Differentially Private Approximations of a Convex Hull in Low Dimensions

Yue Gao
Department of Computing Science, University of Alberta, Edmonton, Canada

Or Sheffet
Faculty of Engineering, Bar-Ilan University, Ramat Gan, Israel

Abstract

We give the first differentially private algorithms that estimate a variety of geometric features of points in the Euclidean space, such as diameter, width, volume of convex hull, min-bounding box, min-enclosing ball, etc. Our work relies heavily on the notion of *Tukey-depth*. Instead of (non-privately) approximating the convex-hull of the given set of points P, our algorithms approximate the geometric features of $D_P(\kappa)$ – the κ-Tukey region induced by P (all points of Tukey-depth κ or greater). Moreover, our approximations are all bi-criteria: for any geometric feature μ our (α,Δ)-approximation is a value "sandwiched" between $(1-\alpha)\mu(D_P(\kappa))$ and $(1+\alpha)\mu(D_P(\kappa-\Delta))$.

Our work is aimed at producing a (α,Δ)-*kernel of* $D_P(\kappa)$, namely a set $\mathcal{S}$ such that (after a shift) it holds that $(1-\alpha)D_P(\kappa)\subset \mathsf{CH}(\mathcal{S})\subset(1+\alpha)D_P(\kappa-\Delta)$. We show that an analogous notion of a bi-critera approximation of a directional kernel, as originally proposed by [1], *fails* to give a kernel, and so we result to subtler notions of approximations of projections that do yield a kernel. First, we give differentially private algorithms that find (α,Δ)-kernels for a "fat" Tukey-region. Then, based on a private approximation of the min-bounding box, we find a transformation that does turn $D_P(\kappa)$ into a "fat" region *but only if* its volume is proportional to the volume of $D_P(\kappa-\Delta)$. Lastly, we give a novel private algorithm that finds a depth parameter κ for which the volume of $D_P(\kappa)$ is comparable to the volume of $D_P(\kappa-\Delta)$. We hope our work leads to the further study of the intersection of differential privacy and computational geometry.

2012 ACM Subject Classification Theory of computation → Theory of database privacy and security

Keywords and phrases Differential Privacy, Computational Geometry, Tukey Depth

Digital Object Identifier 10.4230/LIPIcs.ITC.2021.18

Related Version *Full Version*: `http://arxiv.org/abs/2007.08110`

Funding Much of this work was done when the first author was adviced by the second author, and was supported by grant #2017–06701 of the Natural Sciences and Engineering Research Council of Canada (NSERC). In addition, this work was partially done when O.S. was a participant of the Simons' Institute for the Theory of Computing program of Data-Privacy. O.S. is supported by the BIU Center for Research in Applied Cryptography and Cyber Security in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office, and by ISF grant no. 2559/20.

Acknowledgements Both authors thank the anonymous reviewers for many helpful suggestions in improving this paper.

1 Introduction

With modern day abundance of data, there are numerous datasets that hold the sensitive and personal details of individuals, yet collect only a few features per user. Examples of such *low-dimensional* datasets include locations (represented as points on the 2D-plane), medical data composed of only a few measurements (e.g., [25, 27]), or high-dimensional data restricted to a small subset of features. It is therefore up to us to make sure that the analyses of such sensitive datasets do not harm the privacy of their participants. Differentially private algorithms [10, 12] alleviate such privacy concerns as they guarantee that the presence or absence of any single individual in the dataset has only a limited affect on any outcome.

2nd Conference on Information-Theoretic Cryptography (ITC 2021).
Editor: Stefano Tessaro; Article No. 18; pp. 18:1–18:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

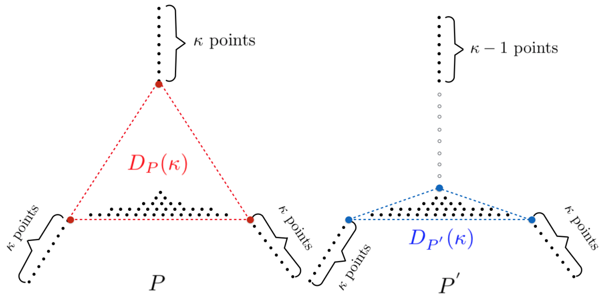

Figure 1 An example showing that $D_P(\kappa)$'s volume, width and min-enclosing triangle can be greatly affected by a single point in the input.

Often (usually motivated by data-visualization), understanding the geometric features of such low-dimensional datasets is a key step in their analysis. Yet, to this day, very little work has been done to establish differentially private algorithms that approximate the data's geometrical features. This should not come as a surprise seeing as most geometric features – such as diameter, width,[1] volume of convex-hull, min-bounding ball radius, etc. – are highly sensitive to the presence / absence of a single datum. Moreover, while it is known that differential privacy generalizes [11, 2], geometrical properties often do not: if the dataset P is composed on n i.i.d. draws from a distribution $\mathcal{P}$ then it might still be likely that, say, $\mathrm{diam}(P)$ and $\mathrm{diam}(\mathcal{P})$ are quite different.[2]

But differential privacy has already overcome the difficulty of large sensitivity in many cases, the leading example being the median – despite the fact that the median may vary greatly by the presence/absence of a single datum, we are still capable of privately approximating the median. The crux in differentially private median approximations [21, 4] is that the quality of the approximation is not measured by the actual distance between the true input-median and the result of the algorithm, but rather by the probability mass of the input's CDF "sandwiched" between the true median and the output of the private algorithm. A similar effect takes place in our work. While we deal with geometric concepts that exhibit large sensitivity, we formulate *robust* approximation guarantees of these concepts, guarantees that do generalize when the data is drawn i.i.d. from some unknown distribution. Yet unlike the private median approximations, our private kernel-approximation algorithm does not *always* return an output. It first verifies that certain niceness assumptions about the input hold; if they don't hold, it is capable of finding a sufficiently "deep" portion of the input which can be privately approximated. Details to follow.

Much like in previous works in differential privacy [3, 19], our approximation rely heavily on the notion of the *depth* of a point. Specifically, our approximation guarantees are with respect to *Tukey depth* [26]. Roughly speaking (see Section 2), a point x has Tukey depth κ w.r.t. a dataset P, denoted $\mathrm{TD}(x, P) = \kappa$, if the smallest set $S \subset P$ one needs to remove from P so that some hyperplane separates x from $P \backslash S$ has cardinality κ. This also allows us to define the *κ-Tukey region* $D_P(\kappa) = \{x \in \mathbb{R}^d : \ \mathrm{TD}(x, P) \geqslant \kappa\}$. So, for example, $D_P(0) = \mathbb{R}^d$ and $D_P(1) = \mathsf{CH}(P)$ (the convex-hull of P). It follows from the definition that for any $1 \leqslant \kappa_1 \leqslant \kappa_2$ we have $\mathsf{CH}(P) = D_P(1) \supset D_P(\kappa_1) \supset D_P(\kappa_2)$. It is known that for any dataset

[1] The min gap between two hyperplanes that "sandwich" the data.
[2] For example, consider $\mathcal{P}$ as a uniform distribution over $2n$ discrete points whose diameter greatly shrinks unless two specific points are drawn into P.

P and depth κ the Tukey-region $D_P(\kappa)$ is a convex polytope, and moreover (see [14]) that for any P of size n it holds that $D_P(n/(d+1)) \neq \emptyset$. Moreover, there exists efficient algorithms (in low-dimensions) that find $D_P(\kappa)$.

One pivotal property of the Tukey depth, which enables differentially private approximations, is that it exhibits low-sensitivity at any given point. As noted by [3], it follows from the very definition of Tukey-depth that if we add to / remove from P any single datum, then the depth of any given $x \in \mathbb{R}^d$ changes by no more than 1. And so, in this work, we give bi-criteria approximations of key geometric features of $D_P(\kappa)$ – where the quality of the approximation is measured both by a multiplicative factor and with respect to a *shallower* Tukey region. Given a measure μ of the convex polytope $D_P(\kappa)$, such as diameter, width, volume etc., we return a (α, Δ)-approximation of μ – a value lower bounded by $(1-\alpha)\mu(D_P(\kappa))$ and upper-bounded by $(1+\alpha)\mu(D_P(\kappa-\Delta))$. This implies that the quality of the approximation depends on *both* the approximation parameters fed into the algorithm and also on the "niceness" properties of the data. For datasets where $\mu(D_P(\kappa-\Delta)) \approx \mu(D_P(\kappa))$, our (α, Δ)-approximation is a good approximation of $\mu(D_P(\kappa))$; but for datasets where $\mu(D_P(\kappa-\Delta)) \gg \mu(D_P(\kappa))$ our guarantee is rather weak. Note that no differentially private algorithm can correctly report for all P whether $\mu(D_P(\kappa))$ and $\mu(D_P(\kappa-\Delta))$ are / are-not similar seeing as, as Figure 1 shows, such proximity can be highly affected by the existence of a single datum in P. Again, this is very much in line with private approximations of the median [21, 4].[3]

Our main goal in this work is to produce an *kernel* for $D_P(\kappa)$. Non privately, a α-kernel [1] of a dataset P is a set $\mathcal{S} \subset P$ where for any direction u it holds that $(1-\alpha)\max_{p,q\in P}\langle p-q, u\rangle \leqslant \max_{p,q\in\mathcal{S}}\langle p-q, u\rangle \leqslant \max_{p,q\in P}\langle p-q, u\rangle$. Agarwal et al. [1] showed that for any P there exists such a kernel whose size is $(1/\alpha)^{O(d)}$. (We thus assume $|P| \gg (1/\alpha)^{O(d)}$ for otherwise the non-private algorithm can trivially output P itself.) More importantly, the fact that $\mathcal{S}$ is a α-kernel implies that $(1-O(\alpha))\mathsf{CH}(P) \subset \mathsf{CH}(\mathcal{S}) \subset \mathsf{CH}(P)$. It is thus tempting to define an analogous notion of (α, Δ)-kernel as "for any direction u we have $(1-\alpha)\max_{p,q\in D_P(\kappa)}\langle p-q, u\rangle \leqslant \max_{p,q\in\mathcal{S}}\langle p-q, u\rangle \leqslant (1+\alpha)\max_{p,q\in D_P(\kappa-\Delta)}\langle p-q, u\rangle$" and hope that it yields that $(1-O(\alpha))D_P(\kappa) \subset \mathsf{CH}(\mathcal{S}) \subset D_P(\kappa-\Delta)$. Alas, that is *not* the case. Having $\mathcal{S} \subset D_P(\kappa)$ turns out to be a crucial component in arguing about the containment of the convex-hulls, and the argument breaks without it. We give a counter example in a later discussion (in Section 3). Therefore, viewing this directional-width approximation property as means to an end, we define the notion of (α, Δ)-kernel directly w.r.t. the containment of the convex bodies.

▶ **Definition 1.** *Given a dataset P and a parameter κ, a set $\mathcal{S}$ is called a* (α, Δ)-kernel *for $D_P(\kappa)$ if there exist two points c_1, c_2 such that $(1-\alpha)(D_P(\kappa)-c_1) \subset \mathsf{CH}(\mathcal{S})-c_1$ and $\mathsf{CH}(\mathcal{S})-c_2 \subset (1+\alpha)(D_P(\kappa-\Delta)-c_2)$.*

Non privately, the "center" points c_1 and c_2 may just as well be the origin, since we can shift the points so that the origin is in the convex-hull; but privately we cannot make such an assumption as it differentiates between two neighboring datasets. Note that in particular, a (α, Δ)-kernel gives the (α, Δ)-approximation of the projection along every direction u proposed earlier (in quotation-marks above). In fact, a (α, Δ)-kernel yields

[3] In particular, in the case where P is drawn from a distribution $\mathcal{P}$, it is known that $\forall x \in \mathbb{R}^d$, $|\frac{1}{n}\mathrm{TD}(x, P) - \mathrm{TD}(x, \mathcal{P})| = O(\sqrt{\frac{d\log(n)}{n}})$ [8], where $\mathrm{TD}(x, \mathcal{P})$ denotes the smallest measure $\mathcal{P}$ places on any halfspace containing x. Thus, if $D_P(\kappa)$ and $D_P(\kappa-\Delta)$ vary drastically, then it follows that the distribution $\mathcal{P}$ is "volatile" at depth $\frac{\kappa}{n}$.

(α, Δ)-approximations of numerous properties of $D_P(\kappa)$, like volume, min-bounding box, min-enclosing / max-enclosed ball radius, surface area, etc. Our work is the first to give a private approximation of *any* of these concepts.

The main caveat of our work is that we are able to output a (α, Δ)-kernel of $D_P(\kappa)$ only when $D_P(\kappa)$ satisfies some "niceness" properties. We briefly describe the structure of our work to better explain these properties and how they relate. We begin with preliminaries in Section 2. In Section 3 we give our algorithm for finding a kernel, which works under the premise that the width of $D_P(\kappa)$ is large. This means that our goal is complete if we are able to assert, using a private algorithm, that $D_P(\kappa)$ has large width (we design a heuristics for this purpose, but it is deferred to the full version of this work); or if we can find a value of κ for which $D_P(\kappa)$ can be privately transformed into a region with large width – a complicated task for which we require multiple "stepping stones" that are detailed in the following sections.

In Sections 4 and 5 we establish some basic privacy-preserving algorithms for tasks we require later.In Section 6, we give a private $(O(1), \Delta)$-approximation of the min-bounding box of $D_P(\kappa)$; and show that this box yields a transformation that turns $D_P(\kappa)$ into a region of large width, *but only if* the volumes of $D_P(\kappa)$ and $D_P(\kappa - \Delta)$ are comparable. So finally, in Section 7, we give an algorithm that finds a value of κ for which is this premise about the volumes of $D_P(\kappa)$ and $D_P(\kappa - \Delta)$ holds, rendering us capable of privately finding a (α, Δ)-kernel for this particular $D_P(\kappa)$.

Providing further details about the private approximation algorithms we introduce in this work requires that we first delve into some background details and introduce some parameters.

The Setting: Low-Dimension and Small Granularity

Differential privacy deals with the trade-offs between the privacy parameters, ε and δ, and an algorithm's utility guarantee. Unlike the majority of works in differential privacy, we don't express these trade-offs based on the size n of the data.[4] Instead, in our work we upper bound the Δ-term of a private (α, Δ)-approximation as a function of the privacy- and accuracy-parameters, as well as additional two parameters. These two parameters are (i) the dimension, d, which we assume to be constant and so $n^{\text{poly}(d)}$ is still considered efficient for our needs; and (ii) the granularity of the grid on which the data resides. In differential privacy, it is impossible to provide useful algorithms for certain basic tasks [6] when the universe of possible entries is infinite. Therefore, we assume that the given input P lies inside the hypercube $[0,1]^d$ and moreover – that its points reside on a grid $\mathcal{G}^d$ whose granularity is denoted as Υ. This means that each coordinate of a point $p \in P$ can be described using $\upsilon = \log_2(1/\Upsilon)$ many bits. We assume here that $1/\Upsilon$ is large (say, all numbers are `ints` in `C`, so $\Upsilon = 2^{-32}$), too large for the grid to be efficiently traversed. And so, for each (ε, δ)-differentially private algorithm we present, an algorithm that returns with a high probability of $1 - \beta$ a (α, Δ)-approximation of some geometric feature of $D_P(\kappa)$, we upper bound the Δ-term as a function of $(\alpha, \beta, \varepsilon, \delta, d, \upsilon)$. (Of course, we must also have that $\kappa > \Delta$ otherwise the algorithm can simply return $[0,1]^d$.) In addition, any algorithm with runtime of $(n \cdot \upsilon \cdot \varepsilon^{-1} \cdot \alpha^{-1} \cdot \log(1/\beta\delta))^{\text{poly}(d)}$ is considered efficient.

[4] Though n comes into play in our work, both in requiring that for large enough κ we have that $D_P(\kappa) \neq \emptyset$ and in bounding Δ, since if $\Delta > n$ then it is trivial to give a (α, Δ)-kernel. Moreover, ideally we would have that $\Delta \leqslant \sqrt{dn\log(n)}$ so that both $D_P(\kappa)$ and $D_P(\kappa - \Delta)$ (roughly) represent the same Tukey-depth region w.r.t to the distribution the dataset was drawn from, based on the above-mentioned bounds of [8].

Lastly, as pre-processing we apply the algorithm of Kaplan et al. [19] that asserts that for sufficiently large κ it holds that $D_P(\kappa)$ is non-empty and non-degenerate (doesn't have 0-volume).

Detailed Contribution and Organization

First, in Section 2 we survey some background in differential privacy and geometry. Our contributions are detailed in the remaining section and are as follows.

- In Section 3 we give our private (α, Δ)-kernel approximation, that much like its non-private equivalent [1], requires some "fatness" condition. In fact, we have two somewhat different conditions. Our first algorithm requires a known (constant) lower bound on $\mathrm{width}(D_P(\kappa))$, and our second algorithm requires a known (constant) lower bound on the ratio $\frac{\mathrm{width}(D_P(\kappa))}{\mathrm{diam}(D_P(\kappa-\Delta))}$. More importantly, the resulting sets from each algorithm do not satisfy an analogous property to the non-private kernel definition of [1], but rather more intricate properties regarding projections along any direction. Thus, Section 3 begins by discussing these two properties and proving that they are sufficient for finding a (α, Δ)-kernel. Due to brevity, we provide here only the high-level ideas of the first (and very simply) algorithm, whereas its full details, as well as the second algorithm and a heuristic that may allow us to tell if a region is "fat",[5] are all deferred to the full version of this work. The remainder of the work presents multiple tools designed in order to privately find a transformation that turns $D_P(\kappa)$ into a fat Tukey-region, each of which may be of independent interest.
- Beimel et al. [3] constructed a function for Tukey-Depth Completion (TDC): given a prefix of $0 \leqslant i < d$ coordinates, each $x \in \mathbb{R}$ is mapped to the max Tukey-depth of a point whose first $i+1$ coordinates are the given prefix concatenated with x. Beimel et al. showed that this TDC-function is quasi-concave (details in Section 4), so (i) by off-the-shelf private approximation algorithms for quasi-concave functions [4, 7] we can find x with high $\mathsf{TDC}(x)$-value; and (ii) repeating this process d times returns a point with high TD. So our first tool is detailed in Section 4 where we present a simple and efficient implementation of the TDC-function in low-dimensions. We also introduce a function that takes an additional parameter ℓ and maps x to $\min\{\mathsf{TDC}(x), \mathsf{TDC}(x+\ell)\}$, which is also quasi-concave and can also be computed efficiently. The two functions play an important role in the construction of *all* following algorithms – we often rotate the space so that some direction v aligns with first axis and then apply TDC to find a good extension of a particular coordinate along v into a point inside $D_P(\kappa)$. While we highlight the main ideas, the full details of this section appear in the full version of this work.
- In Section 5 we give a second batch of rudimentary tools – our efficient private algorithms for $(\alpha, \Delta^{\mathrm{diam}})$-diameter approximation and $(\alpha, \Delta^{\mathrm{width}})$-width approximation. These algorithms are quite standard and rely on the Sparse-Vector Technique; thus their formal descriptions are deferred to the full version of this work.
- In Section 6 we turn our attention to *asserting* that the fatness condition required for the kernel-approximation algorithm holds. We present a private (c, Δ)-approximation of the min bounding box problem – it returns a box B that (a) contains $D_P(\kappa)$ and (b) with volume upper bounded by $c \cdot \mathrm{vol}(D_P(\kappa-\Delta))$. We then show that if $\mathrm{vol}(D_P(\kappa)) \geqslant \frac{\mathrm{vol}(D_P(\kappa-\Delta))}{2}$ then by affinely mapping B to $[0,1]^d$ we turn $D_P(\kappa)$ into a fat Tukey region.

[5] This heuristic allows us to take κ as input to our algorithm: if the heuristic returns "OK" then the niceness conditions hold and we can return a (α, Δ)-approximation of $D_P(\kappa)$; o/w the algorithms of Sections 6 and 7 allow us to replace the value of the given κ with a different value, one for which we can return a (α, Δ)-approximation of $D_P(\kappa)$.

- In Section 7 we give a private algorithm for finding a “good” depth-parameter κ, one for which it does hold that $\mathrm{vol}(D_P(\kappa)) \geqslant \mathrm{vol}(D_P(\kappa-\Delta))/2$. We formulate a certain query q where any κ for which $q_P(\kappa)$ is large must also be a good κ, and then give a private algorithm for finding a κ with a large $q_P(\kappa)$-value. The ε-differentially private algorithm we give is actually rather novel – it is based on a combination of the Exponential-Mechanism with additive Laplace noise. Its privacy is a result of arguing that for any neighboring P and P' where $P' = P \cup \{x\}$ we can *match* κ with $\kappa+1$ so that $|q_P(\kappa) - q_{P'}(\kappa+1)| \leqslant 1$, and then using a few more observations that establish pure ε-differential privacy (rather than (ε,δ)-DP). Again, due to space considerations, the full-details and proofs from Sections 6 and 7 appear in the full version of this work.

Our work thus culminates in the following theorem.

▶ **Theorem 2.** *There exists an efficient (ε,δ)-differentially private algorithm, that for any sufficiently large dataset P, where $|P| \geqslant \tilde{\Omega}(d^4\upsilon \cdot \Delta)$, with probability $\geqslant 1-\beta$ finds a “good” depth parameter κ and a set $\mathcal{S}$ such that $\mathcal{S}$ is a (α,Δ)-kernel of $D_P(\kappa)$ where $\Delta = O(\frac{f(d)}{\varepsilon} \cdot (\frac{1}{\alpha})^{\frac{d}{2}} \sqrt{\log(\frac{1}{\delta})} \log(\frac{1}{\alpha\beta}))$ for some function $f(d) = 2^{d^2 \,\mathrm{poly}\log(d)}$.*

In fact, it is also required that $\Delta \geqslant \Delta^{\mathrm{BB}}(d,\upsilon,\varepsilon,\delta,\beta)$ where Δ^{BB} is guarantee of the private min-bounding-box algorithm, as detailed in Theorem 16; yet this lower-bound holds under a very large regime of parameters.

Additional Works

In addition to the two works [3, 19] that privately find a point inside a convex hull, it is also worth mentioning the works regarding privately approximating the diameter [23, 22] (they return a $O(1)$-approximation of the diameter that may miss a few points) as well as the recent work of [15] which can also be used to approximated the diameter; and the work of [18] that privately approximates a k-edges polygon yet requires a dataset of points where many lie inside the polygon and many lie *outside* the polygon. No additional works that we know of lie in the intersection of differential privacy and computational geometry. Computational geometry, of course, is a rich fied of computer science replete with many algorithms for numerous tasks in geometry. Our work only give private analogs to (a few of) the algorithms of [9, 1], but there are far many more algorithms to be privatized and the reader is referred to [16] for a survey of the field. Many works deal with computing the Tukey-depth and the Tukey region [24, 20], and others give statistical convergence rates for the Tukey-depth when the data is composed of i.i.d. draws from a distribution [28, 8, 5].

2 Preliminaries

Geometry

In this work we use $\langle\cdot,\cdot\rangle$ to denote the inner-product between two vectors in $\mathbb{R}^d$. We use e_j to denote the nature basis element with 1 on jth coordinate and zeros elsewhere. A closed half-space is defined by a vector u and a scalar λ and it is the set $\{x \in \mathbb{R}^d : \langle x,u\rangle \leqslant \lambda\}$. A polytope, which is a convex body, is the intersection of finitely many closed half-spaces. For a polytope P and a point x we define $P - x$ as the shift of P by x (namely $z \in P - x$ iff $\exists y \in P$ s.t. $z = y - x$), and we define by cP the blow-up of P by a scalar c. An inner product $\langle x,u\rangle = \|x\|\|u\|\cos(\angle(x,u))$ is also known a projection of x onto the subspace spanned by u. A projection onto a subspace Π^V maps any $x \in \mathbb{R}^d$ to its closest point in the subspace V. The following fact is well-known.

▶ **Fact 3.** *Let S be a convex body. Let u be any vector and let $\Pi^{\perp u}$ be the projection onto the subspace orthogonal to u. Denote ℓ as the max-length of the intersection of S with any affine line in direction u, and denote A as the volume of the projection of S onto the subspace orthogonal to u, $A = \mathrm{vol}(\Pi^{\perp u}(S))$. Then $\frac{A \cdot \ell}{d} \leqslant \mathrm{vol}(S) \leqslant A \cdot \ell$.*

The fact follows from reshaping S so that it is contained in the "cylinder" whose base is A and height is ℓ, and contains a "pyramid" with base of A and height of ℓ.

The unit-sphere $\mathbb{S}^{d-1}$ is the set of vectors in $\mathbb{R}^d$ of length 1. The *diameter* of the convex body P is defined as $\mathrm{diam}(P) = \max_{p,q \in P} \|p - q\|$, and it is simple to see that $\mathrm{diam}(P) = \max_{u \in \mathbb{S}^{d-1}} \max_{p,q \in P} \langle p - q, u \rangle$. The *width* of a convex body P is analogously defined as $\mathrm{width}(P) = \min_{u \in \mathbb{S}^{d-1}} \max_{p,q \in P} \langle p - q, u \rangle$. A *$\zeta$-angle cover* of the unit sphere is a set of vectors V_ζ such that for any $v \in \mathbb{S}^{d-1}$ there exist u such that $\angle(u, v) \leqslant \zeta$. It is known that each vector in the sphere can be characterized by $d - 1$ angles $\varphi_1, \varphi_2, ..., \varphi_{d-1}$ where $\varphi_i \in [0, 2\pi]$ and for any other j, $\varphi_j \in [0, \pi]$. Therefore by discretizing the interval $[0, \pi]$ we can create a ζ-angle cover of size $2\lceil \pi/\zeta \rceil^{d-1}$.

▶ **Proposition 4** (Proof omitted.). *Let $\zeta < \frac{1}{2}$. Let V_ζ be a ζ-angle cover of $\mathbb{S}^{d-1}$. Then $\forall u \in \mathbb{S}^{d-1}$ the closest $v \in V_\zeta$ satisfies $\|u - v\| \leqslant \sqrt{2}\zeta$.*

Tukey Depth

Given a finite set of points $P \subset \mathbb{R}^d$, the Tukey depth [26] of a point $x \in \mathbb{R}^d$ w.r.t P is defined as $\mathrm{TD}(x, P) = \min_{u \in \mathbb{S}^{d-1}} |\{p \in P : \langle p, u \rangle \leqslant \langle x, u \rangle\}|$. Given P and a depth parameter $\kappa \geqslant 0$ we denote the *κ-Tukey region* as $D_P(\kappa) = \{x \in \mathbb{R}^d : \mathrm{TD}(x, P) \geqslant \kappa\}$. It is known that for any set of points P it holds that $\kappa^* = \max_x \mathrm{TD}(x, P) \in [\frac{|P|}{d+1}, \frac{|P|}{2}]$ (see [14]). It is also known that for all κ, the (non-empty) set $D_P(\kappa)$ is a convex polytope which is the intersection of all closed halfspaces that contain at least $n - \kappa + 1$ points out of P [24], this yields a simple algorithm to compute the κ-Tukey region in time $O(n^{(d-1)\lfloor \frac{d}{2} \rfloor})$. There is a faster algorithm to compute the κ-Tukey region in time $O(n^d \log n)$ [20], and so to compute *all* of the non-empty Tukey-regions in time $O(n^{d+1} \log n)$.

Differential Privacy

The formal definition of differential privacy [10, 12] is as follows.

▶ **Definition 5.** *Two datasets P and P' are called* neighbors *if they differ on a single datum, and in this work we assume that this means that $|P \triangle P'| = 1$. A randomized algorithm $\mathcal{A}$ is said to be (ε, δ)-differentially private (DP) if for any two neighboring datasets P and P' and for any set of possible outputs S it holds that $\Pr[\mathcal{A}(P) \in S] \leqslant e^\varepsilon \Pr[\mathcal{A}(P') \in S] + \delta$. When $\delta = 0$ we say $\mathcal{A}$ is ε-DP or ε-pure DP.*

Differential privacy composes: if $\mathcal{A}$ is (ε, δ)-DP and $\mathcal{B}$ is (ε', δ')-DP, then applying $\mathcal{A}$ and then applying $\mathcal{B}$ sequentially on P is a $(\varepsilon + \varepsilon', \delta + \delta')$-DP algorithm. It is also worth noting the advanced-composition theorem [13], where the sequential application of k (ε, δ)-DP algorithms yields in total an algorithm which is $(O(\varepsilon\sqrt{k \ln(1/k\delta)}), 2k\delta)$-DP (provided $\varepsilon < 1$). Since we deal with a constant dimension d, then whenever we compose poly(d)-many mechanisms, we rely on the basic composition; and whenever we compose exp(d)-many mechanisms, we rely on the advanced composition.

The Laplace additive noise is a ε-DP algorithm that works as follows. Given a function f that maps inputs to real numbers, we first find its global sensitivity $GS(f) = \max_{P,P' \text{neighbors}} |f(P) - f(P')|$, then output $f(P) + \mathsf{Lap}(\mathsf{GS(f)}/\varepsilon)$. It is also worth noting the *Sparse Vector Technique* which is an ε-DP algorithm that allows us to assess t queries $q_1, q_2, .., q_t$, each with $GS(q_i) = 1$, and halt on the very first query that exceeds a certain (noisy) threshold. Our algorithms repeatedly rely on the SVT.

Private Approximations of Quasi-Concave Functions

In our work we use as "building blocks" several known results in differential privacy regarding approximating quasi-concave functions. A function $q : \mathbb{R} \to \mathbb{R}$ is a *quasi-concave function* if for any $x \leqslant y \leqslant z$ it holds that $q(y) \geqslant \min\{q(x), q(z)\}$. Quasi-concave functions that obtain a maximum (namely, there exists some $x \in \mathbb{R}$ such that $\forall y, q(x) \geqslant q(y)$) have the property that the maximum is obtained on a single closed interval $I = [x, y]$ (we allow the case $x = y$, or $I = \{x\}$). Moreover, it follows that on the interval $(-\infty, x)$ the function q is monotone non-decreasing and on the interval (y, ∞) the function q is monotone non-increasing. The following is known about DP-approximations of quasi-concave functions.

▶ **Theorem 6.** *Let q be any function $q : \mathbb{R} \to \mathbb{R}$ satisfying (i) q is quasi-concave, (ii) q has global-sensitivity 1 and (iii) for every closed interval I one can efficiently compute $\max_{x \in I} q(x)$. Let $\mathcal{G} \subset \mathbb{R}$ be a grid of granularity $\Upsilon = 2^{-\upsilon}$, and denote $q^* = \max_{x \in \mathcal{G}} q(x)$. Then, for any $0 < \beta < 1/2$ there exist differentially private algorithms that w.p. $\geqslant 1 - \beta$ return some $x \in \mathcal{G}$ such that $q(x) \geqslant q^* - \alpha^{\mathrm{qc}}(\varepsilon, \delta, \beta)$ where*

$$\alpha^{\mathrm{qc}}(\varepsilon, \delta, \beta) = \begin{cases} O(\frac{\upsilon + \log(1/\beta)}{\varepsilon}), & \textit{using } \varepsilon\textit{-DP binary-search} \\ \tilde{O}(\frac{\log(\upsilon/\beta\varepsilon\delta)}{\varepsilon}), & \textit{using the "Between Thresholds" Algorithm [7]} \\ O\left(\frac{8^{\log^*(\upsilon)} \log^*(\upsilon)}{\varepsilon} \cdot \log(\frac{\log^*(\upsilon)}{\beta\delta})\right) & \textit{using the "RecConvace" algorithm [4]} \end{cases}$$

The first bound is given by standard ε-DP binary search algorithm (folklore). The second bound is given by the rather intuitive "Between Threshold" algorithm of Bun et al. [7] where instead of the standard counting function $f(z) = |\{x : x \leqslant z\}|$ we use the function $f(z) = \max\limits_{x \in (-\infty, z]} q(x) - \max\limits_{x \in [z, \infty)} q(x)$ and set thresholds close to 0 (indicating a maximization point of q). The third is the `RecConcave` algorithm by [4] and is rather involved. (It is unknown[6] whether the recent work [17] is applicable to general quasi-concave functions.)

3 Notions of Kernels and Fatness Suitable for Private Approximation

Prior to presenting our algorithm(s) for finding a kernel of a Tukey-region, we first discuss *our goal* – what it is we wish to output, and *our premise* – the kinds of datasets on which we are guaranteed to release such outputs. Recall, our goal is to give a differentially private algorithm that outputs a collection of points $\mathcal{S}$ which is a (α, Δ)-kernel of $D_P(\kappa)$. Namely, this $\mathcal{S}$ satisfies that (after shifting) $(1 - \alpha)D_P(\kappa) \subset \mathsf{CH}(\mathcal{S}) \subset (1 + \alpha)D_P(\kappa - \Delta)$. Clearly, for any two convex bodies s.t. $\mathcal{A} \subset \mathcal{B}$ and for any projection Π we have that $\Pi(\mathcal{A}) \subset \Pi(\mathcal{B})$. (In fact, this holds for any affine transformation.) So if $\mathcal{S}$ is a (α, Δ)-kernel of $D_P(\kappa)$ then it also holds that

$$\forall u \in \mathbb{S}^{d-1} \quad (1-\alpha) \max_{p,q \in D_P(\kappa)} \langle p-q, u\rangle \leqslant \max_{p,q \in \mathsf{CH}(\mathcal{S})} \langle p-q, u\rangle \leqslant (1+\alpha) \max_{p,q \in D_P(\kappa-\Delta)} \langle p-q, u\rangle \quad (1)$$

In the standard / non-private setting, the definition of kernel [1] is equivalent to $(\alpha, 0)$-kernel (i.e., setting $\Delta = 0$). Moreover, as defined in [1], a $(\alpha, 0)$-kernel must satisfy both the property

[6] Uri Stemmer, private correspondence.

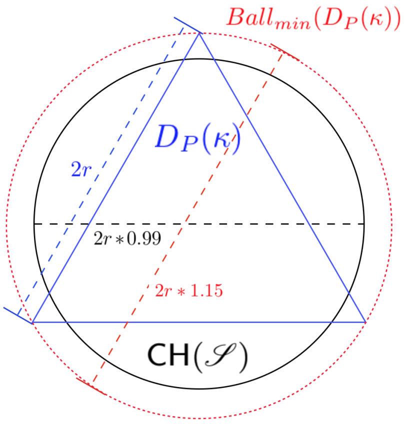

Figure 2 An example showing that the property of Equation (1) doesn't imply that $(1-\alpha)D_P(\kappa) \subset \mathsf{CH}(\mathcal{S})$. Suppose $D_P(\kappa)$ is an equilateral triangle of edge-length $2r$ and $\mathcal{S}$ happens to be a ball of diameter $2 \cdot 0.99 \cdot r$ (and $D_P(\kappa - \Delta)$ is a much larger region). Note that $\mathcal{S}$ does satisfy Equation (1) for $\alpha = 0.01$ yet $0.99 D_P(\kappa) \not\subset \mathsf{CH}(\mathcal{S})$.

in (1) and the property that $\mathcal{S} \subset D_P(\kappa)$; and it is straight-forward to show that together, the two properties yield the containment $(1 - O(\alpha))D_P(\kappa) \subset \mathsf{CH}(\mathcal{S}) \subset D_P(\kappa)$. It turns out that in the private setting, with $\Delta > 0$, since it doesn't necessarily hold that $\mathcal{S} \subset D_P(\kappa)$, then property (1) alone does *not* guarantee that we output an (α, Δ)-kernel. Figure 2 illustrates such a setting. So instead, we give algorithms whose resp. outputs satisfy *variations* of the projection property in (1). Below we state two claims showing that the different variations do yield a kernel. The (far from trivial) proofs of the two claims are deferred to the full version of this work.

▷ Claim 7. Let $\mathcal{S}$ be a set that satisfies the following property in regards to $D_P(\kappa)$ and $D_P(\kappa - \Delta)$:

$$\forall u \in \mathbb{S}^{d-1}, \quad \max_{p \in D_P(\kappa)} \langle p, u \rangle - \alpha \cdot \mathrm{width}_\kappa \leqslant \max_{p \in \mathsf{CH}(\mathcal{S})} \langle p, u \rangle \leqslant \max_{p \in D_P(\kappa - \Delta)} \langle p, u \rangle + \alpha \cdot \mathrm{width}_{\kappa - \Delta} \tag{2}$$

then, denoting $\alpha' = 2\alpha\sqrt{d + \frac{1}{2}}$, there exists two vectors p_1 and p_2 such that we can shift $D_P(\kappa)$ and $D_P(\kappa - \Delta)$ and have that $(1 - \alpha')(D_P(\kappa) - p_1) \subset \mathsf{CH}(\mathcal{S}) - p_1$ and $\mathsf{CH}(\mathcal{S}) - p_2 \subset (1 + \alpha')(D_P(\kappa - \Delta) - p_2)$.

▷ Claim 8. Fix $\alpha < 1/6$ and let $\mathcal{S} \subset D_P(\kappa - \Delta)$ be a set such that there exists a point $c \in D_P(\kappa) \cap \mathcal{S}$ for which

$$\forall u \in \mathbb{S}^{d-1}, \quad (1 - \alpha) \max_{p \in D_P(\kappa)} \langle p - c, u \rangle \leqslant \max_{p \in \mathsf{CH}(\mathcal{S})} \langle p - c, u \rangle + \alpha \cdot \mathrm{width}_\kappa \tag{3}$$

then, denoting $\alpha' = \frac{\alpha}{1-\alpha}(1 + 4\sqrt{d + \frac{1}{2}})$, there exists a vector b such that we can shift $D_P(\kappa)$ and $\mathsf{CH}(\mathcal{S})$ by b and have that $D_P(\kappa) - b \subset (1 + \alpha')(\mathsf{CH}(\mathcal{S}) - b)$.

Definition of Fatness

The algorithms we provide are such that their respective outputs satisfy the premises of Claims 7 and 8. Unfortunately, these algorithms do not return useful sets $\mathcal{S}$ for *any* $D_P(\kappa)$. Much like in the non-private setting [1], in order for each algorithm to output a kernel of $D_P(\kappa)$ we must require that $D_P(\kappa)$ satisfies a certain "fatness" property. In the standard, non-private setting, a convex polytope $D_P(\kappa)$ is called c_d-fat [1] if there exists a constant $c_d \geqslant 1$ (depending solely on the dimension d) where $\mathrm{diam}(D_P(\kappa)) \leqslant c_d \mathrm{width}(D_P(\kappa))$. Alas, our differentially private algorithms require something stronger. Formally, we define the follow various notions of fatness, using the shorthand notation $\mathrm{width}_\kappa \overset{\mathrm{def}}{=} \mathrm{width}(D_P(\kappa))$.

▶ **Definition 9.** *We say $D_P(\kappa)$ is* (c_d, Δ)-fat *if it holds that* $\mathrm{width}_\kappa \geqslant \frac{\mathrm{diam}_{\kappa-\Delta}}{c_d}$. *We say $D_P(\kappa)$ is* c_d-absolutely fat *if* $\mathrm{width}_\kappa \geqslant \frac{1}{c_d}$.

It is clear that the fatness properties can be violated by the addition or removal of a single datapoint to/from P. Therefore, no differentially private algorithm can always assert w.h.p. whether $D_P(\kappa)$ is fat or not, nor estimate its fatness parameter c_d. We comment that in the non-private version [16] the obtained constant is $d^{5/2}2^d(d!)$, whereas our fatness constant is fairly similar: $4d^{5/2}5^d(d!)$.

A Simple Private Kernel Approximation Under "Absolute Fatness"

Under the premise that $D_P(\kappa)$ is c_d-absolutely fat, that is, that $\mathrm{width}_\kappa \geqslant 1/c_d$ (when c_d is known to the algorithm), we are able to give a pretty simple (α, Δ)-kernel approximation algorithm. The algorithm partitions the $[0,1]^d$-cube into subcubes of side length $\frac{2\alpha}{c_d\sqrt{d}}$, and for each subcube C checks whether $\max_{x \in C} \mathrm{TD}(x, P)$ perturbed by Laplace noise is greater than $\kappa' = \kappa - \frac{\Delta}{2}$, and if so – adds C's center to $\mathcal{S}$. Here Δ is set using the union-bound on all Laplace random-variables so that w.h.p. any C where $C \cap D_P(\kappa) \neq \varnothing$ adds its center to $\mathcal{S}$. The full details of the algorithm are deferred to the full version.

▶ **Theorem 10.** *There exists an efficient, (ε, δ)-DP algorithm that returns w.p. $\geqslant 1 - \beta$ a set $\mathcal{S}$ that satisfies that $\forall u \in \mathbb{S}^{d-1}$, $\max_{p \in D_P(\kappa)} \langle p, u \rangle - \alpha \cdot \mathrm{width}_\kappa \leqslant \max_{p \in \mathcal{S}} \langle p, u \rangle \leqslant \max_{p \in D_P(\kappa - \Delta^{\mathrm{kernel}})} \langle p, u \rangle - \alpha \cdot \mathrm{width}_{\kappa - \Delta^{\mathrm{kernel}}}$, where $\Delta^{\mathrm{kernel}} = O(d(\frac{c_d\sqrt{d}}{\alpha})^{d/2}\sqrt{\log(1/\delta)}\log(\frac{c_d d}{\alpha\beta})/\varepsilon)$. So by Claim 7 $\mathcal{S}$ is a kernel for $D_P(\kappa)$.*

Applications

Agarwal et al. [1] define a function μ of a dataset as a *faithful measure* if (i) μ is non-negative, (ii) for every $P \subset \mathbb{R}^d$ we have $\mu(P) = \mu(\mathsf{CH}(P))$, (iii) μ is monotone w.r.t containment of convex bodies, and most importantly, that (iv) for some $c \in (0,1)$ a $(1 - c\alpha)$-kernel of P yields a $(1-\alpha)$-approximation of $\mu(P) = \mu(\mathsf{CH}(P))$. Obviously, any faithful measure μ can be approximated by a (α, Δ)-kernel $\mathcal{S}$ where $(1 - \frac{\alpha}{c})\mu(D_P(\kappa)) \leqslant \mu(\mathsf{CH}(\mathcal{S})) \leqslant (1 + \frac{\alpha}{c})\mu(D_P(\kappa - \Delta))$. Thus, a (α, Δ)-kernel gives suitable approximations for problems such as min/max enclosing ball, min bounding box, John's Ellipsoid, surface-area etc. (all are faithful measures).

3.1 Remainder of this Extended Abstract

In the full version of this work, we also present another (and more complex) algorithm, that works under the premise that $D_P(\kappa)$ is (c_d, Δ)-fat. Similarly, our proposed heuristic for finding whether the data is (c_d, Δ)-fat is also deferred to the full version of this work. This alternative algorithm and heuristics require additional "building blocks" such as privately finding a point inside $D_P(\kappa)$ and privately estimating the diameter, width and various projections. These building blocks are described in Setions 4 and 5 resp. Note that these additional algorithm and heuristics enable us to return a private kernel of $D_P(\kappa)$ for a user-specified value of κ provided the heuristics return "Yes." Yet, should the heuristics return "No," what we do is to find a different value of κ for which a kernel of $D_P(\kappa)$ we can approximated.

In Section 6 we show how to (privately) approximate the bounding box of $D_P(\kappa)$, outputting a box whose volume is comparable to $\mathrm{vol}(D_P(\kappa-\Delta))$. (This algorithm may be of independent interest.) Furthermore, we argue that if it is indeed the case that the volumes of $D_P(\kappa)$ and $D_P(\kappa-\Delta)$ are similar up to a multiplicative factor of 2, then such a bounding box approximation yields a transformation that turns $D_P(\kappa)$ into a $(4d^{\frac{5}{2}}5^d(d!),\Delta)$-fat Tukey region. Thus, in Section 7 we detail an algorithm that returns a value of κ for which $\frac{\mathrm{vol}(D_P(\kappa-\Delta))}{\mathrm{vol}(D_P(\kappa))} \leqslant 2$. So the algorithm from Section 7 returns a value of κ, for which the bounding box approximation algorithm of Section 6 does give a transformation that turns $D_P(\kappa)$ fat; implying that the above-mentioned kernel-approximation algorithm can be successfully applied. The reader should be advised that Sections 4-7 are very succinctly described, where we tried to highlight the main ideas of each algorithm and the (often quite subtle) novelties in each algorithm's design.

4 Tools, Part 1: The Tukey-Depth Completion Function

In this section we discuss the implementation of the following *Tukey Depth Completion* function. This function takes as a parameter an i-long tuple of coordinates, where $0 \leqslant i < d$, and scores each $x \in \mathbb{R}$ with a value κ if the $i+1$ prefix $\bar{y} \circ x$ can be completed to a point with Tukey-depth of κ.

▶ **Definition 11** ([3]). *Fix $d \in \mathbb{N}$ and let P be a collection of points in $\mathbb{R}^d$. For any i-tuple of coordinates $\bar{y} = (y_1, y_2, ..., y_i)$ where $0 \leqslant i \leqslant d-1$ we define the function $\mathsf{TDC}_{\bar{y}} : \mathbb{R} \to \mathbb{R}$ by*

$$\mathsf{TDC}^P_{\bar{y}}(x) = \max_{(z_1,z_2,\ldots,z_{d-1-i})\in\mathbb{R}^{d-i-1}} \mathsf{TD}\big((y_1, .., y_i, x, z_1, .., .z_{d-1-i}), P\big) \qquad (4)$$

For any closed interval $I = [a,b] \subset \mathbb{R}$ we overload the definition of TDC to denote $\mathsf{TDC}^P_{\bar{y}}(I) = \max_{x\in[a,b]} \mathsf{TDC}^P_{\bar{y}}(x)$. Lastly, for any such $\bar{y}$ and any $\ell \in \mathbb{R}$ we denote $\ell\text{-}\mathsf{TDC}^P_{\bar{y}}(x) = \min\{\mathsf{TDC}^P_{\bar{y}}(x), \mathsf{TDC}^P_{\bar{y}}(x+\ell)\}$, and similarly, $\ell\text{-}\mathsf{TDC}^P_{\bar{y}}(I) = \max\limits_{x\in I} \ell\text{-}\mathsf{TDC}^P_{\bar{y}}(x)$. We omit the superscript P whenever the dataset is clear.

In the full version of this work we prove that both the TDC-function and the ℓ-TDC-function are quasi-concave. So it follows that on the real line the values of the TDC-function ascend from 0 to the max-value ($\leqslant n/2$), then descend back to 0. In particular, for any κ (ranging from 0 to the max-value of the $\mathsf{TDC}_{\bar{y}}$-function), there exists an interval $[a_\kappa, b_\kappa]$ such that $x \in [a_\kappa, b_\kappa]$ if and only if $\mathsf{TDC}_{\bar{y}}(x) \geqslant \kappa$. And so, we give a simple, LP-based, algorithm that finds these set of nested intervals $\{[a_\kappa, b_\kappa]\}_{\kappa>0}$, and then – through binary search – finds the maximum κ whose interval intersect the given point x or interval I. (Note that this binary search is over $\leqslant n$ elements so it runs in time $O(\log(n))$.)

Extension

One of the key uses to the TDC-function we rely on is when we rotate directions so that the first axis aligns with a given direction v. In such a case, this is equivalent to rotating the set P, so we use the notation $\mathsf{TDC}^{R_v(P)}_{\bar{y}}$ and on occasion just $\mathsf{TDC}^{R_v}_{\bar{y}}$.

A Technical Point: Grid Refinement

We established that for any $0 \leqslant i \leqslant d-1$ and any prefix $\bar{y}$ there exists an efficient algorithm that computes $\mathsf{TDC}_{\bar{y}}(x)$ and ℓ-$\mathsf{TDC}_{\bar{y}}(x)$. But as by Beimel et al. [3] noted, it is not a-priori clear that the coordinates of the completion lie on the same grid $\mathcal{G}^d$ we start with. Throughout most of this paper we ignore this subtlety,[7] but we do formally show in the full version how to refine $\mathcal{G}$ into a grid $\mathcal{G}'$ with granulatiry of at least $(\Upsilon/d)^{O(d^4)}$ where the output has all of its coordinates in $\mathcal{G}'$. The crux of this result is that all coordinates of all vertices of $D_P(\kappa)$ have granularity $\geqslant (\Upsilon/d)^{O(d^2)}$ (see [19]), and that finding the above-mentioned a_κ, b_κ requires inverting a $(i+1)\times(i+1)$-matrix whose entries are coordinates of vertices of $D_P(\kappa)$. So we end up requiring a refinement of only $(\Upsilon/d)^{O(i\cdot d^2)}$ per $i \in \{1, 2, .., d\}$, so overall our refinement has granularity $(\Upsilon/d)^{O(d^4)}$.

Summary

Now that we refined the grid from $\mathcal{G}$ to $\mathcal{G}'$ with granularity $\Upsilon^{4d^4} = 2^{-\upsilon(4d^4)}$, we can apply any DP-algorithm that w.p.$\geqslant 1-\beta$ returns a point on $\mathcal{G}'$ with roughly the same value of the maximal value. This gives a DP-algorithm that returns w.p.$\geqslant 1-\beta$ a point $x \in \mathcal{G}'$ with either $\mathsf{TDC}_{\bar{y}}$-value or ℓ-$\mathsf{TDC}_{\bar{y}}$-value which is $\alpha^{\mathrm{qc}}(\cdot,\cdot,\cdot)$-close to the max-possible value on the grid. Altogether, we have the following corollary.

▶ **Corollary 12.** *Fix $\varepsilon > 0$, $\delta \geqslant 0$, $\beta \in (0, 1/2)$. There exists an efficient (ε,δ)-DP-algorithm, denoted* `DPPointInTukeyRegion`*, that takes as input a dataset P and a parameter κ where $D_P(\kappa) \neq \emptyset$ and w.p. $\geqslant 1-\beta$ returns a point $\bar{x} \in (\mathcal{G}')^d$ whose Tukey-depth is at least $\kappa - d\alpha^{\mathrm{qc}}(\frac{\varepsilon}{d}, \frac{\delta}{d}, \frac{\beta}{d}) \geqslant \frac{n}{d+1} - d\alpha^{\mathrm{qc}}(\frac{\varepsilon}{d}, \frac{\delta}{d}, \frac{\beta}{d})$. In particular, for any $\kappa \geqslant 0$ we return a point of Tukey-depth $\geqslant \kappa$ provided $n = \Omega(d\kappa + d^2\alpha^{\mathrm{qc}}(\frac{\varepsilon}{d}, \frac{\delta}{d}, \frac{\beta}{d}))$*

$$= \begin{cases} \Omega(d\kappa + d^3\frac{d^4\upsilon + \log(d/\beta)}{\varepsilon}), & \textit{Using the } \varepsilon\textit{-DP binary-search} \\ \tilde{\Omega}(d\kappa + d^3\frac{\log(d\upsilon/\beta\varepsilon\delta)}{\varepsilon}), & \textit{Using the “Between Thresholds” algorithm} \\ \Omega(d\kappa + d^3\frac{8^{\log^*(d\upsilon)}\log^*(d\upsilon)\cdot\log(d\log^*(\upsilon)/\delta\beta)}{\varepsilon}), & \textit{Using the “RecConcave” algorithm} \end{cases} \quad (5)$$

We comment that quantitatively, the results are just as those obtained by [3] (with a minor exception of their granularity level set to Υ^{2^d}), and as such are better than the utility guarantee of [19] when $\delta > 0$. The key improvement of our work is the runtime, decreased to $\mathrm{poly}(\upsilon)$.

5 Tools, Part 2: Approximating the Diameter and Width of a Tukey-Region

The Diameter

In this section our goal is to approximate the diameter of $D_P(\kappa)$, denoted $\mathrm{diam}_\kappa = \max_{a,b\in D_P(\kappa)}$
$\|b-a\|$. Our algorithm returns a (α,Δ)-approximation of diam_κ, namely a value ℓ satisfying $(1-\alpha)\mathrm{diam}_\kappa \leqslant \ell \leqslant \mathrm{diam}_{\kappa-\Delta}$. In order to find such an approximation, we leverage on the idea of discretizing all possible directions, which is feasible in constant-dimension Euclidean space. Denoting V_ζ as a ζ-angle cover of the unit sphere it is straight-forward to show that $(1-\zeta^2)\mathrm{diam}(P) \leqslant \max_{v\in V_\zeta}\max_{a,b\in P}\langle b-a, v\rangle \leqslant \mathrm{diam}(P)$. Based on V_ζ, our approximation merely uses the Sparse-Vector Technique (SVT). For each ℓ we pose the query $q_P(\ell) = \max_{v\in V_\zeta}\max_{x\in\mathbb{R}} \ell\text{-}\mathsf{TDC}^{R_v(P)}(x)$ where R_v is a rotation that sets v as the first vector basis. The details of the algorithm and its proof of correctness are deferred to the full version of this work.

[7] So instead of formally stating “we find a point p inside the convex body” we should say “we find a point p within distance $\sqrt{d}\Upsilon$ from a point inside the convex body.” After all, our work already deals with approximations, so under the (rather benign) premise that the diameter of the convex body is sufficiently larger than Υ, this little additive factor changes very little in the overall scheme.

▶ **Theorem 13.** *There exists an algorithm* DPTukeyDiam *which is an efficient ε-DP algorithm that w.p.* $\geqslant 1-\beta$ *returns a value ℓ which is (α, Δ)-approximation of* diam_κ *for* $\Delta^{\mathrm{diam}}(\varepsilon,\beta) = O(\frac{\log((v+\log(d))/\alpha\beta)}{\varepsilon})$.

The Width

We now turn our attention to the width estimation of the Tukey region $D_P(\kappa)$. Informally, the width of a set is the smallest "sandwich" of parallel hyperplanes that can hold the entire set. Formally, $\mathrm{width}_\kappa = \min_{v\in\mathbb{S}^{d-1}} \max_{a,b\in D_P(\kappa)} |\langle b, v\rangle - \langle a, v\rangle|$. Our private approximation gives a (α, Δ)-approximation of the width – a value w where $(1-\alpha)\mathrm{width}_\kappa \leqslant w \leqslant (1+\alpha)\mathrm{width}_{\kappa-\Delta}$. It is tempting to think that, much like the approach for diameter approximation, a similar discertization/cover of all directions ought to produce a $(1+\alpha)$-approximation of the width. Alas, this approach fails when the width is very small, smaller than the discretization level. But when the discretization is up-to-scale, then we can easily argue the correctness of the discretization approach. The following is proven in the full version of this work.

▶ **Proposition 14.** *Fix any $\alpha > 0$. Given a set $P \subset \mathbb{R}^d$ with diameter D and width w, if we set $\zeta \leqslant \min\{\frac{\alpha w}{\sqrt{2}D}, \frac{1}{2}\}$ and take V_ζ as a ζ-angle cover of the unit-sphere, then we have that $w \leqslant \min_{v\in V_\zeta} \max_{a,b\in P}\langle b-a, v\rangle \leqslant (1+\alpha)w$.*

Following Proposition 14 we present our private approximation of width_κ. This approximation also leverages on the query ℓ-TDC for a decreasing sequence of lengths $\ell_1 > \ell_2 > \ldots$, however, as opposed to diameter approximation, with each smaller ℓ we also use a different discretization of the unit sphere. For each ℓ_i we set $\zeta_i = \frac{\alpha\ell_i}{4D}$ and use the query $q_P(\ell_i) = \min_{v\in V_{\zeta_i}} \max_{x\in\mathbb{R}} \ell_i\text{-TDC}^{R_v(P)}(x)$. We prove that (i) if $\mathrm{width}(D_P(\kappa)) \geqslant \ell_i$ then $q_P(\ell_i) \geqslant \kappa$; and (ii) if $\mathrm{width}(D_P(\kappa)) \leqslant (1-\alpha)\ell_i$ and $\zeta_i \leqslant \frac{\alpha\ell}{4D}$ then $q_P(\ell_i) < \kappa$. Thus our algorithm is merely an application of the SVT with these queries. Algorithm's details and proofs appear in the full version of this work.

▶ **Theorem 15.** *There exists an algorithm* DPApproxWidth *which is a ε-DP algorithm that w.p.* $\geqslant 1-\beta$ *returns a value ℓ which is (α, Δ)-approximation of* width_κ *for* $\Delta^{\mathrm{width}}(\varepsilon,\beta) = O(\frac{\log((v+\log(d))/\alpha\beta)}{\varepsilon})$.

Note that our width-approximation algorithm requires we refine the angle-cover V_ζ with each iteration. Without any a-priori lower bound on the width, the refinement can be as small as Υ, which renders our algorithm inefficient. That is why in our work we rely on having a particular lower bound, of $1/(4d^{\frac{5}{2}} \cdot 5^d \cdot (d!))$ (which is our fatness bound). In addition, our full version also describes here two additional algorithms (also SVT-based) for subroutines we will require later: estimating the max-projection from a point and finding a direction on which some specific scalar has large TDC-value.

6 Private Approximation of the Bounding Box of $D_P(\kappa)$

In this section we give a differentially private algorithm that returns a transformation that turns $D_P(\kappa)$ into a fat Tukey-region. The transformation is based on (privately) finding an approximated bounding-box for $D_P(\kappa)$, and once such a box is found, then the transformation T is merely an affine transformation, composed of rotation and scaling, that maps the bounding box B to the hypercube $[0,1]^d$. We thus focus in this section on a private algorithm that gives a (c_d, Δ)-approximation of the bounding box of $D_P(\kappa)$, so our algorithm's guarantee relates to both the volume of $D_P(\kappa)$ and the volume of $D_P(\kappa-\Delta)$. Formally, we return (w.h.p) a box B which is a bounding box that holds $D_P(\kappa)$ and where $\mathrm{vol}(\mathsf{B}) \leqslant 5^d \cdot (d!) \cdot \mathrm{vol}(D_P(\kappa-\Delta))$.

The algorithm mimics the non-private bounding-box algorithm in [16] (Ch.18). It is a recursive algorithm, where in each level of the recursion we find a segment $\bar{st}$ and an interval I on the line extending this segment where the following three properties must hold: (i) both $s,t \in D_P(\kappa-\Delta)$, (ii) the length of I is $\leqslant 5\|s-t\|$ and (iii) $\forall x \in D_P(\kappa)$, the projection of x onto this line lies inside I. We then project onto the space orthogonal to $\bar{st}$ and recurse. Property (iii) asserts that $D_P(\kappa)$ is contained inside the box we return; property (i) combined with Fact 3 allows us to infer that $\mathrm{vol}(D_P(\kappa-\Delta)) \geqslant \|s-t\| \cdot \mathrm{vol}(\Pi^{\perp st}(D_P(\kappa-\Delta)))/d$ where $\Pi^{\perp st}$ is the projection onto the subspace orthogonal to the line connecting s and t; and property (ii) asserts $\|s-t\| \geqslant |I|/5$ so that recursively we get a $5^d \cdot d!$ approximation of $\mathrm{vol}(D_P(\kappa-\Delta))$. Thus, asserting that these properties hold w.h.p. becomes the goal of our algorithm, which is far from trivial. Details appear in the full version, along with the proof of the algorithm's correctness.

▶ **Theorem 16.** *Let $P \subset \mathcal{G}^d$ be a set of points whose Tukey-region $\kappa + d\alpha^{\mathrm{qc}}(\frac{\varepsilon}{d^2+2d-1}, \frac{\delta}{d^2+2d-1}, \frac{\beta}{d^2+2d-1})$ is non-empty. Then there exists an efficient (ε,δ)-DP algorithm that w.p. $\geqslant 1-\beta$ returns a box B where $D_P(\kappa) \subset \mathsf{B}$ and $\mathrm{vol}(D_P(\kappa)) \leqslant \mathrm{vol}(\mathsf{B}) \leqslant 5^d(d!)\mathrm{vol}(D_p(\kappa-\Delta^{\mathrm{BB}}))$ for*

$$\Delta^{\mathrm{BB}}(\varepsilon,\delta,\beta) = \begin{cases} O(\frac{d^3(\upsilon+\log(d/\beta))}{\varepsilon}), & \textit{Using } \varepsilon\textit{-DP binary search} \\ \tilde{O}(\frac{d^3\log(d\upsilon/\varepsilon\delta\beta)}{\varepsilon}), & \textit{Using the ``Between Threshold'' Alg} \\ O(\frac{d^3\log(d\upsilon/\beta)}{\varepsilon} + \frac{d^3 8^{\log^*(\upsilon)}\log^*(\upsilon)\log(d\log^*(\upsilon)/\delta\beta)}{\varepsilon}), & \textit{Using the ``RecConcave'' algorithm} \end{cases}$$

From a Bounding Box to a "Fat" Input

In classic, non-private, computational geometry, the bounding-box approximation algorithm can be used to design an affine transformation T that turns the input dataset into a fat input, using a rotation and a separate rescaling of each axis so that B is mapped to $[0,1]^d$. Then, finding a kernel for the fat dataset and applying T^{-1} gives a kernel for the original set of points. Unfortunately, we cannot make a similar claim in our setting. Granted, our bounding box is (c_d,Δ)-approximation for any P; but the resulting affine transformation does not, always, guarantee that applying it turns $D_P(\kappa)$ to be (c'_d,Δ)-fat or c'_d-absolutely fat. This should be obvious, since when $D_P(\kappa-\Delta^{\mathrm{BB}})$ is drastically bigger than $D_P(\kappa)$ and B is proportional to $D_P(\kappa-\Delta^{\mathrm{BB}})$, mapping B to $[0,1]^d$ doesn't "stretch" $D_P(\kappa)$ enough to make it fat. Luckily, we show that non-comparable volumes is the only reason this transformation fails to produce a fat Tukey-region.

▶ **Lemma 17.** *Fix $\varepsilon > 0$, $\delta \geqslant 0$ and $\beta > 0$, and define Δ^{BB} as in Theorem 16. Suppose $P \subset \mathcal{G}^d$ is such that for some two parameters $\kappa \geqslant \kappa'$, where $\kappa - \kappa' \geqslant \Delta^{\mathrm{BB}}$, we have that $\mathrm{vol}(D_P(\kappa)) \geqslant \frac{1}{2}\mathrm{vol}(D_P(\kappa'))$. Then there exists a (ε,δ)-differentially private algorithm that w.p. $\geqslant 1-\beta$ computes (i) an affine transformation M that turns $M(D_P(\kappa))$ into a convex polytope which is $(c_d,\kappa-\kappa')$-fat, for $c_d = 4d^{\frac{5}{2}}5^d \cdot (d!)$, and (ii) a transformation $\tilde{M}$ making $\tilde{M}(D_P(\kappa))$ $2d \cdot 5^d \cdot (d!)$-absolutely fat.*

7 Finding a "Good" κ Privately

Our discussion in Section 6 leaves us with the question of finding a "good" κ and $\kappa' = \kappa - \Delta^{\mathrm{kernel}}$ – where $\mathrm{vol}(D_P(\kappa)) \geqslant \mathrm{vol}(D_P(\kappa-\Delta^{\mathrm{kernel}}))/2$. First, we establish that there are many such good pairs. [19] proved that if the volume of a Tukey region is non-zero, then it is at least $(d/\Upsilon)^{-d^3}$. Thus, we set $t = \lceil d^3\upsilon + d^3\log_2(d)\rceil$ and so it must hold for any series $\kappa_1 < \kappa_2 < ... < \kappa_t$ of length t that at least one pair of adjacent κ_i, κ_{i+1} is good, for otherwise the $\mathrm{vol}(D_P(\kappa_t))$ is below the lower bound of [19]. Consider the specific series where $\kappa_i = i \cdot (4\Delta^{\text{kernel}})$ and denote $m = \kappa_t$. Here, a good pair κ_i, κ_{i+1} are $4\Delta^{\text{kernel}}$ apart, therefore many κs in some interval $[\kappa_i, \kappa_{i+1}]$ are good, a fact we rely on in the design of our private algorithm.

To that end, we define the query $q_P(\kappa) \overset{\text{def}}{=} \max\left\{0 \leqslant i \leqslant \min\{\kappa-1, m-\kappa\} : \frac{\mathrm{vol}(D_P(\kappa+i))}{\mathrm{vol}(D_P(\kappa-i))} \geqslant \frac{1}{2}\right\}$. Our goal is to retrieve a κ where $q_P(\kappa) \geqslant \Delta^{\text{kernel}}$ since then $(\kappa, \kappa - \Delta^{\text{kernel}})$ is a good pair. It is obvious that $\forall \kappa, q_p(\kappa) \geqslant 0$ and that $q_P(1) = q_P(m) = 0$, but we also prove in the full version that for any neighboring P and $P' = P \cup \{x\}$ it holds that $|q_P(\kappa) - q_{P'}(\kappa+1)| \leqslant 1$. And so our ε-DP algorithm first picks a value of κ w.p. $\propto \exp(\frac{\varepsilon}{8} q_P(\kappa))$ and then adds Laplace noise (rounded to an integer) to it. Based on all of the above mentioned properties we prove that this "Shifted Exponential Mechanism" is indeed ε-DP. We then argue about its utility, which is far more straight-forward, and obtain the following conclusion.

▶ **Corollary 18.** *Fix $\varepsilon > 0, \delta \geqslant 0, \beta > 0$ and set Δ^{kernel} as in Theorem 10 and $m = 4\lceil d^3 \upsilon + d^3 \log_2(d) \rceil \Delta^{\text{kernel}}$. Let $P \subset \mathcal{G}^d$ be a set of points such that $D_P(m)$ is non-empty and non-degenerate. Then w.p. $\geqslant 1-\beta$, our "Shifted Exponential Mechanism"' returns a value κ such that $\mathrm{vol}(D_P(\kappa))/\mathrm{vol}(D_P(\kappa - \Delta^{\text{kernel}})) \geqslant 1/2$.*

References

1 Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. ACM*, 51(4):606–635, 2004.

2 Raef Bassily, Kobbi Nissim, Adam D. Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In Daniel Wichs and Yishay Mansour, editors, *Symposium on Theory of Computing, STOC*, pages 1046–1059. ACM, 2016.

3 Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. In *COLT*, pages 269–282, 2019.

4 Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, volume 8096 of *Lecture Notes in Computer Science*, pages 363–378. Springer, 2013.

5 Victor-Emmanuel Brunel. Concentration of the empirical level sets of tukey's halfspace depth. *Probability Theory and Related Fields*, 173(3-4):1165–1196, 2019.

6 Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *FOCS*, pages 634–649, 2015.

7 Mark Bun, Thomas Steinke, and Jonathan Ullman. Make up your mind: The price of online queries in differential privacy. In *Symposium on Discrete Algorithms, SODA*, pages 1306–1325. SIAM, 2017.

8 Michael A. Burr and Robert J. Fabrizio. Uniform convergence rates for halfspace depth. *Statistics & Probability Letters*, 124(C):33–40, 2017.

9 Timothy M. Chan. Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus. *Int. J. Comput. Geom. Appl.*, 12(1-2):67–85, 2002.

10 C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

11 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science (New York,*

N.Y.), 349(6248):636–638, August 2015. URL: `http://www.sciencemag.org/content/349/6248/636`.

12 Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 2006.

13 Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60. IEEE Computer Society, 2010.

14 Herbert Edelsbrunner. *Algorithms in combinatorial geometry.* Monographs in Theoretical Computer Science (10). Springer-Verlag, 1 edition, 1987.

15 Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020.

16 Sariel Har-peled. *Geometric Approximation Algorithms.* American Mathematical Society, USA, 2011.

17 Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. In *Conference on Learning Theory, COLT*, volume 100 of *Proceedings of Machine Learning Research.* PMLR, 2020.

18 Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Differentially private learning of geometric concepts. In *International Conference on Machine Learning, ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3233–3241. PMLR, 2019.

19 Haim Kaplan, Micha Sharir, and Uri Stemmer. How to find a point in the convex hull privately. In *International Symposium on Computational Geometry (SoCG)*, volume 164 of *LIPIcs*, pages 52:1–52:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

20 Xiaohui Liu, Karl Mosler, and Pavlo Mozharovskyi. Fast computation of tukey trimmed regions and median in dimension $p > 2$. *Journal of Computational and Graphical Statistics*, 28(3):682–697, 2019.

21 K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84. ACM, 2007. Full version in: `http://www.cse.psu.edu/~asmith/pubs/NRS07`.

22 Kobbi Nissim and Uri Stemmer. Clustering algorithms for the centralized and local models. In *ALT*, pages 619–653, 2018.

23 Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Locating a small cluster privately. In *PODS*, pages 413–427, 2016.

24 Peter J Rousseeuw and Ida Ruts. Constructing the bivariate tukey median. *Statistica Sinica*, 8(3):827–839, 1998.

25 Haruyuki Sanuki, Rui Fukui, Tsukasa Inajima, and Shin'ichi Warisawa. Cuff-less calibration-free blood pressure estimation under ambulatory environment using pulse wave velocity and photoplethysmogram signals. In *BIOSIGNALS*, 2017.

26 J. W. Tukey. Mathematics and the picturing of data. *Proceedings of the International Congress of Mathematicians*, 2:523–531, 1975.

27 Gary M. Weiss, Kenichi Yoneda, and Thaier Hayajneh. Smartphone and smartwatch-based biometrics using activities of daily living. *IEEE Access*, 7:133190–133202, 2019.

28 Yijun Zuo and Robert Serfling. Structural properties and convergence results for contours of sample statistical depth functions. *Ann. Statist.*, 28(2):483–499, April 2000.