
Private Geometric Median in Nearly-Linear Time

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Estimating the geometric median of a dataset is a robust counterpart to mean estimation,
2 and is a fundamental problem in computational geometry. Recently, [HSU24]
3 gave an (ϵ, δ) -differentially private algorithm obtaining an α -multiplicative approx-
4 imation to the geometric median objective, $\frac{1}{n} \sum_{i \in [n]} \|\cdot - \mathbf{x}_i\|$, given a dataset
5 $\mathcal{D} := \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{R}^d$. Their algorithm requires $n \gtrsim \sqrt{d} \cdot \frac{1}{\alpha\epsilon}$ samples, which they
6 prove is information-theoretically optimal. This result is surprising because its
7 error scales with the *effective radius* of \mathcal{D} (i.e., of a ball capturing most points),
8 rather than the worst-case radius. We give an improved algorithm that obtains
9 the same approximation quality, also using $n \gtrsim \sqrt{d} \cdot \frac{1}{\alpha\epsilon}$ samples, but in time
10 $\tilde{O}(nd + \frac{d}{\alpha^2})$. Our runtime is nearly-linear, plus the cost of the cheapest non-private
11 first-order method due to [CLM⁺16]. To achieve our results, we use subsampling
12 and geometric aggregation tools inspired by FriendlyCore [TCK⁺22] to speed up
13 the “warm start” component of the [HSU24] algorithm, combined with a careful
14 custom analysis of DP-SGD’s sensitivity for the geometric median objective.

15 1 Introduction

16 The *geometric median* problem, a.k.a. the Fermat-Weber problem, is one of the oldest problems in
17 computational geometry. In this problem, we are given a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{R}^d$, and our goal
18 is to find a point $\mathbf{x}_* \in \mathbb{R}^d$ that minimizes the average Euclidean distance to points in the dataset:

$$\mathbf{x}_* \in \arg \min_{\mathbf{x} \in \mathbb{R}^d} f_{\mathcal{D}}(\mathbf{x}), \text{ where } f_{\mathcal{D}}(\mathbf{x}) := \frac{1}{n} \sum_{i \in [n]} \|\mathbf{x} - \mathbf{x}_i\|. \quad (1)$$

19 This problem has received widespread interest due to its applications in high-dimensional statistics.
20 In particular, the geometric median of a dataset \mathcal{D} enjoys robustness properties that the mean (i.e.,
21 $\frac{1}{n} \sum_{i \in [n]} \mathbf{x}_i$, the minimizer of $\frac{1}{n} \sum_{i \in [n]} \|\mathbf{x} - \mathbf{x}_i\|^2$) does not. For example, it is known (cf. Lemma 4)
22 that if greater than half of \mathcal{D} lies within a distance r of some $\bar{\mathbf{x}} \in \mathbb{R}^d$, then the geometric median lies
23 within $O(r)$ of $\bar{\mathbf{x}}$. Thus, the geometric median provides strong estimation guarantees even when \mathcal{D}
24 contains outliers. This is in contrast to simpler estimators such as the mean, which can be arbitrarily
25 corrupted by a single outlier. As a result, studying the properties and computational aspects of the
26 geometric median has a long history, see e.g., [Web29, LR91] for some famous examples.

27 We provide improved algorithms for estimating (1) subject to (ϵ, δ) -differential privacy (DP, Defini-
28 tion 1), the de facto notion of provable privacy in modern machine learning. Privately computing the
29 geometric median naturally fits into a recent line of work on designing DP algorithms in the presence
30 of outliers. To explain the challenge of such problems, the definition of DP implies that privacy must
31 hold for *worst-case* datasets. This stringent definition affords DP a variety of desirable properties,
32 most notably *composition* of private mechanisms (cf. [DR14], Section 3.5). However, it also begets
33 challenges: for example, estimating the empirical mean of \mathcal{D} subject to (ϵ, δ) -DP necessarily results

in error scaling $\propto R$, the diameter of the dataset (cf. Section 5, [BST14]). Moreover, the worst-case nature of DP is at odds with typical *average-case* machine learning settings, where most (or all) of \mathcal{D} is drawn from a distribution that we wish to learn about. From an algorithm design standpoint, the question follows: how do we design methods with privacy guarantees for worst-case data, but also with improved utility guarantees for (mostly) average-case data?

Such questions have been successfully addressed for various statistical problems, e.g., parameter estimation [BD14, KV17, BKS19, DFM⁺20, BDKU20, BGS⁺21, AL22, LKJO22, KDH23, BHS23], clustering [NRS07, NSV16, CKM⁺21, TCK⁺22], and more. However, existing approaches for (1) (even non-privately) are based on iterative optimization methods, as the geometric median does not admit a simple, closed-form solution. Much of the DP optimization toolkit is exactly plagued by the aforementioned “worst-case sensitivity” issues, e.g., lower bounds for general stochastic optimization problems again scale with the domain size. This is troubling in the context of (1), because a major appeal of the geometric median is its robustness: its error should not be significantly affected by any small subset of the data. Privately estimating the geometric median thus poses an interesting technical challenge, beyond its potential appeal as a subroutine in downstream robust algorithms.

To explain the distinction between worst-case and average-case error rates in the context of (1), we introduce the following helpful notation: for all quantiles $\tau \in [0, 1]$, we let

$$r^{(\tau)} := \arg \min_{r \geq 0} \left\{ \sum_{i \in [n]} \mathbb{1}_{\|\mathbf{x}_i - \mathbf{x}_\star\| \leq r} \geq \tau n \right\}, \text{ where } \mathbf{x}_\star := \arg \min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{n} \sum_{i \in [n]} \|\mathbf{x} - \mathbf{x}_i\|, \quad (2)$$

when $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{R}^d$ is clear from context. In other words, $r^{(\tau)}$ is the smallest radius describing a ball around the geometric median \mathbf{x}_\star containing at least τn points in \mathcal{D} . We also use R to denote an a priori overall domain size bound, where we are guaranteed that $\mathcal{D} \subset \mathbb{B}^d(R)$. Note that in general, it is possible for, e.g., $r^{(0.9)} \ll R$ if $\approx 10\%$ of \mathcal{D} consists of outliers with atypical norms. Due to the robust nature of the geometric median (i.e., the aforementioned Lemma 4), a natural target is estimation error scaling with the “effective radius” $r^{(\tau)}$ for some quantile $\tau \in (0.5, 1)$. This is a much stronger guarantee than the error rates $\propto R$ that typical DP optimization methods give.

Because a simple argument (Lemma 3) shows $r^{(\tau)} = O(f_{\mathcal{D}}(\mathbf{x}_\star))$ for all $\tau < 1$, in this introduction our goal is to approximate the minimizer of (1) to additive error $\alpha f_{\mathcal{D}}(\mathbf{x}_\star)$ for some $\alpha \in (0, 1)$, i.e., to give α -multiplicative error guarantees on optimizing $f_{\mathcal{D}}$.¹ Again, datasets with outliers may have $f_{\mathcal{D}}(\mathbf{x}_\star) \ll R$, so this is beyond the reach of naively applying DP optimization methods.

In a recent exciting work, [HSU24] bypassed this obstacle and obtained such private multiplicative approximations to the geometric median, and with near-optimal sample complexity. Assuming that \mathcal{D} has size $n \gtrsim \sqrt{d} \cdot \frac{1}{\alpha\epsilon}$,² [HSU24] gave two algorithms for estimating (1) to α -multiplicative error (cf. Appendix A). They also proved a matching lower bound, showing this many samples is information-theoretically necessary.³ From both a theoretical and practical perspective, the main outstanding question left by [HSU24] is computational efficiency: in particular, the [HSU24] algorithms ran in time $\tilde{O}(n^2d + n^3\epsilon^2)$ or $\tilde{O}(n^2d + nd^2 + d^{4.372})$. This leaves a significant gap between algorithms for privately solving (1), and their counterparts in the non-private setting, where [CLM⁺16] showed that (1) could be approximated to α -multiplicative error in nearly-linear time $\tilde{O}(\min(nd, \frac{d}{\alpha^2}))$.

1.1 Our results

Our main contribution is a faster algorithm for privately approximating (1) to α -multiplicative error.

Theorem 1 (informal, see Theorem 4). *Let $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{B}^d(R)$ for $R > 0$, $0 < r \leq r^{(0.9)}$, and $(\alpha, \epsilon, \delta) \in [0, 1]^3$. There is an (ϵ, δ) -DP algorithm that returns $\hat{\mathbf{x}}$ such that with probability $\geq 1 - \delta$, $f_{\mathcal{D}}(\hat{\mathbf{x}}) \leq (1 + \alpha)f_{\mathcal{D}}(\mathbf{x}_\star)$, assuming $n \gtrsim \frac{\sqrt{d}}{\alpha\epsilon}$. The algorithm runs in time $\tilde{O}(nd + \frac{d}{\alpha^2})$.*

¹Our results (and those of [HSU24]) in fact give stronger additive error bounds of $\alpha r^{(\tau)}$ for any $\tau \in (0.5, 1)$.

²In this introduction only, we use $\tilde{O}, \lesssim, \gtrsim$ to hide polylogarithmic factors in problem parameters, i.e., $d, \frac{1}{\alpha}, \frac{1}{\epsilon}, \frac{1}{\delta}$, and $\frac{R}{r}$, where $\mathcal{D} \subseteq \mathbb{B}^d(R)$ and $r \leq r^{(0.9)}$. Our formal theorem statements explicitly state all dependences.

³Intuitively, we require $\alpha \approx d^{-1/2}$ to obtain nontrivial estimation when \mathcal{D} consists of i.i.d. Gaussian data (as a typical radius is $\approx \sqrt{d}$), matching known lower bounds of $n \approx \frac{d}{\epsilon}$ for Gaussian mean estimation [KLSU19].

To briefly explain Theorem 1’s statement, it uses a priori knowledge of $0 < r < R$ such that R upper bounds the domain size of \mathcal{D} , and r lower bounds the “effective radius” $r^{(0.9)}$. However, its runtime only depends polylogarithmically on the aspect ratio $\frac{R}{r}$, rather than polynomially (as naïve DP optimization methods would); we also remark that our sample complexity is independent of $\frac{R}{r}$.

The runtime of Theorem 1 is nearly-linear when $n \gtrsim \frac{1}{\alpha^2}$ (e.g., if $\sqrt{d} \cdot \frac{1}{\epsilon} \gtrsim \frac{1}{\alpha}$), but more generally it does incur an additive overhead of $\frac{d}{\alpha^2}$. This overhead matches the fastest non-private first-order method for approximating (1) to α -multiplicative error, due to [CLM⁺16]. We note that [CLM⁺16] also gave a second-order interior-point method, that non-privately solves (1) in time $\tilde{O}(nd)$, i.e., with polylogarithmic dependence on $\frac{1}{\alpha}$. We leave removing this additive runtime term in the DP setting, or proving this is impossible in concrete query models, as a challenging question for future work.

Our algorithm follows a roadmap by [HSU24], who split their algorithm into two phases: an initial “warm start” phase that computes an $O(1)$ -multiplicative approximation of the geometric median, and a secondary “boosting” phase that uses iterative methods to improve the warm start to an α -multiplicative approximation. The warm start improves the domain size of the boosting phase to scale with the effective radius. However, both the warm start and the boosting phases of [HSU24] required superlinear $\approx n^2 d$ time. Our improvement to the warm start phase of the [HSU24] is quite simple, and may be of independent interest, so we provide a self-contained statement here.

Theorem 2 (informal, see Theorem 3). *Let $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{B}^d(R)$ for $R > 0$, $0 < r \leq r^{(0.9)}$, and $(\epsilon, \delta) \in [0, 1]^2$. There is an (ϵ, δ) -DP algorithm that returns $\hat{\mathbf{x}}$ such that with probability $\geq 1 - \delta$, $f_{\mathcal{D}}(\hat{\mathbf{x}}) = O(f_{\mathcal{D}}(\mathbf{x}_*))$, assuming $n \gtrsim \frac{\sqrt{d}}{\epsilon}$. The algorithm runs in time $\tilde{O}(nd)$.*

1.2 Our techniques

As discussed previously, our algorithm employs a similar framework as [HSU24]. It is convenient to further split the warm start phase of the algorithm into two parts: finding an estimate \hat{r} of the effective radius of \mathcal{D} , and finding an approximate centerpoint at distance $O(\hat{r})$ from the geometric median \mathbf{x}_* .

Radius estimation. Our radius estimation algorithm is almost identical to that in [HSU24], Section 2.1, which uses the sparse vector technique (cf. Lemma 2) to detect the first time an estimate \hat{r} is such that most points have $\geq \frac{3}{4}$ of \mathcal{D} at a distance of $\approx \hat{r}$. The estimate \hat{r} is geometrically updated over a grid of size $O(\log(\frac{R}{r}))$. Naïvely implemented, this strategy takes $\gtrsim n^2 d$ time due to the need for pairwise distance comparisons (cf. Appendix A); even if dimensionality reduction techniques are used, this step appears to require $\Omega(n^2)$ time. We make a simple observation that a random sample of $\approx \log(\frac{1}{\delta})$ points from \mathcal{D} is enough to determine whether a given point has $\gg \beta$ neighbors, or $\ll \gamma$, for appropriate (constant) quantile thresholds β, γ , which is enough to obtain an $\tilde{O}(nd)$ runtime.

Centerpoint estimation. Our centerpoint estimation step departs from [HSU24], Section 2.2, who analyzed a custom variant of DP gradient descent with geometrically-decaying step sizes. We make the simple observation that directly applying the FriendlyCore algorithm of [HSU24] yields the same result. However, the standard implementation of FriendlyCore again uses $\Omega(n^2)$ time to estimate weights for each data point. We show that FriendlyCore can be sped up to run in $\tilde{O}(nd)$ time (independently of $\frac{R}{r}$) using subsampled weights. Our privacy proof is subtle, and based on an argument (Lemma 1) that couples our subsampled algorithm to an idealized algorithm that never fails to be private. We use this to account for the privacy loss due to the failure of our subsampling, i.e., if the estimates are inaccurate. We note that the [HSU24] algorithm for this step already ran in nearly-linear $\approx nd \log(\frac{R}{r})$ time, so we obtain an asymptotic improvement only if $\frac{R}{r}$ is large.

Boosting. The most technically novel part of our algorithm is in the boosting phase, which takes as input a radius and centerpoint estimate from the previous steps, and outputs an α -multiplicative approximation to (1). Like [HSU24], we use iterative methods to implement this phase. However, a major bottleneck to a faster algorithm is the lack of a nearly-linear time DP solver for non-smooth empirical risk minimization (ERM) problems. Indeed, such $\tilde{O}(1)$ -pass optimizers are known only when the objective is convex and sufficiently smooth [FKT20], or $n \gtrsim d^2$ samples are taken [CJJ⁺23]. This is an issue, because while computing the geometric median (1) is a convex ERM problem, it is non-smooth, and multiplicative guarantees are possible even with $n \approx \sqrt{d}$ samples.

We give a custom analysis of DP-SGD, specifically catered to the (non-smooth) ERM objective (1). Our main contribution is a tighter sensitivity analysis of DP-SGD’s iterates, leveraging the structure of the geometric median. To motivate this observation, consider coupled algorithms with iterates \mathbf{z}, \mathbf{z}' , both taking gradient steps with respect to the subsampled function $\|\cdot - \mathbf{x}_i\|$ for some dataset element $\mathbf{x}_i \in \mathcal{D}$. A simple calculation (7) shows these gradients are unit vectors \mathbf{u}, \mathbf{u}' , in the directions of $\mathbf{z} - \mathbf{x}_i$ and $\mathbf{z}' - \mathbf{x}_i$ respectively. It is not hard to formalize (Lemma 11) that updating $\mathbf{z} \leftarrow \mathbf{z} - \eta \mathbf{u}$ and $\mathbf{z}' \leftarrow \mathbf{z}' - \eta \mathbf{u}'$ is always contractive, unless \mathbf{z}, \mathbf{z}' were both already very close to \mathbf{x}_i (and hence, each other) to begin with. We use this structural result to inductively control DP-SGD’s sensitivity, which lets us leverage a prior reduction from private optimization to stable optimization [FKT20].

Our result is the first we are aware of that obtains a nearly-linear runtime for DP-SGD on a structured non-smooth problem. We were inspired by [ALT24], who also gave faster runtimes for (smooth) DP optimization problems with outliers under further assumptions on the objective. We hope that our work motivates future DP optimization methods that harness problem structure for improved rates.

1.3 Related work

Differentially private convex optimization. DP convex optimization has been studied extensively for over a decade [CM08, KST12, BST14, KJ16, BFGT20, FKT20, BGN21, GLL22, GLL⁺23] and inspired the influential DP-SGD algorithm widely adopted in deep learning [ACG⁺16]. In the classic setting, where functions are assumed to be Lipschitz and defined over a convex domain of diameter R , optimal rates have been achieved with linear dependence on R [BFTGT19]. Recent years have seen significant advancements in the gradient complexity of DP stochastic convex optimization [FKT20, AFKT21, KLL21, ZTC22, CJJ⁺23, CCGT24]. Despite these efforts, a nearly-linear gradient complexity has only been established for sufficiently smooth functions [FKT20, ZTC22, CCGT24] and for non-smooth functions [CJJ⁺23] when $\sqrt{n} \gtrsim d$ is satisfied.

Differential privacy with average-case data. Adapting noise to the inherent properties of data, rather than catering to worst-case scenarios, is critical for making differential privacy practical in real-world applications. Several important approaches have emerged in this direction: smooth sensitivity frameworks [NRS07] that refine local sensitivity to make it private; instance optimality techniques [AD20] with tailored guarantees for specific datasets; methods with improved performance under distributional assumptions such as sub-Gaussian or heavy-tailed i.i.d. data [CWZ21, AL23, ALT24]; and data-dependent sensitivity computations that adapt during algorithm execution [ATMR21]. These approaches collectively represent the frontier in balancing privacy and utility beyond worst-case analyses. We view our work as another contribution towards this broader program.

2 Preliminaries

Throughout, vectors are denoted in lowercase boldface, and the all-zeroes and all-ones vectors in dimension d are respectively $\mathbf{0}_d$ and $\mathbf{1}_d$. We use $\|\cdot\|$ to denote the Euclidean (ℓ_2) norm of a vector. We use $[d]$ to denote $\{i \in \mathbb{N} \mid 1 \leq i \leq d\}$. We use $\mathbb{B}^d(\boldsymbol{\mu}, r) := \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x} - \boldsymbol{\mu}\| \leq r\}$ to denote the Euclidean ball of radius $r > 0$ around $\boldsymbol{\mu} \in \mathbb{R}^d$; when $\boldsymbol{\mu}$ is unspecified, then $\boldsymbol{\mu} = \mathbf{0}_d$ by default. For compact $\mathcal{K} \subseteq \mathbb{R}^d$, we use $\Pi_{\mathcal{K}}(\mathbf{x})$ to denote the Euclidean projection $\arg \min_{\mathbf{y} \in \mathcal{K}} \|\mathbf{x} - \mathbf{y}\|$.

We let $\mathbb{I}_{\mathcal{E}}$ denote the 0-1 indicator random variable of an event \mathcal{E} . For two densities μ, ν on the same sample space Ω and $\alpha > 1$, we define the α -Rényi divergence $D_{\alpha}(\mu \parallel \nu) := \frac{1}{\alpha-1} \log(\int (\frac{\mu(\omega)}{\nu(\omega)})^{\alpha} \nu(\omega) d\omega)$. We use $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_d)$ to denote the multivariate normal distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\sigma^2 \mathbf{I}_d$, where \mathbf{I}_d is the $d \times d$ identity. We let $\text{Laplace}(\lambda)$ be the Laplace distribution with scale parameter $\lambda \geq 0$, whose density is $\propto \exp(-\frac{|\cdot|}{\lambda})$. We let $\text{Unif}(S)$ denote the uniform distribution over a set S , and $\text{Bern}(p)$ denote the Bernoulli distribution taking on values $\{0, 1\}$ with mean $p \in [0, 1]$. We refer to a product distribution consisting of k i.i.d. copies of a base distribution \mathcal{D} by $\mathcal{D}^{\otimes k}$. We also use the bounded Laplace distribution with parameters $\lambda, \tau \geq 0$, denoted $\text{BoundedLaplace}(\lambda, \tau)$, i.e., the distribution of $X \sim \text{Laplace}(\lambda)$ conditioned on $|X| \leq \tau$.

Differential privacy. Let \mathcal{X} be some domain, and let $\mathcal{D} \in \mathcal{X}^n$ be a dataset consisting of n elements from \mathcal{X} . We say datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$ are *neighboring* if their symmetric difference has size 1, i.e., they differ in a single element. We use the following definition of differential privacy in this paper.

176 **Definition 1** (Differential privacy). Let $(\epsilon, \delta) \in [0, 1]^2$.⁴ We say that a randomized algorithm
 177 $\mathcal{A} : \mathcal{X}^n \rightarrow \Omega$ satisfies (ϵ, δ) -differential privacy (or is (ϵ, δ) -DP) if for all events $\mathcal{E} \subseteq \Omega$, and for all
 178 neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$, $\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{E}] \leq \exp(\epsilon) \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{E}] + \delta$.

179 DP algorithms satisfy *basic composition* (Theorem B.1, [DR14]), i.e., if $\mathcal{A}_1 : \mathcal{X}^n \rightarrow \Omega_1$ is (ϵ_1, δ_1) -
 180 DP and $\mathcal{A}_2 : \mathcal{X}^n \times \Omega_1 \rightarrow \Omega_2$ is (ϵ_2, δ_2) -DP, then their composition is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -DP. We next
 181 state the Gaussian mechanism. Recall that if $\mathbf{v} : \mathcal{X}^n \rightarrow \mathbb{R}^k$ is a vector-valued function of a dataset,
 182 we say \mathbf{v} has sensitivity Δ if for all neighboring $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$, $\|\mathbf{v}(\mathcal{D}) - \mathbf{v}(\mathcal{D}')\| \leq \Delta$.

183 **Fact 1** (Theorem A.1, [DR14]). Let $\mathbf{v} : \mathcal{X}^n \rightarrow \mathbb{R}^k$ have sensitivity Δ , and let $(\epsilon, \delta) \in [0, 1]^2$. Then,
 184 drawing a sample from $\mathcal{N}(\mathbf{v}(\mathcal{D}), \sigma^2 \mathbf{I}_k)$ is (ϵ, δ) -DP, for any $\sigma \geq \frac{2\Delta}{\epsilon} \cdot \sqrt{\log(\frac{2}{\delta})}$.

185 We also require the bounded Laplace mechanism, which is known to give the following guarantee.

186 **Fact 2** (Lemma 9, [ALT24]). Let $s : \mathcal{X}^n \rightarrow \mathbb{R}$ have sensitivity Δ , and let $(\epsilon, \delta) \in [0, 1]^2$. Then,
 187 drawing $\xi \sim \text{BoundedLaplace}(\frac{\Delta}{\epsilon}, \tau)$ and outputting $s(\mathcal{D}) + \xi$ is (ϵ, δ) -DP for any $\tau \geq \frac{\Delta}{\epsilon} \log(\frac{4}{\delta})$.

188 Fact 2 is proven in [ALT24] using a coupling argument, using the fact that $\text{BoundedLaplace}(\lambda)$ and
 189 $\text{Laplace}(\lambda)$ result in the same sample except with some probability. We appeal to this technique
 190 several times in Section 3, so we explicitly state it here for convenience (with a proof in Appendix B).

191 **Lemma 1.** For $(\epsilon, \delta) \in [0, 1]^2$, $\mathcal{A} : \mathcal{X}^n \rightarrow \Omega$ be an (ϵ, δ) -DP algorithm, and let $\bar{\mathcal{A}}$ be an algorithm
 192 such that on any input $\mathcal{D} \in \mathcal{X}^n$, we have that the total variation distance between $\mathcal{A}(\mathcal{D})$ and $\bar{\mathcal{A}}(\mathcal{D})$
 193 is at most δ' . Then, $\bar{\mathcal{A}}$ is an $(\epsilon, \delta + 4\delta')$ -DP algorithm.

194 We next recall the following well-known result on detecting the first large element in a stream.

195 **Lemma 2** (Theorems 3.23, 3.24, [DR14]). AboveThreshold (Algorithm 4) is $(\epsilon, 0)$ -DP. Moreover,
 196 for $\gamma \in (0, 1)$, let $\alpha = \frac{8\Delta \log(\frac{2T}{\gamma})}{\epsilon}$ and $\mathcal{D} \in \mathcal{X}^n$. AboveThreshold halts at time $k \in [T + 1]$ such that
 197 with probability $\geq 1 - \gamma$: $q_t(\mathcal{D}) \leq \tau + \alpha$ for all $t < k$, and $q_k(\mathcal{D}) \geq \tau - \alpha$ or $k = T + 1$.

198 **Geometric median.** In the rest of the paper, for $R > 0$, we fix a dataset $\mathcal{D} := \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{B}^d(R)$,
 199 i.e., with domain $\mathcal{X} := \mathbb{B}^d(R)$. Our goal is to approximate the *geometric median* of \mathcal{D} , i.e.,

$$\mathbf{x}_\star(\mathcal{D}) := \arg \min_{\mathbf{x} \in \mathbb{R}^d} f_{\mathcal{D}}(\mathbf{x}), \text{ where } f_{\mathcal{D}}(\mathbf{x}) := \frac{1}{n} \sum_{i \in [n]} \|\mathbf{x} - \mathbf{x}_i\| \quad (3)$$

200 is the average Euclidean distance to the dataset. Following e.g., [CLM⁺16, HSU24], we also define
 201 the quantile radii associated with our dataset \mathcal{D} centered at $\bar{\mathbf{x}} \in \mathbb{R}^d$ by

$$r^{(\tau)}(\mathcal{D}; \bar{\mathbf{x}}) := \arg \min_{r \geq 0} \left\{ \sum_{i \in [n]} \mathbb{I}_{\|\mathbf{x}_i - \bar{\mathbf{x}}\| \leq r} \geq \tau n \right\}, \text{ for all } \tau \in [0, 1]. \quad (4)$$

202 When $\bar{\mathbf{x}}$ is unspecified, by default $\bar{\mathbf{x}} = \mathbf{x}_\star(\mathcal{D})$. In our utility analysis we will often suppress the
 203 dependence on \mathcal{D} in \mathbf{x}_\star , $r^{(\tau)}$, etc., as the dataset of interest will not change. In the privacy analysis,
 204 we specify the dependence of these functions on the dataset explicitly when comparing algorithms
 205 run on neighboring datasets. Finally, we include two helper results from prior work.

206 **Lemma 3.** Let $\mathcal{D} := \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{R}^d$. Then, $f_{\mathcal{D}}(\mathbf{x}_\star) \geq (1 - \tau)r^{(\tau)}$ for all $\tau \in [0, 1]$.

207 **Lemma 4** (Lemma 24, [CLM⁺16]). Let $\mathcal{D} := \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{R}^d$ and let $S \subseteq [n]$ have $|S| < \frac{n}{2}$. Then,

$$\|\mathbf{x}_\star - \mathbf{x}\| \leq \left(\frac{2n - 2|S|}{n - 2|S|} \right) \max_{i \notin S} \|\mathbf{x}_i - \mathbf{x}\|, \text{ for all } \mathbf{x} \in \mathbb{R}^d.$$

⁴In principle, ϵ can be larger than 1. However, in this paper, sample complexities are unaffected up to constants for any $\epsilon \geq 1$ if we simply obtain $(1, \delta)$ -DP guarantees rather than (ϵ, δ) -DP guarantees, which are only stronger. Thus we state all results for $\epsilon \in [0, 1]$ for convenience, which simplifies some bounds.

3 Constant-Factor Approximation

In this section, we give our first result: a fast algorithm for computing an $O(1)$ -approximation to the geometric median. We first show how to estimate quantile radii up to constant factors in nearly-linear time using subsampled scores, improving Section 2.1 of [HSU24]. We then adapt a weighted variant of FriendlyCore to give a simple algorithm for approximate centerpoint computation, improving Section 2.2 of [HSU24] for large aspect ratios. Proofs from this section are deferred to Appendix C.

We first present our radius estimation algorithm.

Algorithm 1 FastRadius($\mathcal{D}, r, R, \epsilon, \delta$)

Input: $\mathcal{D} \in \mathbb{B}^d(R)^n$, $0 < r \leq R$, $(\epsilon, \delta) \in [0, 1]^2$

- 1: $(T, k, \tau) \leftarrow (\lceil \log_2(\frac{R}{r}) \rceil, 3 \log(\frac{4T}{\delta}), 0.775n)$
- 2: $\hat{\tau} \leftarrow \tau + \nu_{\text{thresh}}$ for $\nu_{\text{thresh}} \sim \text{Laplace}(\frac{6}{\epsilon})$
- 3: **for** $t \in [T]$ **do**
- 4: **for** $i \in [n]$ **do**
- 5: $S_t^{(i)} \leftarrow \text{Unif}([n])^{\otimes k}$
- 6: $N_t^{(i)} \leftarrow \frac{n}{k} \sum_{j \in S_t^{(i)}} \mathbb{I}_{\|\mathbf{x}_i - \mathbf{x}_j\| \leq r_t}$ for $r_t \leftarrow r \cdot 2^{t-1}$
- 7: **end for**
- 8: $q_t \leftarrow \frac{1}{n} \sum_{i \in [n]} N_t^{(i)}$
- 9: $\nu_t \sim \text{Laplace}(\frac{12}{\epsilon})$
- 10: **if** $q_t + \nu_t \geq \hat{\tau}$ **then**
- 11: **Return:** r_t
- 12: **end if**
- 13: **end for**
- 14: **Return:** R

Algorithm 1 is an instance of AboveThreshold with $\Delta = 3$, where the queries are given on Line 8. However, our queries have random sensitivities depending on the subsampling in Line 5. Nonetheless, Chernoff bounds control this sensitivity with high probability, which yields privacy via Lemma 2.

Lemma 5. *Algorithm 1 is (ϵ, δ) -DP.*

The proof of Lemma 5 is a simple combination of Fact 4, Lemma 1, and Lemma 2. Next, we state a utility bound for Algorithm 1, whose proof is essentially identical to the proof in [HSU24], after applying Chernoff bounds to ensure our estimated counts are accurate.

Lemma 6. *Algorithm 1 runs in time $O(nd \log(\frac{R}{r}) \log(\log(\frac{R}{r}) \frac{1}{\delta}))$. Moreover, if $r \leq 4r^{(0.9)}$ and $n \geq \frac{2400}{\epsilon} \log(\frac{4T}{\delta})$, with probability $\geq 1 - \delta$, Algorithm 1 outputs \hat{r} satisfying $\frac{1}{4}r^{(0.75)} \leq \hat{r} \leq 4r^{(0.9)}$.*

In summary, Lemmas 5 and 6 show that we can privately estimate \hat{r} satisfying $\frac{1}{4}r^{(0.75)} \leq \hat{r} \leq 4r^{(0.9)}$ in nearly-linear time. The upper bound implies (with Lemma 3) that $\hat{r} = O(f_{\mathcal{D}}(\mathbf{x}_*))$; on the other hand, the lower bound will be critically used in our centerpoint estimation procedure, Algorithm 2.

Algorithm 2 is a simplification of the FriendlyCore algorithm of [TCK⁺22], that outputs a noisy weighted average of the dataset, where the weights $\{p_i\}_{i \in [n]}$ linearly interpolate estimated scores $f_i \in [0.5k, 0.75k]$ into the range $[0, 1]$, sending $f_i \geq 0.75k$ to 1, and $f_i \leq 0.5k$ to 0. We first make some basic observations about the points that contribute positively to the weighted combination $\bar{\mathbf{x}}$.

Lemma 7. *Assume that $\hat{r} \geq r^{(0.75)}$ in the context of Algorithm 2. With probability $\geq 1 - \frac{\delta}{18}$, every $i \in [n]$ that is assigned $p_i > 0$ in Algorithm 2 satisfies $\|\mathbf{x}_i - \mathbf{x}_*\| \leq 3\hat{r}$, and $Z \geq 0.6n$.*

Lemma 7 constitutes most of our utility analysis, as all contributing points to $\hat{\mathbf{x}}$ are $O(\hat{r})$ away from \mathbf{x}_* , our final desired error. We next note that whenever the algorithm does not return on Line 10, all surviving points lie in a ball of diameter $O(\hat{r})$, under a high-probability event over our subsampling. Importantly, this holds independently of any assumption on \hat{r} (e.g., we do not require $\hat{r} \geq r^{(0.75)}$).

Lemma 8. *Suppose that it is the case that in the context of Algorithm 2, we have*

$$f_i^* \leq 0.45k \implies f_i \leq 0.5k, \text{ and } f_i^* \leq 0.55k \implies f_i \leq 0.6k, \quad (5)$$

Algorithm 2 FastCenter($\mathcal{D}, \hat{r}, \epsilon, \delta$)

Input: $\mathcal{D} \in \mathbb{B}^d(R)^n$, $\hat{r} \in \mathbb{R}_{>0}$, $(\epsilon, \delta) \in [0, 1]^2$

```
1:  $k \leftarrow 600 \log(\frac{18n}{\delta})$ 
2: for  $i \in [n]$  do
3:    $S_i \leftarrow \text{Unif}([n])^{\otimes k}$ 
4:    $f_i \leftarrow \sum_{j \in S^{(i)}} \mathbb{I}_{\|\mathbf{x}_i - \mathbf{x}_j\| \leq 2\hat{r}}$ 
5:    $p_i \leftarrow \min(\max(0, \frac{f_i - 0.5k}{0.25k}), 1)$ 
6: end for
7:  $Z \leftarrow \sum_{i \in [n]} p_i$ 
8:  $\xi \sim \text{BoundedLaplace}(\frac{24}{\epsilon}, \frac{24}{\epsilon} \log(\frac{24}{\delta}))$ 
9: if  $Z + \xi - \frac{24}{\epsilon} \log(\frac{24}{\delta}) \leq 0.55n$  then
10:   Return:  $\mathbf{0}_d$ 
11: end if
12:  $\bar{\mathbf{x}} \leftarrow \frac{1}{Z} \sum_{i \in [n]} p_i \mathbf{x}_i$ 
13:  $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}_d, \sigma^2 \mathbf{I}_d)$ , for  $\sigma \leftarrow \frac{1600\hat{r}}{n\epsilon} \sqrt{\log(\frac{12}{\delta})}$ 
14: Return:  $\bar{\mathbf{x}} + \boldsymbol{\xi}$ 
```

238 *for all $i \in [n]$. If $Z > 0.55n$, there exists some $\mathbf{x} \in \mathbb{R}^d$ such that $\{\mathbf{x}_i \mid p_i > 0\}_{i \in [n]} \subseteq \mathbb{B}(\mathbf{x}, 4\hat{r})$.*
239 *Moreover, the event (5) occurs with probability $\geq 1 - \frac{\delta}{9}$.*

240 We are now ready to prove a privacy bound on Algorithm 2. Our proof follows the original privacy
241 analysis of FriendlyCore [TCK⁺22], but accounts for the failure of the “accurate subsampling” event
242 (5) by coupling to an algorithm that always ensures (5), and then applying Lemma 1.

243 **Lemma 9.** *If $n \geq 20$, Algorithm 2 is (ϵ, δ) -DP.*

244 Combining our developments gives our constant-factor approximation to the geometric median.

245 **Theorem 3.** *Let $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{B}^d(R)$ for $R > 0$, let $0 < r \leq 4r^{(0.9)}(\mathcal{D})$, and let $(\epsilon, \delta) \in [0, 1]^2$.
246 Suppose that $n \geq C \cdot (\sqrt{d} \cdot \frac{\log(\frac{1}{\delta})}{\epsilon} + \frac{\log(\frac{\log(R/r)}{\delta})}{\epsilon})$, for a sufficiently large constant C . There is an
247 (ϵ, δ) -DP algorithm that returns $(\hat{\mathbf{x}}, \hat{r})$ such that with probability $\geq 1 - \delta$, following notation (3),*

$$f_{\mathcal{D}}(\hat{\mathbf{x}}) \leq (40C' + 1)f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})), \quad \hat{r} \leq 4r^{(0.9)}, \quad (6)$$

248 *for a universal constant C' . Moreover, $\|\mathbf{x}_*(\mathcal{D}) - \hat{\mathbf{x}}\| \leq C'\hat{r}$. The algorithm runs in time*

$$O\left(nd \log\left(\frac{R}{r}\right) \log\left(\frac{n \log(\frac{R}{r})}{\delta}\right)\right).$$

249 We remark that Theorem 3 actually comes with the slightly stronger guarantee that we obtain the
250 optimal value for the geometric median objective $f_{\mathcal{D}}$, up to an additive error scaling as $O(r^{(0.9)})$.
251 In general, while $r^{(0.9)} = O(f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})))$ always holds (Lemma 3), it is possible that $r^{(0.9)} \ll$
252 $f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D}))$ if a small fraction of outliers contributes significantly to $f_{\mathcal{D}}$. We also note that for
253 datasets where we have more a priori information on the number of outliers we expect to see, we can
254 adjust the quantile 0.9 in Theorem 3 to be any quantile > 0.5 by appropriately adjusting constants.

255 4 Boosting Approximations via Stable DP-SGD

256 In this section, we give a DP algorithm that efficiently minimizes the geometric median objective
257 (3) over a domain $\mathbb{B}^d(\bar{\mathbf{x}}, \hat{r})$, given a dataset $\mathcal{D} := \{\mathbf{x}_i\}_{i \in [n]}$. In our final application to the geometric
258 median problem, the optimization domain (i.e., the parameters $\bar{\mathbf{x}} \in \mathbb{R}^d$ and $\hat{r} \in \mathbb{R}_{\geq 0}$) will be
259 privately estimated using Theorem 3, such that with high probability $\hat{r} = O(f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})))$ and
260 $\|\mathbf{x} - \mathbf{x}_*(\mathcal{D})\| \leq \hat{r}$. In the meantime, we treat the domain $\mathbb{B}^d(\bar{\mathbf{x}}, \hat{r})$ as a public input here.

Our strategy is to use a localization framework given by [FKT20], which gives a query-efficient reduction from private DP-SGD to stable DP-SGD executed in phases. Observe that outputting⁵

$$\frac{\mathbf{z} - \mathbf{x}_i}{\|\mathbf{z} - \mathbf{x}_i\|} = \nabla \|\cdot - \mathbf{x}_i\|(\mathbf{z}) \quad (7)$$

for $i \sim \text{Unif}([n])$ is unbiased for a subgradient of $f_{\mathcal{D}}(\mathbf{z})$, leading to the following Algorithm 3.

Algorithm 3 StableDPSGD($\mathcal{D}, \bar{\mathbf{x}}, \hat{r}, \rho, \delta, \eta, T$)

Input: $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{R}^d$, $(\bar{\mathbf{x}}, \hat{r}) \in \mathbb{R}^d \times \mathbb{R}_{\geq 0}$, $\rho > 0$, $\delta \in (0, 1)$, $\eta > 0$, $T = 2^K - 1 \geq n$ for $K \in \mathbb{N}$

```

1:  $m \leftarrow 3(\frac{T}{n} + \log(\frac{8}{\delta}))$ 
2: for  $k \in [K]$  do
3:    $(T^{(k)}, \eta^{(k)}, \sigma^{(k)}) \leftarrow (2^{-k}(T+1), 4^{-k}\eta, 3^{-k}\frac{(2m+1)\eta}{\sqrt{\rho}})$ 
4:   if  $k = 1$  then
5:      $(\mathbf{z}_0^{(k)}, \mathcal{K}^{(k)}) \leftarrow (\bar{\mathbf{x}}, \mathbb{B}^d(\bar{\mathbf{x}}, \hat{r}))$ 
6:   else
7:      $(\mathbf{z}_0^{(k)}, \mathcal{K}^{(k)}) \leftarrow (\hat{\mathbf{x}}^{(k-1)}, \mathbb{B}^d(\hat{\mathbf{x}}^{(k-1)}, 2\sigma^{(k)}\sqrt{d\log(\frac{4K}{\delta})}))$ 
8:   end if
9:   for  $0 \leq t < T^{(k)}$  do
10:     $\mathbf{g}_t^{(k)} \leftarrow \frac{\mathbf{z}_t^{(k)} - \mathbf{x}_i}{\|\mathbf{z}_t^{(k)} - \mathbf{x}_i\|}$  for  $i \sim \text{Unif}([n])$ 
11:     $\mathbf{z}_{t+1}^{(k)} \leftarrow \Pi_{\mathcal{K}^{(k)}}(\mathbf{z}_t^{(k)} - \eta^{(k)}\mathbf{g}_t^{(k)})$ 
12:   end for
13:    $\bar{\mathbf{x}}^{(k)} \leftarrow \frac{1}{T^{(k)}} \sum_{0 \leq t < T^{(k)}} \mathbf{z}_t^{(k)}$ 
14:    $\xi^{(k)} \sim \mathcal{N}(\mathbf{0}_d, (\sigma^{(k)})^2 \mathbf{I}_d)$ 
15:    $\hat{\mathbf{x}}^{(k)} \leftarrow \bar{\mathbf{x}}^{(k)} + \xi^{(k)}$ 
16: end for
17: Return:  $\hat{\mathbf{x}}^{(K)}$ 

```

Remark 1. Several steps in Algorithm 3 are used only in the worst-case utility proof, and do not affect privacy. Practical optimizations can be made while preserving privacy guarantees, e.g., removing projections onto the changing domains $\mathcal{K}^{(k)}$ rather than $\mathcal{K}^{(1)}$, which is not used in the privacy proof.

One optimization we found useful in our experiments (described in Section 5) is replacing the random sampling on Line 10 with deterministic passes through the dataset in a fixed order. By doing so, we know the total number of accesses of any single element is $\leq m := \lceil \frac{T}{n} \rceil$ (rather than the high-probability estimate in Lemma 10 for the randomized variant in Algorithm 3). This lets us tighten the noise level $\sigma^{(k)}$ by a fairly significant constant factor, resulting in improved empirical performance.

We provide a privacy bound on Algorithm 3 in Appendix D.1, and a utility bound in Appendix D.2. Our utility proof is fairly standard, and makes small modifications to the original localization analysis in [FKT20] to obtain a high-probability error guarantee using martingale arguments.

Our main technical novelty lies in our privacy proof, which is unconventional as it requires showing stability of DP-SGD on a non-smooth function (3). This contrasts with [FKT20], which assumed smoothness to achieve fast runtimes. Our main observation to this end is in Lemma 11 and Corollary 1, which show Line 11 implemented with a shared \mathbf{x}_i , but different starting $\mathbf{z}_t^{(k)}, (\mathbf{z}_t^{(k)})'$, is contractive, unless $\mathbf{z}_t^{(k)}, (\mathbf{z}_t^{(k)})'$ were close. This yields a refined bound on the stability of each phase, allowing us to use the [FKT20] reduction for DP. Combining these pieces yields our following main result.

Theorem 4. Let $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{B}^d(R)$ for $R > 0$, let $0 < r \leq 4r^{(0.9)}(\mathcal{D})$, and let $(\alpha, \epsilon, \delta) \in [0, 1]^3$. Suppose that $n \geq C \cdot (\frac{\sqrt{d}}{\alpha\epsilon} \log^{2.5}(\frac{\log(\frac{d}{\alpha\delta\epsilon})}{\delta}))$, for a sufficiently large constant C . There is an (ϵ, δ) -DP algorithm that returns $\hat{\mathbf{x}}$ such that with probability $\geq 1 - \delta$, following notation (3), $f_{\mathcal{D}}(\hat{\mathbf{x}}) \leq (1 + \alpha)f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D}))$. The algorithm runs in time

$$O\left(nd \log\left(\frac{R}{r}\right) \log\left(\frac{d \log(\frac{R}{r})}{\alpha\delta\epsilon}\right) + \frac{d}{\alpha^2} \log\left(\frac{\log(\frac{d}{\alpha\delta\epsilon})}{\delta}\right)\right).$$

⁵By default, if $\mathbf{x} = \mathbf{x}_i$, we let (7) evaluate to $\mathbf{0}_d$, which is a valid subgradient by first-order optimality.

5 Experiments

We present empirical evidence supporting the efficacy of our techniques.⁶ We conduct experiments on Algorithm 1 (the radius estimation step of Section 3) and Algorithm 3, to evaluate how subsampled estimates and DP-SGD respectively improve the performance of our algorithm.

We do not present experiments on Algorithm 2, as our current analysis results in loose constants, which in our preliminary experimentation significantly impacted the performance of Algorithm 2 in practice. We leave optimizing the performance of this step as an important step for future work. Our Algorithm 2 and Section 2.2 of [HSU24] had essentially the same theoretical guarantees, so we find it in line with the conceptual contribution of this work to evaluate the other two components.

We use two types of synthetic datasets with outliers, described in Appendix E: GaussianCluster (used in [HSU24] as well), and HeavyTailed (a multivariate Student’s t distribution).

Subsampling. We performed experiments to show the benefit of subsampling in FastRadius (Algorithm 1) over RadiusFinder (Algorithm 1 from [HSU24]) for differentially private estimation of the quantile radius, which is the first step in differentially private estimation of the geometric median. Across a variety of settings of n , d , and aspect ratios in our GaussianCluster experiment and degrees of freedom in our HeavyTailed, both FastRadius and RadiusFinder consistently resulted in estimates $\approx 1.2\text{-}3\times$ the true quantile radius, with FastRadius’s estimates consistently about 10-20% worse. However, the average wall-clock time for our algorithm was roughly $30\times$ faster, even at $n = 1000$, $d = 10$. We present plots and experimental results for this evaluation in Appendix E.

Boosting. We also evaluate the performance of our boosting algorithm in Section 4, compared to the baseline method from [HSU24]. We in fact evaluate three methods: (1) the baseline method, DPGD (vanilla DP gradient descent), as described in Algorithms 3 and 6, [HSU24], but with an optimized step size selected through ablation studies (cf. Appendix E); (2) StableDPSGD, i.e., our Algorithm 3 implemented as written, and (3) FixedOrderDPSGD, a variant of our Algorithm 3 with the last optimization described in Remark 1. We calibrated our noise level in FixedOrderDPSGD to ensure a fixed level of CDP via a group privacy argument, where we use that each dataset element is deterministically accessed at most $m = \lceil \frac{T}{n} \rceil$ times in FixedOrderDPSGD with T iterations.

We present here a representative experiment, deferring a more comprehensive set of experiments to Appendix E. Here, we vary n only, fixing all other parameters in a GaussianCluster dataset in \mathbb{R}^{50} . We report the performance of all three methods, plotting the passes over the dataset used by the excess error. Our error metric is $\frac{1}{\hat{r}} \cdot (f_{\mathcal{D}}(\hat{\mathbf{x}}) - f_{\mathcal{D}}(\hat{\mathbf{x}}^*))$, i.e., a multiple of the “effective radius” used. This is a more reflective performance metric than the corresponding multiple of $f_{\mathcal{D}}$, as our algorithms achieve this bound (cf. discussion after Theorem 3), and $f_{\mathcal{D}} \geq \hat{r}$ for our datasets due to outliers.

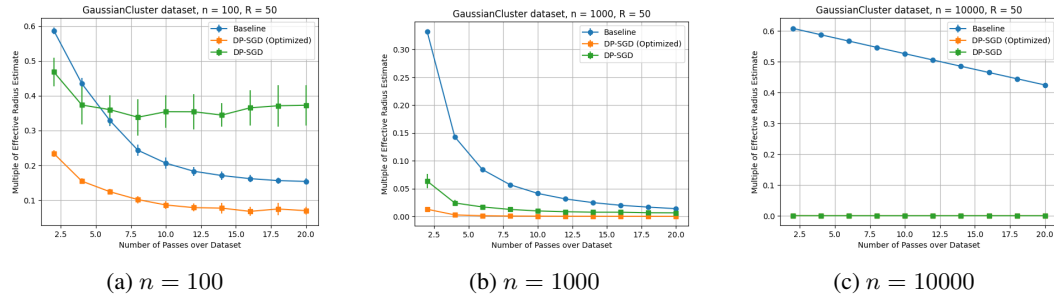


Figure 1: Comparison of DPGD, StableDPSGD, and FixedOrderDPSGD across different datasets over \mathbb{R}^{50} . All plots are averaged across 20 trials and standard deviations are reported as error bars.

Throughout our experiments, we observed consistent trends to Figure 3, where our (optimized) algorithm FixedOrderDPSGD outperformed the baseline by an amount depending on n (with better performance for larger n), and the unoptimized StableDPSGD achieved worse, but competitive, results. We remark that one limitation of our evaluation is that DPGD gradient computations can be parallelized, which leads to wall-clock time savings; we discuss this point further in Appendix E.

⁶Our subsampling experiments were performed on a single Google Colab CPU, and our boosting experiments were performed on a personal Apple M4 with 16GB RAM.

References

- [ACG⁺16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.
- [AD20] Hilal Asi and John C Duchi. Instance-optimality in differential privacy via approximate inverse sensitivity mechanisms. *Advances in neural information processing systems*, 33:14106–14117, 2020.
- [ADV⁺25] Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025*, pages 2005–2039. SIAM, 2025.
- [AFKT21] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in ℓ_1 geometry. In *International Conference on Machine Learning*, pages 393–403. PMLR, 2021.
- [AL22] Hassan Ashtiani and Christopher Liaw. Private and polynomial time algorithms for learning gaussians and beyond. In *Conference on Learning Theory*, pages 1075–1076. PMLR, 2022.
- [AL23] Hilal Asi and Daogao Liu. User-level differentially private stochastic convex optimization: Efficient algorithms with optimal rates. *arXiv preprint arXiv:2311.03797*, 2023.
- [ALT24] Hilal Asi, Daogao Liu, and Kevin Tian. Private stochastic convex optimization with heavy tails: Near-optimality from simple reductions. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024*, 2024.
- [ATMR21] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.
- [BD14] Rina Foygel Barber and John C Duchi. Privacy and statistical risk: Formalisms and minimax bounds. *arXiv preprint arXiv:1412.4451*, 2014.
- [BDKU20] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan R. Ullman. Coinpress: Practical private mean and covariance estimation. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020.
- [BFGT20] Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33:4381–4391, 2020.
- [BFTGT19] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.
- [BGN21] Raef Bassily, Cristóbal Guzmán, and Anupama Nandi. Non-euclidean differentially private stochastic convex optimization. In *Conference on Learning Theory*, pages 474–499. PMLR, 2021.
- [BGS⁺21] Gavin Brown, Marco Gaboardi, Adam Smith, Jonathan Ullman, and Lydia Zakyntinou. Covariance-aware private mean estimation without private covariance estimation. *Advances in neural information processing systems*, 34:7950–7964, 2021.
- [BHS23] Gavin Brown, Samuel B. Hopkins, and Adam D. Smith. Fast, sample-efficient, affine-invariant private mean and covariance estimation for subgaussian distributions. In *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023*, volume 195 of *Proceedings of Machine Learning Research*, pages 5578–5579. PMLR, 2023.

[BKS^W19] Mark Bun, Gautam Kamath, Thomas Steinke, and Steven Z Wu. Private hypothesis selection. *Advances in Neural Information Processing Systems*, 32, 2019.

[BS16] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 635–658, 2016.

[BST14] Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 464–473. IEEE Computer Society, 2014.

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.

[CCGT24] Christopher A Choquette-Choo, Arun Ganesh, and Abhradeep Thakurta. Optimal rates for $o(1)$ -smooth dp-sco with a single epoch and large batches. *arXiv preprint arXiv:2406.02716*, 2024.

[CJJ⁺23] Yair Carmon, Arun Jambulapati, Yujia Jin, Yin Tat Lee, Daogao Liu, Aaron Sidford, and Kevin Tian. Resqueing parallel and private stochastic convex optimization. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023*, pages 2031–2058. IEEE, 2023.

[CKM⁺21] Edith Cohen, Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. Differentially-private clustering of easy instances. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 2049–2059. PMLR, 2021.

[CLM⁺16] Michael B Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 9–21, 2016.

[CM08] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. *Advances in neural information processing systems*, 21, 2008.

[CWZ21] T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *The Annals of Statistics*, 49(5):2825–2850, 2021.

[DFM⁺20] Wenxin Du, Canyon Foot, Monica Moniot, Andrew Bray, and Adam Groce. Differentially private confidence intervals. *arXiv preprint arXiv:2001.02285*, 2020.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[FKT20] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 439–449. ACM, 2020.

[GLL22] Sivakanth Gopi, Yin Tat Lee, and Daogao Liu. Private convex optimization via exponential mechanism. In *Conference on Learning Theory*, pages 1948–1989. PMLR, 2022.

[GLL⁺23] Sivakanth Gopi, Yin Tat Lee, Daogao Liu, Ruoqi Shen, and Kevin Tian. Private convex optimization in general norms. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5068–5089. SIAM, 2023.

[HSU24] Mahdi Haghifam, Thomas Steinke, and Jonathan Ullman. Private geometric median. *Advances in Neural Information Processing Systems*, 37:46254–46293, 2024.

[KDH23] Rohith Kudithipudi, John Duchi, and Saminul Haque. A pretty fast algorithm for adaptive private mean estimation. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 2511–2551. PMLR, 2023.

- [KJ16] Shiva Prasad Kasiviswanathan and Hongxia Jin. Efficient private empirical risk minimization for high-dimensional learning. In *International Conference on Machine Learning*, pages 488–497. PMLR, 2016.
- [KLL21] Janardhan Kulkarni, Yin Tat Lee, and Daogao Liu. Private non-smooth erm and sco in subquadratic steps. *Advances in Neural Information Processing Systems*, 34:4053–4064, 2021.
- [KLSU19] Gautam Kamath, Jerry Li, Vikrant Singhal, and Jonathan R. Ullman. Privately learning high-dimensional distributions. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019*, volume 99 of *Proceedings of Machine Learning Research*, pages 1853–1902. PMLR, 2019.
- [KST12] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1. JMLR Workshop and Conference Proceedings, 2012.
- [KV17] Vishesh Karwa and Salil Vadhan. Finite sample differentially private confidence intervals. *arXiv preprint arXiv:1711.03908*, 2017.
- [LKJO22] Xiyang Liu, Weihao Kong, Prateek Jain, and Sewoong Oh. DP-PCA: statistically optimal and differentially private PCA. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, 2022.
- [LR91] Hendrik P. Lopuhaa and Peter J. Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, 19(1):229–248, 1991.
- [Mir17] Ilya Mironov. Rényi differential privacy. In *30th IEEE Computer Security Foundations Symposium, CSF 2017*, pages 263–275. IEEE Computer Society, 2017.
- [NRS07] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 75–84. ACM, 2007.
- [NSV16] Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Locating a small cluster privately. In Tova Milo and Wang-Chiew Tan, editors, *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016*, pages 413–427. ACM, 2016.
- [Roc76] R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control and Optimization*, 14(5):877–898, 1976.
- [TCK⁺22] Eliad Tsfadia, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. Friend-lycore: Practical differentially private aggregation. In *International Conference on Machine Learning*, pages 21828–21863. PMLR, 2022.
- [Web29] Alfred Weber. *Theory of the Location of Industries*. University of Chicago Press, 1929.
- [ZTC22] Qinzi Zhang, Hoang Tran, and Ashok Cutkosky. Differentially private online-to-batch for smooth losses. In *NeurIPS*, 2022.

A Discussion of [HSU24] runtime

We give a brief discussion of the runtime of the [HSU24] algorithm in this section, as the claimed runtimes in the original paper do not match those described in Section 1. As the [HSU24] algorithm is split into three parts (the first two of which correspond to the warm start phase and the third of which corresponds to the boosting phase), we discuss the runtime of each part separately.

Radius estimation. The radius estimation component of [HSU24] corresponds to Algorithm 1 and Section 2.1 of the paper. The authors claim a runtime of $O(n^2 \log(\frac{R}{r}))$ for this step due to need to do pairwise distance comparisons on a dataset of size n , for $O(\log(\frac{R}{r}))$ times in total. However, we believe the runtime should be $O(n^2 d \log(\frac{R}{r}))$, accounting for the $O(d)$ cost of each comparison.

Centerpoint estimation. The centerpoint estimation component of [HSU24] corresponds to Algorithm 2 and Section 2.2 of the paper. We agree with the authors that this algorithm runs in time $O(nd \log(\frac{R}{r}))$, and in particular, this step does not dominate any runtime asymptotically.

Boosting. The boosting component of [HSU24] corresponds to Algorithms 3 and 4 and Section 3 of the paper. The authors provide two different boosting procedures (based on gradient descent and cutting-plane methods) and state their runtimes as $\tilde{O}(n^2 d)$ and $\tilde{O}(nd^2 + d^{2+\omega})$, where $\omega < 2.372$ is the current matrix multiplication exponent [ADV⁺25]. We agree with the latter runtime analysis; however, we believe there is an additive $\tilde{O}(n^3 \epsilon^2)$ term in the runtime of gradient descent. This follows by noting that Algorithm 3 uses $\approx \frac{n^2 \epsilon^2}{d}$ iterations, each of which takes $O(nd)$ time to implement.

B Deferred material from Section 2

In this appendix, we provide missing material from Section 2. We begin by proving Lemmas 1 and 3.

Lemma 1. *For $(\epsilon, \delta) \in [0, 1]^2$, $\mathcal{A} : \mathcal{X}^n \rightarrow \Omega$ be an (ϵ, δ) -DP algorithm, and let $\bar{\mathcal{A}}$ be an algorithm such that on any input $\mathcal{D} \in \mathcal{X}^n$, we have that the total variation distance between $\mathcal{A}(\mathcal{D})$ and $\bar{\mathcal{A}}(\mathcal{D})$ is at most δ' . Then, $\bar{\mathcal{A}}$ is an $(\epsilon, \delta + 4\delta')$ -DP algorithm.*

Proof. For neighboring datasets $\mathcal{D}, \mathcal{D}'$, and any event $\mathcal{E} \in \Omega$, we have that

$$\begin{aligned} \Pr[\bar{\mathcal{A}}(\mathcal{D}) \in \mathcal{E}] &\leq \Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{E}] + \delta' \\ &\leq \exp(\epsilon) \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{E}] + \delta + \delta' \\ &\leq \exp(\epsilon) \Pr[\bar{\mathcal{A}}(\mathcal{D}') \in \mathcal{E}] + \delta + 4\delta'. \end{aligned}$$

The first and last lines used the assumption between \mathcal{A} and $\bar{\mathcal{A}}$, and the second used that \mathcal{A} is DP. \square

Lemma 3. *Let $\mathcal{D} := \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{R}^d$. Then, $f_{\mathcal{D}}(\mathbf{x}_\star) \geq (1 - \tau)r^{(\tau)}$ for all $\tau \in [0, 1]$.*

Proof. This is immediate from the definition of $f_{\mathcal{D}}$ and nonnegativity of each summand $\|\cdot - \mathbf{x}_i\|$. \square

Next we provide pseudocode for Algorithm 4, analyzed in Lemma 2.

Algorithm 4 AboveThreshold($\mathcal{D}, \{q_t\}_{t \in [T]}, \tau, \epsilon$)

Input: $\mathcal{D} \in \mathcal{X}^n$, sensitivity- Δ queries $\{q_t : \mathcal{X}^n \rightarrow \mathbb{R}\}_{t \in [T]}$, $\tau \in \mathbb{R}$, $\epsilon > 0$

```

1:  $\hat{\tau} \leftarrow \tau + \nu_{\text{thresh}}$  for  $\nu_{\text{thresh}} \sim \text{Laplace}(\frac{2\Delta}{\epsilon})$ 
2: for  $t \in [T]$  do
3:    $\nu_t \sim \text{Laplace}(\frac{4\Delta}{\epsilon})$ 
4:   if  $q_t(\mathcal{D}) + \nu_t \geq \hat{\tau}$  then
5:     Output:  $a_t \leftarrow \top$ 
6:   Halt
7: else
8:   Output:  $a_t \leftarrow \perp$ 
9: end if
10: end for
```

Finally, our developments in Section 4 use the notions of Rényi DP (RDP) and central DP (CDP). We provide a self-contained summary of the definitions and properties satisfied by RDP and CDP here, but refer the reader to [BS16, Mir17] for a more detailed overview.

Definition 2 (RDP and CDP). *Let $\alpha \geq 1$, $\rho \geq 0$. We say that a randomized algorithm $\mathcal{A} : \mathcal{X}^n \rightarrow \Omega$ satisfies (α, ρ) -Rényi differential privacy (or, is (α, ρ) -RDP) if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathcal{X}^n$,*

$$D_\alpha(\mathcal{A}(\mathcal{D}) \parallel \mathcal{A}(\mathcal{D}')) \leq \alpha \rho.$$

If this holds for all $\alpha \geq 1$, we say \mathcal{A} satisfies ρ -central differential privacy (or, is ρ -CDP).

Fact 3 ([Mir17]). *RDP and CDP satisfy the following properties.*

1. (Composition): *If $\mathcal{A}_1 : \mathcal{X}^n \rightarrow \Omega_1$ is (α, ρ_1) -RDP and $\mathcal{A}_2 : \mathcal{X}^n \times \Omega_1 \rightarrow \Omega_2$ is (α, ρ_2) for any fixed choice of input from Ω_1 , the composition of \mathcal{A}_2 and \mathcal{A}_1 is $(\alpha, \rho_1 + \rho_2)$ -RDP.*
2. (RDP to DP): *If \mathcal{A} is (α, ρ) -RDP, it is also $(\alpha\rho + \frac{1}{\alpha-1} \log \frac{1}{\delta}, \delta)$ -DP for all $\delta \in (0, 1)$.*
3. (Gaussian mechanism): *Let $\mathbf{v} : \mathcal{X}^n \rightarrow \mathbb{R}^k$ have sensitivity Δ . Then for any $\sigma > 0$, drawing a sample from $\mathcal{N}(\mathbf{v}(\mathcal{D}), \sigma^2 \mathbf{I}_k)$ is $\frac{\Delta^2}{2\sigma^2}$ -CDP.*

Finally, we require the following standard bound on binomial concentration.

Fact 4 (Chernoff bound). *For all $i \in [n]$, let $X_i \sim \text{Bern}(p_i)$ for some $p_i \in [0, 1]$, and let $\mu := \sum_{i \in [n]} p_i$ and $\hat{\mu} := \sum_{i \in [n]} X_i$. Then, $\Pr[\hat{\mu} > (1 + \epsilon)\mu] \leq \exp(-\frac{\epsilon^2 \mu}{2 + \epsilon})$ for all $\epsilon \geq 0$, and $\Pr[\hat{\mu} < (1 - \epsilon)\mu] \leq \exp(-\frac{\epsilon^2 \mu}{2})$ for all $\epsilon \in (0, 1)$.*

C Deferred material from Section 3

In this appendix, we provide missing material from Section 3.

Lemma 5. *Algorithm 1 is (ϵ, δ) -DP.*

Proof. Fix neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathbb{B}^d(R)^n$, and assume without loss that they differ in the n^{th} entry. Observe that FastRadius (independently) uses randomness in two places: the random subsets in Line 5, and the Laplace noise added as in the original AboveThreshold algorithm in Lines 2 and 9.

We next claim that in any iteration $t \in [T]$, as long as the number of copies of the index n occurring in $\bigcup_{i \in [n-1]} S_t^{(i)}$ is at most $2k$, then the sensitivity of the query q_t is at most 3. To see this, denoting by q_t, q'_t the random queries when Algorithm 1 is run on $\mathcal{D}, \mathcal{D}'$ respectively, and similarly defining $\{N_t^{(i)}, (N_t^{(i)})'\}_{i \in [n]}$, we observe that the sensitivity is controlled as follows:

$$\begin{aligned} q_t - q'_t &\leq \frac{1}{n} \sum_{i \in [n-1]} \left(N_t^{(i)} - (N_t^{(i)})' \right) + \frac{n}{n} \\ &\leq \frac{1}{n} \cdot \frac{n}{k} \cdot \left(\text{number of copies of } n \text{ occurring in } \bigcup_{i \in [n-1]} S_t^{(i)} \right) + 1 \leq 3. \end{aligned}$$

The first line holds because the n^{th} (neighboring) point has $N_t^{(n)} \leq n$ and $(N_t^{(n)})' \geq 0$; the second is because every $\mathbb{I}_{\|\mathbf{x}_i - \mathbf{x}_j\| \leq r_t}$ used in the computation of $N_t^{(i)}$ is coupled except when $j = n$ is sampled.

Now, let $\bar{\mathcal{A}}$ denote Algorithm 1, and let \mathcal{A} denote a variant that conditions on the randomly-sampled $\bigcup_{i \in [n-1]} S_t^{(i)}$ containing at most $2k$ copies of the index n , in all encountered iterations $t \in [T]$. By using Fact 4 (with $\mu \leftarrow k \cdot \frac{n-1}{n}$, $\epsilon \leftarrow 1$), due to our choice of k , $\bigcup_{i \in [n-1]} S_t^{(i)}$ contains at most $2k$ copies of n except with probability $\frac{\delta}{4T}$, so by a union bound, the total variation distance between $\bar{\mathcal{A}}$ and \mathcal{A} is at most $\frac{\delta}{4}$. Moreover, \mathcal{A} is $(\epsilon, 0)$ -DP by using Lemma 2. Thus, $\bar{\mathcal{A}}$ is (ϵ, δ) -DP using Lemma 1. \square

Lemma 6. *Algorithm 1 runs in time $O(nd \log(\frac{R}{r}) \log(\log(\frac{R}{r}) \frac{1}{\delta}))$. Moreover, if $r \leq 4r^{(0.9)}$ and $n \geq \frac{2400}{\epsilon} \log(\frac{4T}{\delta})$, with probability $\geq 1 - \delta$, Algorithm 1 outputs \hat{r} satisfying $\frac{1}{4}r^{(0.75)} \leq \hat{r} \leq 4r^{(0.9)}$.*

526 *Proof.* The first claim is immediate. To see the second, for all $t \in [T]$ denote the “ideal” query by:

$$q_t^* := \frac{1}{n} \sum_{i \in [n]} \sum_{j \in [n]} \mathbb{1}_{\|\mathbf{x}_i - \mathbf{x}_j\| \leq r_t},$$

527 and recall $\mathbb{E}q_t = q_t^*$. Our first claim is that with probability $\geq 1 - \frac{\delta}{2}$, the following guarantees hold
528 for all iterations $t \in [T]$ that Algorithm 1 completes:

$$q_t^* > 0.8n \implies q_t > 0.79n, \text{ and } q_t^* < 0.75n \implies q_t < 0.76n. \quad (8)$$

529 To see the first part of (8), we can view nq_t as a random sum of Bernoulli variables with mean
530 $nq_t^* > 0.8n^2 \geq 20000 \log(\frac{\delta}{4T})$, so Fact 4 with $\epsilon \leftarrow \frac{1}{80}$ yields the claim in iteration t with probability
531 $\geq 1 - \frac{\delta}{2T}$. Similarly, the second part of (8) follows by using Fact 4 with $\mu < 0.7n^2$ and $(1 + \epsilon)\mu \leftarrow$
532 $0.71n^2$, because

$$\exp\left(-\frac{\epsilon^2 \mu}{2 + \epsilon}\right) \leq \exp\left(-\frac{\epsilon^2 \mu}{151\epsilon}\right) = \exp\left(-\frac{\epsilon \mu}{151}\right) \leq \exp\left(-\frac{n^2}{15100}\right) \leq \frac{\delta}{2T} \quad (9)$$

533 for the relevant range of n and $\epsilon \geq \frac{1}{75}$, $\epsilon \mu \geq \frac{n^2}{100}$. We thus obtain (8) after a union bound over all
534 $t \in [T]$.

535 Now, suppose that $t \in [T]$ is the first index where $q_t + \nu_t \geq \hat{r}$, so that $\hat{r} = r_t$ and $\frac{1}{2}\hat{r} = r_{t-1}$, where
536 we let $r_0 := \frac{r}{2}$. If no such query passes, then we set $t = T + 1$ by default. Then by the utility
537 guarantees of Lemma 6, we have that with probability $\geq 1 - \frac{\delta}{2}$,

$$q_t \geq 0.76n, \quad q_{t-1} \leq 0.79n,$$

538 since $\frac{n}{100} \geq \alpha = \frac{24}{\epsilon} \log(\frac{4T}{\delta})$. By taking the contrapositive of (8), we can conclude that with
539 probability $\geq 1 - \delta$, we have $q_t^* \geq 0.75n$ and $q_{t-1}^* \leq 0.8n$. Condition on this event for the rest of
540 the proof.

541 Because $q_t^* \geq 0.75n$, there is clearly some $\mathbf{x}_i \in \mathcal{D}$ such that $|\mathcal{D} \cap \mathbb{B}^d(\mathbf{x}_i, \hat{r})| \geq 0.75n$, as this is the
542 average number of dataset elements in a radius- \hat{r} ball centered at a random $\mathbf{x} \in \mathcal{D}$. Now applying
543 Lemma 4 with S set to the indices of $\mathcal{D} \setminus \mathbb{B}^d(\mathbf{x}_i, \hat{r})$, so that $|S| \leq 0.25n$, gives

$$\|\mathbf{x}_* - \mathbf{x}_i\| \leq 3\hat{r} \implies \|\mathbf{x}_* - \mathbf{x}_j\| \leq 4\hat{r} \text{ for all } \mathbf{x}_j \in \mathcal{D} \cap \mathbb{B}^d(\mathbf{x}_i, \hat{r}).$$

544 This implies $4\hat{r} \geq r^{(0.75)}$ as claimed. Further, because $q_{t-1}^* \leq 0.8n$, we claim $\frac{\hat{r}}{2} > 2r^{(0.9)}$ cannot
545 hold. Assume for contradiction that this happened, and let $S := \mathcal{D} \cap \mathbb{B}^d(\mathbf{x}_*, r^{(0.9)})$. By the triangle
546 inequality, for all of the $0.9n$ choices of $\mathbf{x}_i \in S$, we have that

$$\sum_{j \in [n]} \mathbb{1}_{\|\mathbf{x}_i - \mathbf{x}_j\| \leq \frac{\hat{r}}{2}} \geq 0.9n,$$

547 which implies that $q_{t-1}^* \geq 0.81n$, a contradiction. Thus, we obtain $\hat{r} \leq 4r^{(0.9)}$ as well.

548 We remark that all of this logic handles the case where $2 \leq t \leq T$ is the iteration where Algorithm 1
549 returns. However, it is straightforward to check that the conclusion holds when $t = 1$ (i.e., $\hat{r} = r$)
550 because we assumed $r \leq 4r^{(0.9)}$, and the lower bound logic on \hat{r} is the same as before. Similarly, if
551 $\hat{r} = R$, then the upper bound logic on \hat{r} is the same as before, and $2R \geq r^{(1)} \geq r^{(0.75)} \geq \frac{2}{4}r^{(0.75)}$ is
552 clear. \square

553 **Lemma 7.** Assume that $\hat{r} \geq r^{(0.75)}$ in the context of Algorithm 2. With probability $\geq 1 - \frac{\delta}{18}$, every
554 $i \in [n]$ that is assigned $p_i > 0$ in Algorithm 2 satisfies $\|\mathbf{x}_i - \mathbf{x}_*\| \leq 3\hat{r}$, and $Z \geq 0.6n$.

555 *Proof.* Our proof is analogous to Lemma 6, where for all $i \in [n]$ we define the “ideal score”

$$f_i^* := \frac{k}{n} \sum_{j \in [n]} \mathbb{1}_{\|\mathbf{x}_i - \mathbf{x}_j\| \leq 2\hat{r}},$$

556 such that $\mathbb{E}f_i = f_i^*$. We first claim that with probability $\geq 1 - \frac{\delta}{18}$, the following hold for all $i \in [n]$:

$$f_i^* \geq 0.75k \implies f_i \geq 0.7k, \text{ and } f_i^* \leq 0.45k \implies f_i \leq 0.5k. \quad (10)$$

557 The first claim above is immediate from our choice of k and Fact 4 (with failure probability $\leq \frac{\delta}{18n}$
 558 for each $i \in [n]$); the second follows (with the same failure probability) similarly to (9), i.e.,

$$\exp\left(-\frac{\epsilon^2\mu}{2+\epsilon}\right) \leq \exp\left(-\frac{\epsilon\mu}{20}\right) \leq \exp\left(-\frac{k}{400}\right) \leq \frac{\delta}{18n},$$

559 in our application, with $\epsilon \geq \frac{1}{9}$ and $\epsilon\mu \geq \frac{k}{20}$. Thus a union bound proves (10).

560 To obtain the first claim, observe that any $i \in [n]$ with $\|\mathbf{x}_i - \mathbf{x}_*\| > 3\hat{r}$ must have that $\mathbb{B}(\mathbf{x}_i, 2\hat{r})$ does
 561 not intersect $\mathbb{B}(\mathbf{x}_*, \hat{r})$. However, $\mathbb{B}(\mathbf{x}_*, \hat{r})$ contains $0.75n$ points in \mathcal{D} by assumption, so $f_i^* \leq 0.25k$
 562 and thus as long as the implication (10) holds, then $p_i = 0$ as desired. For the second claim, any \mathbf{x}_i
 563 satisfying $\|\mathbf{x}_i - \mathbf{x}_*\| \leq r^{(0.75)}$ has $|\mathbb{B}(\mathbf{x}_i, 2\hat{r}) \cap \mathcal{D}| \geq 0.75n$, so that $f_i^* \geq 0.7k$. Thus, every such \mathbf{x}_i
 564 has $p_i \geq 0.8$ as long as (10) holds, so the total contribution made by the $\geq 0.75n$ such \mathbf{x}_i to Z is at
 565 least $0.6n$. \square

566 **Lemma 8.** Suppose that it is the case that in the context of Algorithm 2, we have

$$f_i^* \leq 0.45k \implies f_i \leq 0.5k, \text{ and } f_i^* \leq 0.55k \implies f_i \leq 0.6k, \quad (5)$$

567 for all $i \in [n]$. If $Z > 0.55n$, there exists some $\mathbf{x} \in \mathbb{R}^d$ such that $\{\mathbf{x}_i \mid p_i > 0\}_{i \in [n]} \subseteq \mathbb{B}(\mathbf{x}, 4\hat{r})$.
 568 Moreover, the event (5) occurs with probability $\geq 1 - \frac{\delta}{9}$.

569 *Proof.* The first statement in (5) was proven in (10) to hold with probability $\frac{\delta}{18}$, and the second
 570 statement's proof is identical to the first half up to changing constants, so we omit it. Conditioned on
 571 this event, every \mathbf{x}_i with $p_i > 0$ has $f_i^* > 0.45k$. Moreover, because $Z > 0.55n$, there exists some
 572 $j \in [n]$ (i.e., with the maximum value of p_j) such that $p_j > 0.55$, which implies $f_j > 0.6k$ and thus
 573 $f_j^* > 0.55k$.

574 So, we have shown that $\mathbb{B}(\mathbf{x}_j, 2\hat{r})$ contains more than $0.55n$ points in \mathcal{D} , and every surviving $i \in [n]$
 575 (i.e., with positive p_i) contains more than $0.45n$ points in \mathcal{D} . Thus, $\mathbb{B}(\mathbf{x}_j, 2\hat{r})$ and $\mathbb{B}(\mathbf{x}_i, 2\hat{r})$ intersect,
 576 and in particular, $\mathbb{B}(\mathbf{x}_j, 4\hat{r})$ contains every surviving point by the triangle inequality. \square

577 **Lemma 9.** If $n \geq 20$, Algorithm 2 is (ϵ, δ) -DP.

578 *Proof.* Fix neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathbb{B}^d(R)^n$, and assume without loss that they differ in the n^{th}
 579 entry $\mathbf{x}_n \neq \mathbf{x}'_n$. We will define \mathcal{A} , an alternate variant of Algorithm 2, which we denote $\bar{\mathcal{A}}$, where we
 580 condition on the following two events occurring. First, the index n should occur at most $2k$ times in
 581 $\bigcup_{i \in [n-1]} S_i$. Second, the implications (5) must hold. It is clear that the first described event occurs
 582 with probability $\geq 1 - \frac{\delta}{18}$ by using Fact 4 with our choice of k , and we proved in Lemma 8 that the
 583 second described event occurs with probability $\geq 1 - \frac{\delta}{9}$. Thus, the total variation distance between \mathcal{A}
 584 and $\bar{\mathcal{A}}$ is at most $\frac{\delta}{6}$. We will prove that \mathcal{A} is $(\epsilon, \frac{\delta}{3})$ -DP, from which Lemma 1 gives that $\bar{\mathcal{A}}$ is (ϵ, δ) -DP.

585 We begin by showing that according to \mathcal{A} , the statistic $Z + \xi - \frac{24}{\epsilon} \log(\frac{24}{\delta})$ satisfies $(\frac{\epsilon}{2}, \frac{\delta}{6})$ -DP. To do
 586 so, we will prove that Z has sensitivity ≤ 12 , and then apply Fact 2. Recall that by assumption, when
 587 \mathcal{A} is run the number of times n appears in $\bigcup_{i \in [n]} S_i$ is at most $2k$. Thus, for coupled values of Z, Z'
 588 corresponding to $\mathcal{D}, \mathcal{D}'$, where the coupling is over the random indices selected on Line 3,

$$Z - Z' \leq \frac{1}{0.25k} (k - 0) + \frac{1}{0.25k} \left(\text{number of copies of } n \text{ occurring in } \bigcup_{i \in [n-1]} S_i \right) \leq 12.$$

589 In fact, we note that the following stronger unsigned bound holds:

$$\begin{aligned} \sum_{i \in [n]} |p_i - p'_i| &\leq 4 + \sum_{i \in [n-1]} |p_i - p'_i| \\ &\leq 4 + \frac{1}{0.25k} \left(\text{number of copies of } n \text{ occurring in } \bigcup_{i \in [n-1]} S_i \right) \leq 12, \end{aligned} \quad (11)$$

590 because the clipping to the interval $[0, 1]$ in the definitions of p_i, p'_i can only improve $|p_i - p'_i|$, and
 591 the distance between the corresponding f_i, f'_i is at most the number of copies of n occurring in them.

Now, it remains to bound the privacy loss of the rest of \mathcal{A} , depending on whether Line 9 passes. If the algorithm terminates on Line 10, then there is no additional privacy loss.

Otherwise, suppose we enter the branch starting on Line 12. Our next step is to bound the sensitivity of $\bar{\mathbf{x}}$. Observe that whenever this branch is entered, we necessarily have $Z > 0.55n$ (and similarly, $Z' > 0.55n$), because $Z + \xi - \frac{24}{\epsilon} \log(\frac{24}{\delta}) \leq Z$ deterministically. Thus, Lemma 8 guarantees that in \mathcal{A} , all $\mathbf{x}_i \in \mathcal{D}$ with $p_i > 0$ are contained in a ball of radius $4\hat{r}$, and similarly all surviving elements in \mathcal{D}' are contained in a ball of radius $4\hat{r}$. However, there are at least $0.55n$ surviving elements of both \mathcal{D} and \mathcal{D}' , and in particular, for the given range of n there are at least two common surviving elements (one of which must be shared). A ball of radius $8\hat{r}$ around this element, which we denote $\hat{\mathbf{x}}$ in the rest of the proof, contains all surviving elements in \mathcal{D} (according to $\{p_i\}_{i \in [n]}$) and in \mathcal{D}' (according to $\{p'_i\}_{i \in [n]}$).

Now we wish to bound $\bar{\mathbf{x}} - \bar{\mathbf{x}}'$, where $\bar{\mathbf{x}}' := \frac{1}{Z'} \sum_{i \in [n]} p'_i \mathbf{x}'_i$. We have shown that in \mathcal{A} , $|Z - Z'| \leq 12$ and $\min(Z, Z') \geq 0.55n$. For convenience, define $\mathbf{y}_i := \mathbf{x}_i - \hat{\mathbf{x}}$ for all $i \in [n]$, and similarly define \mathbf{y}'_i . Recalling that all surviving elements of $\mathcal{D} \cup \mathcal{D}'$ are contained in $\mathbb{B}(\hat{\mathbf{x}}, 8\hat{r})$,

$$\begin{aligned} \|\bar{\mathbf{x}} - \bar{\mathbf{x}}'\| &= \left\| \frac{1}{Z} \sum_{i \in [n]} p_i \mathbf{y}_i - \frac{1}{Z'} \sum_{i \in [n]} p'_i \mathbf{y}'_i \right\| \\ &\leq \left| \frac{1}{Z} - \frac{1}{Z'} \right| \left\| \sum_{i \in [n-1]} p_i \mathbf{y}_i \right\| + \frac{1}{Z'} \left\| \sum_{i \in [n-1]} (p_i - p'_i) \mathbf{y}_i \right\| + \frac{p_n}{Z} \|\mathbf{y}_n\| + \frac{p'_n}{Z'} \|\mathbf{y}'_n\| \\ &\leq \left| \frac{Z' - Z}{Z'} \right| \left\| \frac{1}{Z} \sum_{i \in [n-1]} p_i \mathbf{y}_i \right\| + \frac{8\hat{r}}{0.55n} \sum_{i \in [n-1]} |p_i - p'_i| + \frac{16\hat{r}}{0.55n} \\ &\leq \frac{96\hat{r}}{0.55n} + \frac{104\hat{r}}{0.55n} + \frac{16\hat{r}}{0.55n} \leq \frac{400\hat{r}}{n}. \end{aligned}$$

The first line shifted both $\bar{\mathbf{x}}$ and $\bar{\mathbf{x}}'$ by $\hat{\mathbf{x}}$, and the second line applied the triangle inequality. The third line applied the triangle inequality to the middle term, and bounded the contribution of \mathbf{y}_n by using that $\|\mathbf{y}_n\| \leq 8\hat{r}$ if $p_n > 0$; a similar bound applies to \mathbf{y}'_n . In the fourth line, we used the triangle inequality on the first term, as well as that $\sum_{i \in [n-1]} |p_i - p'_i| \leq 1 + 12$ by using (11) and accounting for the n^{th} point separately. Thus, $\bar{\mathbf{x}}$ has sensitivity $\frac{400\hat{r}}{n}$ in \mathcal{A} . Fact 1 now guarantees that Line 17 is also $(\frac{\epsilon}{2}, \frac{\delta}{6})$ -DP. \square

Theorem 3. Let $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{B}^d(R)$ for $R > 0$, let $0 < r \leq 4r^{(0.9)}(\mathcal{D})$, and let $(\epsilon, \delta) \in [0, 1]^2$. Suppose that $n \geq C \cdot (\sqrt{d} \cdot \frac{\log(\frac{1}{\delta})}{\epsilon} + \frac{\log(\frac{\log(R/r)}{\delta})}{\epsilon})$, for a sufficiently large constant C . There is an (ϵ, δ) -DP algorithm that returns $(\hat{\mathbf{x}}, \hat{r})$ such that with probability $\geq 1 - \delta$, following notation (3),

$$f_{\mathcal{D}}(\hat{\mathbf{x}}) \leq (40C' + 1)f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})), \quad \hat{r} \leq 4r^{(0.9)}, \quad (6)$$

for a universal constant C' . Moreover, $\|\mathbf{x}_*(\mathcal{D}) - \hat{\mathbf{x}}\| \leq C'\hat{r}$. The algorithm runs in time

$$O\left(nd \log\left(\frac{R}{r}\right) \log\left(\frac{n \log(\frac{R}{r})}{\delta}\right)\right).$$

Proof. Regarding the utility bound, we will only establish that $\|\mathbf{x}_*(\mathcal{D}) - \hat{\mathbf{x}}\| \leq C'\hat{r}$, which also gives (6) upon observing that $f_{\mathcal{D}}$ is 1-Lipschitz, and $f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})) \geq 0.1r^{(0.9)}(\mathcal{D})$, due to Lemma 3.

We first run Algorithm 1 with $(\epsilon, \delta) \leftarrow (\frac{\epsilon}{2}, \frac{\delta}{2})$, which gives for large enough C (via Lemma 6)

$$\frac{1}{4}r^{(0.75)} \leq \hat{r} \leq 4r^{(0.9)}, \quad (12)$$

with probability $\geq 1 - \frac{\delta}{2}$. Next, we run Algorithm 2 with this value of \hat{r} , and parameters $(\epsilon, \delta) \leftarrow (\frac{\epsilon}{2}, \frac{\delta}{2})$. The privacy of composing these two algorithms now follows from Lemmas 5 and 9, and the runtime follows from Lemma 6, because Algorithm 1's runtime does not dominate upon inspection.

It remains to argue about the utility, i.e., that $\|\mathbf{x}_*(\mathcal{D}) - \hat{\mathbf{x}}\| \leq C'\hat{r}$. Conditioned on (12) holding, Lemma 6 guarantees that with probability $\geq 1 - \frac{\delta}{4}$, we have that $\|\bar{\mathbf{x}} - \mathbf{x}_*(\mathcal{D})\| \leq 3\hat{r}$, as a positively-weighted average of points in $\mathbb{B}^d(\mathbf{x}_*(\mathcal{D}), 3\hat{r})$. Finally, for the given value of σ in Algorithm 1, standard Gaussian concentration bounds imply that with probability $\geq 1 - \frac{\delta}{4}$,

$$\|\bar{\mathbf{x}} - \hat{\mathbf{x}}\| = \|\xi\| \leq 3\sigma \sqrt{d \log\left(\frac{4}{\delta}\right)} = O\left(\hat{r} \cdot \frac{\sqrt{d} \log(\frac{1}{\delta})}{n\epsilon}\right) = O(\hat{r}).$$

Thus, $\|\mathbf{x}_*(\mathcal{D}) - \hat{\mathbf{x}}\| \leq C'\hat{r}$ holds for an appropriate C' , except with probability δ . \square

D Deferred material from Section 4

In this appendix, we provide a privacy and utility proof for Algorithm 3, and combine them to prove Theorem 4. To recall, Algorithm 3 proceeds in phases $K \approx \log(T)$. In each phase (loop of Lines 2 to 16) other than $k = 1$, we define a domain $\mathcal{K}^{(k)}$ centered at the output of the previous phase with geometrically shrinking radius $\propto \sigma^{(k)}$; the domain for phase $k = 1$ is simply $\mathbb{B}^d(\bar{\mathbf{x}}, \hat{r})$. After this, we take $T^{(k)}$ steps of SGD over $\mathcal{K}^{(k)}$ with step size $\eta^{(k)}$, and output a noised variant of the average iterate in Lines 13 to 15.

D.1 Privacy of Algorithm 3

We first show that Algorithm 3 satisfies (ϵ, δ) -DP for an appropriate choice of ρ . When the sample functions of interest are smooth (i.e., have bounded second derivative), [FKT20] gives a proof based on the *contractivity* of iterates. This is based on the observation that gradient descent steps with respect to a smooth function are contractive for an appropriate step size (see e.g., Proposition 2.10, [FKT20]). Unfortunately, our sample functions are of the form $\|\cdot - \mathbf{x}_i\|$, which are not even differentiable, let alone smooth. Nonetheless, we inductively prove approximate contractivity of Algorithm 3's iterates by using the structure of the geometric median objective.

Throughout, we fix neighboring $\mathcal{D}, \mathcal{D}' \in (\mathbb{R}^d)^n$, and assume without loss of generality they differ in the n^{th} entry. To simplify notation, we let $\mathcal{I} \in [n]^T$ denote the multiset of T indices sampled in Line 10, across all phases. We prove DP of Algorithm 3 using Lemma 1, where we let $\bar{\mathcal{A}}$ denote Algorithm 3, and we let \mathcal{A} denote a variant of Algorithm 3 conditioned on \mathcal{I} containing at most $m := 3(\frac{T}{n} + \log(\frac{8}{\delta}))$ copies of n . We first bound the total variation distance between \mathcal{A} and $\bar{\mathcal{A}}$ using Fact 4.

Lemma 10. *With probability $\geq 1 - \frac{\delta}{8}$, Algorithm 3 yields \mathcal{I} containing $\leq m$ copies of n .*

Proof. In expectation, we have $\frac{T}{n} \geq 1$ copies, so the result follows from Fact 4 and our choice of m . \square

We will show that \mathcal{A} is $(\epsilon, \frac{\delta}{2})$ -DP, upon which Lemmas 1 and 10 imply that $\bar{\mathcal{A}}$ (Algorithm 3) is (ϵ, δ) -DP. To do so, we control the sensitivity of each iterate $\mathbf{z}_t^{(k)}$, using the following two helper facts.

Fact 5 ([Roc76]). *Let $\mathcal{K} \subset \mathbb{R}^d$ be a compact, convex set. Then for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have*

$$\|\Pi_{\mathcal{K}}(\mathbf{x}) - \Pi_{\mathcal{K}}(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|.$$

Lemma 11. *For any unit vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, and $a, b > 0$, let $\mathbf{x} = a\mathbf{u}$ and $\mathbf{y} = b\mathbf{v}$. Then, letting $\mathbf{x}' \leftarrow (a - \eta)\mathbf{u}$ and $\mathbf{y}' \leftarrow (b - \eta)\mathbf{v}$, we have $\|\mathbf{x}' - \mathbf{y}'\| \leq \max(\|\mathbf{x} - \mathbf{y}\|, 3\eta)$.*

Proof. We claim that

$$\|\mathbf{x}' - \mathbf{y}'\| \leq \|\mathbf{x} - \mathbf{y}\| \iff a + b \geq \eta, \quad (13)$$

from which the proof follows from observing that if $a + b \leq \eta$, then we can trivially bound

$$\|\mathbf{x}' - \mathbf{y}'\| \leq \|\mathbf{x} - \mathbf{y}\| + 2\eta \leq a + b + 2\eta \leq 3\eta.$$

Indeed, (13) follows from a direct expansion:

$$\begin{aligned}\|\mathbf{x} - \mathbf{y}\|^2 - \|\mathbf{x}' - \mathbf{y}'\|^2 &= a^2 + b^2 - 2ab \langle \mathbf{u}, \mathbf{v} \rangle - (a - \eta)^2 - (b - \eta)^2 + 2(a - \eta)(b - \eta) \langle \mathbf{u}, \mathbf{v} \rangle \\ &= 2\eta(a + b) - 2\eta^2 - 2\eta(a + b) \langle \mathbf{u}, \mathbf{v} \rangle + 2\eta^2 \langle \mathbf{u}, \mathbf{v} \rangle \\ &= 2\eta(a + b - \eta)(1 - \langle \mathbf{u}, \mathbf{v} \rangle).\end{aligned}$$

Thus, for $\eta \geq 0$ and $\langle \mathbf{u}, \mathbf{v} \rangle \geq 0$, we conclude that (13) holds. \square

Corollary 1. For any phase $k \in [K]$ in Algorithm 3, condition on the value of $\mathbf{z}_0^{(k)}$, and assume that \mathcal{I} contains at most m copies of n . Then for any $0 \leq t < T_k$, the sensitivity of $\mathbf{z}_t^{(k)}$ is $\leq (2m + 1)\eta^{(k)}$.

Proof. Throughout this proof only, we drop the iteration k from superscripts for notational simplicity, so we let $T := T^{(k)}$ and $\eta := \eta^{(k)}$, referring to the relevant iterates as $\{\mathbf{z}_t\}_{0 \leq t < T} := \{\mathbf{z}_t^{(k)}\}_{0 \leq t < T^{(k)}}$. We also refer to the index i selected on Line 10 in iteration $0 \leq t < T$ by i_t .

Fix two copies of the k^{th} phase of Algorithm 3, both initialized at \mathbf{z}_0 , but using neighboring datasets $\mathcal{D}, \mathcal{D}'$ differing in the n^{th} entry. Also, fix a realization of $\{i_t\}_{0 \leq t < T}$, such that $i_t = n$ at most m choices of t (note that m is actually a bound on how many times $i_t = n$ across all phases, so it certainly bounds the occurrence count in a single phase). Conditioned on this realization, Algorithm 3 is now a deterministic mapping from \mathbf{z}_0 to the iterates $\{\mathbf{z}_t\}_{0 \leq t < T}$, depending on the dataset used.

Denote the iterates given by the dataset \mathcal{D} by $\{\mathbf{z}_t\}_{0 \leq t < T}$ and the iterates given by \mathcal{D}' by $\{\mathbf{z}'_t\}_{0 \leq t < T}$, so that $\mathbf{z}_0 = \mathbf{z}'_0$ by assumption. Also, let $\Phi_t := \|\mathbf{z}_t - \mathbf{z}'_t\|$ for all $0 \leq t < T$. We claim that for all $0 \leq t < T$,

$$\Phi_t \leq \max(2m_t + 1, 3)\eta, \text{ where } m_t := \sum_{0 \leq s < t} \mathbb{1}_{i_s = n}, \quad (14)$$

i.e., m_t is the number of times the index n was sampled in the first t iterations of the phase. If we can show (14) holds, then we are done because $m_t \leq m$ by assumption.

We are left with proving (14), which we do by induction. The base case $t = 0$ is clear. Suppose (14) holds at iteration t . In iteration $t + 1$, if $m_{t+1} = m_t + 1$ (i.e., $i_t = n$ was sampled), then (14) holds by the triangle inequality and the induction hypothesis, because all gradient steps $\eta \mathbf{g}_t$ have $\|\eta \mathbf{g}_t\| \leq \eta$, and projection to \mathcal{K} can only decrease distances (Fact 5). Otherwise, let $i_t = i \neq n$ be the sampled index, using the common point $\mathbf{x}_i \in \mathcal{D} \cap \mathcal{D}'$. Now, (14) follows from applying Lemma 11 with

$$\mathbf{x} \leftarrow \mathbf{z}_t - \mathbf{x}_i, \mathbf{y} \leftarrow \mathbf{z}'_t - \mathbf{x}_i, \mathbf{u} \leftarrow \frac{\mathbf{z}_t - \mathbf{x}_i}{\|\mathbf{z}_t - \mathbf{x}_i\|}, \mathbf{v} \leftarrow \frac{\mathbf{z}'_t - \mathbf{x}_i}{\|\mathbf{z}'_t - \mathbf{x}_i\|}.$$

In particular, we have that $\|\mathbf{x} - \mathbf{y}\| = \Phi_t$, and $\|\mathbf{x}' - \mathbf{y}'\| \geq \Phi_{t+1}$ (due to Fact 5), following notation from Lemma 11. We thus have $\Phi_{t+1} \leq \max(\Phi_t, 3\eta)$, which clearly also preserves (14) inductively. \square

We can now conclude our privacy proof by applying composition to Corollary 1.

Lemma 12. Let $\epsilon \in [0, 1]$. If $\frac{1}{\rho} \geq \frac{4 \log(\frac{2}{\delta})}{\epsilon^2} + \frac{2}{\epsilon}$, Algorithm 3 is (ϵ, δ) -DP.

Proof. We claim that Algorithm 3 satisfies ρ -CDP, conditioned on \mathcal{I} containing at most m choices of t (we denote this conditional variant by \mathcal{A}). By applying the second part of Fact 3 with $\alpha \leftarrow \frac{2 \log(\frac{2}{\delta})}{\epsilon} + 1$, this implies that \mathcal{A} is $(\epsilon, \frac{\delta}{2})$ -DP. Because \mathcal{A} has total variation distance at most $\frac{\delta}{8}$ to Algorithm 3 due to Lemma 10, we conclude using Lemma 1 that Algorithm 3 is (ϵ, δ) -DP.

We are left to show \mathcal{A} satisfies ρ -CDP. In fact, we will show that for all $k \in [K]$, the output of the k^{th} phase of \mathcal{A} , i.e., $\hat{\mathbf{x}}^{(k)}$, satisfies $(\frac{16}{9})^{-k} \cdot \frac{\rho}{2}$ -CDP (treating the starting iterate $\mathbf{z}_0^{(k)} = \hat{\mathbf{x}}^{(k-1)}$ as fixed). Using composition of RDP (the first part of Fact 3), this implies \mathcal{A} is ρ -CDP as desired.

Finally, we bound the CDP of phase $k \in [K]$. Under \mathcal{A} , we showed in Corollary 1 that all iterates of the k^{th} phase have sensitivity $\leq (2m + 1)\eta^{(k)}$. Thus the average iterate $\bar{\mathbf{x}}^{(k)}$ also has sensitivity $\leq (2m + 1)\eta^{(k)}$ by the triangle inequality. We can now bound the CDP of the k^{th} phase using the third part of Fact 3:

$$\frac{((2m + 1)\eta^{(k)})^2}{2(\sigma^{(k)})^2} = \frac{((2m + 1)\eta)^2}{((2m + 1)\eta)^2} \cdot 16^{-k} \cdot 9^k \cdot \frac{\rho}{2} \leq \left(\frac{16}{9}\right)^{-k} \cdot \frac{\rho}{2}.$$

698 D.2 Utility of Algorithm 3

699 We now analyze the error guarantees for Algorithm 3 on optimizing the geometric median objective
700 $f_{\mathcal{D}}$ (3). We begin by providing a high-probability bound on the utility guarantees of each single
701 phase.

702 **Lemma 13.** *Following notation of Algorithm 3, we have with probability $\geq 1 - \frac{\delta}{2}$ that*

$$f_{\mathcal{D}}(\bar{\mathbf{x}}^{(1)}) - f_{\mathcal{D}}(\mathbf{x}_*) \leq \frac{\hat{r}^2}{2\eta^{(1)}T^{(1)}} + \frac{\eta^{(1)}}{2} + 4\hat{r}\sqrt{\frac{2\log(\frac{4K}{\delta})}{T^{(1)}}},$$

703 where $\mathbf{x}_* := \operatorname{argmin}_{\mathbf{x} \in \mathbb{B}^d(\bar{\mathbf{x}}, \hat{r})} f(\mathbf{x})$, and

$$f_{\mathcal{D}}(\bar{\mathbf{x}}^{(k)}) - f_{\mathcal{D}}(\bar{\mathbf{x}}^{(k-1)}) \leq \frac{2(\sigma^{(k)})^2 d \log(\frac{4K}{\delta})}{\eta^{(k)}T^{(k)}} + \frac{\eta^{(k)}}{2} + 8\sigma^{(k)} \log\left(\frac{4K}{\delta}\right) \sqrt{\frac{2d}{T^{(k)}}} \text{ for all } 2 \leq k \leq K.$$

704 *Proof.* First, with probability $\geq 1 - \frac{\delta}{4}$, we have

$$\|\xi^{(k)}\| \leq 2\sigma^{(k)} \sqrt{d \log\left(\frac{4K}{\delta}\right)} \text{ for all } k \in [K],$$

705 by standard Gaussian concentration. Thus, $\bar{\mathbf{x}}^{(k-1)} \in \mathcal{K}^{(k)}$ for all $2 \leq k \leq K$, and $\mathbf{x}_* \in \mathcal{K}^{(1)}$, except
706 with probability $\frac{\delta}{4}$. Next, consider the k^{th} phase of Algorithm 3, and for some $0 \leq t < T_k$, let us
707 denote

$$\tilde{\mathbf{g}}_t^{(k)} := \frac{\mathbf{z}_t^{(k)} - \mathbf{x}_i}{\|\mathbf{z}_t^{(k)} - \mathbf{x}_i\|}$$

708 where $i \in [n]$ is the random index sampled on Line 10 in the t^{th} iteration of phase k . We also denote

$$\mathbf{g}_t^{(k)} := \frac{1}{n} \sum_{i \in [n]} \frac{\mathbf{z}_t^{(k)} - \mathbf{x}_i}{\|\mathbf{z}_t^{(k)} - \mathbf{x}_i\|}.$$

709 We observe that $\mathbb{E}[\tilde{\mathbf{g}}_t^{(k)}] = \mathbf{g}_t^{(k)}$ for any realization of the randomness in all previous iterations.
710 Now, by the standard Euclidean mirror descent analysis, see e.g., Theorem 3.2 of [Bub15], for any
711 $\mathbf{u} \in \mathcal{K}^{(k)}$,

$$\left\langle \eta^{(k)} \tilde{\mathbf{g}}_t^{(k)}, \mathbf{z}_t^{(k)} - \mathbf{u} \right\rangle \leq \frac{\|\mathbf{z}_t^{(k)} - \mathbf{u}\|^2}{2} - \frac{\|\mathbf{z}_{t+1}^{(k)} - \mathbf{u}\|^2}{2} + \frac{(\eta^{(k)})^2}{2}. \quad (15)$$

712 Here we implicitly used that $\|\tilde{\mathbf{g}}_t^{(k)}\| \leq 1$ for all choices of the sampled index $i \in [n]$. Now summing
713 (15) for all iterations $0 \leq t < T^{(k)}$, and normalizing by $\eta^{(k)}T^{(k)}$, we obtain

$$\frac{1}{T^{(k)}} \sum_{0 \leq t < T^{(k)}} \left\langle \tilde{\mathbf{g}}_t^{(k)}, \mathbf{z}_t^{(k)} - \mathbf{u} \right\rangle \leq \frac{\|\mathbf{z}_0^{(k)} - \mathbf{u}\|^2}{2\eta^{(k)}T^{(k)}} + \frac{\eta^{(k)}}{2}.$$

714 Next, we claim that with probability $\geq 1 - \frac{\delta}{4}$,

$$\begin{aligned} \frac{1}{T^{(1)}} \sum_{0 \leq t < T^{(1)}} \left\langle \mathbf{g}_t^{(1)} - \tilde{\mathbf{g}}_t^{(1)}, \mathbf{z}_t^{(1)} - \mathbf{u} \right\rangle &\leq 4\hat{r}\sqrt{\frac{2\log(\frac{4K}{\delta})}{T^{(1)}}}, \\ \frac{1}{T^{(k)}} \sum_{0 \leq t < T^{(k)}} \left\langle \mathbf{g}_t^{(k)} - \tilde{\mathbf{g}}_t^{(k)}, \mathbf{z}_t^{(k)} - \mathbf{u} \right\rangle &\leq 8\sigma^{(k)} \log\left(\frac{4K}{\delta}\right) \sqrt{\frac{2d}{T^{(k)}}} \text{ for all } 2 \leq k \leq K. \end{aligned}$$

715 In each case, this is because $\langle \mathbf{g}_t^{(k)} - \tilde{\mathbf{g}}_t^{(k)}, \mathbf{z}_t^{(k)} - \mathbf{u} \rangle$ is a mean-zero random variable, that is bounded
716 (with probability 1) by twice the diameter of $\mathcal{K}^{(k)}$. Thus we can bound the sub-Gaussian parameter

of their sum, and applying the Azuma-Hoeffding inequality then gives the result. Now, finally by convexity,

$$\begin{aligned} \frac{1}{T^{(1)}} \sum_{0 \leq t < T^{(1)}} \langle \mathbf{g}_t^{(1)}, \mathbf{z}_t^{(1)} - \mathbf{u} \rangle &\geq \frac{1}{T^{(1)}} \sum_{0 \leq t < T^{(1)}} f_{\mathcal{D}}(\mathbf{z}_t^{(1)}) - f_{\mathcal{D}}(\mathbf{u}) \geq f_{\mathcal{D}}(\bar{\mathbf{x}}^{(1)}) - f_{\mathcal{D}}(\mathbf{u}), \\ \frac{1}{T^{(k)}} \sum_{0 \leq t < T^{(k)}} \langle \mathbf{g}_t^{(k)}, \mathbf{z}_t^{(k)} - \mathbf{u} \rangle &\geq \frac{1}{T^{(k)}} \sum_{0 \leq t < T^{(k)}} f_{\mathcal{D}}(\mathbf{z}_t^{(k)}) - f_{\mathcal{D}}(\mathbf{u}) \geq f_{\mathcal{D}}(\bar{\mathbf{x}}^{(k)}) - f_{\mathcal{D}}(\mathbf{u}) \text{ for all } 2 \leq k \leq K. \end{aligned}$$

Combining the above three displays, and plugging in $\mathbf{u} \leftarrow \mathbf{x}_\star$ or $\mathbf{u} \leftarrow \bar{\mathbf{x}}^{(k-1)}$, now gives the conclusion. \square

By summing the conclusion of Lemma 13 across all phases, we obtain an overall error bound.

Lemma 14. *Following notation of Algorithm 3 and Lemma 13, we have with probability $\geq 1 - \delta$ that*

$$f_{\mathcal{D}}(\hat{\mathbf{x}}^{(K)}) - f_{\mathcal{D}}(\mathbf{x}_\star) \leq \frac{\hat{r}^2}{16\eta T} + 19\eta + 8\hat{r} \sqrt{\frac{\log(\frac{4K}{\delta})}{T}} + \frac{1314T\eta d \log^4(\frac{8K}{\delta})}{\rho n^2}.$$

Proof. Throughout this proof, condition on the conclusion of Lemma 13 holding, as well as

$$\|\boldsymbol{\xi}^{(K)}\| \leq 2\sigma^{(K)} \sqrt{d \log\left(\frac{2}{\delta}\right)},$$

both of which hold with probability $\geq 1 - \delta$ by a union bound. Next, by Lemma 13,

$$\begin{aligned} f_{\mathcal{D}}(\hat{\mathbf{x}}_K) - f_{\mathcal{D}}(\mathbf{x}_\star) &= f_{\mathcal{D}}(\bar{\mathbf{x}}^{(1)}) - f_{\mathcal{D}}(\mathbf{x}_\star) + \sum_{k=2}^K f_{\mathcal{D}}(\bar{\mathbf{x}}^{(k)}) - f_{\mathcal{D}}(\bar{\mathbf{x}}^{(k-1)}) + f_{\mathcal{D}}(\hat{\mathbf{x}}_K) - f_{\mathcal{D}}(\bar{\mathbf{x}}_K) \\ &\leq \frac{\hat{r}^2}{2\eta^{(1)}T^{(1)}} + \frac{\eta^{(1)}}{2} + 4\hat{r} \sqrt{\frac{2 \log(\frac{4K}{\delta})}{T^{(1)}}} \\ &\quad + \sum_{k=2}^K \left(\frac{2(\sigma^{(k)})^2 d \log(\frac{4K}{\delta})}{\eta^{(k)}T^{(k)}} + \frac{\eta^{(k)}}{2} + 8\sigma^{(k)} \log\left(\frac{4K}{\delta}\right) \sqrt{\frac{2d}{T^{(k)}}} \right) + \|\boldsymbol{\xi}^{(K)}\| \\ &\leq \frac{\hat{r}^2}{16\eta T} + \frac{\eta}{2} + 8\hat{r} \sqrt{\frac{\log(\frac{4K}{\delta})}{T}} + \frac{144m^2\eta d \log(\frac{4K}{\delta})}{\rho T} + \frac{12\sqrt{dm}\eta \log(\frac{4K}{\delta})}{\sqrt{\rho T}} \\ &\leq \frac{\hat{r}^2}{16\eta T} + \frac{\eta}{2} + 8\hat{r} \sqrt{\frac{\log(\frac{4K}{\delta})}{T}} + \frac{1296T\eta d \log^3(\frac{8K}{\delta})}{\rho n^2} + \frac{36\sqrt{dT}\eta \log^2(\frac{8K}{\delta})}{\sqrt{\rho n}} \\ &\leq \frac{\hat{r}^2}{16\eta T} + 19\eta + 8\hat{r} \sqrt{\frac{\log(\frac{4K}{\delta})}{T}} + \frac{1314T\eta d \log^4(\frac{8K}{\delta})}{\rho n^2}. \end{aligned}$$

The third line used Lipschitzness of $f_{\mathcal{D}}$, the fourth summed parameters using various geometric sequences, the fifth plugged in our value of m , and the last split the fifth term using $2ab \leq a^2 + b^2$ appropriately. \square

By combining Lemmas 12 and 14 with Theorem 3, we obtain our main result on privately approximating the geometric median to an arbitrary multiplicative factor $1 + \alpha$, given enough samples.

Theorem 4. *Let $\mathcal{D} = \{\mathbf{x}_i\}_{i \in [n]} \subset \mathbb{B}^d(R)$ for $R > 0$, let $0 < r \leq 4r^{(0.9)}(\mathcal{D})$, and let $(\alpha, \epsilon, \delta) \in [0, 1]^3$. Suppose that $n \geq C \cdot (\frac{\sqrt{d}}{\alpha\epsilon} \log^{2.5}(\frac{\log(\frac{d}{\alpha\delta\epsilon})}{\delta}))$, for a sufficiently large constant C . There is an (ϵ, δ) -DP algorithm that returns $\hat{\mathbf{x}}$ such that with probability $\geq 1 - \delta$, following notation (3), $f_{\mathcal{D}}(\hat{\mathbf{x}}) \leq (1 + \alpha)f_{\mathcal{D}}(\mathbf{x}_\star(\mathcal{D}))$. The algorithm runs in time*

$$O\left(nd \log\left(\frac{R}{r}\right) \log\left(\frac{d \log(\frac{R}{r})}{\alpha\delta\epsilon}\right) + \frac{d}{\alpha^2} \log\left(\frac{\log(\frac{d}{\alpha\delta\epsilon})}{\delta}\right)\right).$$

734 *Proof.* We first apply Theorem 3 to compute a $(\bar{\mathbf{x}}, \hat{r})$ pair satisfying

$$\hat{r} \leq C' r^{(0.9)}(\mathcal{D}), \quad \|\bar{\mathbf{x}} - \mathbf{x}_*(\mathcal{D})\| \leq \hat{r},$$

735 for a universal constant C' , subject to $(\frac{\epsilon}{2}, \frac{\delta}{2})$ -DP and $\frac{\delta}{2}$ failure probability. We can verify that
 736 Theorem 3 gives these guarantees within the stated runtime, for a large enough C . Next, we call
 737 Algorithm 3 with $\rho \leftarrow \frac{\epsilon^2}{32 \log(\frac{4}{\delta})}$ and $\delta \leftarrow \frac{\delta}{2}$, which is $(\frac{\epsilon}{2}, \frac{\delta}{2})$ -DP by Lemma 12, so this composition
 738 is (ϵ, δ) -DP.

739 Denoting $\hat{\mathbf{x}} := \hat{\mathbf{x}}^{(K)}$ to be the output of Algorithm 3, Lemma 14 guarantees that with probability
 740 $\geq 1 - \frac{\delta}{2}$,

$$f_{\mathcal{D}}(\hat{\mathbf{x}}) - f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})) \leq \frac{\hat{r}^2}{16\eta T} + 19\eta + 8\hat{r} \sqrt{\frac{\log(\frac{16K}{\delta})}{T}} + \frac{5256T\eta d \log^5(\frac{16K}{\delta})}{\epsilon^2 n^2},$$

741 for some choice of η, T and our earlier choices of privacy parameters. Optimizing in η , we have

$$f_{\mathcal{D}}(\hat{\mathbf{x}}) - f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})) \leq 12\hat{r} \sqrt{\frac{\log(\frac{16K}{\delta})}{T}} + \frac{37\hat{r} \sqrt{d \log^5(\frac{16K}{\delta})}}{\epsilon n}.$$

742 Finally, for a large enough C in the definition of n , and $T \geq n + \frac{57600(C')^2 \log(\frac{16K}{\delta})}{\alpha^2}$, we obtain

$$f_{\mathcal{D}}(\hat{\mathbf{x}}) - f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})) \leq 12\hat{r} \sqrt{\frac{\log(\frac{16K}{\delta})}{T}} + \frac{37\hat{r} \sqrt{d \log^5(\frac{16K}{\delta})}}{\epsilon n} \leq \frac{\alpha \hat{r}}{10C'} \leq \frac{\alpha r^{(0.9)}}{10} \leq \alpha f_{\mathcal{D}}(\mathbf{x}_*(\mathcal{D})).$$

743 The last inequality used Lemma 3. Now, the runtime follows from combining Theorem 3 and the fact
 744 that every iteration of Algorithm 3 can clearly be implemented in $O(d)$ time. \square

745 E Deferred material from Section 5

746 We first describe the datasets used for our experiments in Section 5.

747 **GaussianCluster** $(R, n, d, \sigma, \text{frac}_{\text{in}})$: This dataset is described in Appendix H of [HSU24]. We draw
 748 $n_{\text{in}} = \text{frac}_{\text{in}} n$ points i.i.d. from $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_d)$ with $\boldsymbol{\mu}$ uniform on the sphere of radius $\frac{R}{2}$, and
 749 $n_{\text{out}} = n - n_{\text{in}}$ outliers uniformly from the Euclidean ball of radius R .

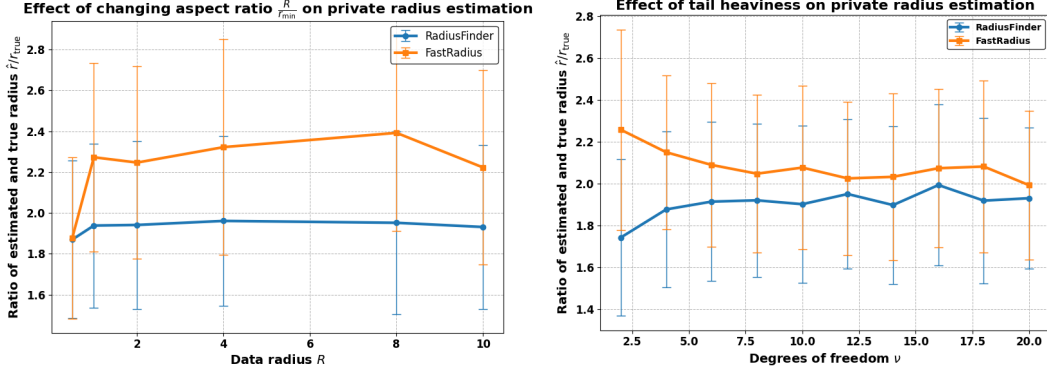
750 **HeavyTailed** (ν, n, d) : This dataset samples n points in \mathbb{R}^d from a zero-mean multivariate Student's t
 751 distribution with identity scale and degrees of freedom ν .

752 To avoid contamination in hyperparameter selection, every experiment is performed with a fresh
 753 dataset.

754 **Subsampling.** We now provide more details and results on our evaluation of Algorithm 1.

755 In the first experiment (Figure 2a), we set $n = 1000$, $d = 10$, inlier fraction $\text{frac}_{\text{in}} = 0.9$, standard
 756 deviation $\sigma = 0.1$, and choose an upper bound R from the set $\{0.5, 1, 2, 4, 8, 10\}$. The dataset is
 757 generated as **GaussianCluster** $(R, n, d, \sigma, \text{frac}_{\text{in}})$ dataset. We set privacy parameters $\epsilon = 1.0$ and
 758 $\delta = 10^{-5}$, quantile fraction $\gamma = 0.75$, and for each trial we sample $r_{\text{min}} \sim \text{Unif}([0.005, 0.02])$ to
 759 randomly initialize our search grid. Since $\gamma < \text{frac}_{\text{in}}$, we estimate the ground-truth quantile radius
 760 $r_{\text{true}} = \sigma \sqrt{d}$, run both algorithms on this dataset, measure the estimated radius \hat{r} and wall-clock
 761 runtime, and report the mean and standard deviation of the estimation ratio \hat{r}/r_{true} and runtime over
 762 100 independent trials.

763 In the second experiment (Figure 2b), we assess the robustness of FastRadius and RadiusFinder
 764 to heavy-tailed data. We set $n = 1000$ and $d = 10$ in the **HeavyTailed** (ν, n, d) dataset with
 765 varying degrees of freedom $\nu \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$. For each trial, we sample $r_{\text{min}} \sim$
 766 $\text{Unif}([0.005, 0.02])$, set privacy parameters $\epsilon = 1.0$, $\delta = 10^{-5}$ and quantile fraction $\gamma = 0.75$. We
 767 estimate the theoretical quantile radius $r_{\text{true}} = \sqrt{d F_{d,\nu}(\gamma)}$ where $F_{d,\nu}$ is the CDF of an $F(d, \nu)$
 768 distribution, which is the Fisher F-distribution with d and ν degrees of freedom, execute both
 769 algorithms, record \hat{r} and runtime, and summarize the mean and standard deviation of the ratio \hat{r}/r_{true}
 770 and runtime across 100 repetitions.



(a) Ratio of the estimated quantile radius to the true radius with varying data radius R . (b) Ratio of the estimated quantile radius to the true radius with varying degrees of freedom ν .

Figure 2: Comparison of RadiusFinder and FastRadius across different data distributions. All plots are averaged across 100 trials and standard deviations are reported as error bars.

We observe that in both cases, across a range of increasingly heavier tails of the distributions, both algorithms achieve reasonable approximation to the true quantile radius, always staying multiplicatively between roughly 1.2 to 3 of the true quantile radius. We further record the average wall-clock time required by both algorithms in Table 1. We observe that FastRadius is significantly faster compared to RadiusFinder, while performing competitively in terms of estimation quality.

We remark that we also experimented with varying $n \in \{500, 1000, 2000\}$ and $d \in \{5, 10, 20\}$ and observed qualitatively similar trends for the performance of both algorithms.

Table 1: Average wall-clock time in seconds over 100 trials for each algorithm in each experiment

Experiment	RadiusFinder	FastRadius
Varying R (Figure 2a)	1.192 ± 0.047	0.0411 ± 0.002
Varying ν (Figure 2b)	1.204 ± 0.171	0.0413 ± 0.007

Boosting. We conclude by describing our evaluation the performance of our boosting algorithm in Section 4 based on a low-pass DP-SGD implementation, compared to the baseline method from [HSU24].

We briefly describe our hyperparameter optimization process for the baseline, DPGD. To satisfy ρ -CDP, Algorithm 3 in [HSU24] recommends a constant step size of $\eta_{\text{base}} = 2\hat{r}\sqrt{\frac{d}{6\rho n^2}}$, where \hat{r} is the estimated radius. We examined performance using various step sizes $\eta = \eta_{\text{base}} \cdot \eta_{\text{multiplier}}$ with multipliers $\eta_{\text{multiplier}} \in \{0.5, 1, 10, 30, 50, 100\}$. Our results indicate that the $\eta_{\text{multiplier}}$ depends significantly on dataset size, in that larger datasets benefit from higher $\eta_{\text{multiplier}}$. Specifically, we observe that as n increases, larger $\eta_{\text{multiplier}}$ values consistently reduce optimization error, but with diminishing returns. Based on these experiments, we select $\eta_{\text{multiplier}} = 30$ for DPGD in our experiments, as this consistently provides nearly the best performance across the range of datasets and parameter settings in our evaluation.

We now describe our setup. In all our experiments, we set $d = 50$, $\rho = 0.5$, and vary $n \in \{100, 1000, 10000\}$. For the GaussianCluster dataset, we set $\sigma = 0.1$ and vary the bounding radius $R \in \{25, 50, 100\}$. We set our estimated initial radius $\hat{r} = 20\sigma\sqrt{d}$ and initialize all algorithms at a uniformly random point on the surface of $\mathbb{B}^d(0.75\hat{r})$.⁷ For the HeavyTailed dataset, we use the same values of d , ρ , and the same range of n . We vary $\nu \in \{2.5, 5.0, 10.0\}$, set our estimated initial radius $\hat{r} = 20\sqrt{dF_{d,\nu}(0.75)}$ to be consistent with our subsampling experiment, and again initialize randomly on the surface of $\mathbb{B}^d(0.75\hat{r})$.

⁷We chose a relatively pessimistic multiple of \hat{r} to create a larger initial loss and account for estimation error.

In our first set of experiments, we used the middle “scale” parameter, i.e., $R = 50$ for the GaussianCluster dataset and $\nu = 5.0$ for the HeavyTailed dataset, varying n only. We report the performance of the three evaluated methods, plotting the passes over the dataset used by the excess error. Our error metric is $\frac{1}{\hat{r}} \cdot (f_{\mathcal{D}}(\hat{\mathbf{x}}) - f_{\mathcal{D}}(\hat{\mathbf{x}}^*))$, i.e., a multiple of the “effective radius” used in the experiment. This is a more reflective performance metric than the corresponding multiple of $f_{\mathcal{D}}$, as our algorithms achieve this bound (see discussion after Theorem 3), and $f_{\mathcal{D}} \geq \hat{r}$ for our datasets due to outliers.

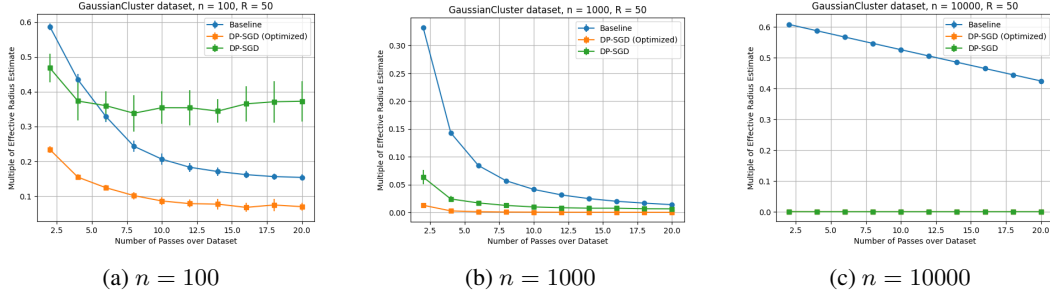


Figure 3: Comparison of DPGD, StableDPSGD, and FixedOrderDPSGD across GaussianCluster data over \mathbb{R}^{50} , varying n . Plots averaged across 20 trials and standard deviations are reported as error bars.

Across GaussianCluster datasets of size $n \in \{100, 1000, 10000\}$, we consistently saw that our optimized FixedOrderDPSGD performed the best, with comparisons between the baseline DPGD and StableDPSGD fluctuating depending on the dataset size. As the theory predicts, the gains of stochastic methods in terms of error-to-pass ratios are more stark when dataset sizes are larger, reflecting the superlinear gradient query complexity (each requiring one pass) that DPGD needs to obtain the optimal utility.

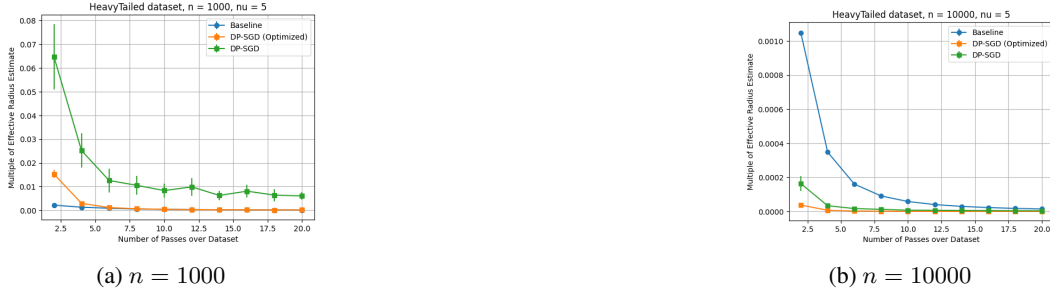


Figure 4: Comparison of DPGD, StableDPSGD, and FixedOrderDPSGD across HeavyTailed data over \mathbb{R}^{50} , varying n . Plots averaged across 20 trials and standard deviations are reported as error bars.

We next present our comparisons for the HeavyTailed dataset. Here, a step size multiple of $\eta_{\text{multiplier}} = 30$ was too aggressive for the baseline DPGD to reliably converge when $n = 100$, but smaller steps impacted the performance of DPGD with larger sample sizes. On the sample sizes where DPGD successfully converged, we saw that our FixedOrderDPSGD consistently outperformed StableDPSGD, whereas the baseline DPGD slightly outperformed FixedOrderDPSGD when $n = 1000$ was moderate. The improved performance of DPGD may be related to the relative simplicity of this dataset, where all the data is distributed in a rotationally symmetric way around the “population geometric median,” i.e., the mean.

In our second set of experiments, we fixed the size of the dataset at $n = 1000$, varying the scale parameter (R for GaussianCluster and ν for HeavyTailed). We observed that the relative performance of our evaluated algorithms was essentially unchanged across the parameter settings we considered.

Finally, we remark that one major limitation of our evaluation is that full-batch gradient methods such as DPGD can be implemented with parallelized gradient computations, leading to wall-clock time savings. In our experiments, DPGD often performed better than FixedOrderDPSGD in terms of wall-clock time (for the same estimation error), even when it incurred significantly larger pass

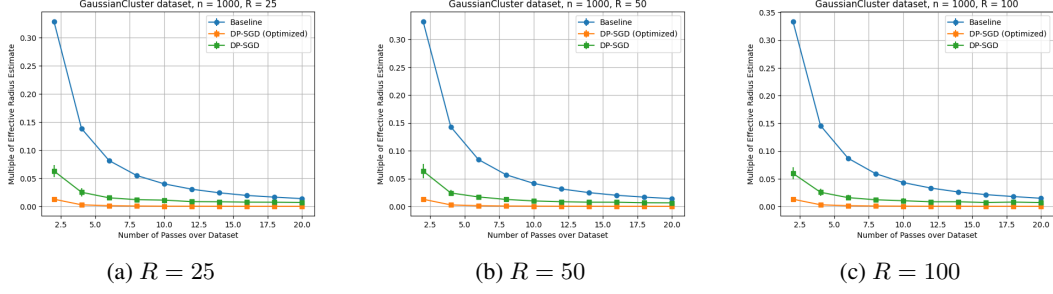


Figure 5: Comparison of DPGD, StableDPSGD, and FixedOrderDPSGD across GaussianCluster data over \mathbb{R}^{50} , varying R . Plots averaged across 20 trials and standard deviations are reported as error bars.

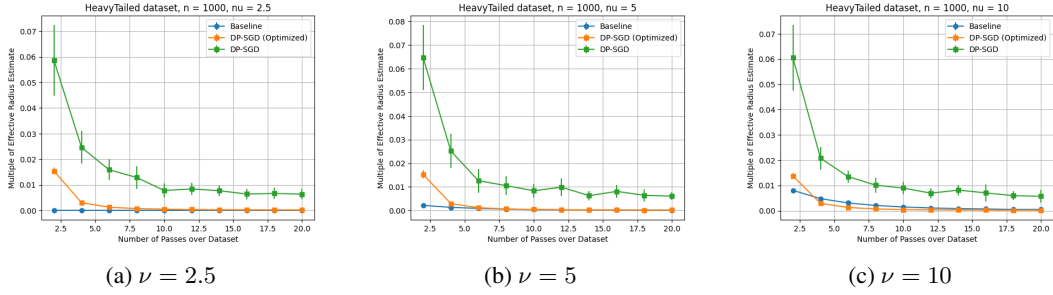


Figure 6: Comparison of DPGD, StableDPSGD, and FixedOrderDPSGD across HeavyTailed data over \mathbb{R}^{50} , varying ν . Plots averaged across 20 trials and standard deviations are reported as error bars.

825 complexities. On the other hand, we expect the gains of methods based on DP-SGD to be larger as
 826 the dataset size and dimension (n, d) grow. There are interesting natural extensions towards realizing
 827 the full potential of private optimization algorithms in practice, such as our Algorithm 3, e.g., the
 828 benefits of using adaptive step sizes or minibatches, which we believe are important and exciting
 829 future directions.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We claimed a theoretical improvement to the private geometric median problem, which we formally prove and provide an experimental evaluation of.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We describe limitations of our theoretical results (e.g., a potentially suboptimal additive runtime term) when introduced in Section 1, and limitations of our experimental evaluation (e.g., accounting for parallelization and heuristics for centerpoint estimation) in Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results except standard ones (e.g., in textbooks) have self-contained proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We give a full description of every experiment in the paper, and provide reproducible code in the supplemental material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: As described above, we provide reproducible code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: As described above, we specify all such details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All reported figures come with at least 20 runs and error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We discuss the computational resources used in our experiments in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We confirm that our research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Our work is primarily theoretical in nature, as it studies a fundamental computational problem. There is likely a significant pipeline for applications of our results to have downstream social implications, which we do not feel is within our scope to comment on in this paper. We remark that ultimately our algorithms are motivated by robustness and privacy desiderata, so they are in line with (potential) positive societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: All of our data is synthetic, so we do not foresee this as a relevant concern.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: We do not use any existing assets in this paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We do not release new assets beyond (simple) synthetic dataset creation.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: These concerns are not relevant to the scope of this paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: These concerns are not relevant to the scope of this paper.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- 1141 • We recognize that the procedures for this may vary significantly between institutions
1142 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
1143 guidelines for their institution.
- 1144 • For initial submissions, do not include any information that would break anonymity (if
1145 applicable), such as the institution conducting the review.

1146 16. **Declaration of LLM usage**

1147 Question: Does the paper describe the usage of LLMs if it is an important, original, or
1148 non-standard component of the core methods in this research? Note that if the LLM is used
1149 only for writing, editing, or formatting purposes and does not impact the core methodology,
1150 scientific rigorousness, or originality of the research, declaration is not required.

1151 Answer: [NA]

1152 Justification: LLMs do not play any significant component in this research.

1153 Guidelines:

- 1154 • The answer NA means that the core method development in this research does not
1155 involve LLMs as any important, original, or non-standard components.
- 1156 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
1157 for what should or should not be described.