Tukey Depth Mechanisms for Practical Private Mean Estimation

Gavin Brown[1], Lydia Zakynthinou[2]

[1]University of Washington, grbrown@cs.washington.edu
[2]University of California, Berkeley, lydiazak@berkeley.edu

1 Introduction

Mean estimation is one of the most ubiquitous tasks in statistics. As a result, it is a focus within differentially private [Dwork et al., 2006] statistical estimation, with a rapidly growing body of work. Of particular interest are private mean estimators for well-concentrated distributions $\mathcal{P}$, such as the class of multivariate Gaussians $\mathcal{N}(\mu, \Sigma)$, where we can achieve comparable accuracy to the non-private case.

In this ongoing work, we focus on estimators based on Tukey depth, and in particular the *Restricted Tukey Depth Mechanism* of Brown et al. [2021], henceforth BGSUZ. The Restricted Tukey Depth Mechanism has several desirable properties: it is robust, it achieves optimal accuracy (up to logarithmic factors) with respect to the tight, affine-invariant Mahalanobis error metric, and it does not require prior knowledge about the parameters of the distribution. (See Appendix F for related work on differentially private Gaussian mean estimation and comparisons.)

Background: The (Restricted) Tukey Depth Mechanism The *Tukey depth* [Tukey, 1975] of a point $y \in \mathbb{R}^d$ with respect to a data set $x \in \mathbb{R}^{n \times d}$, $T_x(y) = \frac{1}{n} \cdot \min_{v \in \mathbb{R}^d} \left| \{x_i \in x : \langle x_i, v \rangle \geq \langle y, v \rangle\} \right|$, is a classic notion of outlyingness in multivariate data, generalizing the notion of quantiles, with a long history within robust statistics. The fastest known algorithms for computing the Tukey depth require n^{d-1} time.

The Tukey depth has sensitivity $\frac{1}{n}$ and thus pairs naturally with the exponential mechanism of McSherry and Talwar [2007], which samples an output $y \in \mathcal{Y}$ from a distribution proportional to $\exp\{\varepsilon q_x(y)/2\}$, where ε is a privacy parameter and $q_x(y)$ is a "score" function that depends on the input dataset x.

To use this approach, we require a bounded output space (so that $\int \exp\{\varepsilon q_x(y)/2\} \mathrm{d}y$ has a finite integral). One solution to this obstacle requires prior knowledge about the data, for example that the dataset lives in a box $[-R, R]^d$.[1] Kaplan et al. [2020] used this approach to solve the problem of finding a point in the convex hull of a dataset. Liu et al. [2021] use the same approach for robust and pure DP mean estimation. We will refer to this algorithm as *BoxEM*.

The *Restricted Exponential Mechanism* (REM) of BGSUZ offers another solution. It restricts the mechanism to never return values of y with $q_x(y) < t$ for some threshold t. This modification does not require a parameter bound R but, since the restriction is data-dependent, the standard privacy proof for the exponential mechanism no longer applies. To work around this, BGSUZ define a notion of "safe" inputs, on which one can run this restricted mechanism while preserving privacy. They compute the Hamming distance from the input data to the space of "unsafe" data sets and apply the propose-test-release (PTR) framework of Dwork and Lei [2009]. This approach requires approximate DP, unlike BoxEM (which satisfies pure DP). This distance check involves a brute-force search over a space of possible datasets and exact enumeration of output distributions.

The main drawback of both algorithms applied to Tukey depth is computational inefficiency. Computing $T_x(y)$ takes time n^{d-1} and is thus intractable in high dimensions. Even in low dimensions, it is not immediately clear how to sample efficiently. One solution involves constructing the Tukey regions (i.e., points with the same Tukey depth $\mathcal{Y}_{=\ell} = \{y : T_x(y) = \ell\}$) and computing their volumes. These take $\tilde{O}(n^d)$ and $n^{O(d^2)}$ time, respectively. Furthermore, REM's brute force distance-to-unsafety check takes $\Omega(d)^{nd}$ time, an obstacle to implementation even in one dimension.

[1]Informally, error bounds for algorithms that satisfy concentrated or pure DP require such assumptions.

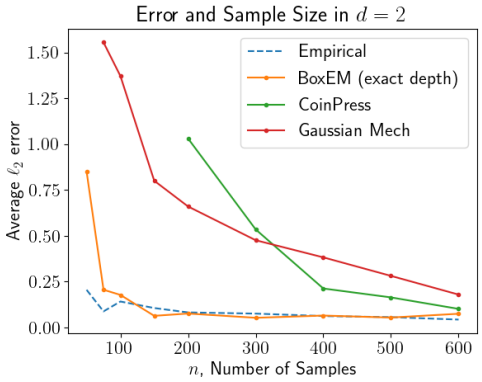

Figure 1: Mechanisms' ℓ_2 error as a function of sample size. "Empirical" represents error due to sampling. Other lines quantify the "cost of privacy," i.e., the difference between the empirical mean and the private estimate. The Tukey mechanism introduces error comparable to the empirical error at small samples sizes.

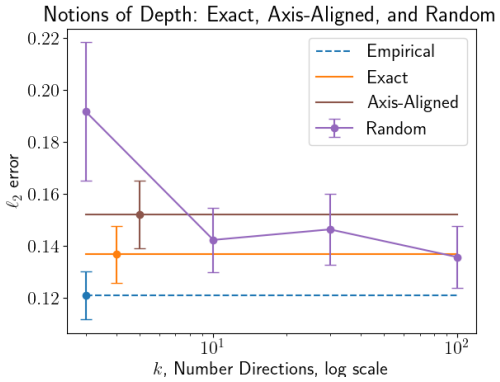

Figure 2: The error of BoxEM under different notions of depth. As more halfspaces are used, the random-depth mechanism has error close to that of exact-depth. This experiment uses $n = 200$, $d = 2$, and 200 trials. "Empirical," "Exact," and "Axis-Aligned" lines each represent a single quantity (which does not depend on k).

Our contributions: practical implementations of the (Restricted) Tukey Depth Mechanism We implement both REM and BoxEM over Tukey depth. We also implement variants that use approximate versions of Tukey depth, trading off accuracy for faster computation. We show their efficiency in practice, establishing that they are viable options for modest dimensions. Given their strong accuracy and robustness guarantees, we contend that they are competitive approaches for mean estimation in this regime.

1. **Exact implementation.** Using advances by Amin et al. [2022], we improve the distance check which now takes $n^{O(d^2)}$ time instead of $\Omega(d^{nd})$. We test this algorithm in 2, 3, and 4 dimensions.
2. **Random Tukey depth.** We show how to replace exact Tukey depth with the *random Tukey depth* [Cuesta-Albertos and Nieto-Reyes, 2008]. Considering k random directions, the depth region is now a polytope defined by k inequalities. We compute the exact volumes of these regions in nk^d time.

In Appendix A, we discuss the natural next step of incorporating approximate polytope volume computation. Existing algorithms achieve an asymptotic running time of $\tilde{O}(nk \cdot \text{poly}(d))$. Appendix E discusses approximate volume computation via rejection sampling, which is practical in low dimensions.

2 Overview of Tools from Prior Work

A natural approach for sampling from the (restricted) exponential mechanism over Tukey depth is to compute the volume of the Tukey upper-level sets $\mathcal{Y}_{\geq \ell} = \{y : T_x(y) \geq \ell\}$, which are convex, select one from an appropriate probability distribution, and sample uniformly from that set. The details of this implementation first appeared in [Kaplan et al., 2020]. This requires constructing the Tukey regions for at most $n/2$ level sets and computing their volumes. The volume of such a polytope can be computed in exponential time via *triangulation*: enumerate the polytope's vertices (i.e., turn the H *representation* into the V *representation*) and partition the body into simplices. The volume of each simplex can be efficiently computed and their sum will give us the volume of the polytope, which corresponds to the Tukey upper-level set.

Approximate Distance-to-Unsafety A key obstacle to implementation of REM is the "distance to unsafety" calculation. Amin et al. [2022] showed how to avoid this search, replacing the exact distance with a carefully

	Random Depth				Exact Depth		
Sample Size	$d = 2$	$d = 3$	$d = 4$	$d = 5$	$d = 2$	$d = 3$	$d = 4$
50	0.6	0.7	2.0	24.1	1.0	0.9	34.6
100	1.1	1.1	4.0	50.0	0.4	14.7	-
500	4.9	5.2	19.8	252.1	6.6	-	-
1000	10.0	9.4	38.1	-	46.0	-	-
2000	21.9	19.4	75.2	-	376.2	-	-

Table 1: Running time (in seconds) of Tukey depth mechanisms across samples size and dimension. "-" means the computation took longer than ten minutes. Random depth used $k = 30$ directions.

constructed approximate distance. This approximation is low sensitivity, so it plugs into the same PTR framework. It is a lower bound on the exact distance to unsafety, which implies that (when the private check passes) the restricted mechanism is still private. Finally, it is easier to compute: it requires only computing the volumes of Tukey level sets on the input dataset. The particular notion of approximate distance comes from a quantity used by BGSUZ in their analysis. Amin et al. [2022] show how it can be used algorithmically. Subsequent work of Dick et al. [2024] gave an improved distance approximation.

Approximate Tukey Depth A Tukey region takes $O(n^d)$ to construct. Outside the privacy literature, there is recent work on computationally efficient approximations to the Tukey depth [Liu et al., 2019, Fojtík et al., 2023]. Some efficient algorithms compute *exact* depth via heuristics; such algorithms do not suit our purposes, since the manner in which they fail may depend on the data and break privacy. However, we can use algorithms which *exactly* calculate *approximate* Tukey depth. In this work, we consider the *random* Tukey depth [Cuesta-Albertos and Nieto-Reyes, 2008], which replaces the minimum over all halfspaces with one over a randomly chosen list of k halfspaces. Amin et al. [2022] considered *axis-aligned* Tukey depth, which is defined as a minimum over the d directions of the canonical basis instead. This implies that every Tukey region is a high dimensional rectangle, whose volumes are trivial to compute. The accuracy guarantee however would be with respect to a hyperrectangle, which provides weaker guarantees for ℓ_2 error.

Improved Algorithms for Tukey Depth and Polytope Volume The core computational tasks in our implementations are computing Tukey regions and computing the volume of a polytope, which require exponential time in the worst case. Significant research effort has gone into producing algorithms that (at least empirically) run faster than brute force. In particular, we rely on the R package TukeyRegion [Liu et al., 2019, Fojtík et al., 2023] and the VINCI software for polytope volumes [Büeler et al., 2000]. See Appendix B for more details.

3 Experiments and Results

For our experiments we consider REM, BoxEM, AxesEM (code by [Amin et al., 2022]), the folklore Gaussian Mechanism and the practice-oriented CoinPress (code by [Biswas et al., 2020]). The simplest of these, the Gaussian Mechanism, clips the data to the $B(0, R)$ ball and releases the resulting empirical mean plus scaled Gaussian noise. CoinPress [Kamath et al., 2019, Biswas et al., 2020] iteratively refines an estimate of the mean; its error depends only polylogarithmically on R.

We present a series of experiments demonstrating these mechanisms and their performance in different regimes. All experiments were conducted on a 2019 MacBook Pro with a 1.4 GHz Quad-Core Intel Core i5. Table 1 lists running times for the Tukey mechanisms in several dimensions and sample sizes. These times are dominated by volume computation, and thus representative for both REM and BoxEM.

Our datasets are generated by selecting a point μ uniformly at random from the sphere of radius 3 and then generating points independently from $\mathcal{N}(0, \mathbb{I})$. Some plots include empirical error $\|\hat{\mu} - \mu\|_2$, where $\hat{\mu}$ is $\frac{1}{n}\sum_i x_i$. To isolate the error introduced by privacy, for the private mechanisms we report the distance $\|\tilde{\mu} - \hat{\mu}\|_2$.

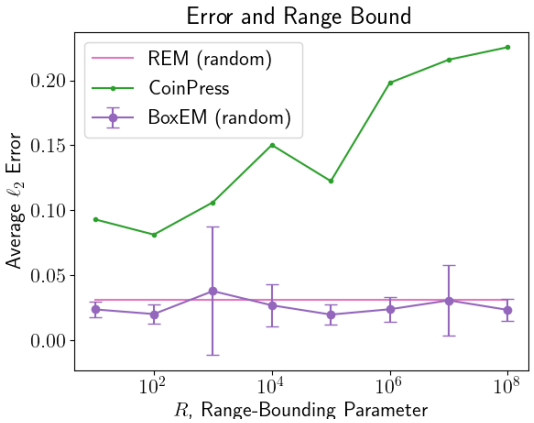

Figure 3: The ℓ_2 error of several algorithms as a function of R, the range-bounding hyperparameter. The error of BoxEM exhibits essentially no dependence on R, as in this regime with high probability it samples from a high-depth region, whose shape is independent of R. The "REM" line represents a single quantity (which does not depend on R). Note the log-log scale. This experiment uses $n = 1000$ and $d = 2$.

We omit average errors that are larger than 3, as the algorithm that returns "0" achieves this error trivially in our setting. Our default algorithmic hyperparameters are: sample size $n = 100$, dimension $d = 2$, number of random halfspaces $k = 30$, and range-bounding parameter $R = 10$. We set $\varepsilon = 1$ and $\delta = 10^{-6}$ throughout. For the pure-DP Tukey mechanism, we simply set it to be 1-DP. Unless otherwise noted, we average over 10 trials. Error bars represent 95% confidence intervals.[2]

The restricted Tukey depth mechanism has a probability of failure that depends on the particular dataset. When the algorithm fails for at least one trial on any particular experimental setting, we do not report the results from the other trials. Given the low number of trials, a single failure event likely represents an infeasible failure rate overall.

Fig. 1 presents our main result, comparing various errors as a function of sample size.

Fig. 2 illustrates how various notions of depth affect the error. The axis-aligned notion of depth used by Amin et al. [2022] admits a very fast algorithm but has higher error than the (much slower) exact-depth algorithm. The algorithm which computes depth with respect to random halfspaces allows us to spend computation and lower error: as the number of halfspaces increases, the error approaches that of exact depth. These results empirically had high variability, so the results represent 200 trials.

Fig. 3 shows the errors of CoinPress and BoxEM mechanism as a function of the range-bounding hyperparameter R. (The error of the Gaussian mechanism scales with poly(R), so it is only useful for small R.) We compare with REM, whose error is independent of R. This represents prior knowledge about the data; dependence on such a quantity is necessary for algorithms satisfying pure or concentrated differential privacy. We see this dependence clearly for CoinPress: polylog(R) shows up in the magnitude of the noise. However, the Tukey mechanism appears to have *no* dependence on R. How can this be? R controls the size of the bounding box, and thus increasing R places more of the exponential mechanism's mass outside the convex hull of the data (i.e., points with Tukey depth zero). When the mechanism samples from this space, we expect high error, roughly $\Omega(\sqrt{d}R)$. However, in the regime of these experiments,[3] with high probability the mechanism samples from a point with non-zero depth. Conditioned on this event, the output of the mechanism is, in fact, independent of R.

[2]Formally, on trial i we obtain compute error $e^{(i)} = \|\mu^{(i)} - \tilde{\mu}^{(i)}\|_2$ and we plot $\hat{e} \pm 1.96\frac{\hat{\sigma}}{\sqrt{100}}$, where $\hat{\sigma}^2 = \frac{1}{100}\sum_i (e^{(i)} - \hat{e})^2$.

[3]Consider $d = 2$, $n = 1000$, and $\varepsilon = 1$, as in this experiment. For $R = 10^{10}$, we draw from outside the convex hull with probability proportional to $e^{\varepsilon \cdot 0} \cdot (2R)^2 \approx 10^{26}$. Empirically, the set of depth $\geq n/4$ has volume around 1, so it receives weight proportional to $e^{\varepsilon n/4} \cdot 1 \approx 10^{108}$, dominating the space outside the convex hull.

References

Ishaq Aden-Ali, Hassan Ashtiani, and Gautam Kamath. On the sample complexity of privately learning unbounded high-dimensional gaussians. In *Algorithmic Learning Theory*, pages 185–216. Proceedings of Machine Learning Research, 2021.

Kareem Amin, Matthew Joseph, Mónica Ribero, and Sergei Vassilvitskii. Easy differentially private linear regression. *arXiv preprint arXiv:2208.07353*, 2022.

C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan Ullman. Coinpress: Practical private mean and covariance estimation. *arXiv preprint arXiv:2006.06618*, 2020.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Gavin Brown, Marco Gaboardi, Adam Smith, Jonathan Ullman, and Lydia Zakynthinou. Covariance-aware private mean estimation without private covariance estimation. *Advances in Neural Information Processing Systems*, 34:7950–7964, 2021.

Gavin Brown, Samuel Hopkins, and Adam Smith. Fast, sample-efficient, affine-invariant private mean and covariance estimation for subgaussian distributions. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 5578–5579. Proceedings of Machine Learning Research, 2023.

Benno Büeler, Andreas Enge, and Komei Fukuda. Exact volume computation for polytopes: a practical study. In *Polytopes—combinatorics and computation*, pages 131–154. Springer, 2000.

Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, 10(1):377–409, December 1993. ISSN 0179-5376. doi: 10.1007/BF02573985.

Juan Antonio Cuesta-Albertos and Alicia Nieto-Reyes. The random tukey depth. *Computational Statistics & Data Analysis*, 52(11):4979–4988, 2008.

Ryan Cumings-Menon. Differentially private estimation via statistical depth. *arXiv preprint arXiv:2207.12602*, 2022.

Travis Dick, Jennifer Gillenwater, and Matthew Joseph. Better private linear regression through better private feature selection. *Advances in Neural Information Processing Systems*, 36, 2024.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, STOC '09, pages 371–380. ACM, 2009.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Rainer Dyckerhoff and Pavlo Mozharovskyi. Exact computation of the halfspace depth. *Computational Statistics & Data Analysis*, 98:19–30, 2016. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2015.12.011. URL https://www.sciencedirect.com/science/article/pii/S0167947315003199.

Ioannis Z Emiris and Vissarion Fisikopoulos. Practical polytope volume approximation. *ACM Transactions on Mathematical Software (TOMS)*, 44(4):1–21, 2018.

Vít Fojtík, Petra Laketa, Pavlo Mozharovskyi, and Stanislav Nagy. On exact computation of tukey depth central regions. *Journal of Computational and Graphical Statistics*, pages 1–26, 2023.

Khashayar Gatmiry, Jonathan Kelner, and Santosh S Vempala. Sampling with barriers: Faster mixing via lewis weights. *arXiv preprint arXiv:2303.00480*, 2023.

Ziyue Huang, Yuting Liang, and Ke Yi. Instance-optimal mean estimation under differential privacy. *Advances in Neural Information Processing Systems*, 34:25993–26004, 2021.

Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan Ullman. Privately learning high-dimensional distributions. In *Conference on Learning Theory*, pages 1853–1902. Proceedings of Machine Learning Research, 2019.

Haim Kaplan, Micha Sharir, and Uri Stemmer. How to find a point in the convex hull privately. *arXiv preprint arXiv:2003.13192*, 2020.

Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94, page 44. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

Rohith Kuditipudi, John Duchi, and Saminul Haque. A pretty fast algorithm for adaptive private mean estimation. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 2511–2551. PMLR, 12–15 Jul 2023. URL https://proceedings.mlr.press/v195/kuditipudi23a.html.

Aditi Laddha, Yin Tat Lee, and Santosh Vempala. Strong self-concordance and sampling. In *Proceedings of the 52nd annual ACM SIGACT symposium on theory of computing*, pages 1212–1222, 2020.

Jean B Lasserre. An analytical expression and an algorithm for the volume of a convex polyhedron in r n. *Journal of optimization theory and applications*, 39:363–377, 1983.

Yin Tat Lee and Santosh S Vempala. Convergence rate of riemannian hamiltonian monte carlo and faster polytope volume computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1115–1121, 2018.

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.

Xiyang Liu, Weihao Kong, Sham Kakade, and Sewoong Oh. Robust and differentially private mean estimation. *Advances in Neural Information Processing Systems*, 34:3887–3901, 2021.

László Miklós Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Struct. Algorithms*, 4:359–412, 1993. URL https://api.semanticscholar.org/CorpusID:12538090.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

Andrés Muñoz Medina and Jenny Gillenwater. Duff: A dataset-distance-based utility function family for the exponential mechanism. *arXiv preprint arXiv:2010.04235*, 2020.

Kelly Ramsay and Shoja'eddin Chenouri. Differentially private depth functions and their associated medians. *arXiv preprint arXiv:2101.02800*, 2021.

Eliad Tsfadia, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Friendlycore: Practical differentially private aggregation. In *International Conference on Machine Learning*, pages 21828–21863. Proceedings of Machine Learning Research, 2022.

John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.

A Future Work: Approximate Volume and Sampling

There is a long line of work on Markov Chain Monte Carlo algorithms for approximate sampling from uniform distributions over convex bodies, and specifically polytopes. Many of these algorithms are practical and appear well-suited to our setting [see Lee and Vempala, 2018, Gatmiry et al., 2023, Laddha et al., 2020, for recent work and discussions]. Existing analyses, however, involve large constants. Theoreticians are interested in the asymptotic behavior while practitioners focus on empirical performance. For our application, however, we require practical *and formal* guarantees on the mixing times of these algorithms. We work through the privacy analysis of REM given formal PAC guarantees for a volume and sampling oracle in Appendix D.

In terms of implementation, future work could include optimizations to amortize workload across repetitions, given the specific structure of our polytopes (Tukey level sets with respect to the same data set).

Task	**Shorthand**	**Time**	**References**
Exact Tukey Depth $T_x(y)$	T_{point}	n^{d-1}	Dyckerhoff and Mozharovskyi [2016]
Construct One Tukey Polytope	T_{region}	n^d	Liu et al. [2019]
Construct All Tukey Polytopes	$T_{all-regions}$	n^d	Fojtík et al. [2023]
Exact Volume of Polytope	$T_{vol_{k,d}}$	k^d	Chazelle [1993]
Construct All Random Tukey Polytopes	$T^k_{all-regions}$	nkd	
Approx. Polytope Volume (Ball Walk)	$T_{BallWalk_{d,R}}$	d^4R^2	Lovász and Simonovits [1993]
Approx. Polytope Volume (Best-Known)	$T_{\widetilde{vol}_{k,d}}$	$k^{4/3}d^{10/3}$	Gatmiry et al. [2023]

Table 2: Collected asymptotic running times for various operations, omitting log factors. n: sample size, d: dimension, k: number of directions with respect to which we compute random Tukey depth, R: radius of enclosing ball when convex body is in isotropic position.

B Our Implementations

Our first implementation uses exact Tukey depth, exact volume computation, and exact sampling. The first two tasks are accomplished by the R package TukeyRegion [Liu et al., 2019, Fojtík et al., 2023]. With the volumes in hand, selecting the level set requires drawing a sample from a weighted list. Some care must be taken around numerical stability; see Appendix C for more discussion. We sample from the chosen level set with rejection sampling, with proposals draw uniformly from a bounding box.

Our second implementation is identical except for the substitution of random Tukey depth. We begin the algorithm by choosing a number k of halfspaces uniformly at random and, crucially, independently of the data. Each halfspace $j \in [k]$ is determined by a vector $a_j \in \mathbb{R}^d$. This means the ℓ-th level set is now a polytope $P = \{x : Ax \leq b_i\}$, where the j-th row of A is the vector a_j. There are several existing implementations which admit exact computation of polytopes, including QHull [Barber et al., 1996]. For our application, the state-of-the-art appears to be Lasserre's algorithm [Lasserre, 1983], implemented in VINCI [Büeler et al., 2000]. See Emiris and Fisikopoulos [2018] for further comparisons and discussion.

C Computational Details

Algorithm 1 $\mathcal{A}_{\varepsilon,\delta,t}(x)$ via Exact Volume Computations and Sampling

Require: Data set $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^{n\times d}$. Privacy parameters: $\varepsilon, \delta > 0$. Minimum threshold t.

Adjust parameters $\varepsilon_p \leftarrow \varepsilon/4$, $\varepsilon_e \leftarrow \varepsilon/2$, $\delta_p \leftarrow \delta$, $\delta_e \leftarrow \delta/e^{2\varepsilon_p}$. ▷ Split ε equally among PTR, $\mathcal{M}$.

Volume Computations

for $\ell = \{1, \ldots, \lfloor n/2 \rfloor\}$ **do**

  $V_{\geq \ell} \leftarrow \mathrm{Vol}(\mathcal{Y}_{\geq \ell})$ where $\mathcal{Y}_{\geq \ell} = \{y \in \mathbb{R}^d : T_x(y) \geq \ell\}$.

  $V_{=\ell} \leftarrow V_{\geq \ell} - V_{\geq \ell+1}$

PTR check

$\tilde{h}(x) \leftarrow \max\left\{\left\{k : 0 \leq k < t \text{ and } \exists g > 0 \text{ s.t. } \frac{V_{\geq t-k-1}}{V_{\geq t+k+g+1}} \cdot e^{-g\varepsilon_e/2} \leq \delta_e \cdot (4e^{\varepsilon_e})^{-1}\right\}, -1\right\}$

if $\tilde{h}(x) + z < \frac{\log(1/2\delta_p)}{\varepsilon_p}$ for $z \sim \mathrm{Lap}(1/\varepsilon_p)$ **then return** `FAIL`.

Sampling from REM: $\mathcal{M}$

Draw random level $L \in \{t, \ldots, \lfloor n/2 \rfloor\}$ with $\Pr[L = \ell] \propto V_{=\ell} e^{\varepsilon_e \ell/2}$ ▷ Equivalently, draw ℓ w.p. p_ℓ (Eq. (1)).

return $\tilde{\mu} \sim \mathrm{Uniform}(\mathcal{Y}_{\geq L} \setminus \mathcal{Y}_{\geq L+1})$ ▷ Then, sample uniformly from the convex set $\mathcal{Y}_{\geq L}$ instead.

Implementing sampling from REM. The following calculation is almost verbatim in Kaplan et al. [2020]. Pick a point y with depth in x above t; that is, $T_x(y) \geq t$. We want to sample this with probability proportional to $e^{\varepsilon T_x(y)}$, where ε will later be replaced by $\varepsilon_e/2$ in our implementation. We might think to pick a depth ℓ with probability proportional to $e^{\varepsilon\ell} \cdot \mathrm{Vol}(\mathcal{Y}_{=\ell})$ and then sample uniformly from this set. But this set is not practical as $\mathcal{Y}_{=\ell}$ is not convex. Our goal is to achieve

$$\Pr[\text{output from depth } m] = C \cdot e^{\varepsilon m} \cdot \mathrm{Vol}(\mathcal{Y}_{=m}).$$

Instead, we will pick a depth ℓ with probability p_ℓ and then sample uniformly from $\mathcal{Y}_{\geq \ell}$, since that is the more natural set to sample from. What should the p_ℓ probabilities be? The probability of picking a point at depth m is a sum over the depths from $\ell = t$ to m:

$$\begin{aligned}\Pr[\text{output depth } m] &= \sum_{\ell=t}^{m} \Pr[\text{output depth } m \mid \text{draw from } \mathcal{Y}_{\geq \ell}] \Pr[\text{draw from } \mathcal{Y}_{\geq \ell}] \\ &= \sum_{\ell=t}^{m} \frac{\mathrm{Vol}(\mathcal{Y}_{=m})}{\mathrm{Vol}(\mathcal{Y}_{\geq \ell})} \cdot p_\ell.\end{aligned}$$

For some C to be set later, then, we want

$$\mathrm{Vol}(\mathcal{Y}_{=m}) \sum_{\ell=t}^{m} p_\ell / \mathrm{Vol}(\mathcal{Y}_{\geq \ell}) = C \cdot \mathrm{Vol}(\mathcal{Y}_{=m}) e^{\varepsilon m}.$$

The volumes on both sides cancel. Let $x_\ell \stackrel{\text{def}}{=} p_\ell/\mathrm{Vol}(\mathcal{Y}_{\geq \ell})$ and $C' = Ce^{\varepsilon t}$. We have the triangular sum:

$$\begin{aligned} x_t &= Ce^{\varepsilon t} = C' \\ x_{t+1} + x_t &= Ce^{\varepsilon(t+1)} = C'e^{\varepsilon} \\ x_{t+1} &= C'e^{\varepsilon} - C' = C'e^{\varepsilon}(1 - e^{-\varepsilon}) \\ x_{t+2} + x_{t+1} + x_t &= Ce^{\varepsilon(t+2)} \\ x_{t+2} &= C'e^{2\varepsilon}(1 - e^{-\varepsilon}). \end{aligned}$$

In general, we have $x_\ell = Ce^{\varepsilon\ell}(1-e^{-\varepsilon})$, or

$$\Pr[\text{draw from } \mathcal{Y}_{\geq\ell}] = p_\ell = C \cdot \mathrm{Vol}(\mathcal{Y}_{\geq\ell}) \cdot e^{\varepsilon\ell}(1-e^{-\varepsilon}), \tag{1}$$

where $C = \left((1-e^{-\varepsilon})\sum_{\ell=t}^{\lfloor n/2\rfloor} e^{\varepsilon\ell}\mathrm{Vol}(\mathcal{Y}_{\geq\ell})\right)^{-1}$ is the normalizing constant so that $\sum_{\ell=t}^{\lfloor n/2\rfloor} p_\ell = 1$.

Note that this calculation is also useful for the exponential mechanism over a box: the volume of depth 0 gets $p_0 = C \cdot \mathrm{Vol}(\mathcal{Y}_{\geq 0}) = C(2R)^d$, for $\ell \geq 1$ $p_\ell = C \cdot \mathrm{Vol}(\mathcal{Y}_{\geq\ell})e^{\varepsilon\ell}(1-e^{-\varepsilon})$, and the normalizing constant is adjusted accordingly.

Racing Sampling for Numerical Stability As in Amin et al. [2022], see Medina and Gillenwater [2020] for a reference and proof of correctness. They attribute the algorithm to Ilya Mironov. We want to select depth ℓ with probability proportional to $\mathrm{Vol}(\mathcal{Y}_{\geq\ell}) \cdot e^{\varepsilon_e\ell/2} \cdot (1-e^{-\varepsilon_e/2})$. To do this, for each ℓ we can independently draw $U_\ell \in_R [0,1]$ and compute

$$Z_\ell = \log\log\frac{1}{U_\ell} - \log\mathrm{Vol}(\mathcal{Y}_{\geq\ell}) - \frac{\varepsilon_e\ell}{2} - \log(1-e^{-\varepsilon_e/2}).$$

Then $L = \arg\min Z_\ell$ will be correctly distributed.

C.1 Approximate Volumes and Sampling

As in Kaplan et al. [2020], the same algorithm works when we have only approximations to the volumes and we sample almost-uniformly from the convex sets. Of course, we have to track the effects of the approximations.

For the volume approximation, we assume access to a "probably approximately correct" volume oracle (Definition C.1), which replaces the exact volume computations of Lines 2-4.

Definition C.1. A *PAC volume oracle* $\mathcal{V}_{\eta,\beta}$ accepts a dataset $x \in \mathbb{R}^{n\times d}$ and parameters $\eta, \beta \in (0,1)$ and returns a list $V'_{\geq 1}, \ldots, V'_{\geq\lfloor n/2\rfloor} \in \mathbb{R}$ that satisfies the following guarantees. For any inputs x, η, β, with probability at least $1-\beta$ over the coins of the algorithm we have for all ℓ,

$$(1-\eta)\mathrm{Vol}(\mathcal{Y}_{\geq\ell,x}) \leq V'_{\geq\ell} \leq (1+\eta)\mathrm{Vol}(\mathcal{Y}_{\geq\ell,x}).$$

In the exact case, our algorithm draws $L = \ell$ in Line 7 with probability p_ℓ as computed in Eq. (1) above. Similarly, we will now use these approximate volumes to choose $L = \ell$ in Line 7 with probability

$$p'_\ell = C' \cdot V'_{\geq\ell} e^{\varepsilon_e\ell/2}(1-e^{-\varepsilon_e/2}), \tag{2}$$

where $C' = \left((1-e^{-\varepsilon_e/2})\sum_{\ell=t}^{\lfloor n/2\rfloor} e^{\varepsilon_e\ell/2}V'_{\geq\ell}\right)^{-1}$.

After choosing level L, the algorithm will use an approximate uniform sampler to sample a point from $\mathcal{Y}_{\geq L}$.

Definition C.2. Let $\mathcal{Y}$ be a convex body in $\mathbb{R}^d$ and $\tau, \zeta \in (0,1)$. An *approximate uniform sampler* $\mathcal{S}_{\tau,\zeta}$ is an algorithm which, with probability $1-\zeta$, returns a random point from $\mathcal{Y}$ with pdf U' whose total variation distance from $\mathrm{Uniform}(\mathcal{Y})$ is at most τ.

Algorithm 2 $\mathcal{A}'_{\varepsilon,\delta,t}(x)$ via Approximate Volume Computations and Sampling

Require: Data set $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^{n\times d}$. Privacy parameters: $\varepsilon, \delta > 0$. Minimum threshold t.

1: Adjust parameters $\varepsilon_p \leftarrow \varepsilon/6$, $\varepsilon_e \leftarrow 2\varepsilon/5$, $\delta_p \leftarrow \delta/2$, $\delta_e \leftarrow \delta/(4e^{11\varepsilon/20})$ ▷ Split ε equally among PTR, $\mathcal{M}'$.

2: Set volume approximation parameters $\eta \leftarrow \frac{e^{\varepsilon/20}-1}{e^{\varepsilon/20}+1}$, $\beta \leftarrow \min\{1/4, \delta/(8(2e^{\epsilon}+1))\}$.

3: Set approximate sampling parameters $\tau \leftarrow \delta/(4e^{11\varepsilon/20}(1+e^{\varepsilon/2}))$, $\zeta \leftarrow \min\{1/4, \delta/(8(2e^{\epsilon}+1))\}$.

Volume Computations

4: $\{V'_{\geq \ell}\}_{\ell\in[\lfloor n/2\rfloor]} \leftarrow \mathcal{V}_{\eta,\beta}(x)$.

PTR check

5: $\tilde{h}'(x) \leftarrow \max\left\{\left\{k : 0 \leq k < t \text{ and } \exists g > 0 \text{ s.t. } \frac{V'_{\geq t-k-1}}{V'_{\geq t+k+g+1}} \cdot e^{-g\varepsilon_e/2} \leq \delta_e \cdot (4e^{\varepsilon_e + \log\frac{1+\eta}{1-\eta}})^{-1}\right\}, -1\right\}$

6: **if** $\tilde{h}'(x) + z < \frac{\log(1/2\delta_p)}{\varepsilon_p}$ for $z \sim \mathrm{Lap}(1/\varepsilon_p)$ **then return** `FAIL`.

Sampling from REM: $\mathcal{M}'$

7: Draw random level $L \in \{t, \ldots, \lfloor n/2\rfloor\}$ with $\Pr[L = \ell] = p'_\ell$ as in Eq. (2).

8: **return** $\tilde{\mu}' \leftarrow \mathcal{S}_{\tau,\zeta}(\mathcal{Y}_{\geq L})$

The privacy guarantees of Algorithm 2, which uses approximate volume computations and sampling, are in Appendix D.

D Privacy Analyses

D.1 BoxEM, Exact Volume, Exact Sampling

The privacy analysis in this setting is just that of the standard exponential mechanism, which samples y from $p(y) \propto \exp\{\varepsilon T_x(y)/2\}$ over the domain $[-R, R]^d$.

D.2 REM, Exact Volume, Exact Sampling

The analysis is almost completely contained in BGSUZ, with the exception that $\tilde{h}$ is now 2-sensitive instead of 1-sensitive, so the total privacy guarantee is $(2\varepsilon_p + \varepsilon_e, \max\{e^{2\varepsilon_p}\delta_e, \delta_p\})$.

D.3 REM, Approximate Volume, Approximate Sampling

We can show the following equivalent of Lemma 5.7 from Kaplan et al. [2020] for the REM.

Lemma D.1. *Let data set x. Let $\mathcal{M}$ be the REM of Lines 7-8 in Algorithm 1 and $\mathcal{M}'$ the REM of Lines 7-8 in Algorithm 2 which uses approximate volume oracle $\mathcal{V}_{\eta,\beta}$ and uniform sampler $\mathcal{S}_{\tau,\zeta}$. Let G be the event that the guarantees of Def. C.1, C.2 hold and $B \subseteq \mathbb{R}^d$ be any measurable set. Then*

$$\frac{1-\eta}{1+\eta}\Pr[\mathcal{M}(x) \in B] - \tau \leq \Pr[\mathcal{M}'(x) \in B \mid G] \leq \frac{1+\eta}{1-\eta}\Pr[\mathcal{M}(x) \in B] + \tau.$$

Proof Sketch. To prove the lemma, as in Kaplan et al. [2020], we condition on G and expand $\Pr[\mathcal{M}'(x) \in B] = \sum_{\ell=t}^{\lfloor n/2\rfloor} \Pr[\mathcal{M}'(x) \in B \mid \mathcal{M}'(x) \in \mathcal{Y}_{\geq\ell}] \cdot \Pr[\mathcal{M}'(x) \in \mathcal{Y}_{\geq\ell}] = \sum_{\ell=t}^{\lfloor n/2\rfloor}(\frac{\mathrm{Vol(B)}}{\mathrm{Vol}(\mathcal{Y}_{\geq\ell})} \pm \tau) \cdot p'_\ell$, where the latter holds by the guarantees of the approximate uniform sampler. By the guarantees of the PAC volume oracle, we also get that

$$\frac{1-\eta}{1+\eta}p_\ell \leq p'_\ell \leq \frac{1+\eta}{1-\eta}p_\ell,$$

which would complete the proof. □

Lemma D.1 allows us to connect the privacy parameters of $\mathcal{M}'$ to the ones of $\mathcal{M}$: if G_x and G_y are the events that the guarantees of Def. C.1 and C.2 hold for the executions of Algorithm 2 on neighboring data sets x and y, respectively, and $G = G_x \wedge G_y$, then $\mathcal{M}(x) \approx_{\varepsilon_e,\delta_e} \mathcal{M}(y) \Rightarrow \mathcal{M}'(x)_{|G} \approx_{\varepsilon_2,\delta_2} \mathcal{M}'(y)_{|G}$ for

$$\varepsilon_2 = \varepsilon_e + 2\log\frac{1+\eta}{1-\eta}, \quad \delta_2 = \frac{1+\eta}{1-\eta}\delta_e + \left(1 + e^{\varepsilon_e + 2\log\frac{1+\eta}{1-\eta}}\right)\tau. \tag{3}$$

Lemma D.2. *Let G_x and G_y be the events that the guarantees of Def. C.1 and C.2 hold for the executions of Algorithm 2 on neighboring data sets x and y, respectively and condition on those events. The sensitivity of the approximate distance function $\tilde{h}'$ (Line 6 of Algorithm 2) is $|\tilde{h}'(x) - \tilde{h}'(y)| \le 2\left(\frac{4\log\frac{1+\eta}{1-\eta}}{\varepsilon_e} + 1\right)$. The (regular) PTR check then satisfies $(8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta} + 2\varepsilon_p)$-DP. Moreover, $D_H(x, \texttt{UNSAFE}_{\varepsilon_e,\delta_e,t}) > \tilde{h}'(x)$.*

Proof Sketch. The proof follows Lemma 3.6 of Amin et al. [2022]. We can show that for neighboring data sets x, y (in the add/remove model), and fixed $k_x > 0, g_x > 0$,

$$\frac{V'_{\ge t-k_y-1,y}}{V'_{\ge t+k_y+g_y+1,x}} \cdot e^{-g_y\varepsilon_e/2} \le \frac{V'_{\ge t-k_x-1,x}}{V'_{\ge t+k_x+g_x+1,x}} \cdot e^{-g_x\varepsilon_e/2},$$

for $g_y = g_x + \frac{4\log\frac{1+\eta}{1-\eta}}{\varepsilon_e} > 0$ and $k_y = k_x - \frac{4\log\frac{1+\eta}{1-\eta}}{\varepsilon_e} - 1$. So $\tilde{h}'(y) \ge \tilde{h}'(x) - \frac{4\log\frac{1+\eta}{1-\eta}}{\varepsilon_e} - 1$. The remaining steps of the proof are the same (and we account for a factor of 2 due to switching to the swap model).

Moreover, we can relate the approximate distance using exact volumes $\tilde{h}(x)$ (Line 6 in Algorithm 1) to $\tilde{h}'(x)$. Specifically, $\tilde{h}'(x) = k$ means that there exists some $g > 0$ so that $\frac{V'_{\ge t-k-1,x}}{V'_{\ge t+k+g+1,x}} \cdot e^{-g\varepsilon_e/2} \le \frac{1-\eta}{1+\eta}\delta_e(4e^{\varepsilon_e})^{-1}$. Under G_x, this implies that $\frac{V_{\ge t-k-1,x}}{V_{\ge t+k+g+1,x}} \cdot e^{-g\varepsilon_e/2} \le \frac{1+\eta}{1-\eta}\frac{1-\eta}{1+\eta}\delta_e(4e^{\varepsilon_e})^{-1} = \delta_e(4e^{\varepsilon_e})^{-1}$ so $\tilde{h}(x) \ge k = \tilde{h}'(x)$. By Lemma 3.8 in [Brown et al., 2021], $D_H(x, \texttt{UNSAFE}_{\varepsilon_e,\delta_e,t}) > \tilde{h}(x)$ and this completes the proof. □

We derive the overall privacy guarantee of Algorithm 2: we first condition on the event G that the approximate volume computations and sampling are successful and then account for this event, using the following general proposition:

Proposition D.3. *If $\mathcal{A}(x)_{|G} \approx_{\varepsilon,\delta} \mathcal{A}(y)_{|G}$, then $\mathcal{A}(x) \approx_{\varepsilon,\delta+\Pr[\bar{G}](e^{\varepsilon}\Pr[G]^{-1}+1)} \mathcal{A}(y)$.*

Proof. Let x, y be neighboring data sets and B be any measurable set.

$$\begin{aligned}
\Pr[A(x) \in B] &\le \Pr[A(x) \in B \mid G] + \Pr[\bar{G}] \\
&\le e^{\varepsilon}\Pr[A(y) \in B \mid G] + \delta + \Pr[\bar{G}] && \text{(by assumption)} \\
&\le e^{\varepsilon}\Pr[A(y) \in B]\Pr[G]^{-1} + \delta + \Pr[\bar{G}] \\
&= e^{\varepsilon}\big(\Pr[A(y) \in B] + \Pr[A(y) \in B](\Pr[G]^{-1} - 1)\big) + \delta + \Pr[\bar{G}] \\
&\le e^{\varepsilon}\Pr[A(y) \in B] + e^{\varepsilon}(\Pr[G]^{-1} - 1) + \delta + \Pr[\bar{G}] \\
&= e^{\varepsilon}\Pr[A(y) \in B] + e^{\varepsilon}\Pr[\bar{G}]\Pr[G]^{-1} + \delta + \Pr[\bar{G}]
\end{aligned}$$

□

Theorem D.4. *Algorithm 2 is (ε, δ)-DP for*

$$\varepsilon = \left(8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta} + 2\varepsilon_p\right) + \left(\varepsilon_e + 2\log\frac{1+\eta}{1-\eta}\right)$$

and

$$\delta = \max\left\{\delta_p, e^{8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta}+2\varepsilon_p}\left(\frac{1+\eta}{1-\eta}\delta_e + \left(1 + e^{\varepsilon_e+2\log\frac{1+\eta}{1-\eta}}\right)\tau\right)\right\} + 2(2e^{\varepsilon}+1)(\beta+\zeta).$$

Proof Sketch. Denote the algorithm by $\mathcal{A}'$ for short. Let G the event that the guarantees of Def. C.1 and C.2 hold for the executions of $\mathcal{A}'$, on neighboring data sets x and y simultaneously.

We follow the proof of Proposition D.1 from Brown et al. [2021] and split the proof into two cases: $\mathcal{M}(x) \not\approx_{\varepsilon_e,\delta_e} \mathcal{M}(y)$ and $\mathcal{M}(x) \approx_{\varepsilon_e,\delta_e} \mathcal{M}(y)$. The former case is easy: it implies that $D_H(x, \texttt{UNSAFE}_{\varepsilon_e,\delta_e,t}) = D_H(y, \texttt{UNSAFE}_{\varepsilon_e,\delta_e,t}) = 0$ so $\tilde{h}'(x) = \tilde{h}'(y) = -1$ by Lemma D.2. The PTR check will then guarantee $(0, \delta_p)$-DP in that case since the probability of failure is the same.

Now, assume $\mathcal{M}(x) \approx_{\varepsilon_e,\delta_e} \mathcal{M}(y)$. By Lemma D.2 the probability of failing under x, y can differ by at most a factor of e^{ε_1} where $\varepsilon_1 = 8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta} + 2\varepsilon_p$. For a measurable set B, and denoting $F = \texttt{FAIL}$, we break down the probability:

$$\begin{aligned}
\Pr[\mathcal{A}'(x) \in B \mid G] &= \Pr[\mathcal{A}'(x) \in B \cap F \mid G] + \Pr[\mathcal{A}'(x) \in B \setminus F \mid G] \\
&= \Pr[\mathcal{A}'(x) \in B \mid \mathcal{A}'(x) \in F \wedge G]\Pr[\mathcal{A}'(x) \in F \mid G] \\
&\quad + \Pr[\mathcal{A}'(x) \in B \mid \mathcal{A}(x) \notin F \wedge G]\Pr[\mathcal{A}'(x) \notin F \mid G] \\
&\le e^{\varepsilon_1}\Big(\Pr[\mathcal{A}'(x) \in B \mid \mathcal{A}(x) \in F \wedge G]\Pr[\mathcal{A}'(y) \in F \mid G] \\
&\quad + \Pr[\mathcal{A}'(x) \in B \mid \mathcal{A}'(x) \notin F \wedge G]\Pr[\mathcal{A}'(y) \notin F \mid G]\Big).
\end{aligned}$$

Since B either contains $\texttt{FAIL}$ or it doesn't, $\Pr[\mathcal{A}'(x) \in B \mid \mathcal{A}'(x) \in F \wedge G] = \Pr[\mathcal{A}'(y) \in B \mid \mathcal{A}(y) \in F \wedge G]$. Furthermore, not failing means we run $\mathcal{M}'(x)$. Let $\varepsilon_2 = \varepsilon_e + 2\log\frac{1+\eta}{1-\eta}$ and $\delta_2 = \frac{1+\eta}{1-\eta}\delta_e + \left(1 + e^{\varepsilon_e + 2\log\frac{1+\eta}{1-\eta}}\right)\tau$. By our assumption and Eq. (3), we have

$$\begin{aligned}
&\Pr[\mathcal{A}'(x) \in B \mid G] \\
&\le e^{\varepsilon_1}(\Pr[\mathcal{A}'(y) \in B \cap F \mid G] + \Pr[\mathcal{M}'(x) \in B \mid G]\Pr[\mathcal{A}'(y) \notin F \mid G]) \\
&\le e^{\varepsilon_1}(\Pr[\mathcal{A}'(y) \in B \cap F \mid G] + (e^{\varepsilon_2}\Pr[\mathcal{M}'(y) \in B \mid G] + \delta_2)\Pr[\mathcal{A}'(y) \notin F \mid G]).
\end{aligned}$$

To finish the proof, we simplify: $\Pr[\mathcal{A}'(x) \in B \mid G] \le e^{\varepsilon_1+\varepsilon_2}\Pr[\mathcal{A}'(y) \in B \mid G] + e^{\varepsilon_1}\delta_2$.

The overall privacy guarantee for both cases, conditioned on G, is then $(\varepsilon_G, \delta_G)$-DP for

$$\varepsilon_G = \left(8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta} + 2\varepsilon_p\right) + \left(\varepsilon_e + 2\log\frac{1+\eta}{1-\eta}\right)$$

and

$$\delta_G = \max\left\{\delta_p, e^{8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta}+2\varepsilon_p}\left(\frac{1+\eta}{1-\eta}\delta_e + \left(1 + e^{\varepsilon_e+2\log\frac{1+\eta}{1-\eta}}\right)\tau\right)\right\}.$$

By Proposition D.3, we take into account the event G, which has $\Pr[\bar{G}] \le 2(\beta + \zeta) < 1/2$. We conclude that the overall privacy parameters are

$$\varepsilon = \varepsilon_G = \left(8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta} + 2\varepsilon_p\right) + \left(\varepsilon_e + 2\log\frac{1+\eta}{1-\eta}\right)$$

and

$$\delta \le \delta_G + \Pr[\bar{G}](\Pr[G]^{-1}e^{\varepsilon} + 1) \le \max\left\{\delta_p, e^{8\frac{\varepsilon_p}{\varepsilon_e}\log\frac{1+\eta}{1-\eta}+2\varepsilon_p}\left(\frac{1+\eta}{1-\eta}\delta_e + \left(1 + e^{\varepsilon_e+2\log\frac{1+\eta}{1-\eta}}\right)\tau\right)\right\} + 2(2e^{\varepsilon} + 1)(\beta + \zeta).$$

□

E Approximate Volume Computation via Rejection Sampling

A simple and common approach for sampling and approximating the volume of a polytope P is *rejection sampling*. In our work, we can apply it to Tukey regions $\mathcal{Y}_{\ge\ell,x} \subseteq \mathbb{R}^d$.

Sampling is straightforward. However in our case, volume approximations require formal accuracy guarantees, which affect the privacy of our algorithms. We establish how many samples one needs to draw so that the volume approximation satisfies the formal (η, β)-accuracy guarantee of Definition C.1.

Producing Inner and Outer Balls/Boxes Consider polytope $P = \{x : Ax \leq b\}$. Rejection sampling requires a vector c and real numbers r, R such that $B(c, r) \subseteq P \subseteq B(c, R)$. (The balls do not need to have the same center, but we can make this assumption for simplicity.)

There exists a linear program which finds c and r such that r is maximized. In particular, c is called the *Chebyshev center* (see [Boyd and Vandenberghe, 2004, Section 8.5]). As finding the minimum R such that $P \in B(c, R)$ is in general intractable when P is in H-representation, we find the ball around the convex hull of the data instead.

Estimating the Volume We will run rejection sampling with proposals from $B(c, R)$ and count the number of points that lie in P. Our volume estimate will then be (fraction accepted)×(volume of $B(c, R)$). We have a formula for the latter quantity, so it remains to estimate the probability of acceptance. Call this quantity $q = \mathrm{Vol}(P)/\mathrm{Vol}(B(c, R))$. We want to produce an estimate $\hat{q} \in (1 \pm \eta)q$ for some parameter η. This is equivalent to asking that $|q - \hat{q}| \leq \eta q$. The proposals are independent, so by a Chernoff bound, for m proposals we have

$$\Pr[|q - \hat{q}| \geq \eta q] \leq 2 \exp\{-2m\eta^2 q^2\},$$

which is at most β when $m \geq \frac{\ln 2/\beta}{2\eta^2 q^2}$.

How Many Proposals? We know that q, the true acceptance probability, is no smaller than $\frac{\mathrm{Vol}(B(c,r))}{\mathrm{Vol}(B(c,R))} = (r/R)^d$. Thus, it suffices to draw

$$m \geq \left(\frac{R}{r}\right)^{2d} \cdot \frac{\ln 2/\beta}{2\eta^2}.$$

This would satisfy the guarantees of a PAC volume oracle as in Definition C.1 with parameters $(\eta, n\beta/2)$.

When $r \approx R$, the running time of this approach will be small. We expect that r/R should be approximately the condition number of the covariance matrix of the data, which for example is $\kappa = 1$ for spherical Gaussians.

F Related work

There is a great deal of work on private Gaussian mean estimation. We focus on the central model of differential privacy. [Karwa and Vadhan, 2018] first established approaches for learning univariate Gaussians with optimal error. In the multivariate case with known covariance, there is a folklore multivariate mean estimator, based on [Karwa and Vadhan, 2018], which knows or privately computes an approximation of a ball where the data lie $B(c, R)$, clips them to that ball, and releases the resulting empirical mean using the Gaussian mechanism. Its error scales linearly with R. [Kamath et al., 2019, Aden-Ali et al., 2021] and the practice-oriented [Biswas et al., 2020] give estimators with near-optimal error for the known covariance case, whose error depends logarithmically on a priori bounds on the range R of the data. If the covariance is unknown, the same estimators can still provide affine-invariant guarantees, albeit with a worse sample complexity with respect to the dimension d and a logarithmic dependence on the condition number of the covariance matrix κ.

Liu et al. [2021] and BGSUZ were the first approaches for the unknown covariance case whose sample complexity depends optimally on the dimension d. The Tukey Depth Mechanism of Liu et al. [2021] requires prior knowledge of parameter bounds, and its accuracy depends logarithimcally on these parameter, whereas the Restricted Tukey Depth Mechanism of BGSUZ is free of any dependence on parameter bounds. Both approaches are computationally inefficient. Brown et al. [2023] (and Kuditipudi et al. [2023] with a slightly worse sample complexity) give the first polynomial-time algorithms whose sample complexity has no dependence on parameter bounds and almost matches the exponential-time Restricted Tukey Depth Mechanism (Theorem F.1). Other work on computing depth functions privately (such as the Tukey depth) includes [Ramsay and Chenouri, 2021, Cumings-Menon, 2022].

[Huang et al., 2021, Tsfadia et al., 2022] give polynomial-time algorithms for private aggregation which are not designed specifically for Gaussians but aim to minimize the dependence of the error on the range of

the data. These are also suitable for learning multivariate Gaussian means and would have guarantees similar to Biswas et al. [2020] for the known covariance case.

Properties of the (Restricted) Tukey Depth Mechanism The Restricted Tukey Depth Mechanism has certain desirable properties. First, it does not require any prior knowledge of parameters of the distribution, such as the covariance matrix Σ, its condition number κ, or a range R such that $\|\mu\|_2 \leq R$. This allows the data analyst to use it without spending privacy budget to estimate these hyperparameters (or guessing them, affecting its accuracy). Additionally, not only does the algorithm not need to know R, but also its accuracy does not depend on it at all.

Second, it has asymptotically optimal accuracy guarantees (Theorem F.1). The guarantee holds with respect to Mahalanobis distance $\|\hat{\mu} - \mu\|_\Sigma = \|\Sigma^{-1/2}(\hat{\mu} - \mu)\|_2$, an affine-invariant error metric which tightly captures the uncertainty of the true mean, and characterizes the total variation distance between $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\hat{\mu}, \Sigma)$ up to constants. It can be relaxed to a euclidean ball guarantee $\|\hat{\mu} - \mu\|_2 \leq \alpha\sqrt{\|\Sigma\|_2}$. Furthermore, the dependence on δ in the sample complexity is decoupled from d, which implies that we can ask for δ to be very small, in the order of e^{-d}. Finally, the constant C is not too large, overall making it an algorithm whose accuracy is expected to be good in practice.

Theorem F.1 (Theorem 3.2 [Brown et al., 2021])**.** *For any $\varepsilon, \delta > 0$, the Restricted Tukey Depth Mechanism is (ε, δ)-differentially private. There exists an absolute constant C such that, for any $0 < \alpha, \beta, \varepsilon < 1$, $0 < \delta \leq \frac{1}{2}$, mean μ, and positive definite Σ, if $x \sim \mathcal{N}(\mu, \Sigma)^{\otimes n}$ and*

$$n \geq C\left(\frac{d + \log(1/\beta)}{\alpha^2} + \frac{d + \log(1/\alpha\varepsilon\beta)}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right),$$

then with probability at least $1 - \beta$, $\|\mathcal{A}(x) - \mu\|_\Sigma \leq \alpha$.

Third, the algorithm is robust to data corruptions in the strong contamination model, which means that arbitrary adversarial changes to at most αn points in its input, do not affect its accuracy (up to constants).

For our experiments, we examine the simplest apporach of the Gaussian Mechanism, the practice-oriented CoinPress [Biswas et al., 2020], and Tukey-based Mechanisms: the Tukey Depth Mechanism over the hypercube of [Liu et al., 2021] (BoxEM), the Restricted Tukey Depth Mechanism of BGSUZ (REM), and the Axis-Aligned Tukey Depth Mechanism of [Amin et al., 2022] (AxesEM).