

# Differentially Private Quasi-Concave Optimization: Bypassing the Lower Bound and Application to Geometric Problems

Kobbi Nissim\*

Eliad Tsfadia\*

Chao Yan\*

April 29, 2025

## Abstract

We study the sample complexity of differentially private optimization of quasi-concave functions. For a fixed input domain  $\mathcal{X}$ , Cohen et al. [9] (STOC 2023) proved that any generic private optimizer for low sensitive quasi-concave functions must have sample complexity  $\Omega(2^{\log^* |\mathcal{X}|})$ .

We show that the lower bound can be bypassed for a series of “natural” problems. We define a new class of *approximated* quasi-concave functions, and present a generic differentially private optimizer for approximated quasi-concave functions with sample complexity  $\tilde{O}(\log^* |\mathcal{X}|)$ . As applications, we use our optimizer to privately select a center point of points in  $d$  dimensions and *probably approximately correct* (PAC) learn  $d$ -dimensional halfspaces. In previous works, Bun et al. [7] (FOCS 2015) proved a lower bound of  $\Omega(\log^* |\mathcal{X}|)$  for both problems. Beigel et al. [4] (COLT 2019) and Kaplan et al. [17] (NeurIPS 2020) gave an upper bound of  $\tilde{O}(d^{2.5} \cdot 2^{\log^* |\mathcal{X}|})$  for the two problems, respectively. We improve the dependency of the upper bounds on the cardinality of the domain by presenting a new upper bound of  $\tilde{O}(d^{5.5} \cdot \log^* |\mathcal{X}|)$  for both problems. To the best of our understanding, this is the first work to reduce the sample complexity dependency on  $|\mathcal{X}|$  for these two problems from exponential in  $\log^* |\mathcal{X}|$  to  $\log^* |\mathcal{X}|$ .

---

\*Department of Computer Science, Georgetown University. E-mails: [kobbi.nissim@georgetown.edu](mailto:kobbi.nissim@georgetown.edu), [eliadtsfadia@gmail.com](mailto:eliadtsfadia@gmail.com), [cy399@georgetown.edu](mailto:cy399@georgetown.edu). Work is partially funded by NSF grant No. 2217678 and by a gift to Georgetown University.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Existing Results . . . . .	1
1.2	Our Results . . . . .	2
1.2.1	Private Optimization for Approximated Quasi-Concave Functions . . . . .	2
1.2.2	Applications . . . . .	3
1.3	Open Questions . . . . .	4
<b>2</b>	<b>Preliminaries</b>	<b>4</b>
2.1	Notations . . . . .	4
2.2	Learning Theory . . . . .	4
2.3	Differential Privacy . . . . .	5
2.4	Halfspaces . . . . .	5
2.5	Notation . . . . .	5
<b>3</b>	<b>Our Private Quasi-Concave Optimization Scheme</b>	<b>6</b>
3.1	One-Dimensional Case . . . . .	6
3.2	Extending to Higher Dimension . . . . .	7
<b>4</b>	<b>Differentially Private Tukey Median Approximation</b>	<b>9</b>
4.1	Additional Preliminaries for Tukey Median . . . . .	9
4.1.1	Domain Extension . . . . .	10
4.2	Privately Estimating the Tukey Median . . . . .	10
<b>5</b>	<b>Privately Learning Halfspaces via Privately Solving Linear Feasibility Problems</b>	<b>11</b>
5.1	Additional Preliminaries for Linear Feasibility Problems . . . . .	11
5.2	Privately Solving Linear Feasibility Problems . . . . .	13
5.3	Privately Learning Halfspaces . . . . .	14
<b>A</b>	<b>Approximated Functions Have Low Sensitivity</b>	<b>16</b>
<b>B</b>	<b>VC Dimension of Linear Feasibility Problem</b>	<b>16</b>
<b>C</b>	<b>Huang et al. [15]’s Algorithm is Not Differentially Private</b>	<b>17</b>

# 1 Introduction

The training of machine learning models often uses sensitive personal information that requires privacy protection. We explore privacy-preserving techniques in machine learning, a line of research initiated by Kasiviswanathan et al [18], where a probably approximately correct (PAC) learner is required to preserve differential privacy with respect to its training data. Differential privacy ensures that a privacy attacker would not be able to detect the presence of an input datum. Formally,

**Definition 1** (Differential Privacy [11]). *Let  $\mathcal{X}$  be a data domain and  $\mathcal{Y}$  be an output domain. A (randomized) mechanism  $M: \mathcal{X}^n \rightarrow \mathcal{Y}$  is  $(\epsilon, \delta)$ -differentially private if for any pair of neighboring datasets  $S, S' \in \mathcal{X}^n$  (i.e., datasets that differ on a single entry), and any event  $E \subseteq \mathcal{Y}$ , it holds that*

$$\Pr[M(S) \in E] \leq e^\epsilon \cdot \Pr[M(S') \in E] + \delta,$$

where the probability is over the randomness of  $M$ .

*Sample complexity*, i.e., the required dataset size for a learning task, is one of the main questions in learning theory. Without privacy requirements, it is well-known that the sample complexity of PAC learning a concept class  $C$  is  $\Theta(VC(C))$  [24], where  $VC(C) \leq \log |C|$  is the Vapnik–Chervonenkis dimension of  $C$ . Kasiviswanathan et al. [18] provide a generic private learner showing that a sample complexity  $O(\log |C|)$  suffices for private learning. The characterization of the sample complexity of private learning is still an open problem.

The main motivation of this work is to advance our understanding of the sample complexity of two fundamental geometric problems: privately finding a center point and privately learning halfspaces, both with input points over a finite Euclidean space  $\mathcal{X}^d$ . In particular, we focus on how the sample complexity depends on the cardinality of  $\mathcal{X}$ . Towards this goal, we revisit a primitive that was introduced in [5] and used in prior work on learning halfspaces: optimizing private quasi-concave functions.

## 1.1 Existing Results

Under *pure* differential privacy (i.e., with  $\delta = 0$ ), it is known that the sample complexity of privately learning halfspaces with input points over a finite  $d$ -dimensional domain  $\mathcal{X}^d$  is  $\Theta(d^2 \log |\mathcal{X}|)$ .<sup>1</sup> Beimel et al. [5] showed that the sample complexity of privately learning thresholds (i.e., 1-dimensional halfspaces) under *approximate* differential privacy (i.e., with  $\delta > 0$ ) can be significantly smaller than under pure differential privacy. They constructed a general differentially private algorithm  $\mathcal{A}_{RecConcave}$  that, given a quasi-concave low-sensitivity target function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$ , requires only  $\tilde{O}\left(2^{O(\log^* |\tilde{\mathcal{X}}|)}\right)$  data elements to optimize it. I.e., given a dataset  $S \in \mathcal{X}^*$  of that size,  $\mathcal{A}_{RecConcave}$  computes  $\tilde{x} \in \tilde{\mathcal{X}}$  such that  $Q(S, \tilde{x})$  is close to  $\max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\}$ . The properties that  $Q$  should satisfy are:

1. *Quasi-Concave*: For any dataset  $S \in \mathcal{X}^*$  and any  $x' < x < x''$  in  $\tilde{\mathcal{X}}$  it holds that  $Q(S, x) \geq \min\{Q(S, x'), Q(S, x'')\}$ ,
2. *Low-Sensitivity*: For any neighboring  $S, S' \in \mathcal{X}^*$  and any  $x \in \tilde{\mathcal{X}}$  it holds that  $|Q(S, x) - Q(S', x)| \leq 1$ .

Bun et al. [7] and Kaplan et al. [16] reduced learning thresholds to the *interior point* problem, where given a dataset  $S \in \mathcal{X}^n$  for a finite  $\mathcal{X} \subset \mathbb{R}$ , the goal is to output  $x$  such that  $\min S \leq x \leq \max S$ . Note that the latter problem can be solved privately by optimizing the following quasi-concave function:

$$Q_{IP}(S = \{x_1, \dots, x_n\}, x) := \min \{ |\{i \in [n] : x_i \leq x\}|, |\{i \in [n] : x_i \geq x\}| \}. \quad (1)$$

Beimel et al. [4] and Kaplan et al. [17] extended this approach to optimize high-dimensional functions. More specifically, suppose that we have a low-sensitivity quasi-concave  $d$ -dimensional function  $Q: (\mathcal{X}^d)^* \times$

<sup>1</sup>The upperbound follows from the  $O(\log |C|)$  upperbound of [18] (considering that a hyperplane in  $d$  dimensions can be represented by  $d$  points). The lowerbound follows from [13] who showed that the Littlestone dimension of halfspaces is  $d^2 \log |\mathcal{X}|$ .

$\mathbb{R}^d \rightarrow \mathbb{R}$  that we would like to optimize.<sup>2</sup> Then we can apply the 1-dimensional optimizer  $\mathcal{A}_{RecConcave}$  coordinate-by-coordinate: In step  $i \in [d]$ , given the results  $\tilde{x}_1, \dots, \tilde{x}_{i-1}$  of the previous optimizations, we compute  $\tilde{x}_i$  by applying  $\mathcal{A}_{RecConcave}$  to optimize the 1-dimensional function

$$Q_{\tilde{x}_1, \dots, \tilde{x}_{i-1}}(S, x_i) := \max_{x_{i+1}, \dots, x_d \in \mathbb{R}} Q(S, (\tilde{x}_1, \dots, \tilde{x}_{i-1}, x_i, x_{i+1}, \dots, x_d)), \quad (2)$$

where  $x_i$  is chosen from a proper finite domain  $\tilde{\mathcal{X}}_i$  with  $\log^* |\tilde{\mathcal{X}}_i| \approx \log^* |\mathcal{X}|$ .

Beimel et al. [4] applied this approach to privately find a center point by privately optimizing the Tukey-Depth function (Definition 13), and Kaplan et al. [17] reduced private learning of halfspaces to privately optimizing a quasi-concave function  $cdepth$  (Definition 16). The resulting upper bounds for both problems are  $\tilde{O}(d^{2.5} \cdot 2^{O(\log^* |\mathcal{X}|)})$ , where the term  $2^{O(\log^* |\mathcal{X}|)}$  in both results is due to the application of the 1-dimensional quasi-concave optimizer  $\mathcal{A}_{RecConcave}$ . Towards better understanding the sample complexity of these two problems, the question we ask in this paper is:

Are there differentially private algorithms for center points and for PAC learning halfspaces with sample complexity  $O(\text{poly}(d) \cdot \log^* |\mathcal{X}|)$ ?

A series of works by Bun et al. [7], Beimel et al. [4] and Alon et al. [2] give a lower bound of  $\Omega(d + \log^* |\mathcal{X}|)$  on the sample complexity of privately selecting a center point and privately learning halfspaces. In particular, in terms of the dependency in  $\log^* |\mathcal{X}|$ , there is an exponential gap between the upper and lower bounds established prior to this work.

In the 1-dimensional case, this gap was recently closed. Kaplan et al. [16] presented an upper bound of  $\tilde{O}((\log^* |\mathcal{X}|)^{1.5})$  for learning thresholds, and more recently, Cohen et al. [9] improved this upper bound to  $\tilde{O}(\log^* |\mathcal{X}|)$ . However, it remained unclear how to extend these upper bounds to the high-dimensional case. The main issue is that these methods are tailored to optimize the specific interior point function  $Q_{IP}$  (Equation 1) and do not provide a general method to optimize any quasi-concave function as  $\mathcal{A}_{RecConcave}$  does. Therefore, it was unclear how to apply the ideas there to the more general functions that are induced by optimizing high dimensional tasks coordinate-by-coordinate (Equation 2).

One could hope that a general private quasi-concave optimization can be done using an exponentially smaller sample complexity. Unfortunately, Cohen et al. [9] proved that a sample complexity of  $\Omega(2^{\log^* |\mathcal{X}|})$  is necessary, in general. They interpreted this lowerbound as follows:

*“We view this lower bound as having an important conceptual message, because private quasi-concave optimization is the main workhorse (or more precisely, the only known workhorse) for several important tasks, such as privately learning (discrete) halfspaces [4, 17]. As such, current bounds on the sample complexity of privately learning halfspaces are exponential in  $\log^* |\mathcal{X}|$ , but it is conceivable that this can be improved to a polynomial or a linear dependency. The lower bound means that either this is not true, or that we need to come up with fundamentally new algorithmic tools in order to make progress w.r.t. halfspace.” [9]*

We show how to bypass the lower bound of Cohen et al. [9] by only focusing on ‘natural’ quasi-concave functions which include the functions that are induced in Equation 2 for the high dimensional optimization tasks of [4, 17]. As a result, we achieve the first exponential improvement in the dependency on  $\log^* |\mathcal{X}|$  and establish a new upper bound of  $\tilde{O}(d^{5.5} \log^* |\mathcal{X}|)$  for privately learning halfspaces and a selecting center point.<sup>3</sup>

## 1.2 Our Results

### 1.2.1 Private Optimization for Approximated Quasi-Concave Functions

Our first contribution is a new method to bypass the lower bound of Cohen et al. [9] for a natural class of quasi-concave functions which we call *approximated* quasi-concave functions.

<sup>2</sup>A  $d$ -dimensional  $Q$  is quasi-concave if for any dataset  $S$ , any points  $x_1, \dots, x_k$  and any  $x$  in their convex hull, it holds that  $Q(S, x) \geq \min_i \{Q(S, x_i)\}$ .

<sup>3</sup>We are aware that a recent paper by Huang et al. [15] claims to privately learn halfspaces using sample complexity of  $\tilde{O}(d \cdot \log^* |\mathcal{X}|)$ . In Appendix C, we show that their algorithm is not differentially private.

More formally, we say that a function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$  can be  $(\alpha, \beta, m)$ -approximated if a random subset  $S' \subseteq S$  of size  $|S'| = m$  satisfies  $\Pr[\forall x, |Q(S, x) - Q(S', x)| \leq \alpha] \geq 1 - \beta$ . We consider  $\alpha$  as an ‘acceptable’ additive error (in particular,  $\alpha \ll \max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\}$ ), and  $\beta$  as a small enough confidence error.

We show how to reduce the task of optimizing such functions  $Q$  to the 1-dimensional interior point problem. The idea is to use the sample and aggregate approach of [20]: We partition the  $n$ -size input dataset  $S$  into random  $m$ -size subsets, compute (non-privately) the optimal solution with respect to each subset, and then aggregate the  $n/m$  solutions using a private interior point algorithm. By the approximation property of  $Q$ , with probability  $1 - \beta n/m \approx 1$ , all the  $n/m$  solutions have high value over  $Q(S, \cdot)$  (i.e., at most  $\alpha$ -far from the optimum), and since the function is quasi-concave, then any interior point also has a high value. By using the  $\tilde{O}(\log^* |\tilde{\mathcal{X}}|)$  interior-point algorithm of Cohen et al. [9], we obtain the following theorem:

**Theorem 1** (Informal). *There exists a  $(\varepsilon, \delta)$ -differentially private algorithm `IPConcave` that given a target function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$  that is quasi-concave and can be  $(\alpha, \beta, m)$ -approximated, and given a dataset  $S$  of size at least  $\tilde{O}_{\varepsilon, \delta}(m \log^* |\tilde{\mathcal{X}}|)$ , the algorithm outputs  $\hat{x} \in \tilde{\mathcal{X}}$  such that*

$$\Pr \left[ |Q(S, \hat{x}) - \max_x Q(S, x)| \leq O(\alpha) \right] \geq 1 - \tilde{O}(\beta \log^* |\tilde{\mathcal{X}}|).$$

Similar to the work of Beimel et al. [4] and Kaplan et al. [17], we extend this method to high-dimensional functions by applying `IPConcave` iteratively coordinate-by-coordinate.

**Theorem 2** (Informal). *There exists a  $(\varepsilon, \delta)$ -differentially private algorithm `IPConcaveHighDim` that given a high-dimensional target function  $Q: (\mathcal{X}^d)^* \times \mathbb{R}^d \rightarrow \mathbb{R}$  that is quasi-concave and can be  $(\alpha, \beta, m)$ -approximated, and given a dataset  $S$  of size at least  $\tilde{O}_{\varepsilon, \delta, \alpha, \beta}(m \log^* |\mathcal{X}| \sqrt{d})$ , the algorithm outputs  $\hat{x} \in \mathbb{R}^d$  such that*

$$\Pr \left[ |Q(S, \hat{x}) - \max_{x \in \mathbb{R}^d} Q(S, x)| \leq O(\alpha d) \right] \geq 1 - \tilde{O}(\beta (\log^* |\mathcal{X}| + d)).$$

Note that the approximation requirement here is a very strong guarantee. It requires that a random subset has a value similar to this of the original dataset for ‘every’ input  $x$ . Fortunately, VC theory can provide such a guarantee for the functions that were used by [4, 17]. As applications, we close the exponential gap of sample complexity for private center point and learning halfspaces, as described next.

## 1.2.2 Applications

**Private Center Point** For privately approximating the center point, we follow the approach of Beimel et al. [4] to optimize the Tukey depth function, but now with our new `IPConcaveHighDim` method. By VC theory, the Tukey depth function (normalized by the dataset size) can be  $(\alpha, \beta, m)$ -approximated with  $m = \tilde{O}((d + \log(1/\beta))/\alpha^2)$  (Lemma 1). Furthermore, the Tukey center of every  $n$ -size dataset  $S$  has Tukey depth at least  $\frac{n}{d+1}$ . Hence, by substituting  $\alpha$  with  $\alpha/d^2$  in Theorem 2, we obtain the following result.

**Theorem 3** (Privately approximating the center point). *Let  $\mathcal{X} \subset \mathbb{R}$  be a finite domain. There exists an  $(\varepsilon, \delta)$ -differentially private algorithm  $\mathbf{A}: (\mathcal{X}^d)^n \rightarrow \mathbb{R}^d$  that given an  $n$ -size dataset  $S \in (\mathcal{X}^d)^n$ , for  $n \geq \tilde{O}_{\alpha, \beta, \varepsilon, \delta}(d^{5.5} \cdot \log^* |\mathcal{X}|)$ ,  $\mathbf{A}(S)$  outputs, with probability  $1 - \beta$ , a point with Tukey depth in  $S$  of at least  $\frac{1-\alpha}{d+1} \cdot n$ .*

**Private Halfspace Learning** For privately learning halfspaces, we follow the approach of Kaplan et al. [17] to reduce the task to private feasibility problem and solve this problem by optimizing a quasi-concave function *cdepth* (Definition 16), but now using our `IPConcaveHighDim` method. Similarly to the Tukey depth function, the *cdepth* function (normalized by the dataset size) can also be  $(\alpha, \beta, m)$ -approximated for  $m = \tilde{O}((d + \log(1/\beta))/\alpha^2)$  (Corollary 3). An  $\alpha/d$  error in the *cdepth* function is translated to an  $\alpha$  error for learning halfspace and the linear feasibility problem. Therefore, we achieve our learner by substituting  $\alpha$  with  $\alpha/d^2$  in Theorem 2.

**Theorem 4** (Privately learning halfspaces). *Let  $\mathcal{X} \subset \mathbb{R}$  be a finite domain. There exists an  $(\varepsilon, \delta)$ -differentially private  $(\alpha, \beta)$ -PAC learner for halfspaces over examples from  $\mathcal{X}^d$  with sample complexity  $n = \tilde{O}_{\alpha, \beta, \varepsilon, \delta}(d^{5.5} \cdot \log^* |\mathcal{X}|)$ .*

**Learning over concept class of VC dimension 1.** We remark that since every concept class with VC dimension 1 can be embedded in 3-dimensional halfspaces class [3, 1], our results imply that every concept class with VC dimension 1 can be privately learned with sample size  $\tilde{O}(\log^* |\mathcal{X}|)$ .

**Corollary 1.** *For any concept class  $\mathcal{C}$  with VC dimension 1 and input domain  $\mathcal{X}$ , there exists an  $(\varepsilon, \delta)$ -differentially private algorithm that can  $(\alpha, \beta)$ -PAC learn it with sample size  $\tilde{O}_{\varepsilon, \delta, \alpha, \beta}(\log^* |\mathcal{X}|)$ .*

### 1.3 Open Questions

In this work, we present a general private quasi-concave optimization method that bypasses the lower bound of Cohen et al. [9] and enables to achieve the first  $d^{O(1)} \cdot \log^* |\mathcal{X}|$  sample complexities for fundamental problems such as privately approximating the center point and privately learning halfspaces over a finite euclidean input domain  $\mathcal{X}^d$ . But while our result bridge the exponential gap in  $\log^* |\mathcal{X}|$ , the dependency on the dimension  $d$  has increased from  $d^{2.5}$  ([4, 17]) to  $d^{5.5}$  (Theorems 3 and 4). It remains open to understand if it is possible to reduce the dependency on  $d$  while avoiding an exponential blow-up in  $\log^* |\mathcal{X}|$ .

## 2 Preliminaries

### 2.1 Notations

We denote a subset  $S$  of  $X$  by  $S \subseteq X$ , and a multi-set  $S$  of elements in  $X$  by  $S \in X^*$  (or  $S \in X^n$  if  $S$  is of size  $n$ ). For a multi-set  $S \in X^*$  and a set  $T \subseteq X$ , we let  $S \cap T$  be the multi-set of all the elements in  $S$  (including repetitions) that belong to  $T$ . A subset  $S'$  of a multi-set  $S$  (denoted by  $S' \subseteq S$ ) refers to a sub-multiset, i.e., if  $S = \{x_1, \dots, x_n\}$  then there exists a set of indices  $I = \{i_1, \dots, i_m\} \subseteq [n]$  such that  $S' = \{x_{i_1}, \dots, x_{i_m}\}$ . A random  $m$ -size subset  $S'$  of a multi-set  $S$  is specified by a random  $m$ -size set of indices  $I \subseteq [n]$ .

Throughout this paper, a dataset refers to a multi-set.

### 2.2 Learning Theory

**Definition 2** (Vapnik-Chervonenkis dimension [25, 14]). *Let  $X$  be a set and  $R$  be a set of subsets of  $X$ . Let  $S \subseteq X$  be a subset of  $X$ . Define  $\Pi_R(S) = \{S \cap r \mid r \in R\}$ . If  $|\Pi_R(S)| = 2^{|S|}$ , then we say  $S$  is shattered by  $R$ . The Vapnik-Chervonenkis dimension of  $(X, R)$  is the largest integer  $d$  such that there exists a subset  $S$  of  $X$  with size  $d$  that is shattered by  $R$ .*

**Definition 3** ( $\alpha$ -approximation<sup>4</sup> [25, 14]). *Let  $X$  be a set and  $R$  be a set of subsets of  $X$ . Let  $S \in X^*$  be a finite multi-set of elements in  $X$ . For any  $0 \leq \alpha \leq 1$  and  $S' \subseteq S$ ,  $S'$  is an  $\alpha$ -approximation of  $S$  for  $R$  if for all  $r \in R$ , it holds that  $\left| \frac{|S \cap r|}{|S|} - \frac{|S' \cap r|}{|S'|} \right| \leq \alpha$ .*

**Theorem 5** ([25, 14]). *Let  $(X, R)$  have VC dimension  $d$ . Let  $S \subseteq X$  be a subset of  $X$ . Let  $0 < \alpha, \beta \leq 1$ . Let  $S' \subseteq S$  be a random subset of  $S$  with size at least  $O\left(\frac{d \cdot \log \frac{d}{\alpha} + \log \frac{1}{\beta}}{\alpha^2}\right)$ . Then with probability at least  $1 - \beta$ ,  $S'$  is an  $\alpha$ -approximation of  $S$  for  $R$ .<sup>5</sup>*

**Definition 4** (Generalization and empirical error). *Let  $\mathcal{D}$  be a distribution,  $c$  be a concept and  $h$  be a hypothesis. The error of  $h$  w.r.t.  $c$  over  $\mathcal{D}$  is defined as*

$$\text{error}_{\mathcal{D}}(c, h) = \Pr_{x \sim \mathcal{D}}[c(x) \neq h(x)].$$

<sup>4</sup>Commonly called  $\varepsilon$ -approximation. We use  $\alpha$ -approximation as  $\varepsilon$  is used as a parameter of differential privacy.

<sup>5</sup>Although the theorem is stated for sets  $S \subseteq X$ , it also holds for multisets  $S \in X^*$ . To see it, fix an  $n$ -size multiset  $S \in X^n$  and transform the domain set  $X = \{x_1, \dots, x_k\}$  to an extended domain set  $X' = \{x_{1,1}, \dots, x_{1,n}, \dots, x_{k,1}, \dots, x_{k,n}\}$ . For every  $r \in R$ , we can transform it to  $r' = \{x_{i,1}, \dots, x_{i,n} : x_i \in r\}$ , and let  $R' = \{r' : r \in R\}$ . We have  $VC(X', R') = VC(X, R)$ . Now each element  $x_i$  that appears  $t$  times in  $S$  can be replaced by  $x_{i,1}, x_{i,2}, \dots, x_{i,t}$ . This reduction transforms all multiset setting to a corresponding set setting.

For a finite dataset  $S$ , the error of  $h$  w.r.t.  $c$  over  $S$  is defined as

$$\text{error}_S(c, h) = \frac{|\{x \in S \mid c(x) \neq h(x)\}|}{|S|}.$$

**Theorem 6** ([6, 17]). Let  $\mathcal{C}$  be a concept class and let  $\mathcal{D}$  be a distribution. Let  $\alpha, \beta > 0$ , and  $m \geq \frac{48}{\alpha} \left(10VC(\mathcal{C}) \log(\frac{48\epsilon}{\alpha}) + \log(\frac{5}{\beta})\right)$ . Let  $S$  be a sample of  $m$  points drawn i.i.d. from  $\mathcal{D}$ . Then

$$\Pr[\exists c, h \in \mathcal{C} \text{ s.t. } \text{error}_S(c, h) \leq \alpha/10 \text{ and } \text{error}_{\mathcal{D}}(c, h) \geq \alpha] \leq \beta.$$

## 2.3 Differential Privacy

**Definition 5** (Differential Privacy [11]). Let  $\mathcal{X}$  be a data domain and  $\mathcal{Y}$  be an output domain. A (randomized) mechanism  $M$  mapping  $\mathcal{X}^n$  to  $\mathcal{Y}$  is  $(\epsilon, \delta)$ -differentially private if for any pair of inputs  $S, S' \in \mathcal{X}^n$  where  $S$  and  $S'$  differ on a single entry, and any event  $E \subseteq \mathcal{Y}$ , it holds that

$$\Pr[M(S) \in E] \leq e^\epsilon \cdot \Pr[M(S') \in E] + \delta,$$

where the probability is over the randomness of  $M$ .

**Theorem 7** (Advanced composition [12]). Let  $M_1, \dots, M_k : \mathcal{X}^n \rightarrow \mathcal{Y}$  be  $(\epsilon, \delta)$ -differentially private mechanisms. Then the algorithm that on input  $S \in \mathcal{X}^n$  outputs  $(M_1(S), \dots, M_k(S))$  is  $(\epsilon', k\delta + \delta')$ -differentially private, where  $\epsilon' = \sqrt{2k \ln(1/\delta')} \cdot \epsilon$  for every  $\delta' > 0$ .

We use differentially private algorithms for the Interior Point problem:

**Definition 6** (Interior Point [7]). Let  $\mathcal{X}$  be a (finite) ordered domain. We say that  $p \in \mathcal{X}$  is an interior point of a dataset  $S = \{x_1, \dots, x_n\} \in \mathcal{X}^n$  if  $\min_{i \in [n]} \{x_i\} \leq p \leq \max_{i \in [n]} \{x_i\}$ .

Bun et. al. [7] proved that the sample complexity for solving the interior point problem with differential privacy must grow proportionally to  $\log^* |\mathcal{X}|$  [7]. Cohen et. al. provide a nearly optimal algorithm solving the interior point privately on a finite domain [9].

**Theorem 8** ([9]). Let  $\mathcal{X}$  be a finite ordered domain. There exists an  $(\epsilon, \delta)$ -differentially private algorithm `PrivateIP` that on input  $S \in \mathcal{X}^n$  outputs an interior point with probability  $1 - \beta$  provided that  $n > n_{IP}(|\mathcal{X}|, \beta, \epsilon, \delta)$  for  $n_{IP}(|\mathcal{X}|, \beta, \epsilon, \delta) \in O\left(\frac{\log^* |\mathcal{X}| \cdot \log^2(\frac{\log^* |\mathcal{X}|}{\beta\delta})}{\epsilon}\right)$ .

## 2.4 Halfspaces

**Definition 7** (Halfspaces and hyperplanes). Let  $\mathcal{X} \subset \mathbb{R}^d$ . For  $a_1, \dots, a_d, w \in \mathbb{R}$ , the halfspace predicate  $h_{a_1, \dots, a_d, w} : \mathcal{X} \rightarrow \{\pm 1\}$  is defined as  $h_{a_1, \dots, a_d, w}(x_1, \dots, x_d) = 1$  if and only if  $\sum_{i=1}^d a_i x_i \geq w$ . Define the concept class  $\text{HALFSPACE}_d(\mathcal{X}) = \{h_{a_1, \dots, a_d, w}\}_{a_1, \dots, a_d, w \in \mathbb{R}}$ .

## 2.5 Notation

We use the notation  $(x_1, \dots, x_i) \times \mathcal{X}^{d-i}$  for the space of points with a fixed prefix of  $i$  coordinates:

$$(x_1, \dots, x_i) \times \mathcal{X}^{d-i} = \{y \in \mathcal{X}^d \mid y_j = x_j \text{ for } j \in [i]\}.$$

### 3 Our Private Quasi-Concave Optimization Scheme

In this section, we present our main algorithm `IPConcave` that for privately optimizing approximated quasi-concave functions.

We first give the definition of the quasi-concave function and the sensitivity of functions.

**Definition 8** (Quasi-Concave). *Let  $\mathcal{X}$  be an ordered domain. A function  $f: \mathcal{X} \rightarrow \mathbb{R}$  is quasi-concave if  $f(\ell) \geq \min(\{f(i), f(j)\})$  for every  $i < \ell < j$ .*

**Definition 9** (Sensitivity). *The sensitivity of a function  $f: \mathcal{X}^* \rightarrow \mathbb{R}$  is the smallest  $k$  such that for every pair of neighboring datasets  $S, S' \in \mathcal{X}^*$  (i.e., differ in exactly one entry), we have  $|f(S) - f(S')| \leq k$ . A function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$  is called a sensitivity- $k$  function if for every  $x \in \tilde{\mathcal{X}}$ , the function  $Q(\cdot, x)$  has sensitivity  $\leq k$ .*

#### 3.1 One-Dimensional Case

Beimel et al. [5] provide an algorithm  $\mathcal{A}_{\text{RecConcave}}$  that given as inputs a dataset  $S \in \mathcal{X}^*$  and a sensitivity-1 function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$  such that  $Q(S, \cdot)$  is quasi-concave, the algorithm privately finds a point  $\hat{x} \in \tilde{\mathcal{X}}$  such that  $|Q(S, \hat{x}) - \max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\}| \leq \tilde{O}\left(2^{O(\log^* |\tilde{\mathcal{X}}|)}\right)$ . In fact, the exponential dependency in  $2^{O(\log^* |\tilde{\mathcal{X}}|)}$  is necessary in general since Cohen et al. [9] proved a matching lower bound.

In this work, we bypass the lower bound of [9] by showing that if the quasi-concave function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$  has the property that given a dataset  $S \in \mathcal{X}^*$ , the function  $Q(S, \cdot)$  can be well approximated by  $Q(S', \cdot)$  for an  $m$ -size random subset  $S' \subset S$ , then we can privately optimize it using sample complexity  $\tilde{O}(m \cdot \log^* |\tilde{\mathcal{X}}|)$ .

Here we define  $(\alpha, \beta, m)$ -approximation with respect to  $Q$ .

**Definition 10** (Approximation with respect to  $Q$ ). *For any dataset  $S \in \mathcal{X}^*$  and function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$ , we say that  $S' \subseteq S$  is an  $\alpha$ -approximation of  $S$  with respect to  $Q$  if for any  $x \in \tilde{\mathcal{X}}$ , we have  $|Q(S, x) - Q(S', x)| \leq \alpha$ . We say  $(S, Q)$  can be  $(\alpha, \beta, m)$ -approximated, if by randomly selecting a subset  $S' \subseteq S$  of size at least  $m$ , then with probability  $1 - \beta$ ,  $S'$  is an  $\alpha$ -approximation of  $S$  with respect to  $Q$ .*

Note that  $\alpha$ -approximation with respect to a function  $Q$  (Definition 10) is similar to  $\alpha$ -approximation with respect to a set of binary value functions  $R$  (Definition 3). Indeed, in Sections 4 and 5 we exploit this similarity and apply Theorem 5 for upper bounding the subset size of approximating specific functions.

In Appendix A, we show that approximated function are, in particular, low-sensitivity functions.

In the following, we present our new private optimization algorithm for quasi-concave functions that can be approximated by a random subset.

---

#### Algorithm 1: IPConcave

---

**Parameter:** Confidence parameter  $\beta > 0$ , privacy parameter  $\varepsilon, \delta > 0$ , and number of subsets  $t$ .

**Inputs:** A dataset  $S \in \mathcal{X}^n$ , for  $n \geq t$ , and a function  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$  where  $\tilde{\mathcal{X}} \subseteq \mathbb{R}$  and is finite.

**Operation:**

1. Randomly partition  $S$  into  $S_1, \dots, S_t$ , each with size at least  $\lfloor n/t \rfloor$ .
  2. For  $i \in [t]$ : Compute  $y_i = \operatorname{argmax}_{x \in \tilde{\mathcal{X}}} \{Q(S_i, x)\}$ .
  3. Compute and output  $\hat{x} \sim \text{PrivateIP}_{\tilde{\mathcal{X}}, \beta, \varepsilon, \delta}(y_1, \dots, y_t)$  (the algorithm from Theorem 8 that solves the 1-dimensional interior point problem over the domain  $\tilde{\mathcal{X}}$  with confidence parameter  $\beta$  and privacy parameters  $\varepsilon, \delta$ ).
- 

**Theorem 9** (Restatement of Theorem 1). *Let  $\varepsilon > 0$ ,  $\delta, \alpha, \beta \in (0, 1)$ ,  $n, t \in \mathbb{N}$  with  $n \geq t$ , let  $\mathcal{X}$  be a domain of data elements, let  $\tilde{\mathcal{X}} \subseteq \mathbb{R}$  be a finite domain, and let  $Q: \mathcal{X}^* \times \tilde{\mathcal{X}} \rightarrow \mathbb{R}$ . Then the following holds:*

1. **Privacy:** Algorithm  $\text{IPConcave}_{\alpha,\beta,\varepsilon,\delta,t}(\cdot, Q): \mathcal{X}^n \rightarrow \tilde{\mathcal{X}}$  is  $(\varepsilon, \delta)$ -differentially private.
2. **Accuracy:** Let  $S \in \mathcal{X}^n$  and assume that  $Q(S, \cdot)$  is quasi-concave and that  $(S, Q)$  can be  $(\alpha, \beta', \lfloor n/t \rfloor)$ -approximated (Definition 10). If  $t \geq n_{IP}(|\mathcal{X}|, \beta, \varepsilon, \delta) \in \tilde{O}_{\beta,\varepsilon,\delta}(\log^* |\mathcal{X}|)$  (the sample complexity of  $\text{PrivatelP}$  from Theorem 8), then

$$\Pr_{\hat{x} \sim \text{IPConcave}_{\alpha,\beta,\varepsilon,\delta,t}(S,Q)} \left[ \left| Q(S, \hat{x}) - \max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\} \right| \leq 2\alpha \right] \geq 1 - t \cdot \beta' - \beta$$

*Proof. Privacy:* A change of one input point affect one of the points  $y_i$ . So  $\text{PrivatelP}$  guarantees that the output is  $(\varepsilon, \delta)$ -differentially private.

*Accuracy:* The  $(\alpha, \beta', \lfloor n/t \rfloor)$ -approximation guarantees that for any subset  $S_i$ , with probability  $1 - \beta'$ ,  $S_i$  is an  $\alpha$ -approximation of  $S$  with respect to  $Q$  (Definition 10). Let  $x_{opt} = \operatorname{argmax}_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\}$ . Under the condition of  $\alpha$ -approximation with respect to  $Q$ , we have  $Q(S_i, x_{opt}) \geq Q(S, x_{opt}) - \alpha = \max_{x \in \mathcal{X}} \{Q(S, x)\} - \alpha$ . Then we have  $Q(S_i, y_i) \geq Q(S_i, x_{opt}) \geq \max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\} - \alpha$ . Using the  $\alpha$ -approximation w.r.t  $Q$  again, we have  $Q(S, y_i) \geq Q(S_i, y_i) - \alpha \geq \max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\} - 2\alpha$ . Since  $\text{PrivatelP}$  succeeds with probability  $1 - \beta$ , then by the union bound, with probability  $1 - t \cdot \beta' - \beta$ , for all  $i \in [t]$  we have  $Q(S, y_i) \geq \max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\} - 2\alpha$  and that  $\hat{x}$  is an interior point of  $\{y_1, \dots, y_t\}$ . Since  $Q$  is quasi-concave, the above implies that  $Q(S, \hat{x}) \geq \min_{i \in [t]} \{Q(S, y_i)\} \geq \max_{x \in \tilde{\mathcal{X}}} \{Q(S, x)\} - 2\alpha$ , as required.  $\square$

## 3.2 Extending to Higher Dimension

Beimel et al. [4] and Kaplan et al. [17] use  $\mathcal{A}_{\text{RecConcave}}$  to optimize high dimension functions. The method is to iteratively select good coordinates using  $\mathcal{A}_{\text{RecConcave}}$ . Thus the sample size of their result must depend on  $2^{\log^* |\mathcal{X}|}$ . This work shows that our new optimizer can also be extended to higher dimensions if the target function  $Q$  is (high-dimensional) quasi-concave and has proper finite-size domains. Notice that the 1-dimensional concavity property is equivalent to quasi-concave.

**Definition 11** ((High-Dimensional) Quasi-Concave). *We say a function  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  is quasi-concave if for any points  $p_1, \dots, p_k$  and any point  $p$  in the convex hull of  $\{p_i\}_{i \in [k]}$ , it holds that  $f(p) \geq \min(\{f(p_1), \dots, f(p_k)\})$ .*

Since the interior point can only be privately found in a finite domain, we need to construct such domain that contains a point with a high value of  $Q$ .

**Definition 12** (Proper Finite Domains). *We say that  $\tilde{\mathcal{X}}_1, \dots, \tilde{\mathcal{X}}_d$  are proper finite domains for  $Q: \mathcal{X}^* \times \mathbb{R}^d \rightarrow \mathbb{R}$  if for any  $i \in [d]$ ,  $S \in \mathcal{X}^*$  and  $\hat{x}_1 \in \tilde{\mathcal{X}}_1, \dots, \hat{x}_{i-1} \in \tilde{\mathcal{X}}_{i-1}$ , there exists  $\hat{x}_i \in \tilde{\mathcal{X}}_i$ , such that*

$$\max_{x_i, \dots, x_d \in \mathbb{R}} Q(S, (\hat{x}_1, \dots, \hat{x}_{i-1}, x_i, \dots, x_d)) = \max_{x_{i+1}, \dots, x_d \in \mathbb{R}} Q(S, (\hat{x}_1, \dots, \hat{x}_i, x_{i+1}, \dots, x_d)).$$

*The construction of  $\tilde{\mathcal{X}}_i$  may depend on  $\hat{x}_1, \dots, \hat{x}_{i-1}$ , so  $\tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_d$  are treated as functions.*

We next describe our algorithm  $\text{IPConcaveHighDim}$  that extends  $\text{IPConcave}$  for high-dimensional quasi-concave functions.

**Remark 1.** *Note that in Algorithm 2, we chose to use the same partition  $S_1, \dots, S_t$  for all iterations rather than letting  $\text{IPConcave}$  to sample a fresh partition in each invocation. The main reason we chose to do that is to increase the confidence guarantee, since once  $S_1, \dots, S_t$  are all a good approximation of  $S$  w.r.t.  $Q$ , they are good for all the iterations and therefore, there is no need to re-sample.*

**Theorem 10** (Restatement of Theorem 2). *Let  $\varepsilon > 0$ ,  $\delta, \alpha, \beta \in (0, 1)$ ,  $n, t, d \in \mathbb{N}$  with  $n \geq t$ , let  $\mathcal{X}$  be a domain of data elements, let  $Q: \mathcal{X}^* \times \mathbb{R}^d \rightarrow \mathbb{R}$  be a function with finite domains  $\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_d$ , let  $X = \max_{i \in [n]} \left\{ |\tilde{\mathcal{X}}_i| \right\}$ . Then the following holds:*

1. **Privacy:** Algorithm  $\text{IPConcaveHighDim}_{\alpha,\beta,\varepsilon,\delta,t}(\cdot, Q): \mathcal{X}^n \rightarrow \mathbb{R}^d$  is  $(\varepsilon \cdot \sqrt{2d \ln(1/\delta')}, d\delta + \delta')$ -differentially private for any choice of  $\delta' > 0$ .

---

**Algorithm 2:** IPConcaveHighDim
 

---

**Parameter:** Utility parameters  $\alpha, \beta > 0$ , privacy parameter  $\varepsilon, \delta > 0$ , and number of subsets  $t$ .

**Inputs:** A dataset  $S \in \mathcal{X}^n$ , for  $n \geq t$ , and a function  $Q: \mathcal{X}^* \times \mathbb{R}^d \rightarrow \mathbb{R}$  with proper finite domains  $\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_d$  (Definition 12).

**Operation:**

1. Randomly partition  $S$  into  $S_1, \dots, S_t$ , each with size at least  $\lfloor n/t \rfloor$ .

2. For  $i = 1, \dots, d$ :

(a) Let  $\tilde{\mathcal{X}}_i = \tilde{\mathcal{X}}_i(\hat{x}_1, \dots, \hat{x}_{i-1})$ , and define the function  $\hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-1}}: \mathcal{X}^n \times \tilde{\mathcal{X}}_i \rightarrow \mathbb{R}$  as

$$\hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-1}}(S, x) := \max_{x_{i+1}, \dots, x_d \in \mathbb{R}} Q(S, (\hat{x}_1, \dots, \hat{x}_{i-1}, x, x_{i+1}, \dots, x_d)).$$

(b) Compute  $\hat{x}_i \sim \text{IPConcave}_{\alpha, \beta, \varepsilon, \delta, t}(S, \hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-1}})$ , where we fix the partition in Step 1 of IPConcave (Algorithm 1) to  $S_1, \dots, S_t$ .

3. Output  $(\hat{x}_1, \dots, \hat{x}_d)$ .

---

2. **Accuracy:** Let  $S \in \mathcal{X}^n$  and assume that: (1)  $Q(S, \cdot)$  is quasi-concave (Definition 11), (2)  $(S, Q)$  can be  $(\alpha, \beta', \lfloor n/t \rfloor)$ -approximated (Definition 10), and (3)  $\tilde{\mathcal{X}}_1, \dots, \tilde{\mathcal{X}}_d$  are proper for  $Q$  (Definition 12). Then for  $t = n_{IP}(X, \beta, \varepsilon, \delta) \in \tilde{O}_{\alpha, \beta, \varepsilon, \delta}(\log^* X)$  (the sample complexity of PrivateIP from Theorem 8 over domain of size  $X$ ), it holds that

$$\Pr_{\hat{x} \sim \text{IPConcaveHighDim}_{\alpha, \beta, \varepsilon, \delta, t}(S, Q)} \left[ |Q(S, \hat{x}) - \max_{x \in \mathbb{R}^d} \{Q(S, x)\}| \leq 2\alpha d \right] \geq 1 - t \cdot \beta' - d \cdot \beta$$

*Proof. Privacy of Algorithm 2:* Each call to  $\text{IPConcave}_{\alpha, \beta, \varepsilon, \delta, t}$  (Step 2b) is  $(\varepsilon, \delta)$ -differentially private (Theorem 9). Using advanced composition (Theorem 7), we get that Algorithm 2 is  $(\varepsilon \cdot \sqrt{2d \ln(1/\delta')}, d\delta + \delta')$ -differentially private for any choice of  $\delta' > 0$ .

**Accuracy of Algorithm 2** By Definition 10 and the union bound, with probability  $1 - t \cdot \beta'$ , all  $S_1, \dots, S_t$  are  $\alpha$ -approximation of  $S$  with respect to  $Q$ , and in the following we assume that this event occurs. This means  $|Q(S, x) - Q(S_j, x)| \leq \alpha$  for all  $x \in \mathbb{R}^d$  and  $j \in [t]$ . In particular, this implies that for every  $i \in [d]$ , in the  $i$ 'th iteration of IPConcaveHighDim, the subset  $S_j$ , for every  $j \in [t]$ , is an  $\alpha$ -approximation of  $S$  with respect to the (one-dimensional) function  $\hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-1}}$  (defined in Step 2a). Furthermore, since  $Q(S, \cdot)$  is quasi-concave (Definition 11), then each function  $\hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-1}}(S, \cdot)$  is also quasi-concave. Thus, by the accuracy guarantee of IPConcave (Theorem 9), in the  $i$ 'th iteration of IPConcaveHighDim, with probability  $1 - \beta$ , the algorithm computes  $\hat{x}_i$  such that

$$\begin{aligned} \hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-1}}(S, \hat{x}_i) &\geq \max_{x_i \in \mathcal{X}_i, x_{i+1}, \dots, x_d \in \mathbb{R}} Q(S, (\hat{x}_1, \dots, \hat{x}_{i-1}, x_i, \dots, x_d)) - 2\alpha \\ &= \max_{x_i, \dots, x_d \in \mathbb{R}} Q(S, (\hat{x}_1, \dots, \hat{x}_{i-1}, x_i, \dots, x_d)) - 2\alpha, \end{aligned} \quad (3)$$

where the equality holds since  $\mathcal{X}_1, \dots, \mathcal{X}_d$  are proper finite domains for  $Q$  by assumption. Thus, with probability  $1 - d \cdot \beta$ , Equation 3 holds for any iteration  $i$  during the execution of IPConcaveHighDim, which means that  $\hat{Q}(\hat{x}_1) \geq \max_{x \in \mathbb{R}^d} Q(S, x) - 2\alpha$ , and for every  $i \in \{2, \dots, d\}$ :

$$\hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-1}}(S, \hat{x}_i) \geq \max_{x_i, \dots, x_d \in \mathbb{R}} Q(S, (\hat{x}_1, \dots, \hat{x}_{i-1}, x_i, \dots, x_d)) - 2\alpha = \hat{Q}_{\hat{x}_1, \dots, \hat{x}_{i-2}}(S, \hat{x}_{i-1}) - 2\alpha,$$

and in the following, we assume that this event occurs. We conclude that the output  $\hat{x} = (\hat{x}_1, \dots, \hat{x}_d)$  satisfies  $Q(S, \hat{x}) \geq \max_{x \in \mathbb{R}^d} Q(S, x) - 2\alpha d$ , as required.  $\square$

## 4 Differentially Private Tukey Median Approximation

In this section, we show how to use IPConcaveHighDim (Algorithm 2) to privately approximate the Tukey median. We give additional preliminaries in Section 4.1, and describe our Tukey median approximation algorithm in Section 4.2.

### 4.1 Additional Preliminaries for Tukey Median

**Definition 13** (Tukey depth, Tukey median [23]). *Let  $S \in (\mathbb{R}^d)^n$  be a dataset of  $n$  points and let  $p \in \mathbb{R}^d$  be a point (not necessarily in  $S$ ). The Tukey depth of  $p$  in  $S$  is the minimum over all hyperplanes  $h$  going through  $p$  of the number of points in  $S$  on one side of  $h$ . A point is a Tukey median of  $S$  if it has the maximum Tukey depth in  $S$ .*

We use  $TD_S(p)$  for the Tukey depth of point  $p$  in point set  $S$ .

**Fact 1.** *Let  $S \in (\mathbb{R}^d)^n$  and let  $p_1, \dots, p_k$  be points in  $\mathbb{R}^d$  with Tukey depth at least  $\gamma n$  in  $S$ . Then any point in the convex hull of  $p_1, \dots, p_k$  has Tukey depth at least  $\gamma n$  in  $S$ .*

*Proof.* Let  $p'$  be a point inside the convex hull of  $p_1, \dots, p_k$  and assume  $p'$  has Tukey depth  $T < \gamma n$ . Then there exists a hyperplane  $h$  that goes through  $p'$  with less than  $\gamma n$  points on one of its sides. Denote by  $S_1$  and  $S_2$  be two half-spaces on the two sides of  $h$ . Without loss of generality,  $S_1$  contains  $T$  points of  $S$ . Note that since  $p'$  is in the convex hull of  $p_1, \dots, p_k$ , it must be that at least one of the points  $p_1, \dots, p_k$  is inside  $S_1$ . Without loss of generality, assume  $p_1 \in S_1$ .

Let  $h'$  be a hyperplane that goes through  $p_1$  and is parallel to  $h$ . Then there are less than  $T$  points on one side of  $h'$ , and hence  $p_1$  has Tukey depth  $T < \gamma n$ , a contradiction.  $\square$

Helly's theorem [10] implies that a point with a high Tukey depth always exists.

**Fact 2** ([10]). *For every  $S \in (\mathbb{R}^d)^n$ , the center point (and hence also the Tukey median) of  $S$  has Tukey depth at least  $\frac{n}{d+1}$  in  $S$ .*

In the following lemma, we apply Theorem 5 (the  $\alpha$ -approximation theory of Vapnik and Chervonenkis [25]) to determine an upper bound on the subset size  $m$  such that a given set of points  $S$  can be  $(\alpha, \beta, m)$ -approximated with respect to the Tukey depth function (Definition 10).<sup>6</sup>

**Lemma 1.** *Let  $S \in (\mathbb{R}^d)^n$  and let  $S' \subseteq S$  be a random subset of  $S$  with cardinality  $m = |S'| > O\left(\frac{d \cdot \log(d/\alpha) + \log(1/\beta)}{\alpha^2}\right)$ . Then, with probability at least  $1 - \beta$ , for all  $p \in \mathbb{R}^d$ , if  $TD_{S'}(p) = \gamma' m$  and  $TD_S(p) = \gamma n$  then  $|\gamma - \gamma'| \leq \alpha$ .*

*Proof.* This proof uses the well-known result that  $d$  dimensional half-space has VC-dimension  $d$ . More exactly, for  $X = \mathbb{R}^d$  and the set of all halfspaces  $R$ , the VC dimension  $(X, R)$  is  $d$ . So by Theorem 5, with probability  $1 - \beta$ , an  $m$ -size random subset  $S'$  is an  $\alpha$ -approximation of  $S$  with respect to  $R$  (Definition 3), which implies that for any hyper-plane, if there are  $\gamma_1 n$  points of  $S$  and  $\gamma_2 m$  points of  $S'$  on one side of  $h$ , then it holds that  $|\gamma_1 - \gamma_2| \leq \alpha$ . In the following, we assume that this  $1 - \beta$  probability event occurs (denote by  $E$ ).

Fix a point  $p$  with  $TD_S(p) = \gamma n$ , and assume towards a contradiction that  $TD_{S'}(p) = \gamma' m$  for  $\gamma' < \gamma - \alpha$  (the direction  $\gamma' > \gamma + \alpha$  follows similarly). This implies that there exists a hyperplane that goes through  $p$  such that there are at most  $\gamma' m$  points of  $S'$  on one side of it. Since event  $E$  occurs, the same side of the hyperplane contains at most  $(\gamma' + \alpha)n$  points of  $S$ . The latter implies that  $TD_S(p) \leq (\gamma' + \alpha)n < \gamma n$ , contradiction. We thus conclude that  $|\gamma - \gamma'| \leq \alpha$ .  $\square$

<sup>6</sup>Similar statements are also provided in some works about approximating center points (e.g. [8]).

### 4.1.1 Domain Extension

In our algorithm, we find an approximate Tukey median one coordinate at a time, and it is possible that a point would fall outside the input domain  $\mathcal{X}^d$ . We use the domain extension technique by [4].

**Lemma 2** (domain extension of  $\mathcal{X}$  [4]). *Given a finite  $\mathcal{X} \subset \mathbb{R}$ , there exist sets  $\tilde{\mathcal{X}}_1, \dots, \tilde{\mathcal{X}}_d$ , where  $|\tilde{\mathcal{X}}_i| \leq (d|\mathcal{X}|^{d^2(d+1)})^{2^d}$ , such that for all  $1 \leq i \leq d$ , for all  $S \subset \mathcal{X}^d$ , and for all  $(x_1^*, \dots, x_{i-1}^*) \in \tilde{\mathcal{X}}_1 \times \dots \times \tilde{\mathcal{X}}_{i-1}$ , there exist  $(\tilde{x}_i, \dots, \tilde{x}_d) \in \tilde{\mathcal{X}}_i \times \dots \times \tilde{\mathcal{X}}_d$  satisfying*

$$\max_{x_i, \dots, x_d \in \mathbb{R}} TD_S(x_1^*, \dots, x_{i-1}^*, x_i, \dots, x_d) = TD_S(x_1^*, \dots, x_{i-1}^*, \tilde{x}_i, \dots, \tilde{x}_d).$$

## 4.2 Privately Estimating the Tukey Median

We use `IPConcaveHighDim` (Algorithm 2) to find a point with high Tukey depth privately. Define the function

$$Q_{TD}(S, x) = \frac{TD_S(x)}{|S|}.$$

Here we verify that  $Q_{TD}(S, x)$  satisfies the accuracy requirements in Theorem 10.

1. **Approximation by a random subset:** By Lemma 1, a random subset  $S' \subseteq S$  of size

$$m(\alpha, \beta) \in O\left(\frac{d \cdot \log(d/\alpha) + \log(1/\beta)}{\alpha^2}\right)$$

is an  $\alpha$ -approximation of  $S$  with respect to  $Q_{TD}$  with probability  $1 - \beta$ .

2. **Concavity:** It is guaranteed by Fact 1.

3. **Proper finite domains:** The domain extension  $\tilde{\mathcal{X}}_1, \dots, \tilde{\mathcal{X}}_d$  in Lemma 2 provide proper finite domains for  $Q_{TD}$  (Definition 12) with maximal domain size  $X$  where  $\log^* X \in O(\log^*(|\mathcal{X}| + d))$ .

Thus, the following theorem is an immediate corollary of Theorem 10.

**Theorem 11.** *Let  $\varepsilon > 0$ ,  $\delta, \alpha, \beta \in (0, 1)$ ,  $n, d, t \in \mathbb{N}$  with  $n \geq t$  and let  $\mathcal{X} \subseteq \mathbb{R}$  be a finite domain. Let  $\mathbf{A}: (\mathcal{X}^d)^n \rightarrow \mathbb{R}^d$  be the algorithm that on input  $S$ , computes `IPConcaveHighDim` $_{\alpha, \beta, \varepsilon, \delta, t}(S, Q_{TD})$  with the finite domains  $\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_d$  from Lemma 2 and outputs its output, and let  $X = \max_{i \in [d]} \{|\tilde{\mathcal{X}}_i|\}$ . Then*

1. **Privacy:**  $\mathbf{A}$  is  $(\varepsilon \cdot \sqrt{2d \ln(1/\delta')}, d\delta + \delta')$ -differentially private for any choice of  $\delta' > 0$ .
2. **Accuracy:** Let  $\lambda_{opt} |S|$  be the Tukey depth of the Tukey median in  $S \in \mathcal{X}^n$ . Assuming that  $t = n_{IP}(X, \beta, \varepsilon, \delta)$  (the sample complexity of `PrivateIP` in Theorem 8) and that

$$\begin{aligned} n &\geq m(\alpha, \beta) \cdot n_{IP}(X, \beta, \varepsilon, \delta) \in O\left(\frac{d \log(d/\alpha) + \log(1/\beta)}{\alpha^2} \cdot \frac{\log^*(|\mathcal{X}| + d) \cdot \log^2\left(\frac{\log^*(|\mathcal{X}| + d)}{\beta\delta}\right)}{\varepsilon}\right) \\ &= \tilde{O}\left(\frac{\log^* |\mathcal{X}| \cdot (d + \log(1/\beta)) \cdot \log^2(1/\delta)}{\varepsilon \alpha^2}\right), \end{aligned}$$

then with probability  $1 - (t + d)\beta$ ,  $\mathbf{A}(S)$  outputs a point  $\hat{x} \in \mathbb{R}^d$  with  $TD_S(\hat{x}) \geq (\lambda_{opt} - 2d\alpha)n$ .

Recall that by Fact 2, the Tukey median of any  $S \in \mathcal{X}^d$  has Tukey depth at least  $\lambda_{opt}n$  for  $\lambda_{opt} = \frac{1}{d+1}$ . Thus, by substituting  $\varepsilon$  by  $\frac{\varepsilon}{\sqrt{2d \ln(2/\delta)}}$ ,  $\delta$  by  $\frac{\delta}{2d}$ ,  $\delta'$  by  $\delta/2$ ,  $\alpha$  by  $\frac{\alpha}{2d(d+1)}$ , and  $\beta$  by  $\frac{\beta}{t+d}$ , we obtain our main theorem for estimating the Tukey median.

**Theorem 12** (Restatement of Theorem 3). *Let  $\varepsilon > 0$ ,  $\delta, \alpha, \beta \in (0, 1)$ ,  $n, d \in \mathbb{N}$  and let  $\mathcal{X} \subseteq \mathbb{R}$  be a finite domain. There exists an  $(\varepsilon, \delta)$ -differentially private algorithm  $A: (\mathcal{X}^d)^n \rightarrow \mathbb{R}^d$  and a value*

$$n_{min} \in \tilde{O}\left(\frac{d^{4.5} \cdot \log^* |\mathcal{X}| \cdot (d + \log(1/\beta)) \cdot \log^{2.5}(1/\delta)}{\varepsilon \alpha^2}\right) = \tilde{O}_{\alpha, \beta, \varepsilon, \delta}(d^{5.5} \cdot \log^* |\mathcal{X}|),$$

such that if  $n \geq n_{min}$ , then for any  $S \in (\mathcal{X}^d)^n$  it holds that

$$\Pr_{\hat{x} \sim A(S)} \left[ TD_S(\hat{x}) \geq \frac{1 - \alpha}{d + 1} \cdot n \right] \geq 1 - \beta.$$

## 5 Privately Learning Halfspaces via Privately Solving Linear Feasibility Problems

In this section, we show how to use `IPConcaveHighDim` (Algorithm 2) to solve linear feasibility problems privately and thus imply a private half-spaces learner. We give additional preliminaries in Section 5.1, and describe our algorithm for solving linear feasibility problems in Section 5.2.

### 5.1 Additional Preliminaries for Linear Feasibility Problems

**Notation.** For  $a \in \mathbb{Z}^+$  we use  $[[a]]$  to denote the set  $\{-a, -a + 1, \dots, a\}$ . For two sets of integers  $A, B$  and a scalar  $c$ , we define  $A/B = \{a/b : a \in A \wedge b \in B\}$  and  $c \cdot A = \{ca : a \in A\}$ . For  $x = (x_1, \dots, x_d)$  and  $y = (y_1, \dots, y_d)$ , we denote by  $\langle x, y \rangle = \sum_{i=1}^d x_i y_i$  the inner-product of  $x$  and  $y$ .

Recall that for  $a \in \mathbb{R}^d$  and  $w \in \mathbb{R}$ , we define the predicate  $h_{a,w}(x) = (\langle a, x \rangle \geq w)$ . Let  $H_{a,w}$  be the halfspace  $\{x \in \mathbb{R}^d : h_{a,w}(x) = 1\}$ .

**Linear Feasibility Problem [17].** Let  $X \in \mathbb{N}$  be a parameter and  $\mathcal{X} = [[X]]$ . In a linear feasibility problem, we are given a feasible collection  $S \in (\mathcal{X}^{d+1})^n$  of  $n$  linear constraints over  $d$  variables  $x_1, \dots, x_d$ . The target is to find a solution in  $\mathbb{R}^d$  that satisfies all (or most) constraints. Each constraint has the form  $h_{a,w}(x) = 1$  for some  $a \in \mathcal{X}^d$  and  $w \in \mathcal{X}$ . I.e., a linear feasibility problem is an LP problem with integer coefficients between  $-X$  and  $X$  and without an objective function.

**Definition 14** ( $(\alpha, \beta)$ -solving  $(X, d, n)$ -linear feasibility [17]). *We say that an algorithm  $(\alpha, \beta)$ -solves  $(X, d, n)$ -linear feasibility if for every feasible collection of  $n$  linear constraints over  $d$  variables with coefficients in  $\mathcal{X}$ , with probability  $1 - \beta$  the algorithm finds a solution  $(x_1, \dots, x_d)$  that satisfies at least  $(1 - \alpha)n$  constraints.*

Notice that there exists a reduction from PAC learning of halfspaces in  $d$  dimensions when the domain of labeled examples is  $\mathcal{X}^d$  to solving linear feasibility. Each input labeled point  $((x_1, \dots, x_d), y) \in \mathcal{X}^d \times \{-1, 1\}$  is transformed to a linear constraint  $h_{y \cdot (x_1, \dots, x_d, -1), 0}$ . By Theorem 6, if we can  $(\alpha, \beta)$ -solve  $(X, d + 1, n)$  linear feasibility problems with  $n \geq \tilde{O}_{\alpha, \beta}(d)$ , then the reduction results in an  $(O(\alpha), O(\beta))$ -PAC learner for  $d$  dimensional halfspaces.<sup>7</sup>

For any point, it is natural to consider how many linear constraints it satisfies. We define it as the *depth* of this point. However, as mentioned by [17], the *depth* function is not quasi-concave, i.e., for some points  $p_1, \dots, p_k$ , the *depth* of the point inside the convex hull of  $p_1, \dots, p_k$  cannot guarantee to be as high as  $p_1, \dots, p_k$  (see the left figure of Figure 1,  $A$  and  $B$  have *depth* 1, but  $C$  only has *depth* 0). Kaplan et. al. [17] consider the convexification of the *depth*, which is called *cdepth*. The property of *cdepth* guarantees that the *cdepth* of any point inside the convex hull of  $p_1, \dots, p_k$  is as high as that of  $p_1, \dots, p_k$  (see the right figure of Figure 1,  $A$  and  $B$  have *cdepth* 1,  $C$  also has *cdepth* 1).

<sup>7</sup>We remark that the reduction only works smoothly when the points are assumed to be in general position, but this assumption can be eliminated (see Section 5.1 in [17]).

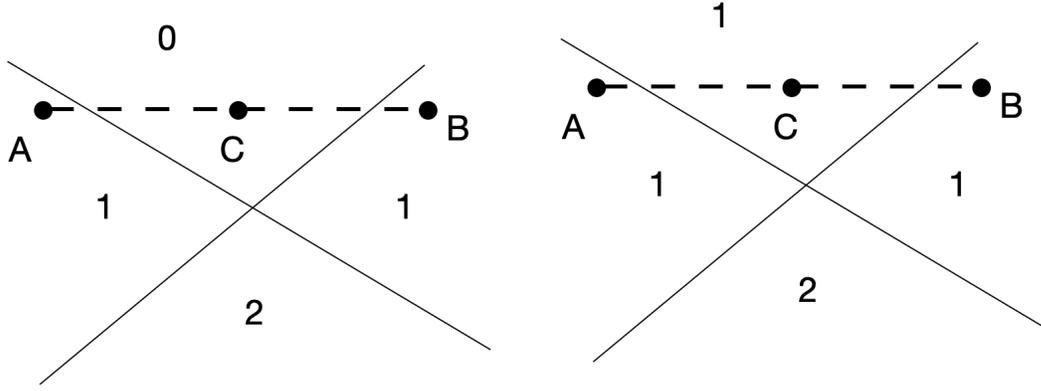


Figure 1: Depth and cdepth

**Definition 15** (Convexification of a function  $f : (\mathcal{X}^{d+1})^* \times \mathbb{R}^d \rightarrow \mathbb{R}$  [17]). For  $S \in (\mathcal{X}^{d+1})^*$  and  $y \in \mathbb{R}$ , let  $\mathcal{D}_S(y) = \{z \in \mathbb{R}^d : f(S, z) \geq y\}$ . The convexification of  $f$  is the function  $f_{Conv} : \mathcal{X}^* \times \mathbb{R}^d \rightarrow \mathbb{R}$  defined by  $f_{Conv}(S, x) = \max\{y \in \mathbb{R} : x \in \text{ConvexHull}(\mathcal{D}_S(y))\}$ .

I.e., if  $x$  is in the convex hull of points  $Z \subset \mathbb{R}^d$  then  $f_{Conv}(S, x)$  is at least  $\min_{z \in Z}(f(S, z))$ .

**Definition 16** (depth and cdepth [17]). Let  $S$  be a collection of predicates. Define  $\text{depth}_S(x)$  to be the number of predicates  $h_{a,w}$  in  $S$  such that  $h_{a,w}(x) = 1$ . Let  $\text{cdepth}_S(x) = f_{Conv}(S, x)$  for the function  $f(S, x) = \text{depth}_S(x)$ .

Similarly to Fact 1, we have

**Fact 3.** Let  $S$  be a set of  $n$  linear constraints and  $p_1, \dots, p_k$  be points with  $\text{cdepth}_S(p_i) \geq \lambda n$  for all  $i \in [k]$ . Then any point  $p$  in the convex hull of  $\{p_i\}_{i \in [k]}$  satisfies  $\text{cdepth}_S(p) \geq \lambda n$ .

*Proof.* By the assumption, each of the points  $p_i$  can be written as a convex combination  $p_i = \sum_{j \in [k_i]} \eta_{i,j} y_{i,j}$  of  $k_i$  points  $y_{i,1}, \dots, y_{i,k_i}$  with  $\text{depth}_S(y_{i,j}) \geq \lambda n$ . Since  $p$  is in the convex hull of  $p_1, \dots, p_k$ , it is also in the convex hull of  $\{y_{i,j}\}$ .  $\square$

**Fact 4** ([17]). For any  $S \in (\mathbb{R}^d \times \mathbb{R})^*$  and any  $x \in \mathbb{R}^d$ , it holds that

$$\text{depth}_S(x) \geq (d+1) \cdot \text{cdepth}_S(x) - d|S|.$$

By the above fact, if we can find a point with  $\text{cdepth}_S \geq (1-\alpha)|S|$  where  $\alpha \ll 1/(d+1)$ , then this point has  $\text{depth}_S \approx |S|$ .

Analogously to Lemma 2, we define the domain extension for linear feasibility problems. Unlike in Lemma 2, where the extension did not depend on the input for the problem of approximating the Tukey median, the extension for linear feasibility depends on the input. Given a collection  $S$  of predicates, the extension is computed iteratively, coordinate by coordinate, where the input for the  $i^{\text{th}}$  iteration is a prefix  $(x_1^*, \dots, x_{i-1}^*)$  in the extension obtained in iterations 1 to  $i-1$ .

**Definition 17** (domain extension for linear feasibility [17]). We define the  $d$  domains  $\{\tilde{\mathcal{X}}_i\}_{i=1}^d$  iteratively. For  $i = 1$ , let  $\tilde{\mathcal{X}}_1 = \tilde{\mathcal{X}}'_1 / \tilde{\mathcal{X}}''_1$  where  $\tilde{\mathcal{X}}'_1 := [[(d \cdot d!) \cdot X^d]]$  and  $\tilde{\mathcal{X}}''_1 := ([[d! \cdot X^d]]) \setminus \{0\}$ . For  $i > 1$  and given  $x_{i-1}^* = s_{i-1}/t_{i-1} \in \tilde{\mathcal{X}}_{i-1}$  where  $s_{i-1} \in \tilde{\mathcal{X}}'_{i-1}$  and  $t_{i-1} \in \tilde{\mathcal{X}}''_{i-1}$ , let  $\tilde{\mathcal{X}}_i = \tilde{\mathcal{X}}'_i / \tilde{\mathcal{X}}''_i$  where  $\tilde{\mathcal{X}}'_i = [[(d \cdot d!)^i \cdot X^{di}]]$  and  $\tilde{\mathcal{X}}''_i = ([[d! \cdot t_{i-1} \cdot X^d]]) \setminus \{0\}$ .

**Theorem 13** ([17]). Let  $X \in \mathbb{N}$ ,  $\mathcal{X} \in [[\pm X]]$ ,  $S \in (\mathcal{X}^d \times \mathcal{X})^*$ ,  $i \in [d]$ . Assume that  $\tilde{\mathcal{X}}_j$  is according to Definition 17 for all  $j \in [i]$  where for  $j < i$  the coordinates  $x_j^*$  are picked from  $\tilde{\mathcal{X}}_j$ . Then there exists  $x_i^* \in \tilde{\mathcal{X}}_i$  such that

$$\max_{\tilde{x}_{i+1}, \dots, \tilde{x}_d \in \mathbb{R}} \text{cdepth}_S(x_1^*, \dots, x_{i-1}^*, x_i^*, \tilde{x}_{i+1}, \dots, \tilde{x}_d) = \max_{\tilde{x}_i, \dots, \tilde{x}_d \in \mathbb{R}} \text{cdepth}_S(x_1^*, \dots, x_{i-1}^*, \tilde{x}_i, \tilde{x}_{i+1}, \dots, \tilde{x}_d)$$

The following lemma gives the VC dimension of the linear feasibility problem so that we can apply the VC theory on it. The lemma is closely related to the fact that the VC dimension of  $d$ -dimensional halfspaces is  $d$  [25, 19, 21, 22], and we give the full proof details in Appendix B.

**Lemma 3.** [VC dimension of linear feasibility] Let  $X_{\text{Halfspace}} = \{H_{a,w} \mid (a,w) \in \mathbb{R}^{d+1}\}$ . For a point  $p \in \mathbb{R}^d$ , let  $r_p = \{H_{a,w} \mid p \in H_{a,w}\}$ , and let  $R_{\text{Points}} = \{r_p \mid p \in \mathbb{R}^d\}$ . The VC dimension of  $(X_{\text{Halfspace}}, R_{\text{Points}})$  is  $d$ .

Note that for a dataset  $S = \{(a_1, w_1), \dots, (a_n, w_n)\} \in (\mathbb{R}^{d+1})^n$  and a point  $p \in \mathbb{R}^d$ , the depth of  $p$  in  $S$  is  $|\{H_{a_1, w_1}, \dots, H_{a_n, w_n}\} \cap r_p|$ . Hence, the following statement is an immediate corollary of Theorem 5 and Lemma 3.

**Corollary 2.** Let  $S \subseteq (\mathbb{R}^d)^n$  and let  $S' \subseteq S$  be a random subset of  $S$  with cardinality  $m = |S'| \geq O\left(\frac{d \cdot \log(\frac{d}{\alpha}) + \log \frac{1}{\beta}}{\alpha^2}\right)$ . Then, with probability at least  $1 - \beta$ , for all  $p \in \mathbb{R}^d$ , if  $\text{depth}_{S'}(p) = \gamma' m$  and  $\text{depth}_S(p) = \gamma n$  then  $|\gamma - \gamma'| \leq \alpha$ .

In the following, we show that the same approximation guarantee holds also with respect to the  $\text{cdepth}$  function.

**Corollary 3.** Let  $S \subseteq (\mathbb{R}^d)^n$  and let  $S' \subseteq S$  be a random subset of  $S$  with cardinality  $m = |S'| \geq O\left(\frac{d \cdot \log(\frac{d}{\alpha}) + \log \frac{1}{\beta}}{\alpha^2}\right)$ . Then, with probability at least  $1 - \beta$ , for all  $p \in \mathbb{R}^d$ , if  $\text{cdepth}_{S'}(p) = \gamma' m$  and  $\text{cdepth}_S(p) = \gamma n$  then  $|\gamma - \gamma'| \leq \alpha$ .

*Proof.* Assume  $p$  is a point with  $\text{cdepth}_S(p) = \gamma n$ . By the definition of  $\text{cdepth}$ , there exists points  $p_1, \dots, p_k$ , such that  $\text{depth}_S(p_i) \geq \gamma n$  for all  $i \in [k]$  and  $p \in \text{ConvexHull}(p_1, \dots, p_k)$ . By Corollary 2, with probability  $1 - \beta$ ,  $\text{depth}_{S'}(p_i) \geq (\gamma - \alpha)m$  for all  $i \in [k]$ . Therefore,  $\text{cdepth}_{S'}(p) \geq (\gamma - \alpha)m$ . The proof for the other direction is similar.  $\square$

## 5.2 Privately Solving Linear Feasibility Problems

We use `IPConcaveHighDim` (Algorithm 2) to solve linear feasibility problems privately. Define the function

$$Q_{LF}(S, x) = \frac{\text{cdepth}_S(x)}{|S|}.$$

Here we verify that  $Q_{LF}(S, x)$  satisfies the accuracy requirements in Theorem 10.

1. **Approximation by a random subset:** By Corollary 3, a random subset  $S' \subseteq S$  of size

$$m(\alpha, \beta) \in O\left(\frac{d \cdot \log(d/\alpha) + \log(1/\beta)}{\alpha^2}\right)$$

is an  $\alpha$ -approximation of  $S$  with respect to  $Q_{LF}$  with probability  $1 - \beta$ .

2. **Concavity:** It is guaranteed by Fact 3.

3. **Proper finite domains:** The domain extension  $\tilde{\mathcal{X}}_1, \dots, \tilde{\mathcal{X}}_d$  in Definition 17 and Theorem 13 provide proper finite domains for  $Q_{LF}$  (Definition 12) with maximal domain size  $X$  where  $\log^* X \in O(\log^*(|\mathcal{X}| + d))$ .

Thus, the following theorem is an immediate corollary of Theorem 10.

**Theorem 14.** *Let  $\varepsilon > 0$ ,  $\delta, \alpha, \beta \in (0, 1)$ ,  $n, d, t \in \mathbb{N}$  with  $n \geq t$  and let  $\mathcal{X} \subseteq \mathbb{R}$  be a finite domain. Let  $\mathbf{A}: (\mathcal{X}^d)^n \rightarrow \mathbb{R}^d$  be the algorithm that on input  $S$ , computes  $\text{IPConcaveHighDim}_{\alpha, \beta, \varepsilon, \delta, t}(S, Q_{LF})$  with the finite domains  $\tilde{\mathcal{X}}_1, \tilde{\mathcal{X}}_2, \dots, \tilde{\mathcal{X}}_d$  from Definition 17 and outputs its output, and let  $X = \max_{i \in [d]} \{|\tilde{\mathcal{X}}_i|\}$ . Then*

1. **Privacy:**  $\mathbf{A}$  is  $(\varepsilon \cdot \sqrt{2d \ln(1/\delta')}, d\delta + \delta')$ -differentially private for any choice of  $\delta' > 0$ .
2. **Accuracy:** Assuming that  $t = n_{IP}(X, \beta, \varepsilon, \delta)$  (the sample complexity of  $\text{PrivateIP}$  in Theorem 8) and that

$$\begin{aligned} n \geq m(\alpha, \beta) \cdot n_{IP}(X, \beta, \varepsilon, \delta) &\in O\left(\frac{d \log(d/\alpha) + \log(1/\beta)}{\alpha^2} \cdot \frac{\log^*(|\mathcal{X}| + d) \cdot \log^2\left(\frac{\log^*(|\mathcal{X}| + d)}{\beta\delta}\right)}{\varepsilon}\right) \\ &= \tilde{O}\left(\frac{\log^*|\mathcal{X}| \cdot (d + \log(1/\beta)) \cdot \log^2(1/\beta\delta)}{\varepsilon\alpha^2}\right), \end{aligned}$$

Then with probability  $1 - (t + d)\beta$ ,  $\mathbf{A}(S)$  outputs a point  $\hat{x} \in \mathbb{R}^d$  with  $\text{cdepth}_S(\hat{x}) \geq (1 - 2d\alpha)n$ .

Combining the corollary with Fact 4, we have the following result.

**Corollary 4.** *With probability  $1 - (t + d)\beta$ , the point  $\hat{x}$  satisfies  $\text{depth}_S(\hat{x}) \geq (1 - 2d\alpha - 2d^2\alpha) \cdot |S|$ .*

Thus, by substituting  $\varepsilon$  by  $\frac{\varepsilon}{\sqrt{2d \ln(2/\delta)}}$ ,  $\delta$  by  $\frac{\delta}{2d}$ ,  $\delta'$  by  $\delta/2$ ,  $\alpha$  by  $\frac{\alpha}{4d^2}$ , and  $\beta$  by  $\frac{\beta}{t+d}$ , we obtain our main theorem for solving linear feasibility problem.

**Theorem 15.** *Let  $\varepsilon > 0$ ,  $\delta, \alpha, \beta \in (0, 1)$ ,  $n, d \in \mathbb{N}$  and let  $\mathcal{X} \subseteq \mathbb{R}$  be a finite domain. There exists an  $(\varepsilon, \delta)$ -differentially private algorithm  $\mathbf{A}: (\mathcal{X}^d)^n \rightarrow \mathbb{R}^d$  and a value*

$$n_{min} \in \tilde{O}\left(\frac{d^{4.5} \cdot \log^*|\mathcal{X}| \cdot (d + \log(1/\beta)) \cdot \log^2(1/\beta\delta) \cdot \log^{0.5}(1/\delta)}{\varepsilon\alpha^2}\right) = \tilde{O}_{\alpha, \beta, \varepsilon, \delta}(d^{5.5} \cdot \log^*|\mathcal{X}|),$$

such that if  $n \geq n_{min}$ , then for any  $S \in (\mathcal{X}^d)^n$  it holds that

$$\Pr_{\hat{x} \sim \mathbf{A}(S)}[\text{depth}_S(\hat{x}) \geq (1 - \alpha) \cdot n] \geq 1 - \beta.$$

### 5.3 Privately Learning Halfspaces

Combining Theorems 15 and 6, we get the following result.

**Theorem 16** (Restatement of Theorem 4). *Let  $\mathcal{X} \subset \mathbb{R}$  be a finite domain. There exists an  $(\varepsilon, \delta)$ -differentially private  $(\alpha, \beta)$ -PAC learner for halfspaces over examples from  $\mathcal{X}^d$  with sample complexity  $n = \tilde{O}_{\alpha, \beta, \varepsilon, \delta}(d^{5.5} \cdot \log^*|\mathcal{X}|)$ .*

## References

- [1] N. Alon, D. Haussler, and E. Welzl. Partitioning and geometric embedding of range spaces of finite vapnik-chervonenkis dimension. In *Proceedings of the Third Annual Symposium on Computational Geometry*, SCG '87, page 331–340, New York, NY, USA, 1987. Association for Computing Machinery.
- [2] N. Alon, R. Livni, M. Malliaris, and S. Moran. Private PAC learning implies finite littlestone dimension. In M. Charikar and E. Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 852–860. ACM, 2019.

- [3] N. Alon, S. Moran, and A. Yehudayoff. Sign rank versus vapnik-chervonenkis dimension. *Sbornik: Mathematics*, 208(12):1724, 2017.
- [4] A. Beimel, S. Moran, K. Nissim, and U. Stemmer. Private center points and learning of halfspaces. In *Conference on Learning Theory*, pages 269–282. PMLR, 2019.
- [5] A. Beimel, K. Nissim, and U. Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, pages 363–378, 2013.
- [6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, Oct. 1989.
- [7] M. Bun, K. Nissim, U. Stemmer, and S. P. Vadhan. Differentially private release and learning of threshold functions. In *FOCS*, pages 634–649, 2015.
- [8] K. L. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with iterated radon points. In *Proceedings of the Ninth Annual Symposium on Computational Geometry, SCG '93*, page 91–98, New York, NY, USA, 1993. Association for Computing Machinery.
- [9] E. Cohen, X. Lyu, J. Nelson, T. Sarlós, and U. Stemmer. Optimal differentially private learning of thresholds and quasi-concave optimization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023*, pages 472–482, 2023.
- [10] L. Danzer, B. Grünbaum, and V. Klee. Helly’s theorem and its relatives. 1963.
- [11] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.
- [12] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.
- [13] V. Feldman and D. Xiao. Sample complexity bounds on differentially private learning via communication complexity. *SIAM J. Comput.*, 44(6):1740–1764, 2015.
- [14] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. In *SCG '86*, 1986.
- [15] Y. H. Huang, W.-H. Chen, and S.-C. Tsai. On the sample complexity of privately learning half-spaces. In *Proceedings of the 16th Asian Conference on Machine Learning*, volume 260 of *Proceedings of Machine Learning Research*, pages 655–670. PMLR, 2025.
- [16] H. Kaplan, K. Ligett, Y. Mansour, M. Naor, and U. Stemmer. Privately learning thresholds: Closing the exponential gap. In *COLT*, pages 2263–2285, 2020.
- [17] H. Kaplan, Y. Mansour, U. Stemmer, and E. Tsfadia. Private learning of halfspaces: Simplifying the construction and reducing the sample complexity. *Advances in Neural Information Processing Systems*, 33:13976–13985, 2020.
- [18] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011.
- [19] J. Matousek. *Lectures on Discrete Geometry*. Graduate Texts in Mathematics. Springer New York, 2013.
- [20] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84. ACM, 2007.
- [21] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145 – 147, 1972.

- [22] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.
- [23] J. W. Tukey. *Mathematics and the picturing of data*. 1975.
- [24] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, Nov. 1984.
- [25] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

## A Approximated Functions Have Low Sensitivity

In this section, we show that any approximated function has low sensitivity.

**Theorem 17.** *If  $(S, Q)$  can be  $(\alpha, 1/3, m)$ -approximated for any  $S$  (Definition 10), then for any neighboring datasets  $S_1, S_2$  with size at least  $4m$ , we have  $|Q(S_1, x) - Q(S_2, x)| \leq 2\alpha$ .*

*Proof.* Assume  $|S_1| = |S_2| = n$  and  $x_1$  and  $x_2$  are the different entries in the two datasets. For each of  $S_1$  and  $S_2$ , the number of  $m$ -size subsets is  $\binom{n}{m}$ . So for each of  $S_1$  and  $S_2$ , the number of  $\alpha$ -approximation is at least  $\frac{2\binom{n}{m}}{3}$ . Since the number of  $m$ -size subsets containing  $x_1$  or  $x_2$  are at most  $\binom{n-1}{m-1}$ , thus the number of  $\alpha$ -approximation that does not containing  $x_1$  or  $x_2$  is at least  $\left(\frac{2\binom{n}{m}}{3} + \frac{2\binom{n}{m}}{3}\right) - \binom{n}{m} - \binom{n-1}{m-1} = \left(\frac{1}{3} - \frac{m}{n}\right) \cdot \binom{n}{m} > 0$ . Let  $S'$  be one of these  $\alpha$  approximations. Then  $|Q(S_1, x) - Q(S_2, x)| \leq |Q(S_1, x) - Q(S', x)| + |Q(S', x) - Q(S_2, x)| \leq 2\alpha$   $\square$

## B VC Dimension of Linear Feasibility Problem

In this section, we give a proof of Lemma 3, restated below.

**Lemma 4** (Restatement of Lemma 3). *Let  $X_{Halfspace} = \{H_{a,w} \mid (a,w) \in \mathbb{R}^{d+1}\}$ . For a point  $p \in \mathbb{R}^d$ , let  $r_p = \{H_{a,w} \mid p \in H_{a,w}\}$ , and let  $R_{Points} = \{r_p \mid p \in \mathbb{R}^d\}$ . The VC dimension of  $(X_{Halfspace}, R_{Points})$  is  $d$ .*

The proof has two parts. Recall that for  $a \in \mathbb{R}^d$  and  $w \in \mathbb{R}$  we define the predicate  $h_{a,w}(x) = (\langle a, x \rangle \geq w)$ . Let  $H_{a,w}$  be the halfspace  $\{x \in \mathbb{R}^d \mid h_{a,w}(x) = 1\}$ .

**1. There exists  $d$  halfspaces that can be shattered.** Consider  $d$  halfspaces  $S = \{H_{e_1,0}, \dots, H_{e_d,0}\}$ , where  $e_i = 0^{i-1} \times 1 \times 0^{d-i}$ . Note that for every  $x \in \{-1, 1\}^d$  and every  $i \in [d]$ :  $H_{e_i,0} \in r_x \iff x_i = 1$ , where recall that  $r_x = \{H_{a,w} \mid x \in H_{a,w}\}$ . Therefore,  $S$  is shattered by  $\{r_x\}_{x \in \{-1, 1\}^d}$ .

**2. Every  $d + 1$  halfspaces cannot be shattered.** For any halfspace  $H_{a,w}$ , and any point  $p$  on one side of the hyperplane  $\{x \mid \langle a, x \rangle + w = 0\}$ , the function  $r_p$  give the same label to  $H_{a,w}$ . Given  $d + 1$  halfspaces, consider the hyperplane arrangement of their corresponding hyperplanes. Each cell of the hyperplane arrangement can be uniquely projected to one dichotomy of  $d + 1$  halfspaces (see an example in Figure 2). By Theorem 18, the number of all possible dichotomies is at most  $\sum_{i=0}^d \binom{d+1}{d} < 2^{d+1}$ . Thus  $d + 1$  halfspaces cannot be shattered.

**Theorem 18** ([19] Page 127, Proposition 6.1.1). *The maximum number of cells for  $n$  hyperplanes with  $n > d$  in  $\mathbb{R}^d$  is  $\sum_{i=0}^d \binom{n}{d}$ .*

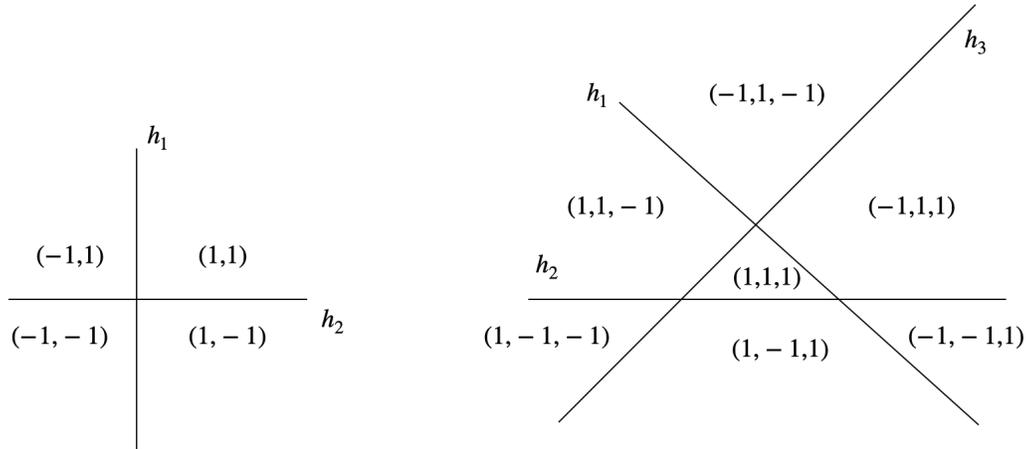


Figure 2: 2 halfspaces can be shattered (left) and 3 halfspaces cannot be shattered (right) in the 2-D plane

## C Huang et al. [15]’s Algorithm is Not Differentially Private

Huang et al. [15] present an algorithm that they claim is differentially private and learns halfspaces over examples from  $\mathcal{X}^d$  with sample complexity  $\tilde{O}(d \cdot \log^* |\mathcal{X}|)$ . In this section, we explain why their algorithm is inherently not differentially private. For simplicity, we even focus on their 2-dimensional version (described in their Section 3.1).

Huang et al. [15]’s 2-dimensional learning algorithm is focused on learning halfspaces that go through the origin. Each such halfspace can be uniquely represented by an angle  $\phi \in [0, 2\pi)$  (denote this halfspace by  $h_\phi$ ), so their goal is to describe an algorithm  $\mathbf{A}: (\mathcal{X}^2 \times \{-1, 1\})^* \rightarrow [0, 2\pi)$  that is: (1)  $(\epsilon, \delta)$ -differentially private, and (2) Given a dataset  $S \in (\mathcal{X}^2 \times \{-1, 1\})^n$  with  $n \geq \tilde{\Theta}_{\alpha, \beta, \epsilon, \delta}(\log^* |\mathcal{X}|)$  that is realizable by some  $h_\phi$  (unknown  $\phi \in [0, 2\pi)$ ), with probability  $1 - \beta$ ,  $\mathbf{A}(S)$  outputs  $\phi^*$  such that  $|\{(x, y) \in S: h_{\phi^*}(x) = y\}| \geq (1 - \alpha)n$  (i.e.,  $\mathbf{A}$  is  $(\alpha, \beta)$ -empirical learner).

Huang et al. [15] first define a finite discretization  $\mathcal{H}_\gamma$  of  $[0, 2\pi)$  with size  $O(|\mathcal{X}|^2)$  such that if  $S \in \mathcal{X}^d$  is realizable over  $[0, 2\pi)$ , then it is also realizable over the grid  $\mathcal{H}_\gamma$ .

---

### Algorithm 3: MakeData

---

**Inputs:**  $\epsilon > 0$ ,  $\mathcal{H}_\gamma \subseteq [0, 2\pi)$ ,  $S \in (\mathcal{X}^2 \times \{-1, 1\})^*$ .

**Operation:**

1.  $S_{\mathcal{H}} \leftarrow \emptyset$ .
  2. for  $\phi \in \mathcal{H}_\gamma$  do:
    - (a)  $n_\phi = |\{(x, y) \in S: |\phi(x) - \phi| < \gamma \text{ and } h_\phi(x) = y\}|$ .
    - (b) Add  $\max\{\lceil n_\phi + \text{Lap}(1/\epsilon) \rceil, 1\}$  copies of  $\phi$  to  $S_{\mathcal{H}}$ .
  3. Return  $S_{\mathcal{H}}$ .
-

---

**Algorithm 4:** MakeThrData

---

**Inputs:**  $S_{\mathcal{H}} \in ([0, 2\pi])^*$ ,  $S \in (\mathcal{X}^2 \times \{-1, 1\})^*$ ,  $C \in \mathbb{N}$ .

**Operation:**

1. Calculate  $q(S, \phi) = |\{(x, y) \in S : h_{\phi}(x) = y\}|$  for every  $\phi \in S_{\mathcal{H}}$ .
  2. Let  $\max_C(S_{\mathcal{H}})$  be the  $C$  largest elements in  $S_{\mathcal{H}}$  according to the lexicographic order of  $(q(S, \phi), \phi)$ ;
  3. Randomly select  $\phi' \in S_{\mathcal{H}} \setminus \max_C(S_{\mathcal{H}})$  and rotate the coordinate so that  $\phi' = 0$ ;
  4. Let  $\phi^* := \operatorname{argmax}_{\phi \in \max_C(S_{\mathcal{H}})} \{q(S, \phi)\}$ .
  5.  $S_{Thr} \leftarrow \emptyset$ .
  6. For  $\phi \in \max_C(S_{\mathcal{H}})$  do
    - (a)  $y \leftarrow 1$  if  $\phi \leq \phi^*$ ; otherwise,  $y \leftarrow -1$ .
    - (b) Add  $(\phi, y)$  to  $S_{Thr}$ .
- return  $S_{Thr}$ .
- 

---

**Algorithm 5:**  $A_{SimpleH}$ 

---

**Inputs:**  $\varepsilon, \delta, \alpha, \beta > 0$ ,  $S \in (\mathcal{X}^2 \times \{-1, 1\})^*$ ,  $\gamma \in [0, 2\pi)$ , and  $(\varepsilon, \delta)$ -differentially private  $(\alpha, \beta)$ -empirical learner  $A_{Thr}$  that privately learn thresholds over  $\mathcal{X}_{Thr}$  with sample complexity  $n_{Thr} = n_{Thr}(\mathcal{X}_{Thr}, \varepsilon, \delta, \alpha, \beta)$ .

**Operation:**

1.  $\mathcal{H}_{\gamma} \leftarrow \text{Discretize}(\gamma)$ ;
  2.  $S_{\mathcal{H}} \leftarrow \text{MakeData}(\varepsilon, \mathcal{H}_{\gamma}, S)$ ;
  3.  $S_{Thr} \leftarrow \text{MakeThrData}(S_{\mathcal{H}}, S, n_{Thr})$ ;
  4. Apply  $A_{Thr}$  with input  $S_{Thr}$ , parameters  $\varepsilon, \delta, \alpha, \beta$ , and get  $\phi^*$ .
  5. Output  $\phi^*$ .
-

**Theorem 19** (Theorem 14 in [15]). *For any  $\varepsilon, \delta, \alpha, \beta \in (0, 1)$ , if there is an  $(\varepsilon, \delta)$ -differentially private  $(\alpha, \beta)$ -empirical learner  $A_{Thr}$  that learns thresholds on a finite domain  $\mathcal{X}_{Thr}$  with  $n_{Thr}(\mathcal{X}_{Thr}, \varepsilon, \delta, \alpha, \beta)$  samples, then with sample complexity*

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

$A_{SimpleH}$  (Algorithm 5) is an  $(\varepsilon, \delta)$ -differentially private  $(\alpha, \beta)$ -empirical learner for 2-dimensional half-spaces.

In the following, we prove that Theorem 19 is wrong by showing that Algorithm 5 is blatantly not differentially private.

**Counterexample to Theorem 19.** Let  $S$  be a dataset that consists of  $n/2 + 1$  copies of  $((1, 0), -1)$  (i.e., the point  $(1, 0)$  with a label  $-1$ ), and  $n/2 - 1$  copies of  $((-1, 0), -1)$ , and let  $S'$  be the neighboring dataset that is obtained by replacing one of the  $((1, 0), -1)$  in  $S$  with  $((-1, 0), -1)$ . We remark that although  $S$  and  $S'$  are not realizable (i.e., no halfspace agree on the labeled points), differential privacy is a *worst-case* guarantee and so  $A_{SimpleH}(S)$  (with privacy parameters  $\varepsilon, \delta$ ) should be  $(\varepsilon, \delta)$ -indistinguishable from  $A_{SimpleH}(S')$  by Theorem 19.<sup>8</sup>

Let  $S_{\mathcal{H}}, S_{Thr}$ , and  $\phi^*$  be the values computed in the execution  $A_{SimpleH}(S)$ , and let  $S'_{\mathcal{H}}, S'_{Thr}$ , and  $\phi'^*$  be corresponding values in the execution of  $A_{SimpleH}(S')$ . Note that both  $S_{\mathcal{H}}$  and  $S'_{\mathcal{H}}$  contain at least 1 copy of every angle in the grid  $\mathcal{H}_{\gamma}$ . Furthermore, note that in algorithm *MakeThrData*, for any grid angle  $\phi \in (0, \pi)$  we have  $q(S, \phi) = n/2 + 1$  and  $q(S', \phi) = n/2 - 1$ , and for any grid angle  $\phi \in (\pi, 2\pi)$  we have  $q(S, \phi) = n/2 - 1$  and  $q(S', \phi) = n/2 + 1$ . Therefore, the output  $S_{Thr}$  of *MakeThrData*( $S_{\mathcal{H}}, S$ ) is a dataset of angles in  $(0, \pi)$ , and the output  $S'_{Thr}$  of *MakeThrData*( $S'_{\mathcal{H}}, S'$ ) is a dataset of angles in  $(\pi, 2\pi)$ . Thus, the output  $\phi^*$  of  $A_{Thr}(S_{\mathcal{H}})$  is in  $(0, \pi)$ , and the output  $\phi'^*$  of  $A_{Thr}(S'_{\mathcal{H}})$  is in  $(\pi, 2\pi)$ , and thus we conclude that  $A_{SimpleH}(S)$  and  $A_{SimpleH}(S')$  are clearly distinguishable.

**What is wrong in [15]’s Privacy Analysis?** The main issue in [15]’s privacy analysis is the (wrong) argument that the composition of *MakeThrData* on top of *MakeData* is “differentially private” just because *MakeData* is differentially private, which is wrong since *MakeThrData* also uses the input dataset. As we demonstrated in our counter example, this composition is clearly not differentially private, and in fact it can result in completely disjoint outputs on neighboring datasets.

---

<sup>8</sup>We remark that it is also not hard to determine realizable datasets  $S, S'$  that break the privacy guarantee of  $A_{SimpleH}$ , but we chose the unrealizable ones as they make the arguments cleaner and simpler.