependence on σ_{scale}. If we use this sampling process, we will get smaller noise rate (σ_{noise}) for the noisy discrete Gaussian with smaller period, and larger noise rate for the one with larger period. For the ease of analysis, we instead consider $(1/\sigma_{\mathrm{scale}}) \circ D^{\mathcal{N}}_{\mathbf{x}+\mathbf{x}_{\mathrm{add}}+\mathbb{Z}^n, \sigma_{\mathrm{scale}}} \mid (y = y')$, where $\mathbf{x}_{\mathrm{add}} \sim D^{\mathcal{N}}_{\mathbb{R}^n, \sigma_{\mathrm{add}}}$ is independent

noise. The purpose of this extra term term is to compensate for the noise rate, so it can remain constant independent of the period.

The last bit of technical detail is that instead of doing the rejection for $y = y'$ which has 0 acceptance probability, we will only reject if y is not corresponding to any discrete Gaussian we need. Then we do another rejection to make sure the magnitude of discrete Gaussians in the mixture is correct. In the next subsection, we introduce the complete rejection sampling method.

3.3 The Full Hard Instance Construction

We first introduce the complete rejection algorithm and then explain how the hard instance is produced using it. Below we provide proof overviews; omitted proofs can be found in Appendix C.

3.3.1 The Complete Rejection Algorithm

The rejection sampling algorithm is the following. The sampling process produces the noisy variant of the clean distribution which for some carefully selected set $B \subseteq [0, 1]$ has PDF function $\frac{1}{\lambda(B)} \int_B D^{\mathcal{N}}_{k+(t+k-\psi)\mathbb{Z},1} dk$ in the hidden direction, as we will see in Lemma 3.5.

Algorithm 1 Rejection Sampling Algorithm

Inputs: A sample $(\mathbf{x}, y) \in \mathbb{R}^n_1 \times \mathbb{R}_1$ and the input parameters are $t, \epsilon, \psi \in \mathbb{R}_{>0}$, where $\psi + \epsilon \leq t$, $B \subseteq [\psi, \psi + \epsilon]$, $\delta \in (0, 1)$. In addition, the parameters satisfy items (i)-(iii) of Condition 3.1.
Output: REJECT or a sample $\mathbf{x}' \in \mathbb{R}^n$.

1. Reject unless there is a $k \in B$ such that $y = \frac{k}{t+k-\psi}$.
2. Furthermore, reject with probability $1 - \frac{t^2}{(t+k-\psi)^2}$.
3. Let $\mathrm{SR} = 1 - 4(t + \epsilon)^2\sigma^2, \sigma_{\mathrm{scale}} = \frac{\mathrm{SR}}{(t+k-\psi)\sqrt{n}}$ and $\sigma_{\mathrm{add}} = \sqrt{\frac{(1-\mathrm{SR})\sigma^2_{\mathrm{scale}} - \mathrm{SR}(\sigma/\sqrt{n})^2}{\mathrm{SR}}}$. Then, sample independent noise $\mathbf{x}_{\mathrm{add}} \sim D^{\mathcal{N}}_{\mathbb{R}^n,\sigma_{\mathrm{add}}}$ and output $\mathbf{x}' \sim (1/\sigma_{\mathrm{scale}}) \circ D^{\mathcal{N}}_{\mathbf{x}+\mathbf{x}_{\mathrm{add}}+\mathbb{Z}^n,\sigma_{\mathrm{scale}}}$.

Notice that the parameter SR does *not* depend on y, whereas $\sigma_{\mathrm{scale}}, \sigma_{\mathrm{add}}$ do depend on y.

For convenience, let us use the following notations for the output distributions.

Definition 3.4 (Output Distribution of Rejection Sampling)**.** Let $D^{\mathrm{alternative}}_{t,\epsilon,\psi,B,\delta}$ be the distributions of $\mathbf{x}'$ produced by Algorithm 1 (conditioned on the algorithm accepts[3]) given that $(\mathbf{x}, y)$ are sampled: let $\mathbf{x} \sim U(\mathbb{R}^n_1), z \sim D^{\mathcal{N}}_{\sigma}$ and then let $y = \mathrm{mod}_1(\langle \mathbf{x}, \mathbf{s}\rangle + z)$, where $\mathbf{s} \in \{\pm 1\}^n$ is the secret. Furthermore, let $D^{\mathrm{null}}_{t,\epsilon,\psi,B,\delta}$ be a similar distribution but when $\mathbf{x} \sim U(\mathbb{R}^n_1), y \sim U(\mathbb{R}_1)$ are independent.

Note that $D^{\mathrm{alternative}}_{t,\epsilon,\psi,B,\delta}$ depends on $\mathbf{s}$ but we do not explicitly denote this in our notations.

Alternative Hypothesis Analysis. The main properties of $D^{\mathrm{alternative}}_{t,\epsilon,\psi,B,\delta}$ are summarized below.

Lemma 3.5. *Let $\mathbf{x}' \sim D^{\mathrm{alternative}}_{t,\epsilon,\psi,B,\delta}$. Then, we have that:*

(i) For $\mathbf{x}'^{\mathbf{s}}$, there is a "clean" distribution D' on $\mathbb{R}$ such that $P_{D'}(u) = (1 \pm O(\delta))P_Z(u)$, where $Z = \frac{1}{\lambda(B)} \int_B D^{\mathcal{N}}_{k+(t+k-\psi)\mathbb{Z},\sigma_{\mathrm{signal}}} dk$, and the distribution of $\mathbf{x}'^{\mathbf{s}}$ is $D_{\mathbf{x}'^{\mathbf{s}}} = D' \star D^{\mathcal{N}}_{\sigma_{\mathrm{noise}}}$, where $\sigma_{\mathrm{signal}} = \sqrt{\mathrm{SR}}$, and $\sigma_{\mathrm{noise}} = \sqrt{1 - \mathrm{SR}} = 2(t + \epsilon)\sigma$.

(ii) $\mathbf{x}'^{\perp\mathbf{s}}$ is "nearly independent" of $\mathbf{x}'^{\mathbf{s}}$; namely, for any $l \in \mathbb{R}$ and $\mathbf{u} \in \mathbb{R}^{n-1}$, we have that $P_{\mathbf{x}'^{\perp\mathbf{s}}|\mathbf{x}'^{\mathbf{s}}=l}(\mathbf{u}) = (1 \pm O(\delta))P_{D^{\mathcal{N}}_{\mathbb{R}^{n-1},1}}(\mathbf{u})$.

The crux of the proof is the observation that the two following processes of generating $(\mathbf{w}, \mathbf{y})$ result in the same (joint) distribution:

- Sample $\mathbf{x} \sim U(\mathbb{R}^n_1), \mathbf{x}_{\mathrm{add}} \sim D^{\mathcal{N}}_{\mathbb{R}^n,\sigma_{\mathrm{add}}}, z \sim D^{\mathcal{N}}_{\sigma}$ and let $\mathbf{w} \sim (1/\sigma_{\mathrm{scale}}) \circ D^{\mathcal{N}}_{\mathbf{x}+\mathbf{x}_{\mathrm{add}}+\mathbb{Z}^n,\sigma_{\mathrm{scale}}}, y = \mathrm{mod}_1(\langle \mathbf{x}, \mathbf{s}\rangle + z)$ (as in the $\mathbf{x}' \sim D^{\mathrm{alternative}}_{t,\epsilon,\psi,B,\delta}$).

[3] For brevity, we say that the algorithm *accepts* if it does not reject.

- Sample $\mathbf{x}_{\text{new}} \sim U(\mathbb{R}_1^n), z_{\text{new}} \sim D^{\mathcal{N}}_{\sqrt{\sigma^2+\|\mathbf{s}\|_2^2\sigma_{\text{add}}^2}}$ and let $\mathbf{w} \sim (1/\sigma_{\text{scale}}) \circ D^{\mathcal{N}}_{\mathbf{x}_{\text{new}}+\mathbb{Z}^n,\sigma_{\text{scale}}}, y = \text{mod}_1(\sigma_{\text{scale}}\|\mathbf{s}\|_2\mathbf{w}^{\mathbf{s}} + z_{\text{new}})$

The latter is quite simple. For example, observe that $\sigma_{\text{scale}}\mathbf{w}$ is simply distributed as $D^{\text{expand}}_{\mathbb{R}_1^n,\sigma_{\text{scale}}}$. Using Fact A.5 in the Appendix, $\mathbf{w}$ is point-wise close to a (continuous) Gaussian. Therefore, $\mathbf{w}'^{\perp\mathbf{s}}$ and $\mathbf{w}'^{\mathbf{s}}$ are nearly independent. Finally, observing that rejecting sampling based on y does not change the conditional distribution $\mathbf{w}'^{\perp\mathbf{s}} \mid \mathbf{w}'^{\mathbf{s}} = l$. This suffices to show property (ii).

As for property (i), we first note that the rejection sampling ensures that k (the parameter in Step 1 of Algorithm 1) is uniform over B. Recall also that $\mathbf{w}$ is pointwise close to a continuous Gaussian; this implies the same for $\mathbf{w}^{\mathbf{s}}$. In fact, $\mathbf{w}^{\mathbf{s}}$ is pointwise close to $D_1^{\mathcal{N}}$. Furthermore, when we condition on a particular fixed k (or equivalently fixed y), the noise term z_{new} can only take values $i-\sigma_{\text{scale}}\|\mathbf{s}\|_2\mathbf{w}^{\mathbf{s}}$, for some $i \in \mathbb{Z} + \frac{k}{t+(k-\psi)}$. Roughly speaking, we may view i as our "signal" and $\sigma_{\text{scale}}\|\mathbf{s}\|_2\mathbf{w}^{\mathbf{s}} - i$ as our "noise". The signal-to-noise ratio is thus $\sqrt{\frac{(\sigma_{\text{scale}}\sqrt{n})^2}{(\sigma_{\text{scale}}\sqrt{n})^2+(\sigma^2+\|\mathbf{s}\|_2^2\sigma_{\text{add}}^2)}} = \sqrt{SR}$, where the equality follows from $\|\mathbf{s}\|_2^2 = n$ and our choice of σ_{add}.

Null Hypothesis Analysis. For $D^{\text{null}}_{t,\epsilon,\psi,B,\delta}$, we can show that it is point-wise close to $D^{\mathcal{N}}_{\mathbb{R}^n,1}$:

Lemma 3.6. *For any $\mathbf{u} \in \mathbb{R}^n$, we have $P_{D^{\text{null}}_{t,\epsilon,\psi,B,\delta}}(\mathbf{u}) = (1 \pm O(\delta))P_{D^{\mathcal{N}}_{\mathbb{R}^n,1}}(\mathbf{u})$.*

The proof of this case is relatively simple: $\mathbf{x}, y$ are independent, so even after conditioning on a specific value of y, $\mathbf{x}$ remains uniformly distributed over $\mathbb{R}_1^n$. This means that $\text{mod}_1(\mathbf{x} + \mathbf{x}_{\text{add}})$ is uniformly random over $\mathbb{R}_1^n$ and the distribution $D^{\mathcal{N}}_{\mathbf{x}+\mathbf{x}_{\text{add}}+\mathbb{Z}^n,\sigma'_{\text{scale}}}$ simplifies to $D^{\text{expand}}_{\mathbb{R}_1^n,\sigma'_{\text{scale}}}$. At this point, an application of Fact A.5 in the appendix concludes the proof.

3.3.2 The Reduction Algorithm

With the rejection sampling algorithm (Algorithm 1) at our disposal, we can now give the full construction of the hard instance. We use $D_{t,\epsilon,\psi_+,B_+,\delta}$ for $\mathbf{x}|y = +1$, $D_{t,\epsilon,\psi_-,B_-,\delta}$ for $\mathbf{x}|y = -1$ (with a carefully chosen pair of (B_+, ψ_+) and (B_-, ψ_-), as we discussed in Subsection 3.2) and take a proper marginal distribution of y to build a joint distribution of $(\mathbf{x}, y)$. We introduce the reduction algorithm that given samples from our LWE problem (either from the null or alternative hypothesis) produces samples $(\mathbf{x}, y)$ i.i.d. form a joint distribution with the following properties:

1. If the input LWE problem is the null hypothesis, then $\mathbf{x}|y = +1$ and $\mathbf{x}|y = -1$ are close in total variation distance. Therefore no hypothesis for predicting y in terms of $\mathbf{x}$ can do much better than the best constant hypothesis.
2. If the input LWE problem is the alternative hypothesis, then the joint distribution of $(\mathbf{x}, y)$ we build is close to a distribution D that satisfies $O(\eta)$ Massart condition w.r.t a degree-$O(t/\epsilon)$ PTF, and there is a degree-$O(t/\epsilon)$ PTF with small error on D.

We formalize the idea in Subsection 3.2 here. For $\mathbf{x}|y = +1$, we will use $\psi_+ \stackrel{\text{def}}{=} 0$ and $B_+ \stackrel{\text{def}}{=} [0, \epsilon]$, this is equivalent to the [DK21] construction. For $\mathbf{x}|y = -1$, we take $\psi_- \stackrel{\text{def}}{=} t/2$ which is also the same as [DK21], but we will need to carve out the slots on B_-. First, we define a mapping $g(it + t/2 + b) = \frac{b}{i+1} + t/2$ where $i \in \mathbb{Z}$ and $b \in \mathbb{R}_t$. This mapping maps a location $it + t/2 + b$ to the corresponding place we need to carve out on B_-, which is defined in Algorithm 2. These intervals are chosen in the way so the decaying density part of $+1$ can fit in as we discussed in Subsection 3.2. Now we introduce the algorithm that reduces LWE to learning Massart PTFs.

The following observation is immediate from the algorithm.

Observation 3.7. *In the alternative (resp. null) hypothesis case, the output distribution of Algorithm 2, conditioned on not failing, is the same as m' i.i.d. samples drawn from $D^{\text{alternative}}_{\text{PTF}}$ (resp. $D^{\text{null}}_{\text{PTF}}$).*

Again, we define the output distributions of the algorithms in the two cases as follows:

Definition 3.8. Let $D^{\text{alternative}}_{\text{PTF}}$ be mixture of $D^{\text{alternative}}_{t,\epsilon,\psi_+,B_+,\delta}$ and $D^{\text{alternative}}_{t,\epsilon,\psi_-,B_-,\delta}$ with $+1$ and -1 labels and weights $1 - \eta$ and η respectively. Similarly, let Let $D^{\text{null}}_{\text{PTF}}$ be mixture of $D^{\text{null}}_{t,\epsilon,\psi_+,B_+,\delta}$ and $D^{\text{null}}_{t,\epsilon,\psi_-,B_-,\delta}$ with $+1$ and -1 labels and weights $1 - \eta$ and η respectively.

Algorithm 2 Reducing LWE to Learning PTFs with Massart Noise

Inputs: m samples from an instance of $\mathrm{LWE}(m, \mathbb{R}_1^n, \{\pm 1\}^n, \mathcal{N}_{\sigma^2}, \mathrm{mod}_1)$. The input parameters are $m' \in \mathbb{N}, t, \epsilon \in \mathbb{R}_{>0}$, $\delta \in (0,1)$, $\eta > 0$ be a sufficiently small value. In addition, the parameters satisfy Condition 3.1.
Output: m' many samples in $\mathbb{R}^n \times \{\pm 1\}$ or FAIL.

1. We take $\psi_+ = 0$, $B_+ = [0, \epsilon]$, $\psi_- = t/2$ and

$$\begin{aligned} B_- \stackrel{\text{def}}{=} & [t/2, t/2 + \epsilon] \\ & - \bigcup_{i=\frac{t}{2\epsilon}-1}^{\frac{t}{\epsilon}-1} g([it - 2c'\epsilon, it]) - \bigcup_{i=\frac{t}{2\epsilon}-1}^{\frac{t}{\epsilon}-1} g([it + (i+1)\epsilon, it + (i+1)\epsilon + 2c'\epsilon]) \\ & - \bigcup_{i=-\frac{t}{\epsilon}-1}^{-\frac{t}{2\epsilon}-1} g([it + (i+1)\epsilon - 2c'\epsilon, it + (i+1)\epsilon]) - \bigcup_{i=-\frac{t}{\epsilon}-1}^{-\frac{t}{2\epsilon}-1} g([it, it + 2c'\epsilon]) \ . \end{aligned}$$

2. Repeat the following m' times. If at any point the algorithm attempts to use more than m LWE samples from the input, then output FAIL.
   (a) With probability $1 - \eta$, repeat the following until Algorithm 1 accepts and output $\mathbf{x}'$: run Algorithm 1 with the next unused LWE sample from the input and parameters $t, \epsilon, \psi = \psi_+, B = B_+, \delta$. Add $(\mathbf{x}', +1)$ to the output samples.
   (b) With probability η, repeat the following until Algorithm 1 accepts and output $\mathbf{x}'$: run Algorithm 1 with the next unused LWE sample from the input and parameters $t, \epsilon, \psi = \psi_-, B = B_-, \delta$. Add $(\mathbf{x}', -1)$ to the output samples.

Alternative Hypothesis Analysis. We prove that there must be a degree-$O(t/\epsilon)$ PTF s.t. $D_{\mathrm{PTF}}^{\mathrm{alternative}}$ satisfies the $O(\eta)$ Massart noise condition w.r.t this PTF and this PTF has small error w.r.t. $D_{\mathrm{PTF}}^{\mathrm{alternative}}$.

Lemma 3.9. *There is a degree-$O(t/\epsilon)$ PTF $\mathrm{sign}(p(\mathbf{x}))$ such that:*

1. $\mathbf{E}_{(\mathbf{x},y)\sim D_{\mathrm{PTF}}^{\mathrm{alternative}}}[\mathrm{sign}(p(\mathbf{x})) \neq y] \leq \exp(-\Omega(t^4/\epsilon^2)) + O(\delta/m')$*; and*
2. $D_{\mathrm{PTF}}^{\mathrm{alternative}}$ *is $O(\delta/m')$ close in total variation distance to a distribution that satisfies the $O(\eta)$ Massart noise condition with respect to $\mathrm{sign}(p(\mathbf{x}))$.*

To prove this, we extend the proof of [DK21] to be able to handle the noise from the previous step. We start by truncating the noise so that it belongs to $[-c'\epsilon, c'\epsilon]$. Due to the last condition of parameters in Algorithm 2, the probability mass truncated is at most $O(\delta/m')$ which corresponds to these terms in the above lemma. Once we have considered the truncated version of the noise, our carving ensures that, at any point in the support of +1, the probability mass of +1 is at least $\Omega(1)$ times that of -1 (and vice versa). The remainder of the proof then works similarly to [DK21].

Putting Everything Together. Having reduced LWE to learning Massart PTFs , we can apply a Veronese mapping on the samples; this PTF becomes an LTF on the Veronese mapping. Since we use degree-$O(t/\epsilon)$ Veronese mapping, the dimension for the Massart LTF problem is $N = (n+1)^{O(t/\epsilon)}$.

4 Discussion

The fact that we need a *subexponential* hardness of LWE is not a coincidence: the known SQ lower bounds for learning Massart halfspaces [DK21, NT22] only rule out quasi-polynomial time algorithms. (In fact, this also holds in the much more challenging agnostic model [Dan16].) Therefore, if we only assume polynomial time/advantage hardness for LWE, our reduction would have to run in polynomial time. Such a reduction, assuming LWE to be exponentially hard, would also give candidate exponentially SQ-hard instances for learning Massart halfspaces, significantly improving on the aforementioned super-polynomial results [DK21, NT22].

References

[ALNS20] D. Aggarwal, J. Li, P. Q. Nguyen, and N. Stephens-Davidowitz. Slide reduction, revisited - filling the gaps in SVP approximation. In *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 274–295. Springer, 2020.

[Blu03] A. Blum. Machine learning: My favorite results, directions, and open problems. In *44th Symposium on Foundations of Computer Science (FOCS 2003)*, pages 11–14, 2003.

[CKMY20] S. Chen, F. Koehler, A. Moitra, and M. Yau. Classification under misspecification: Halfspaces, generalized linear models, and connections to evolvability. *CoRR*, abs/2006.04787, 2020. Appeared in NeurIPS'20.

[Coh97] E. Cohen. Learning noisy perceptrons by a perceptron in polynomial time. In *Proceedings of the Thirty-Eighth Symposium on Foundations of Computer Science*, pages 514–521, 1997.

[Dan16] A. Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the 48th Annual Symposium on Theory of Computing, STOC 2016*, pages 105–117, 2016.

[DGT19] I. Diakonikolas, T. Gouleakis, and C. Tzamos. Distribution-independent PAC learning of halfspaces with massart noise. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 4751–4762, 2019.

[DIK+21] I. Diakonikolas, R. Impagliazzo, D. M. Kane, R. Lei, J. Sorrell, and C. Tzamos. Boosting in the presence of massart noise. In *Conference on Learning Theory, COLT 2021*, volume 134 of *Proceedings of Machine Learning Research*, pages 1585–1644. PMLR, 2021.

[DK21] I. Diakonikolas and D. M. Kane. Near-optimal statistical query hardness of learning halfspaces with massart noise. *arXiv preprint arXiv:2012.09720*, 2021.

[DKT21] I. Diakonikolas, D. Kane, and C. Tzamos. Forster decomposition and learning halfspaces with noise. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 7732–7744, 2021.

[GVV22] A. Gupte, N. Vafa, and V. Vaikuntanathan. Continuous lwe is as hard as lwe & applications to learning gaussian mixtures. *arXiv preprint arXiv:2204.02550*, 2022.

[Kea98] M. J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for lwe-based encryption. In *Cryptographers' Track at the RSA Conference*, pages 319–339. Springer, 2011.

[MG02] D. Micciancio and S. Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, 2002.

[Mic18] D. Micciancio. On the hardness of learning with errors with binary secrets. *Theory Comput.*, 14(1):1–17, 2018.

[MN06] P. Massart and E. Nedelec. Risk bounds for statistical learning. *Ann. Statist.*, 34(5):2326–2366, 10 2006.

[MP68] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, MA, 1968.

[MR07] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[Nov62] A. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.

[NT22] R. Nasser and S. Tiegel. Optimal SQ lower bounds for learning halfspaces with massart noise. *CoRR*, abs/2201.09818, 2022.

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, 2009*, pages 333–342. ACM, 2009.

[Reg09] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[Ros58] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[Slo88] R. H. Sloan. Types of noise in data for concept learning. In *Proceedings of the First Annual Workshop on Computational Learning Theory*, COLT '88, pages 91–96, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

[Val84] L. G. Valiant. A theory of the learnable. In *STOC*, pages 436–445. acm, 1984.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] *See Section 4.*
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] *See Assumption 2.3.*
   (b) Did you include complete proofs of all theoretical results? [Yes] *See Appendix.*
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]