# On the Sample Complexity of Privately Learning Half-spaces

**Yi-Hsiang Huang** shawn880421.cs11@nycu.edu.tw
**Wei-Hong Chen** a34123211@gmail.com
**Shi-Chun Tsai**[*] sctsai@nycu.edu.tw
*Computer Science Department, National Yang Ming Chiao Tung University, Hsinchu, Taiwan*

**Editors:** Vu Nguyen and Hsuan-Tien Lin

## Abstract

We present a differentially private learner for half-spaces over a finite domain $\mathcal{X}^d \subseteq \mathbb{R}^d$ with sample complexity $\tilde{O}(d \cdot \log^* |\mathcal{X}|)$, which improves over $\tilde{O}(d^{2.5} \cdot 8^{\log^* |\mathcal{X}|})$, the state-of-the-art result of [Kaplan et al., 2020]. The building block of our result is the reformulation from privately learning half-spaces to a composition of privately learning thresholds.

**Keywords:** Machine Learning, Differential Privacy, Computational Learning Theory, Sample Complexity, Half-spaces

## 1. Introduction

Machine learning plays a pivotal role in modern technology by empowering systems to autonomously learn from data, recognize patterns, and make informed decisions with minimal human intervention, etc. However, as some of the applications involve sensitive individual data, models trained on such data could potentially disclose private information, raising concerns about data privacy protection. Therefore, an emerging demand is how to learn from datasets while preserving data privacy.

Dwork et al. (2006) proposed Differential Privacy for data privacy studies. Due to its robust theoretical framework and practical effectiveness in protecting individual privacy, it has become a popular paradigm in related research. Building on the foundation of Probably Approximately Correct (PAC) learning by Valiant (1984), Kasiviswanathan et al. (2011) introduced the concept of private learning. This area merges PAC learning with differential privacy to protect individual information within datasets. The core principle of differential privacy is to ensure minimal impact on the output when a single data point changes, which is inherently linked to stability. This connection highlights the natural synergy between learnability, privacy, and stability in machine learning algorithms Dwork et al. (2014); Alon et al. (2022). Private learning research tackles the challenge of balancing data usage (sample complexity) with achieving both privacy and accuracy. Stricter privacy often demands more data to maintain accuracy, increasing computational and logistical burdens. In this paper, we focus on the sample complexity of privately learning half-spaces, a fundamental task in computational learning theory that holds significant importance due to its versatility and wide-ranging applications across various domains. We prove the following as our main contribution in this work:

**Main result: (informal)** Given $\epsilon, \delta, \alpha, \beta \in (0, 1)$ and a finite domain $\mathcal{X}^d \subseteq \mathbb{R}^d$, a realizable sample with size $\tilde{O}(d \cdot \log^* |\mathcal{X}|)$, there is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner for $d$-dimensional half-spaces.

[*] This paper is supported in part by the National Science and Technology Council of Taiwan under grants NSTC 112-2221-E-A49-050-MY3 and 113-2634-F-A49-001-MBK.

## 1.1. Related Work

Kasiviswanathan et al. (2011) showed that any finite class $\mathcal{C}$ can be adequately learned with pure differential privacy, achieving a sample complexity bound of $O(\log|\mathcal{C}|)$. Based on this, Beimel et al. (2014) established a matching lower bound for proper pure-private learners, confirming the optimality of the sample complexity bound in this setting. Extending this line of work, Feldman and Xiao (2014) showed that similar bounds hold even for improper pure-private learners. In general, learning with pure differential privacy requires sample complexity proportional to the logarithm of the size of the hypothesis class.

### 1.1.1. Privately Learning Thresholds

Regarding the task of privately learning one-dimensional half-spaces, also known as thresholds, over a finite domain $\mathcal{X} \subset \mathbb{R}$, Beimel et al. (2013) showed that by relaxing the privacy guarantees from pure to approximate differential privacy, the upper bound can be significantly improved from $O(log|\mathcal{X}|)$ to $\tilde{O}(8^{\log^*|\mathcal{X}|})$. After that, a line of research dedicated to improving the sample complexity bound (Bun et al. (2015, 2018); Kaplan et al. (2020a); Bun et al. (2015); Cohen et al. (2023), where the current knowledge of the bound is $\tilde{\Theta}(\log^*|\mathcal{X}|)$. It demonstrates a gap between private and non-private settings, where the sample complexity of non-private learning is $\Theta(1)$.

### 1.1.2. Privately Learning Half-spaces

For privately learning half-spaces over $\mathcal{X}^d$, Beimel et al. (2019) presented an upper bound of $\tilde{O}(d^{4.5} \cdot 8^{\log^*|\mathcal{X}|})$ and significantly improved over previous bound of $O(d^2 \cdot \log|\mathcal{X}|)$ by Kasiviswanathan et al. (2011) in terms of the domain size $|\mathcal{X}|$. Their method is based on a reduction to the task of privately finding a point in the convex hull of a given dataset. Subsequently, Kaplan et al. (2020b) improved the bound to $\tilde{O}(d^{2.5} \cdot 8^{\log^*|\mathcal{X}|})$ with a different reduction to the task of privately solving the linear feasibility problem, which is the best-known upper bound. In contrast, learning with non-private setting only requires $\Theta(d)$.

## 1.2. Other Related Works

Based on different assumptions, there are several related works on the sample complexity. For example, a line of works are based on the large margin assumption (Blum et al. (2005); Bun et al. (2014); Bassily et al. (2014); Jain and Thakurta (2014)). Le Nguyen et al. (2020) showed that the sample complexity bound could depend on the margin instead of the dimension, and they proved that their bound is optimal in the setting. Dagan and Feldman (2020), Su et al. (2023) studied the task of PAC learning half-spaces under a different privacy constraint called non-interactive local differential privacy introduced by Kasiviswanathan et al. (2011) and Evfimievski et al. (2003), where they presented an algorithm with the bound on sample complexity linear in dimension. Ghazi et al. (2021) showed an upper bound on sample complexity polynomial in the Littlestone dimension of the concept class (Littlestone (1988)), which is a combinational measurement of online learnability.

### 1.3. Organizations

The rest of this paper is organized as follows. In section 2, we introduce notations and preliminaries. Section 3 shows our algorithm for privately learning half-spaces and its analysis. We conclude with some remarks in section 4.

## 2. Preliminaries

We use calligraphic letters to denote sets and boldface for vectors and matrices. For $\mathbf{x} = (x_1, x_2, ...x_d) \in \mathbb{R}^d$ and $\mathbf{x}' = (x'_1, x'_2, ..., x'_d) \in \mathbb{R}^d$, let $\langle \mathbf{x}, \mathbf{x}' \rangle = \sum_{i=1}^{d} x_i x'_i$ be the inner product of $x$ and $x'$ and $\|\mathbf{x}\|_2$ be the $l_2$-norm. Given a set $\mathcal{X}$, we denote $\mathcal{X}^*$ as the set of all possible multi-sets whose elements are taken from $\mathcal{X}$, and let $\mathcal{X}^n \in \mathcal{X}^*$ be a multi-set with size $n$. Given an integer $n$, $log^*(n)$ denotes the iterated logarithm, which is a way to express how many times it needs to take the logarithm of $n$ until it gets a value at most 1, i.e., $log^*(n) = 1 + log^*(log(n))$ if $n > 1$ and zero otherwise.

### 2.1. Private Learning

The concept of differential privacy is defined as follows.

**Definition 1 (Differential Privacy Dwork et al. (2006))** *A randomized algorithm $A$ with domain $\mathcal{X}$ is $(\epsilon, \delta)$-differentially private if for all $\mathcal{E} \subseteq Range(A)$ and for all neighboring datasets $D,\ D' \in \mathcal{X}^*$ such that:*

$$\Pr[A(D) \in \mathcal{E}] \leq \exp(\epsilon) \Pr[A(D') \in \mathcal{E}] + \delta.$$

If $\delta = 0$, then $A$ satisfies *pure*-differential privacy; otherwise, it satisfies *approximate*-differential privacy. Such a definition also yields a property that it is immune to any post-processing.

**Lemma 2 (Post-Processing Dwork et al. (2014))** *Consider some domains $\mathcal{X}, \mathcal{H}$ and $\mathcal{H}'$. Let $M : \mathcal{X} \to \mathcal{H}$ be an $(\epsilon, \delta)$-differentially private mechanism and $f : \mathcal{H} \to \mathcal{H}'$ be any arbitrary mapping. The concatenation $f \circ M$ is $(\epsilon, \delta)$-differentially private.*

The Laplace mechanism safeguards individual privacy in numerical data analysis by adding noise that hides the influence of any single participant.

**Definition 3 (Laplace Distribution)** *A random variable has the probability distribution $Lap(b)$ if its probability density function is $f(x) = \frac{1}{2b} exp(-\frac{|x|}{b})$, where $x \in \mathbb{R}$.*

**Definition 4 (Sensitivity)** *A function $f : \mathcal{X}^* \to \mathbb{R}^n$ has sensitivity $k$ if for every neighboring samples $S, S' \in X^*$, it holds that $|f(S) - f(S')|_1 \leq k$.*

**Lemma 5 (The Laplace Mechanism Dwork et al. (2006))** *Let $f : \mathcal{X}^* \to \mathbb{R}^n$ be a sensitivity-$k$ function. For an input dataset $D$, the mechanism $A$ that adds independent noise with distribution $Lap(\frac{k}{\epsilon})$ to each of the $n$ outputs of $f(D)$ preserves $\epsilon$-differential privacy.*

Consider a PAC learner $A$ (Valiant (1984)), that tries to learn a target concept $c \in \mathcal{C}$. We say that $A$ is a private learner if it also satisfies differential privacy with respect to its training data.

**Definition 6 (Private PAC Learning Kasiviswanathan et al. (2011))** *Let $A$ be an algorithm with input $S \in \mathcal{X}^n$. $A$ is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-PAC learner with sample complexity $n$ for a concept class $\mathcal{C}$ over $\mathcal{X}$ using hypothesis class $\mathcal{H}$ if*

1. *$A$ is $(\epsilon, \delta)$-differentially private;*

2. *$A$ is an $(\alpha, \beta, n)$-PAC learner with hypothesis class $\mathcal{H}$.*

If $\mathcal{H} \subseteq \mathcal{C}$, then $A$ is called a proper learner; otherwise, it is called an improper learner.

We use Empirical Learning to measure the performance of a trained model.

**Definition 7** *The empirical error of a hypothesis $h$ for a sample $S = (x_i, y_i)_{i=1}^n$ is defined as $error_S(h) = \frac{1}{n}|\{i : h(x_i) \neq y_i\}|$.*

**Definition 8 (Empirical Learning)** *An algorithm $A$ is an $(\alpha, \beta, n)$-empirical learner using hypothesis class $\mathcal{H}$ if for all distributions $D$ on $\mathcal{X} \times \{-1, 1\}$, given an input sample $S$ of size $n$ drawn i.i.d. from $D$, algorithm $A$ outputs a hypothesis $h \in \mathcal{H}$ satisfying $\Pr[error_S(h) \leq \alpha] \geq 1 - \beta$.*

This task is simpler to handle than standard PAC learning, a distributional error minimization task. Replacing PAC learning with this task does not lose generality, which is implied by the result of Bun et al. (2015).

## 2.2. A Private Learner for Interior Points Problem - TreeLog

The algorithm TreeLog was initially proposed by Kaplan et al. (2020a) for solving the interior points problem[1], with a sample complexity bound of $\tilde{O}((\log^* |\mathcal{X}|)^{1.5})$. Cohen et al. (2023) further improved it to $\tilde{O}(\log^* |\mathcal{X}|)$ by introducing a novel technique called Reorder-Slice-Compute paradigm. This paradigm provided a more refined analysis of privacy, overcoming the bottleneck caused by the advanced composition theorem Dwork et al. (2010) and hence eliminating the polynomial dependency on $\log^* |\mathcal{X}|$.

Although the interior point problem is trivial without privacy constraints (since any input point is a valid output), solving it with differential privacy is much more challenging. In particular, Bun et al. (2015) have shown that privately solving this problem is equivalent to privately learning thresholds (properly) [2]. We summarize the result about TreeLog as follows.

[1] An algorithm $A$ solves the interior point problem over a domain $\mathcal{X} \subseteq \mathbb{R}$ with sample complexity $n$ and failure probability at most $\beta$ if for every dataset $S \in \mathcal{X}^n$, $\Pr[\min S \leq A(S) \leq \max S] \geq 1 - \beta$, where the probability is taken over the coins of $A$.

[2] For $u \in \mathcal{X} \subseteq \mathbb{R}$, a threshold is defined as $h_u : \mathcal{X} \to \{-1, 1\}$ such that for $x \in \mathcal{X}$, $h_u(x) = 1$ if $x \leq u$, and $-1$ otherwise. An algorithm learns thresholds if given a sample $(\mathcal{X} \times \{-1, 1\})^n$ , it outputs a hypothesis $h$ such that with probability at least $1 - \beta$, $error_S(h)$ is at most $\alpha$.

**Lemma 9 (TreeLog Cohen et al. (2023))** *For any privacy parameters $\epsilon, \delta \in (0,1)$, any finite and totally ordered domain $\mathcal{X}$, any desired accuracy parameters $\alpha, \beta \in (0,1)$, given a realizable sample of size*

$$n = O(\frac{\log^* |\mathcal{X}| \cdot \log^2 (\frac{\log^* |\mathcal{X}|}{\beta\delta})}{\alpha\epsilon}),$$

*there is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner that learns thresholds.*

## 3. Privately Learning Half-spaces

Following Kaplan et al. (2020b), we consider the task of privately learning half-spaces over a finite space $\mathcal{X}^d$, where $\mathcal{X} = \{x \in \mathbb{Z} : |x| \leq X\}$ for some $X \in \mathbb{N}$. We define the half-space as follows.

**Definition 10 (Half-space)** *For $\mathbf{a} = (a_1, a_2, ..., a_d) \in \mathbb{R}^d \setminus (0, 0, ..., 0)$ and $b \in \mathbb{R}$, a half-space is denoted as $h_{\mathbf{a},b} : \mathcal{X}^d \to \{-1, 1\}$ where $h_{\mathbf{a},b}(\mathbf{x}) = 1$ if $\langle \mathbf{a}, \mathbf{x} \rangle + b \geq 0$, and $-1$ otherwise.*

Without loss of generality, we assume that $b = 0$ and denote a half-space as $h_{\mathbf{a}}$, where every half-space passes through the origin [3]. Our goal is that given a realizable sample $S \in (\mathcal{X}^d \times \{-1, 1\})^*$ [4], the algorithm is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner with parameters $\epsilon, \delta, \alpha, \beta \in (0, 1)$.

Inspired by the construction of Sadigurschi and Stemmer (2021) on privately learning axis-aligned rectangles, we aim to reformulate the task to a composition of privately learning thresholds.

### 3.1. Learning Half-spaces In Two Dimensions

As defined above, each half-space can be indicated with a unit normal vector $\mathbf{n} = (n_1, n_2, ..., n_d)$, where $\|\mathbf{n}\|_2 = 1$. In the two-dimensional case, learning half-spaces can be reformulated as learning a 2-dimensional unit vector, where the set of all possible half-spaces forms a circle. Therefore, by considering the vector as rotating from a reference direction, the task of learning half-spaces can be reformulated as learning their corresponding angles. Formally, We transform a vector $\mathbf{n} = (n_1, n_2)$ into the corresponding polar coordinate $(r, \phi)$, where $r > 0$ is the distance from the origin and $\phi \in [0, 2\pi)$ is the rotate angle from the reference vector. With $\mathbf{n}$ being a unit vector, we have $r = 1$, and our target is to find the angle $\phi$. For convenience, we abuse the notation and denote a half-space as $h_\phi$.

The target is now an angle $\phi$ such that $h_\phi$ correctly classifies all elements in the sample $S$. To evaluate how close the angle we found to the target one is, we define a quality function as follows.

$$q(S, \phi) = |\{(\mathbf{x}, y) \in S \mid h_\phi(\mathbf{x}) = y\}|.$$

Intuitively, an approximate maximizer for this function implies a hypothesis that correctly classifies a significant proportion of the dataset. In addition, this function is suitable for private algorithms as we can verify that the sensitivity of the function is 1 (in terms of the

[3] When $b \neq 0$, we can consider it as the $(d+1)^{th}$ coordinate of $\mathbf{a}$, and extend each $\mathbf{x} \in \mathcal{X}^d$ to $(\mathbf{x}, 1)$.

[4] We say $S$ is a realizable sample if there exists a concept $c \in \mathcal{C}$ that is consistent with the sample.

first input). In particular, each element in $S$ can affect the resulting quality by at most 1. To this end, one might consider simply applying an optimization function for the task, while the continuous space of angles is not privately learnable (Alon et al. (2019)). Therefore, discretizing the angle domain becomes a crucial step in the reformulation process. We define the function as follows.

$$Discretize(\gamma) = \{i \cdot \gamma : i = 0, 1, ..., \lfloor 2\pi/\gamma \rfloor\}.$$

That is, Discretize takes an input angle $\gamma \in [0, 2\pi)$ and outputs a set of $(\lfloor 2\pi/\gamma \rfloor + 1)$ evenly spaced angles, denoted as $\mathcal{H}_\gamma$. We aim to set $\gamma$ appropriately for the discretization while preserving accuracy. In particular, We aim to guarantee that at least one of the half-spaces with the maximum quality is included in $\mathcal{H}_\gamma$.

**Lemma 11** *Given the domain $\mathcal{X} = \{x \in \mathbb{Z} : |x| \leq k\}$, consider the directional vectors passing through the origin and points in $\mathcal{X}^2$. The minimum (positive) angle $\gamma$ between any two non-collinear directional vectors satisfies* $\sin(\gamma) \geq 1/2k^2$*.*

**Proof** Let $\boldsymbol{v_1}, \boldsymbol{v_2}$ be two non-collinear directional vectors achieving the minimum angle $\gamma$, we have $\sin(\gamma) = ||\boldsymbol{v_1} \times \boldsymbol{v_2}||/(||\boldsymbol{v_1}|| \cdot ||\boldsymbol{v_2}||)$, where $\boldsymbol{v_1} \times \boldsymbol{v_2}$ is the cross product of the two vectors if we interpret $\boldsymbol{v}$ as $(x_1, x_2, 0)$. That is, we have $\sin(\gamma) = |det((\boldsymbol{v_1}, \boldsymbol{v_2}))|/(||\boldsymbol{v_1}|| \cdot ||\boldsymbol{v_2}||)$. Since the vectors contain only integer elements, the minimum positive value of the numerator is 1. Furthermore, the length of any vector $||\boldsymbol{v}||$ is at most $\sqrt{2}k$. Thereby, we have $\sin(\gamma) \geq 1/2k^2$. ■

Therefore, setting $\gamma$ as guaranteed in Lemma 11 is sufficient for preparing the dataset. ensures that for every pair of non-collinear directional vectors passing through the origin and a point in $\mathcal{X}^2$, there exists at least one half-space in $\mathcal{H}_\gamma$ that passes through these points. Moreover, given the assumption that the sample is realizable, this setting also guarantees the existence of half-spaces with the maximum quality in the resulting dataset. With the fact that $sin(\gamma) \approx \gamma$ for sufficiently small $\gamma$, the size of $\mathcal{H}_\gamma$ is $O(|\mathcal{X}|^2)$. Remark that to ensure the existence of half-spaces with the maximum quality in the domain, Beimel et al. (2019) used an arbitrary domain with the size of $O((d \cdot |\mathcal{X}|^{d^4})^{2^d})$. While Kaplan et al. (2020b) provided a complicated construction of domain with the size of $O(d^{d^d} \cdot |\mathcal{X}|^d)$ to achieve the same assurance. Therefore, our construction significantly reduces the size of the domain, enhancing the computational efficiency and scalability.

After obtaining the set of half-spaces $\mathcal{H}_\gamma$, we observe a nice property: each element in this set has a higher quality if it is closer to the one with the maximum quality in terms of angles. That is, with the assumption of data being realizable, all possible error points for any half-space can only be positioned between the half-space and the target one. Furthermore, this property is also possessed by thresholds. This gives a hint of the problem reformulation we aim to achieve.

Consider that we cut apart the circle formed by the set of half-spaces and flatten it. In this way, finding a half-space on the flattened line can be closely related to finding thresholds on a line, as we desired. Specifically, we can cut apart the circle by eliminating the elements with the lowest qualities, since they are adjacent and are dispensable without significantly compromising performance. However, this direct approach is not feasible as it can violate the privacy.

To see that, consider the scenario where the target half-space is aligned with the second axis. Given a sample $S = \{(\mathbf{x} = (0,1), 1)\}$, there are half of the half-spaces in $\mathcal{H}_\gamma$ that correctly classify the point $(\mathbf{x}, 1)$ and therefore have the quality value of 1, while the other half, denoted as $M$, have the quality value of 0 and hence are the elements to be eliminated. Suppose we increase the multiplicity of the point $(\mathbf{x}, 1)$ by 1, then the subset to be eliminated is unchanged. However, adding another point $((0,-1), 1)$ can increase the quality values of all but one half-space in $M$ by 1, resulting in a subset differing from $M$ in nearly half of the elements.

The result shows that with two neighboring input samples, the resulting sets of half-spaces are not necessarily neighboring. Therefore, we need to perform a preprocessing step to transform the set into a suitable input for learning thresholds while preserving privacy. Let $\phi(\mathbf{x})$ denote the angle corresponding to $\mathbf{x}$, and we define the function as follows.

```
Algorithm 1: $MakeData(\epsilon, \mathcal{H}_\gamma, S)$
Input: $\epsilon > 0, \mathcal{H}_\gamma \subseteq [0, 2\pi), S \in (\mathcal{X}^2 \times \{-1,1\})^*$
$S_\mathcal{H} \leftarrow \{\}$;
for $\phi \in \mathcal{H}_\gamma$ do
    $n_\phi = |\{(\mathbf{x}, y) \in S : |\phi(\mathbf{x}) - \phi| < \gamma \text{ and } h_\phi(\mathbf{x}) = y\}|$;
    add $max(\lceil n_\phi + Lap(\frac{1}{\epsilon})\rceil, 1)$ copies of $\phi$ to $S_\mathcal{H}$;
end
return $S_\mathcal{H}$;
```

**Lemma 12** *Algorithm 1 satisfies $(\epsilon, 0)$-differential privacy. Furthermore, there is at least one half-space (with the angle) $\phi^* \in S_\mathcal{H}$ with quality $q(S, \phi^*) = max_{\phi \in [0,2\pi)} q(S, \phi)$.*

The function builds a connection between the resulting dataset $S_\mathcal{H}$ and the input sample $S$ such that each half-space represents a portion of the input points. This connection enable us to transform the dataset without violating the privacy. We define the function as follows.

```
Algorithm 2: $MakeThrData(S_\mathcal{H}, S, C)$
Input: $S_\mathcal{H} \in ([0, 2\pi))^*, S \in (\mathcal{X}^2 \times \{-1,1\})^*, C \in \mathbb{N}$
Calculate $q(S, \phi)$ for every $\phi \in S_\mathcal{H}$;
Let $\max_C(S_\mathcal{H})$ be the $C$ largest elements in $S_\mathcal{H}$ according to the lexicographical order
  of $(q(S, \phi), \phi)$;
Randomly select $\phi' \in S_\mathcal{H} \setminus \max_C(S_\mathcal{H})$ and rotate the coordinate so that $\phi' = 0$;
Let $\phi^* := argmax_{\phi \in \max_C(S_\mathcal{H})}\{q(S, \phi)\}$;
$S_{Thr} \leftarrow \{\}$;
for $\phi \in \max_C(S_\mathcal{H})$ do
    $y \leftarrow 1$ if $\phi \leq \phi^*$; otherwise, $y \leftarrow -1$;
    add $(\phi, y)$ to $S_{Thr}$;
end
return $S_{Thr}$
```

The function first selects the $C$ largest elements from $S_\mathcal{H}$ according to their qualities. To ensure the selected subset $max_C(S_\mathcal{H})$ contains adjacent elements, angles are also taken into consideration. This ordering implies that the larger elements have higher quality and are closer to the target half-space simultaneously. This adjacency criterion is significant since the selected elements will later be reordered by randomly choosing an element from the subset $S_{\mathcal{H}} \setminus \max_C(S_{\mathcal{H}})$ to serve as the anchor for rotating the coordinate. Therefore, the elements are ordered properly such that the ones with lower quality are more distant from the threshold, or vice versa. Furthermore, by our construction, the threshold $\phi^*$ is a maximizer for the quality function $q$, which implies that an algorithm that approximately learns thresholds also approximately solves the original task. We summarize the properties of MakeThrData as follows.

**Lemma 13** *There is at least one labeled half-space (with the angle) $(\phi^*, y)$, $y \in \{-1, 1\}$ in the output of Algorithm 2 with quality $q(S, \phi^*) = max_{\phi \in [0,2\pi)} q(S, \phi)$.*

Let $A_{Thr}$ be any differentially private algorithm for learning thresholds over a domain $\mathcal{X}_{Thr} \subseteq \mathbb{R}$, we can learn 2-dimensional half-spaces as follows.

```
Algorithm 3: $A_{SimpleH}(\epsilon, \delta, \alpha, \beta, S, \gamma, A_{Thr})$
Input: $\epsilon, \delta, \alpha, \beta \in (0,1), S \in (\mathcal{X}^2 \times \{-1,1\})^*$, $\gamma \in [0, 2\pi)$, an $(\epsilon, \delta)$-differentially private
       $(\alpha, \beta)$-empirical learner $A_{Thr}$ that privately learns thresholds over $\mathcal{X}_{Thr}$ with
       sample complexity $n_{Thr} = n_{Thr}(\mathcal{X}_{Thr}, \epsilon, \delta, \alpha, \beta)$
$\mathcal{H}_\gamma \leftarrow Discretize(\gamma)$;
$S_{\mathcal{H}} \leftarrow MakeData(\epsilon, \mathcal{H}_\gamma, S)$;
$S_{Thr} \leftarrow MakeThrData(S_{\mathcal{H}}, S, n_{Thr})$;
Apply $A_{Thr}$ with input $S_{Thr}$, parameters $\epsilon, \delta, \alpha, \beta$ and get $\phi^*$;
return $h_{\phi^*}$
```

The following theorem shows the correctness of $A_{SimpleH}$.

**Theorem 14** *For any $\epsilon, \delta, \alpha, \beta \in (0,1)$, if there is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner $A_{Thr}$ that learns thresholds on a finite domain $\mathcal{X}_{Thr}$ with $n_{Thr}(\mathcal{X}_{Thr}, \epsilon, \delta, \alpha, \beta)$ samples, then with sample complexity*

$$n = O(n_{Thr}(\mathcal{X}_{Thr}, \frac{\epsilon}{2}, \delta, \alpha, \beta)),$$

*Algorithm 3 is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner for 2-dimensional half-spaces.*

Based on the above, we establish a connection between the task of privately learning half-spaces and thresholds. This connection is indeed valuable for addressing the known gap. For instance, with the construction and analysis above, we can apply the works on the task of privately learning thresholds (Lemma 9) to half-spaces immediately, which significantly improves the state-of-the-art result (Kaplan et al. (2020b)). Formally,

**Corollary 15** *For any privacy parameters $\epsilon, \delta \in (0,1)$, any desired accuracy parameters $\alpha, \beta \in (0,1)$, given a realizable sample with size*

$$n = O(\frac{\log^* |\mathcal{X}| \cdot \log^2 (\frac{\log^* |\mathcal{X}|}{\beta\delta})}{\alpha\epsilon}),$$

*there is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner $A : (\mathcal{X}^2 \times \{-1,1\})^n \rightarrow \mathcal{H}$ for 2-dimensional half-spaces.*

Since $\mathcal{X}_{Thr} \subseteq \mathcal{H}_\gamma$ according to our construction, we have $|\mathcal{X}_{Thr}| = O(|\mathcal{X}|^2)$. Therefore, the result follows by setting the privacy and accuracy parameters as Theorem 14 and applying $n_{Thr}$ as Lemma 9. The resulting sample complexity bound remains since the parameters only increase in a constant factor, which shows that we can actually reformulate the task of privately learning two-dimensional half-spaces as learning one-dimensional thresholds.

### 3.2. Learning Half-spaces In High Dimensions

We now generalize the above results to higher dimension $d > 2$. In the $d$-dimensional space, we can represent a unit vector $\mathbf{n} = (n_1, n_2, ...n_d) \in \mathbb{R}^d$ with another vector $(r, \phi_1, \phi_2, ...\phi_{d-1})$ in spherical coordinate, Blumenson (1960), where $r$ is the length of $\mathbf{n}$ and $\phi_i$ the angles relative to the reference vector of the corresponding coordinate. Formally,

$$\begin{aligned} &n_1 = r\cos(\phi_1), \\ &n_2 = r\sin(\phi_1)\cos(\phi_2), \\ &... \\ &n_d = r\sin(\phi_1)...\sin(\phi_{d-2})\sin(\phi_{d-1}). \end{aligned}$$

This transformation enables us to indicate any half-space in $d$-dimensional space by their angles $(\phi_1, \phi_2, ...\phi_{d-1})$. We next define a suitable quality function as above. Motivated by Beimel et al. (2019) and Kaplan et al. (2020b), for all $i \in \{1, ..., d-1\}$, define a quality function $Q_{\phi_1^*,...\phi_{i-1}^*}(S, \phi_i) := \max_{\tilde{\phi}_{i+1},...,\tilde{\phi}_{d-1} \in \mathcal{H}_\gamma} q(S, (\phi_1^*, ..., \phi_{i-1}^*, \phi_i, \tilde{\phi}_{i+1}, ..., \tilde{\phi}_{d-1}))$. Note that this definition differs from the previous ones: Beimel et al. (2019) maximizes the Tukey depth, and Kaplan et al. (2020b) maximizes the convexification of a quality function instead of the function itself. Moreover, instead of searching among a continuous space as before, our approach uses the quality function that searches within the discrete subspace $\mathcal{H}_\gamma$ and significantly reduces the computational complexity.

As mentioned by Kaplan et al. (2020b), the function $Q$ is not guaranteed to be quasi-concave. In order to overcome this issue, they presented a technique to 'convexify' a function, making it quasi-concave and thereby solvable by leveraging the method of Beimel et al. (2013). However, this reformulation of the function introduces additional errors, eventually increasing the sample complexity in terms of dimension $d$. We can avoid these issues by directly solving the optimization problem on $Q$ without reducing it to a quasi-concave optimization problem. In order to meet the privacy requirements, we first analyze the sensitivity of $Q$. Note that the proof of this statement differs from those in Beimel et al. (2019) and Kaplan et al. (2020b) as it requires no additional knowledge and is applicable to both of their constructions.

**Lemma 16** $Q_{\phi_1^*,...,\phi_{i-1}^*}(\cdot, \phi_i)$ *is a sensitivity-*$1$ *function in terms of the first input.*

**Proof** Consider two neighboring samples $S, S' = S \cup \{(\mathbf{x}, y)\}$. Given $\phi_1^*, ..., \phi_{i-1}^*$, suppose there exists a $\phi_i$ such that $Q_{\phi_1^*,...,\phi_{i-1}^*}(S', \phi_i) \geq Q_{\phi_1^*,...,\phi_{i-1}^*}(S, \phi_i) + 2$. Let $\phi_{i+1}, ..., \phi_{d-1}$ be the maximum completion of $Q_{\phi_1^*,...,\phi_{i-1}^*}(S, \phi_i)$, and $\phi'_{i+1}, ..., \phi'_{d-1}$ for $Q_{\phi_1^*,...,\phi_{i-1}^*}(S', \phi_i)$, respectively. By the definition of $Q$,

$$q(S', \phi_1^*, ..., \phi_{i-1}^*, \phi_i, \phi'_{i+1}, ..., \phi'_{d-1}) \geq q(S, \phi_1^*, ..., \phi_{i-1}^*, \phi_i, \phi_{i+1}, ..., \phi_{d-1}) + 2,$$

which implies

$$q(S', (\phi_1^*, ..., \phi_{i-1}^*, \phi_i, \phi_{i+1}', ..., \phi_{d-1}')) \geq q(S, (\phi_1^*, ..., \phi_{i-1}^*, \phi_i, \phi_{i+1}', ..., \phi_{d-1}')) + 2$$

since the latter value is at most that of $q(S, \phi_1^*, ..., \phi_{i-1}^*, \phi_i, \phi_{i+1}, ..., \phi_{d-1})$. With $q(S, \cdot)$ being a sensitivity-1 function in terms of the first input, the statement is proved by contradiction. ■

As we aim to leverage the aforementioned constructions and generalize them to the high-dimensional setting, we first show that the error introduced in each coordinate does not explode with the increased dimension.

**Lemma 17** *Consider* $\phi_1^*, ..., \phi_{i-1}^*$ *and* $\phi_i' = argmax_{\phi_i \in \mathcal{H}_\gamma} Q_{\phi_1^*,...,\phi_{i-1}^*}(S, \phi_i)$*, then*

$$Q_{\phi_1^*,...,\phi_{i-1}^*}(S, \phi_i') = max_{\tilde{\phi}_i, \tilde{\phi}_{i+1},...,\tilde{\phi}_{d-1} \in [0,2\pi)} q(S, (\phi_1^*, ..., \phi_{i-1}^*, \tilde{\phi}_i, \tilde{\phi}_{i+1}, ..., \tilde{\phi}_{d-1}).$$

**Proof** Let $\phi_{i+1}', ..., \phi_{d-1}'$ be the maximum completion for $\phi_i'$, and $\phi_{i+1}^*, ..., \phi_{d-1}^*$ for the target value $\phi_i^*$. Suppose that for some $j \in \{i+1, ..., d-1\}$, there exists a point $\mathbf{x} \in S$ between $\phi_j'$ and $\phi_j^*$ such that the half-space with angle $\phi_j'$ mis-classifies it. Then

$$q(S, (\phi_1^*, ..., \phi_{i-1}^*, \phi_i', \phi_{i+1}', ..., \phi_j', ..., \phi_{d-1}')) = q(S, (\phi_1^*, ..., \phi_{i-1}^*, \phi_i^*, \phi_{i+1}^*, ..., \phi_j^*, ..., \phi_{d-1}^*)) - 1.$$

However, by our construction, we can identify an angle $\tilde{\phi}_j \in \mathcal{H}_\gamma$, which is closer to $\phi_j^*$ such that there are no points between them. That is, if $\phi_j^*$ correctly classifies $\mathbf{x}$, then $\tilde{\phi}_j$ also correctly classifies it, where

$$q(S, (\phi_1^*, ..., \phi_{i-1}^*, \phi_i', \phi_{i+1}', ..., \phi_j', ..., \phi_{d-1}')) = q(S, (\phi_1^*, ..., \phi_{i-1}^*, \phi_i^*, \phi_{i+1}', ..., \tilde{\phi}_j, ..., \phi_{d-1}')) - 1.$$

Therefore, by the definition of $Q$, the claim is proved by contradiction. ■

With the quality function, our goal is to identify a half-space, coordinate by coordinate, which approximately maximizes the quality. Specifically, suppose we identify a value $\phi_1^*$ for the first coordinate, which approximately maximizes $Q$. By the definition of $Q$, this guarantees that there exists a completion $(\tilde{\phi}_2, ..., \tilde{\phi}_{d-1})$ such that $q(S, \phi_1^*, \tilde{\phi}_2, ..., \tilde{\phi}_{d-1})$ is close to the maximum quality. Thus, in every iteration, we find a value for the coordinate such that there is a completion in which we do not lose too much from the maximum attainable quality.

To generalize the previous result in 2-dimensional case, we review the functions defined above. Since Discretize has its construction independent of the input samples, the function remains unchanged and is actually applicable to each coordinate. For the function MakeData to be applicable for each iteration, let $\phi_i(\mathbf{x})$ be the $i$-th angle corresponding to $\mathbf{x}$, we modify the function as follows.

```
Algorithm 4: $MakeHighDimData(\epsilon, \mathcal{H}_\gamma, S, i)$
Input: $\epsilon > 0, \mathcal{H}_\gamma \subseteq [0, 2\pi), S \in (\mathcal{X}^d \times \{-1, 1\})^*, i \in \{1, 2, ..., d-1\}$
$S_\mathcal{H} \leftarrow \{\}$;
for $\phi \in \mathcal{H}_\gamma$ do
    $n_\phi = |\{(\mathbf{x}, y) \in S : |\phi_i(\mathbf{x}) - \phi| < \gamma \text{ and } h_\phi(\mathbf{x}) = y\}|$;
    add $\phi$ with $max(\lceil n_\phi + Lap(\frac{1}{\epsilon}) \rceil, 1)$ copies to $S_\mathcal{H}$;
end
return $S_\mathcal{H}$;
```

We summarize the properties of $MakeHighDimData$ as follows.

**Corollary 18** *Algorithm 4 satisfies $\epsilon$-differential privacy. Moreover, there is at least one half-space (with angle) $\phi_i^*$ in the dataset $S_{\mathcal{H}}$ with quality*

$$Q_{\phi_1^*,\dots,\phi_{i-1}^*}(S,\phi^*) = max_{\phi\in[0,2\pi)}(Q_{\phi_1^*,\dots,\phi_{i-1}^*}(S,\phi)).$$

The proof is similar to that of Lemma 12 with the guarantee of Lemma 17. With this modification, the function can be applied for each iteration and outputs the corresponding dataset of half-spaces for the coordinate.

Next, we modify MakeThrData as in Algorithm 5, where we introduce a constant parameter that determines the initial size of the output dataset and incorporates angles from previous iterations. Instead of employing the function $q$, we utilize $Q$ as the quality metric suitable for high-dimensional settings. Additionally, we introduce a noise addition step at Line 2. We also record the points in the original sample $S$ that corresponds to the dataset $S_{Thr}$.

```
Algorithm 5: $MakeHighDimThrData(\epsilon, S_{\mathcal{H}}, S, C, \phi_1^*, ..., \phi_{i-1}^*)$
Input: $\epsilon > 0, S_{\mathcal{H}} \in ([0,2\pi))^*, S \in (\mathcal{X}^d \times \{-1,1\})^*, C \in \mathbb{N}, \phi_1^*, ..., \phi_{i-1}^* \in \mathcal{H}_\gamma$
Calculate $Q_{\phi_1^*,\dots,\phi_{i-1}^*}(S,\phi_i)$ for every $\phi_i \in S_{\mathcal{H}}$;
Let $max_{\tilde{C}}(S_{\mathcal{H}})$ be the $C + Geom(1-e^{-\epsilon})$ largest elements in $S_{\mathcal{H}}$ according to the
  lexicographical order of $(Q_{\phi_1^*,\dots,\phi_{i-1}^*}(S,\phi_i), \phi_i)$;
$S_{Thr} = \emptyset$;
Randomly choose $\phi_i' \in S_{\mathcal{H}} \setminus \max_{\tilde{C}}(S_{\mathcal{H}})$ and rotate the coordinate so that $\phi_i' = 0$;
Let $\phi_i^* := argmax_{\phi_i \in \max_{\tilde{C}}(S_{\mathcal{H}})}\{Q_{\phi_1^*,\dots,\phi_{i-1}^*}(S,\phi_i)\}$;
for $\phi_i \in \max_{\tilde{C}}(S_{\mathcal{H}})$ do
    $y \leftarrow 1$ if $\phi_i \leq \phi^*$; otherwise, $y \leftarrow -1$;
    add $(\phi_i, y)$ to $S_{Thr}$;
end
Let $S_{\tilde{C}} \subseteq S$ be the points corresponding to the elements in $S_{Thr}$ without labels;
return $S_{Thr}, S_{\tilde{C}}$
```

The function is summarized as follows.

**Corollary 19** *There is at least one labeled half-space (with the angle) $(\phi_i^*, y), y \in \{-1,1\}$ in the output of Algorithm 5 with the quality function*

$$Q_{\phi_1^*,\dots,\phi_{i-1}^*}(S,\phi_i^*) = max_{\phi\in[0,2\pi)}(Q_{\phi_1^*,\dots,\phi_{i-1}^*}(S,\phi)).$$

The proof is similar to that of Lemma 13 with the guarantee of Lemma 17. Now we describe the generalized algorithm as follows.

```
Algorithm 6: $A_{HighH}(\epsilon, \delta, \alpha, \beta, S, \gamma, A_{Thr})$
Input: $\epsilon, \delta, \alpha, \beta \in (0,1), S \in (\mathcal{X}^d \times \{-1,1\})^*$, $\gamma \in [0, 2\pi)$, an $(\epsilon, \delta)$-differentially private
       $(\alpha, \beta)$-empirical learner $A_{Thr}$ that privately learns thresholds over $\mathcal{X}_{Thr}$ with
       sample complexity $n_{Thr}(\mathcal{X}_{Thr}, \epsilon, \delta, \alpha, \beta)$
$\mathcal{H}_\gamma \leftarrow Discretize(\gamma)$;
$S_0 \leftarrow S$;
for $i = 1, 2, ...d-1$ do
    $S_{\mathcal{H}} \leftarrow MakeHighDimData(\mathcal{H}_\gamma, S_{i-1}, i)$;
    $S_{Thr}, S_{\tilde{C}} \leftarrow MakeHighDimThrData(\epsilon, S_{\mathcal{H}}, S_{i-1}, n_{Thr}, \phi_1^*, ..., \phi_{i-1}^*)$;
    Apply $A_{Thr}$ with input $S_{Thr}$ and get $\phi_i^*$;
    $S_i \leftarrow S_{i-1} \setminus S_{\tilde{C}}$;
end
return $h_{(\phi_1^*, ..., \phi_{d-1}^*)}$
```

For the estimation of the total privacy cost, we adopt a paradigm called the Reorder-Slice-Compute paradigm proposed by Cohen et al. (2023). Let $Geom(p)$ be the geometric distribution with parameter $0 < p \leq 1$. Formally, $Pr[Geom(p) = k] = (1-p)^k p$ for integer $k \geq 0$. We apply a simplified version of the paradigm as follows.

```
Algorithm 7: $ReorderSliceCompute(\epsilon, \delta, S, \tau, m, E_1, ...E_\tau, A)$ (Cohen et al. (2023))
Input: $\epsilon, \delta \in (0,1), S \in (\mathcal{X})^n$, $\tau, m \in \mathbb{N}$, a sequence of sorters $E_1, ...E_\tau$, and an
       $(\epsilon, \delta)$-differentially private algorithm $A$
$S_0 = S$;
for $i = 1, 2, ...\tau$ do
    $\tilde{m} \leftarrow m + Geom(1 - e^{-\epsilon})$;
    $max_{\tilde{m}}(S_{i-1}) \leftarrow$ the largest $\tilde{m}$ elements in $E_i(S_{i-1})$;
    $S_i \leftarrow S_{i-1} \setminus max_{\tilde{m}}(S_{i-1})$;
    $r_i \leftarrow A(max_{\tilde{m}}(S_{i-1}))$;
end
return $r_1, ...r_\tau$
```

The main idea of this paradigm is that during the iterative process, it is possible to eliminate the single differing element and thereby reduce the privacy cost. In particular, let $S, S' = S \cup \{(x', y')\}$ be neighboring samples and we execute *ReorderSliceCompute* in parallel with the inputs. Let $S_i, S_i'$ be the respective datasets for the two executions during the $i$-th iteration. If during some iteration $j$, $x'$ is included in the subset $max_{\tilde{m}}(S'_{j-1})$ and the corresponding noises $w_j$ and $w_j'$ satisfy $w_j = w_j' - 1$, then the extra element $x'$ will be eliminated from $S'$. The remaining processes are now 'synchronized' and no more privacy cost introduced. Consequently, by carefully analyzing the privacy cost incurred by the differing element, the algorithm preserves privacy requirements as follows:

**Lemma 20 (Privacy of ReorderSliceCompute Cohen et al. (2023))** *For any $\delta' > 0$, Algorithm 7 is $(O(\epsilon log(1/\delta')), \delta' + 2\delta\tau)$-differentially private.*

This implies that the total privacy cost avoids the dependence on $\tau$ in terms of $\epsilon$, ensuring our improvement of the sample complexity without sacrificing the privacy parameters in higher dimensions. We prove the correctness of $A_{HighH}$ in Theorem 21.

**Theorem 21** *For any $\epsilon, \delta, \alpha, \beta \in (0,1)$ and $\delta' > 0$, if there is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner $A_{Thr}$ that learns thresholds on a finite domain $\mathcal{X}_{Thr}$ with sample complexity $n_{Thr}(\mathcal{X}_{Thr}, \epsilon, \delta, \alpha, \beta)$, then with sample complexity*

$$n = O(n_{Thr}(\mathcal{X}_{Thr}, O(\frac{\epsilon}{log(\frac{1}{\delta'})}), \frac{\delta - \delta'}{2(d-1)}, \frac{\alpha}{d-1}, \frac{\beta}{d-1})),$$

*Algorithm 6 is an $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner for d-dimensional half-spaces.*

While a direct estimation of privacy composition introduces a $\frac{1}{d^{0.5}}$ term to the privacy parameter, we avoid it with a tighter bound on the privacy cost induced by the differing element. With this result, we obtain the following:

**Corollary 22** *For any privacy parameters $\epsilon, \delta \in (0,1)$, any desired accuracy parameters $\alpha, \beta \in (0,1)$, given a realizable sample with size*

$$n = O(d \cdot \frac{\log^* |\mathcal{X}| \cdot \log^2 (\frac{d \cdot \log^* |\mathcal{X}|}{\beta\delta})}{\alpha\epsilon}),$$

*there is a $(\epsilon, \delta)$-differentially private $(\alpha, \beta)$-empirical learner $A : (\mathcal{X}^d \times \{-1, 1\})^n \to \mathcal{H}$ for d-dimensional half-spaces.*

To this end, we have improved the upper bound on the sample complexity of privately learning half-spaces over $\mathcal{X}^d$ to $\tilde{O}(d \cdot \log^* |\mathcal{X}|)$, where the remaining $d$ term is introduced by the setting of the error bound $\alpha$.

## 4. Conclusion

In this paper, we proposed a private learner for half-spaces over a finite grid $\mathcal{X}^d \in \mathbb{R}^d$ with sample complexity linear in both $d$ and $\log^* |\mathcal{X}|$, which improves the previous known bound $\tilde{O}(d^{2.5} \cdot 8^{\log^* |\mathcal{X}|})$ by Kaplan et al. (2020b), and answers an open problem raised by them. Moreover, this improves over the generic bound $O(d^2 \cdot \log |\mathcal{X}|)$ in terms of both the dimension and the domain size. However, the optimal sample complexity for learning half-spaces with (approximate) differential privacy remains open. Besides, via the known transformation from the realistic to the agnostic setting (Beimel et al. (2021), Alon et al. (2020)), the sample complexity increases up to $\tilde{O}(d \cdot \log^* |\mathcal{X}| + d^2)$. It is not clear whether this gap can be closed. Lastly, the dominant complexity of our algorithm is $O(d^2 \cdot \mathcal{X}^{2(d+1)} \cdot |S|)$, where the computation of the quality functions can be the bottleneck. We leave it as a future work.

## References

Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860, 2019.

Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. In *Conference on Learning Theory*, pages 119–152. PMLR, 2020.

Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and online learnability are equivalent. *ACM Journal of the ACM (JACM)*, 69(4):1–34, 2022.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 363–378. Springer, 2013.

Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94:401–437, 2014.

Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. In *Conference on Learning Theory*, pages 269–282. PMLR, 2019.

Amos Beimel, Kobbi Nissim, and Uri Stemmer. Learning privately with labeled and unlabeled examples. *Algorithmica*, 83:177–215, 2021.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.

LE Blumenson. A derivation of n-dimensional spherical coordinates. *The American Mathematical Monthly*, 67(1):63–66, 1960.

Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 1–10, 2014.

Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 634–649. IEEE, 2015.

Mark Bun, Cynthia Dwork, Guy N Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–86, 2018.

Edith Cohen, Xin Lyu, Jelani Nelson, Tamás Sarlós, and Uri Stemmer. Optimal differentially private learning of thresholds and quasi-concave optimization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 472–482, 2023.

Yuval Dagan and Vitaly Feldman. Interaction is necessary for distributed learning with privacy or communication constraints. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 450–462, 2020.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222, 2003.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, pages 1000–1019. PMLR, 2014.

Badih Ghazi, Noah Golowich, Ravi Kumar, and Pasin Manurangsi. Sample-efficient proper pac learning with approximate differential privacy. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 183–196, 2021.

Prateek Jain and Abhradeep Guha Thakurta. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pages 476–484. PMLR, 2014.

Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. In *Conference on Learning Theory*, pages 2263–2285. PMLR, 2020a.

Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. Private learning of half-spaces: Simplifying the construction and reducing the sample complexity. *Advances in Neural Information Processing Systems*, 33:13976–13985, 2020b.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Huy Le Nguyen, Jonathan Ullman, and Lydia Zakynthinou. Efficient private algorithms for learning large-margin halfspaces. In *Algorithmic Learning Theory*, pages 704–724. PMLR, 2020.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2:285–318, 1988.

Menachem Sadigurschi and Uri Stemmer. On the sample complexity of privately learning axis-aligned rectangles. *Advances in Neural Information Processing Systems*, 34:28286–28297, 2021.

Jinyan Su, Jinhui Xu, and Di Wang. On pac learning halfspaces in non-interactive local privacy model with public unlabeled data. In *Asian Conference on Machine Learning*, pages 927–941. PMLR, 2023.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.