

Concentrated Geo-Privacy

Yuting Liang

Hong Kong University of Science and Technology
Hong Kong, China
yliangbs@cse.ust.hk

Ke Yi

Hong Kong University of Science and Technology
Hong Kong, China
yike@cse.ust.hk

ABSTRACT

This paper proposes *concentrated geo-privacy (CGP)*, a privacy notion that can be considered as the counterpart of *concentrated differential privacy (CDP)* for geometric data. Compared with the previous notion of geo-privacy [1, 5], which is the counterpart of standard differential privacy, CGP offers many benefits including simplicity of the mechanism, lower noise scale in high dimensions, and better composability known as *advanced composition*. The last one is the most important, as it allows us to design complex mechanisms using smaller building blocks while achieving better utilities. To complement this result, we show that the previous notion of geo-privacy inherently does not admit advanced composition even using its approximate version. Next, we study three problems on private geometric data: the identity query, k nearest neighbors, and convex hulls. While the first problem has been previously studied, we give the first mechanisms for the latter two under geo-privacy. For all three problems, composability is essential in obtaining good utility guarantees on the privatized query answer.

CCS CONCEPTS

• Security and privacy → Privacy-preserving protocols.

KEYWORDS

privacy, geometric data

ACM Reference Format:

Yuting Liang and Ke Yi. 2023. Concentrated Geo-Privacy. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23)*, November 26–30, 2023, Copenhagen, Denmark. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3576915.3623068>

1 INTRODUCTION

Differential privacy (DP) is the de facto privacy model for protecting personal information. Under DP, the output of a query is perturbed randomly such that its probability distribution does not change much on datasets that are “similar”. In standard DP, this similarity is defined in terms of *neighboring inputs*:

Definition 1 (Differential Privacy [11]). Let $\epsilon, \delta \geq 0$. A randomized mechanism $M : U \rightarrow V$ is (ϵ, δ) -differentially private, or simply

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CCS '23, November 26–30, 2023, Copenhagen, Denmark

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0050-7/23/11...\$15.00
<https://doi.org/10.1145/3576915.3623068>

(ϵ, δ) -DP, if for any neighboring inputs $x \sim x' \in U$ and any measurable $S \subseteq V$,

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

Smaller values of ϵ, δ correspond to better privacy. Typically, ϵ is set to a constant, but may be set smaller for individual mechanisms during compositions (as discussed later). Roughly speaking¹, δ is the probability with which M breaches privacy, so it should be negligible. In particular, the mechanism is said to satisfy *pure-DP* if $\delta = 0$; the $\delta > 0$ case is then called *approximate-DP*.

Depending on how the neighboring relationship \sim is defined, we arrive at different DP variants. In the *central* model of DP, $x \sim x'$ iff they differ by a single record. This models the situation where there is a trusted data curator who collects and publishes privatized query results. Alternatively, if we set $x \sim x'$ between all pairs of distinct inputs, then the model becomes what is known as *local DP* [15]. This corresponds to the scenario where each data owner privatizes his/her input before sending it to the curator, who is potentially malicious.

In local DP, any single message satisfying Definition 1 has little utility, as the output distributions on any two inputs are required to be indistinguishable. So all mechanisms under local DP has to aggregate messages from a large number of users for the result to be useful. However, for many applications involving geometric data, e.g., the current or past locations of the user, we are interested in obtaining some privatized version of the data on a per user basis. The standard DP definition is thus not appropriate in such scenarios.

1.1 From DP to GP

In standard DP², two inputs x, x' are either neighbors or not. Such a binary relationship cannot capture the more quantitative relationships on geometric data. In the example above, requiring two faraway points x, x' (e.g., in different cities) to also have indistinguishable output distributions is likely an overkill. In seeing this, Andrés et al. [1] and Chatzikokolakis et al. [5] extend the neighboring relationship to a metric space (U, dist) , and stipulate that privacy should degrade gracefully as $\text{dist}(x, x')$ increases. They have only considered the pure case $\delta = 0$, but their definition extends to the approximate version, i.e., $\delta \geq 0$, in the natural way:

Definition 2 (Geo-Privacy³ [1, 5]). Let $\epsilon, \delta \geq 0$. A randomized mechanism $M : U \rightarrow V$ is (ϵ, δ) -geo-private, or simply (ϵ, δ) -GP, if

¹To be precise, the probabilistic interpretation is slightly stronger than the stated definition, where it implies the definition and is implied by it up to a small loss in the privacy parameters [23, 40].

²For an account of differences between standard (ϵ, δ) -DP and other DP-related definitions, the reader is referred to [10].

³This notion is called *geo-indistinguishability* in [1] and d_χ -privacy in [5], respectively.

for any $x, x' \in U$ and any measurable $S \subseteq V$,

$$\Pr[M(x) \in S] \leq e^{\varepsilon \cdot \text{dist}(x, x')} \cdot \Pr[M(x') \in S] + \delta.$$

Note that $(\varepsilon, 0)$ -GP incorporates both central DP and local DP by choosing an appropriate distance function $\text{dist}(\cdot, \cdot)$. To recover the former, we use the Hamming metric⁴; for the latter, we use the discrete metric $\text{dist}(x, x') = 1$ for all $x \neq x'$. However, the power of GP lies in the flexibility in choosing a distance function suitable for the input domain U . For example, when $U = \mathbb{R}^d$, the Euclidean metric $\text{dist}(x, x') := \|x - x'\|_2$ is arguably a more appropriate distance function than the Hamming or the discrete metric. Note that in terms of privacy, GP is at least as strong as DP for all x, x' such that $\text{dist}(x, x') \leq 1$, while weaker as $\text{dist}(x, x')$ becomes larger than 1. This is also why the ε parameter in GP measures the privacy loss per unit distance, as explained in [1, 5]. However, it is exactly due to this weakening of privacy (for faraway inputs) that allows GP to enjoy higher utility. In particular, due to the use of the discrete metric in local DP, the data collected from any single user has little utility by definition; only the aggregated result from many users may yield some useful information. In contrast, GP mechanisms offer much better utility. For example, the noise scale of the Laplace mechanism, the canonical GP mechanism, is⁵ $\tilde{O}(d)$ for the identity query (see Section 2.3 for details).

1.2 Our Proposal: CGP

We propose a new definition of geo-privacy using Rényi divergences, which we call *concentrated geo-privacy (CGP)*:

Definition 3 (Concentrated Geo-Privacy). Let $\rho \geq 0$. A mechanism $M : U \rightarrow V$ satisfies ρ -concentrated-geo-privacy, or simply ρ -CGP, if for any $x, x' \in U$ and all $\alpha > 1$

$$D_\alpha(M(x) \| M(x')) \leq \alpha \rho \cdot \text{dist}(x, x')^2.$$

In the definition above, $M(x)$ and $M(x')$ are the output distributions of the mechanism on x and x' , respectively, and $D_\alpha(\cdot \| \cdot)$ denotes the Rényi divergence of order α (see Section 2.2 for the detailed definition). Our definition can be considered as the counterpart and generalization of *concentrated differential privacy (CDP)* [3] in the geometric setting. Note that the definition of CGP stipulates that the Rényi divergence grows quadratically as $\text{dist}(x, x')$. This is due to the relationship between the privacy parameter ε used in GP and the ρ used in CGP (see Lemma 3.6 and 3.7). More precisely, we show that the privacy of ρ -CGP is at least as strong as that of $(\tilde{O}(\sqrt{\rho}), \delta)$ -GP, except for faraway points (see Lemma 3.7 for the formal statement). Correspondingly, ρ measures the privacy loss per unit distance squared.

Compared with GP, CGP offers the following benefits:

- (1) The canonical mechanism under CGP is the Gaussian mechanism, which is much easier to implement than the Laplace mechanism in $d \geq 2$ dimensions.
- (2) The noise scale of the Gaussian mechanism is $\tilde{O}(\sqrt{d})$, while that of the Laplace mechanism is $\tilde{O}(d)$.

⁴This follows from the group privacy property of DP [13]. However, (ε, δ) -GP under the Hamming metric is stronger than (ε, δ) -DP, as the group size allowed in DP group privacy is limited to $O(\log(1/\delta)/\varepsilon)$ for $\delta > 0$.

⁵The $\tilde{O}(\cdot)$ notation suppresses dependency on ε, ρ , and polylogarithmic factors in $n, d, 1/\delta, 1/\beta$.

- (3) Gaussians have many nice properties, in particular, linearity. Many geometric measurements are assumed to have some Gaussian noise already. Then the linearity of Gaussians makes it easy to add more noise, if needed, to achieve any desired level of privacy.
- (4) CGP admits *advanced composition*, i.e., the noise scale is proportional to \sqrt{k} as opposed to k for GP, in any adaptive k -fold composition.

Point (4) above requires some elaboration. Composition theorems are an important tool for privacy-preserving data analytics, as they allow us to ask multiple queries, possibly adaptively, on the same input under a given privacy budget. Furthermore, it makes the modular design of complex mechanisms easy, as we can combine smaller building blocks while tracking the overall privacy loss. Indeed, one of the most important reasons why $(\varepsilon, \delta > 0)$ -DP is of interest is that it allows advanced composition [13], while basic composition (i.e., noise scale grows linearly in k) is the best one can do under pure-DP.

However, prior work on GP [1, 5] has omitted the $\delta > 0$ case in Definition 2. In fact, our initial goal was to derive an advanced composition theorem for GP. In failing to do so, we actually proved that basic composition is the best possible for GP under the Euclidean metric, *even for* $\delta > 0$ (see Theorem 3.5). This may have explained why [1, 5] did not consider the $\delta > 0$ case.

The intuitive reason why (ε, δ) -DP admits advanced composition, but (ε, δ) -GP (under the Euclidean metric) does not, is the following. The advanced composition theorem for DP [13] actually consists of two terms⁶:

$$\varepsilon = \sqrt{2k \log(1/\delta)} \cdot \varepsilon' + k\varepsilon'(\varepsilon' - 1), \quad (1)$$

where ε' is the privacy parameter for each of the k mechanisms and ε, δ are the privacy parameters of the composed mechanism. For the typical parameter regime $\varepsilon \leq O(1)$, we need to set $\varepsilon' = O\left(\varepsilon/\sqrt{k \log(1/\delta)}\right)$, so that the second term is dominated by the first, yielding the square-root growth often quoted for advanced composition: $\varepsilon = \tilde{O}(\sqrt{k} \cdot \varepsilon')$. However, in Definition 2 where ε is replaced by $\varepsilon \cdot \text{dist}(x, x')$, (1) should hold for all values of $\varepsilon \cdot \text{dist}(x, x')$, which is unbounded. Thus, the second term will eventually dominate as x, x' get farther away from each other, breaking this square-root (actually, any sub-linear) relationship. It is possible to fix this issue by adding a third privacy parameter Δ to limit $\text{dist}(x, x')$ (see Definition 6), but our CGP definition using Rényi divergences solves the problem in a more elegant way using a single parameter ρ .

1.3 Problems under GP and CGP

In Section 4, we develop GP and CGP mechanisms for queries where the input is an n -tuple of points in \mathbb{R}^d . Such a tuple may represent a trajectory of a user, a collection of points visited, or the features of objects belonging to the user. This setting corresponds to the local model, where each user privatizes his/her data before sending it to a malicious data curator. Prior works under GP [1, 25, 39] have only studied the identity query or applied post-processing to it. In this paper, in addition to the identity query, we have also designed algorithms for the k nearest neighbor (k NN) and the convex hull

⁶We use \log to denote the natural logarithm with base e throughout this paper.

problem. Both these problems can be solved using the identity query with post-processing, but our new algorithms offer much better utility. Our algorithms can be made to satisfy either GP or CGP, but thanks to the benefits of CGP, there are some polynomial improvements in the utility as we switch from GP to CGP:

- (1) For the identity query, the GP algorithm has an error of $\tilde{O}(dn)$, while the CGP algorithm has error $\tilde{O}(\sqrt{dn})$.
- (2) For the k NN problem, the GP algorithm has an error of $\tilde{O}(k)$, while the CGP algorithm has error $\tilde{O}(\sqrt{k})$. Both error bounds hold for any d .
- (3) For the two-dimensional convex hull problem, the GP algorithm has an error of $\tilde{O}(\sqrt{\omega(x)}+1)$, while the CGP algorithm has an error of $\tilde{O}(\omega(x)^{1/3}+1)$, where $\omega(x)$ denotes the diameter (the farthest distance between any two points) in x . The gap reduces in higher dimensions, though.

Finally, we performed a set of experiments using trajectory data. The results have confirmed the utility improvement of our k NN and convex hull algorithms over the baseline method, as well as the difference between the GP and the CGP versions.

1.4 Related Work

While [1, 5] are the most related to our work, privacy over geometric data has attracted much attention in the literature, with the following works also relevant. [25] privatizes each location on the trajectory under GP, and applies post-processing steps on the privatized trajectory for travel time prediction. [39] privatizes a collection of points in \mathbb{R}^2 and performs k -means clustering on the privatized data. [41] proposes a variant of DP by defining the neighboring relationship between two points within a given distance of r . However, their definition does not impose any relationship between the privacy and the distance, as GP and CGP do. Another definition more closely related to standard DP is differential privacy on δ -location set [38], which requires a location set X_δ to be given in advance at each timestamp t . X_δ is the set of locations such that with probability $1 - \delta$, the user is at one of the locations in X_δ at timestamp t . X_δ defines neighborhood in the sense that every pair of locations in X_δ needs to satisfy the standard DP requirement.

Many problems on geometric data have been studied under the central DP model. However, for the privatized output to have any utility, the query must have low *sensitivity*, i.e., the maximum amount of change in the query output under an (arbitrary) change in one record. Examples of such queries include range counting⁷ [20, 31, 32], Tukey depth⁸ [2, 16, 22], and the mean⁹ [19, 24]. However, problems like k NN or convex hull have unbounded sensitivity: For the former, observe that a k NN query degenerates into the maximum problem in one dimension when the query point is at $+\infty$, which already has unbounded sensitivity. For the latter, moving a point to ∞ will cause the convex hull to enlarge infinitely. Note that [2, 22] only return some point inside the convex hull while

⁷Given a collection of points and a range space, e.g., all halfspaces or axis-parallel rectangles, the problem is to release a privatized data structure from which the number of points inside each range can be approximately counted.

⁸Given a set of points, the Tukey depth of a given point p is the minimum number of points on one side of a hyperplane that contains p . The vertices on the convex hull thus has Tukey depth 1.

⁹For the mean problem to have low sensitivity, we need to further assume that the domain is bounded, e.g., the unit ball.

[16] aims at producing an approximate convex hull in terms of the Tukey depth. Their algorithms do not return an approximate convex hull in terms of geometric distances, which is only possible under GP or CGP, which restrict (more precisely, lower the privacy requirement of) such large changes in the locations of the points.

There are several works for privatizing trajectory data in the central DP model [6, 17, 18]. There, the input is a database of trajectories and two inputs are considered neighbors if they differ in a single trajectory. These works aim to generate a collection of synthetic trajectories resembling the original collection in distribution. To this end, they construct a privatized representation of the empirical distribution of chains of locations via private spatial decomposition techniques [8] which ultimately privatizes counts [12]. The problem has also been studied in the local DP model where the input consists of a single trajectory [9]. However, their mechanisms do not actually satisfy the local DP requirement, since as mentioned, no mechanism can offer meaningful utility under local DP. By relaxing the privacy model to GP or CGP, our algorithms return privatized trajectories with meaningful utility guarantees.

2 PRELIMINARIES

2.1 Notation

Any (nontrivial) mechanism $M : U \rightarrow V$ under differential privacy must be randomized. More formally, such a randomized mechanism is a map $\mathcal{M} : U \rightarrow \mathbb{D}(V)$, where $\mathbb{D}(V)$ denotes the space of all probability distributions over V . We shall use uppercase letters in scripts to denote probability distributions, e.g., $\mathcal{M}(x)$ denotes the output probability distribution of mechanism \mathcal{M} on input x . The same letter in uppercase (without scripts) represents a random variable drawn from this distribution, i.e., $M(x)$ is a random variable drawn from $\mathcal{M}(x)$. The same letter in lowercase will be used to denote the corresponding probability density function (pdf), i.e., the pdf of $\mathcal{M}(x)$ is $m(x)(\cdot)$.

2.2 Differential Privacy

In addition to Definition 1, another popular variant for DP is based on Rényi divergences:

Definition 4 (Rényi Divergences [34, 37]). Let \mathcal{P}, \mathcal{Q} be distributions on domain R with pdf $p(\cdot)$ and $q(\cdot)$, respectively. The Rényi divergence of order $\alpha \in (0, 1) \cup (1, \infty)$ is defined as

$$D_\alpha(\mathcal{P} \parallel \mathcal{Q}) = \frac{1}{\alpha - 1} \log \left(\int_R p(y)^\alpha q(y)^{1-\alpha} dy \right).$$

The max-divergence is

$$D_\infty(\mathcal{P} \parallel \mathcal{Q}) = \lim_{\alpha \rightarrow \infty} D_\alpha(\mathcal{P} \parallel \mathcal{Q}) = \sup_{y \in R} \log \left(\frac{p(y)}{q(y)} \right).$$

The following properties of Rényi divergences are well known:

LEMMA 2.1 (PROPERTIES OF RÉNYI DIVERGENCES [37]). *For $\alpha \geq 1$, Rényi divergences satisfy the following properties:*

- (1) [Monotonicity] $D_\alpha(\mathcal{P} \parallel \mathcal{Q}) \leq D_{\alpha'}(\mathcal{P} \parallel \mathcal{Q})$ for $1 \leq \alpha \leq \alpha' \leq \infty$.
- (2) [Additivity] $D_\alpha(\mathcal{P}_1 \times \dots \times \mathcal{P}_k \parallel \mathcal{Q}_1 \times \dots \times \mathcal{Q}_k) = \sum_{j=1}^k D_\alpha(\mathcal{P}_j \parallel \mathcal{Q}_j)$ for pairs of probability distributions $(\mathcal{P}_j, \mathcal{Q}_j)$ on σ -algebra \mathcal{F}_j , $j \in [k]$.

- (3) [Data Processing Inequality] $D_\alpha(P_{|\mathcal{G}}\|Q_{|\mathcal{G}}) \leq D_\alpha(\mathcal{P}\|Q)$ for any sub- σ -algebra $\mathcal{G} \subseteq \mathcal{F}$.
- (4) [Gaussian Distributions] $D_\alpha(\mathcal{N}(\mu_1, \sigma^2 I_{d \times d})\|\mathcal{N}(\mu_2, \sigma^2 I_{d \times d})) = \frac{\alpha \|\mu_1 - \mu_2\|_2^2}{2\sigma^2}$ for $\mu_1, \mu_2 \in \mathbb{R}^d$.

Concentrated differential privacy is defined in terms of Rényi divergences:

Definition 5 (Concentrated Differential Privacy¹⁰ [3]). Let $\rho \geq 0$. A mechanism $M : U \rightarrow V$ satisfies concentrated differential privacy, denoted ρ -CDP, if for any neighboring inputs $x \sim x' \in U$, any measurable $S \subseteq V$, and all $\alpha > 1$,

$$D_\alpha(M(x)\|M(x')) \leq \alpha\rho.$$

It is known that the privacy guarantee of CDP sits between that of pure and approximate DP [3]:

LEMMA 2.2 (RELATIONSHIP BETWEEN DP DEFINITIONS [3]). Let M be a randomized mechanism. Then

- (1) if M is $(\epsilon, 0)$ -DP, then it is also ρ -CDP where $\rho = \frac{1}{2}\epsilon^2$;
- (2) if M is ρ -CDP, then it is also (ϵ, δ) -DP, where $\epsilon = \rho + \sqrt{4\rho \log(1/\delta)}$ for all $\delta > 0$.

The above relationships imply that ϵ and ρ are related quadratically in the typical parameter regime $\epsilon \leq O(1)$.

We can also characterize (ϵ, δ) -DP in terms of the max-divergence.

LEMMA 2.3 (CHARACTERIZATION OF DIFFERENTIAL PRIVACY [13]). Let $\epsilon, \delta \geq 0$. A randomized mechanism M is (ϵ, δ) -differentially private iff

$$D_\infty^\delta(M(x)\|M(x')) \leq \epsilon \text{ and } D_\infty^\delta(M(x')\|M(x)) \leq \epsilon,$$

where

$$D_\infty^\delta(\mathcal{P}\|Q) := \sup_{S \subseteq \mathcal{R}, \Pr[P \in S] > \delta} \log \left(\frac{\Pr[P \in S] - \delta}{\Pr[Q \in S]} \right),$$

for all $x \sim x'$.

For a query $g : U \rightarrow V = \mathbb{R}^d$, the simplest mechanism for achieving DP is to add noise drawn from a distribution with scale proportional to its *sensitivity* Δg to each dimension of the true query answer:

LEMMA 2.4 (CANONICAL DP MECHANISMS [3, 11]). Let $g : U \rightarrow \mathbb{R}^d$ such that $\Delta g := \sup_{x \sim x'} \|g(x) - g(x')\|_p < \infty$ exists. Then the mechanism $M(x) := g(x) + b \cdot [Z_1, \dots, Z_d]^T$ is

1. [Laplace mechanism] ϵ -DP for $p = 1$, where $b = \frac{\Delta g}{\epsilon}$ and $Z_j \sim \text{Lap}(1)$ for $j \in [d]$;
2. [Gaussian mechanism] ρ -CDP for $p = 2$, where $b = \frac{\Delta g}{\sqrt{2\rho}}$ and $Z_j \sim \mathcal{N}(0, 1)$ for $j \in [d]$.

Note that the noise scales are proportional to $1/\epsilon$ and $1/\sqrt{\rho}$, respectively, echoing the quadratic relationship between ϵ and ρ .

LEMMA 2.5 (COMPOSITIONS OF DP MECHANISMS [3, 13]). Let M_1, \dots, M_k be DP mechanisms. Let M be a k -fold adaptive composition of the M_j 's. Then M is

- (1) (ϵ, δ) -DP, where $\epsilon := \sum_{j \in [k]} \epsilon_j$ and $\delta := \sum_{j \in [k]} \delta_j$, if each M_j is (ϵ_j, δ_j) -DP for $j \in [k]$;
- (2) $(\epsilon, k\delta' + \delta)$ -DP for all $\delta > 0$, where $\epsilon := \sqrt{2k \log(1/\delta)}\epsilon' + k\epsilon'(e^{\epsilon'} - 1)$, if each M_j is (ϵ', δ') -DP for $j \in [k]$;
- (3) ρ -CDP, where $\rho := \sum_{j \in [k]} \rho_j$, if each M_j is ρ_j -CDP for $j \in [k]$.

Note that composition theorems are usually applied in the opposite direction, namely, for a given total privacy budget ϵ , how to allocate it to the k mechanisms. Thus, the factor- k growth in ϵ in Lemma 2.5 (1) implies that the noise scale of each mechanism is k times larger than that from a single invocation of the mechanism. For this reason, it is often referred to as *basic composition*. In contrast, (2) is termed *advanced composition*, as it yields a (quasi) square root growth. However, advanced composition leads to a $\delta > 0$ for the composed mechanism, even if each individual M_j satisfies pure DP. In (3), the privacy loss grows linearly in terms of ρ . But since $\rho = \tilde{\Theta}(\epsilon^2)$, the growth in terms of ϵ , or equivalently the noise scale, is still square root. In fact, by combining with Lemma 2.2, when each M_j is the Laplace (or any pure-DP) mechanism or the Gaussian mechanism, the composition result implied by (3) is no worse than that of (2).

2.3 Geo-Privacy

Now suppose the output space is also a metric equipped with distance function $\text{dist}_V(\cdot, \cdot)$. A query $g : U \rightarrow V$ is K -Lipschitz if $\text{dist}_V(g(x), g(x')) \leq K \cdot \text{dist}(x, x')$ for all $x, x' \in U$. A canonical mechanism for achieving ϵ -GP is the following, which can be considered as the instantiation of the *exponential mechanism* [28] with $\text{dist}_V(g(x), y)$ as the utility score for $y \in V$:

LEMMA 2.6 ([5]). Let $g : U \rightarrow V$ be K -Lipschitz. The mechanism that, on input x , draws a $y \in V$ from a distribution with pdf $\propto e^{-\frac{\epsilon}{K} \cdot \text{dist}_V(g(x), y)}$, is ϵ -GP.

However, Lemma 2.6 does not lend itself to an efficient implementation. Andrés et al. [1] considered the special case where $U = V = \mathbb{R}^2$ (under the Euclidean metric) and $g(x) = x$ (which is 1-Lipschitz), namely, one would like to privatize a single point in \mathbb{R}^2 , and showed that Lemma 2.6 instantiates into the planar Laplace mechanism [1]. In Appendix A, we further show how this can be extended to d dimensions.

By arguments similar to those for standard DP, a basic composition theorem can be proved for GP.

LEMMA 2.7 (BASIC COMPOSITION FOR GP¹¹ [1]). Let M_j be (ϵ_j, δ_j) -GP for $j = 1, \dots, k$, and let M be a k -fold adaptive composition of the M_j 's. Then M is (ϵ, δ) -GP, where $\epsilon := \sum_{j \in [k]} \epsilon_j$ and $\delta := \sum_{j \in [k]} \delta_j$.

3 CONCENTRATED GEO-PRIVACY

The definition of CGP is given in Definition 3. In this section, we analyze its various properties.

¹⁰More commonly known as *zero-concentrated differential privacy*. Note that there is a related two-parameter definition (μ, τ) -mCDP, also known by the name *mean-concentrated differential privacy* [14], where $(\tau^2/2, \tau)$ -mCDP implies $\tau^2/2$ -CDP [3].

¹¹[1] only proved the theorem for the pure case $\delta_j = 0$, but extension to $\delta_j > 0$ is straightforward.

3.1 The Gaussian Mechanism

The Gaussian mechanism is the canonical mechanism for CDP. We can extend it to CGP for queries $g : U \rightarrow \mathbb{R}^d$ equipped with the Euclidean metric after factoring in the Lipschitz constant:

THEOREM 3.1. *Let $g : U \rightarrow \mathbb{R}^d$ be a K -Lipschitz function. Then the mechanism G defined by $G(x) := g(x) + \frac{K}{\sqrt{2\rho}}Z$, where $Z \sim \mathcal{N}(0, I_{d \times d})$, is ρ -CGP.*

PROOF. Let $x, x' \in U$, let $\mathcal{G}(x)$ and $\mathcal{G}(x')$ denote the distributions of $G(x)$ and $G(x')$, respectively. For $\alpha > 1$,

$$\begin{aligned} D_\alpha(\mathcal{G}(x) \parallel \mathcal{G}(x')) &= D_\alpha\left(\mathcal{N}\left(g(x), \frac{K^2}{2\rho}I_{d \times d}\right) \parallel \mathcal{N}\left(g(x'), \frac{K^2}{2\rho}I_{d \times d}\right)\right) \\ &= \frac{\alpha\rho}{K^2} \|g(x) - g(x')\|_2^2 \quad (\text{By Lemma 2.1 (4)}) \\ &\leq \alpha\rho \text{dist}(x, x')^2. \end{aligned}$$

□

Compared with the d -dimensional planar Laplace distribution, $\mathcal{N}(0, I_{d \times d})$ is a distribution much easier to draw from: just generate each coordinate from the standard Gaussian. Furthermore, the noise scale (i.e., $\|G(x) - g(x)\|_2$) of the Gaussian mechanism is $\tilde{O}(\sqrt{d})$ with high probability assuming $K = 1$, while that of the Laplace mechanism is $\tilde{O}(d)$ (see Appendix A).

3.2 Compositions

We can show that our notion of CGP enjoys the same (adaptive) composition property as CDP. We only state and prove the result for composing two mechanisms (proof follows similar arguments as those in [29] and is included in Appendix B); a simple induction leads to k -fold compositions. Consequently, by Lemma 3.1, the noise scale of each mechanism is \sqrt{k} times larger than that on a single mechanism.

THEOREM 3.2. *Let $M_1 : U \rightarrow V_1$ be ρ_1 -CGP, let $M_2 : U \times V_1 \rightarrow V_2$ be ρ_2 -CGP w.r.t. its first argument. Then the mechanism $M : U \rightarrow V_1 \times V_2$ defined by $M(x) = (M_1(x), M_2(x, M_1(x)))$ is $(\rho_1 + \rho_2)$ -CGP.*

Note that if $\rho_2 = 0$ in Lemma 3.2, which means that $M_2(x, y_1)$ does not depend on x , then the lemma degenerates into the post-processing property of CGP, which also follows from property (3) of Lemma 2.1.

A negative result on the composability of (ϵ, δ) -GP. It turns out that basic composition (Lemma 2.7) is the best we can do for GP, i.e., the privacy loss in a k -fold composition has to grow linearly in k , even if the composed mechanism is allowed to have $\delta > 0$. This stands in contrast with (ϵ, δ) -DP, where advanced composition theorems are known to achieve an $\tilde{O}(\sqrt{k})$ growth¹² for any negligible $\delta > 0$.

To formalize this result, we first note that (ϵ, δ) -GP can also be characterized by Rényi divergences, by replacing ϵ with $\epsilon \cdot \text{dist}(x, x')$ in the characterization of standard DP:

¹²Thus, composition results tighter than that of Dwork and Roth [13] under standard (ϵ, δ) -DP (e.g., [21, 30]) also do not have counterparts under (ϵ, δ) -GP.

LEMMA 3.3. *A mechanism $M : U \rightarrow V$ is (ϵ, δ) -GP iff for all $x, x' \in U$, $D_\infty^\delta(\mathcal{M}(x) \parallel \mathcal{M}(x')) \leq \epsilon \cdot \text{dist}(x, x')$ and $D_\infty^\delta(\mathcal{M}(x') \parallel \mathcal{M}(x)) \leq \epsilon \cdot \text{dist}(x, x')$.*

For a negative result, it suffices to consider the special case $U = V = \mathbb{R}$ with $g(x) = x$, where we use the canonical GP mechanism in Lemma 2.6. In this case, on every input x , the mechanism returns an output $M(x)$ drawn from a distribution $\mathcal{M}(x)$ with pdf $m(x)(y) \propto e^{-\epsilon \cdot |y-x|}$.

LEMMA 3.4. *Let each $M_j : \mathbb{R} \rightarrow \mathbb{R}$ be the ϵ -GP mechanism defined above. Define $M = (M_1(x), \dots, M_k(x))$ for any $k \geq 2$. Then for any $\epsilon > 0$, $0 \leq \delta \leq \frac{1}{4}$, $0 < \lambda < 1$, $x \in \mathbb{R}$, there is $x' \in \mathbb{R}$ such that $D_\infty^\delta(\mathcal{M}(x) \parallel \mathcal{M}(x')) > (1 - \lambda)k\epsilon \cdot \|x - x'\|$.*

PROOF. Let $x' = x - \Delta$ where $\Delta := \frac{2}{\epsilon\lambda} \log(k)$. Let s be a point between x' and x given by $s := x - \frac{\lambda}{2}\Delta$, let $S^* := \{(y_1, \dots, y_k) \in \mathbb{R}^k : y_j \geq s \forall j \in [k]\}$. Let $Y = (Y_1, \dots, Y_k)$, $Y' = (Y'_1, \dots, Y'_k)$ be random vectors corresponding to $M(x)$ and $M(x')$, respectively. Then $\Pr[Y \in S^*] = \prod_{j \in [k]} \Pr[Y_j \geq s] = (1 - \frac{1}{2}e^{-\epsilon \frac{\lambda}{2}\Delta})^k = (1 - \frac{1}{2k})^k \geq \frac{1}{2}$ and $\Pr[Y' \in S^*] = \prod_{j \in [k]} \Pr[Y'_j \geq s] = (\frac{1}{2}e^{-\epsilon(1-\frac{\lambda}{2})\Delta})^k$. Thus,

$$\begin{aligned} D_\infty^\delta(\mathcal{M}(x) \parallel \mathcal{M}(x')) &= \max_{S \subseteq \mathbb{R}^k : \Pr[Y \in S] > \delta} \log\left(\frac{\Pr[Y \in S] - \delta}{\Pr[Y' \in S]}\right) \\ &\geq \log\left(\frac{\Pr[Y \in S^*] - \delta}{\Pr[Y' \in S^*]}\right), \\ e^{D_\infty^\delta(\mathcal{M}(x) \parallel \mathcal{M}(x'))} &\geq \frac{\Pr[Y \in S^*] - \delta}{\Pr[Y' \in S^*]} = \frac{(1 - \frac{1}{2}e^{-\epsilon \frac{\lambda}{2}\Delta})^k - \delta}{(\frac{1}{2}e^{-\epsilon(1-\frac{\lambda}{2})\Delta})^k} \\ &\geq \frac{1/2 - \delta}{(1/2)^k e^{-k\epsilon(1-\lambda/2)\Delta}} \\ &\geq \frac{1}{4} 2^k e^{k\epsilon(1-\lambda/2)\Delta} > e^{k\epsilon(1-\lambda)\|x-x'\|}. \end{aligned}$$

□

Combining Lemma 3.3 and 3.4, we obtain:

THEOREM 3.5. *For any $\epsilon > 0$, $0 \leq \delta \leq \frac{1}{4}$, $0 < \lambda < 1$, and any $k \geq 2$, there is an $(\epsilon, 0)$ -GP mechanism M such that its k -fold composition does not satisfy $((1 - \lambda)k\epsilon, \delta)$ -GP.*

As a negative result, Theorem 3.5 does not preclude the possibility that a particular GP mechanism in a particular metric space may have better composability. In fact, a general impossibility result is not true, since a GP mechanism under the discrete metric satisfies local DP, for which advanced composition holds. Nevertheless, the negative result means that one cannot use GP mechanisms as black boxes to compose a more complex GP mechanism with a privacy loss better than linear. In particular, this is the case for the canonical GP mechanism in Lemma 2.6.

3.3 Relationships between GP and CGP

In terms of the relationship between GP and CGP, we have a partial analogue of Lemma 2.2 in geo-privacy.

LEMMA 3.6. *Any $(\epsilon, 0)$ -GP mechanism is also $\frac{\epsilon}{2}$ -CGP.*

PROOF. It has been shown in [3] that given $e^{-\varepsilon_0} \leq \frac{p(y)}{q(y)} \leq e^{\varepsilon_0}$ for all y , then $e^{(\alpha-1)D_\alpha(\mathcal{P}\|\mathcal{Q})} \leq e^{(\alpha-1)\alpha\frac{\varepsilon_0^2}{2}}$ for all $\alpha > 1$. Let $x, x' \in U$. Let $\varepsilon_0 := \varepsilon \cdot \text{dist}(x, x')$, then $e^{-\varepsilon_0} \leq \frac{m(x)(y)}{m(x')(y)} \leq e^{\varepsilon_0}$, which implies $e^{(\alpha-1)D_\alpha(\mathcal{M}(x)\|\mathcal{M}(x'))} \leq e^{(\alpha-1)\alpha\frac{\varepsilon_0^2}{2}} = e^{(\alpha-1)\alpha\frac{\varepsilon^2 \cdot \text{dist}(x, x')^2}{2}}$. I.e. $D_\alpha(\mathcal{M}(x)\|\mathcal{M}(x')) \leq \alpha\frac{\varepsilon^2}{2} \cdot \text{dist}(x, x')^2$ for all $\alpha > 1$. \square

However, ρ -CGP does not imply $\tilde{O}(\sqrt{\rho}, \delta)$ -GP: If it did, we would be able to compose k ε -GP mechanisms, which also satisfy $O(\varepsilon^2)$ -CGP by Lemma 3.6, into an $O(k\varepsilon^2)$ -CGP mechanism by Theorem 3.2, hence an $(\tilde{O}(\sqrt{k\varepsilon}, \delta)$ -GP mechanism, violating Theorem 3.5. Fundamentally, it is exactly due to this weakening in privacy that gives CGP the better composability.

Nevertheless, below we show that this weakening in privacy only happens for faraway inputs. Specifically, we show that CGP is at least as strong as the following natural relaxation of (ε, δ) -GP:

Definition 6 ($(\varepsilon, \delta, \Delta)$ -GP). Let $\varepsilon, \delta \geq 0$ and $\Delta > 0$. A mechanism $M : U \rightarrow V$ is $(\varepsilon, \delta, \Delta)$ -GP, if for any measurable $S \subseteq V$ and any $x, x' \in U$ satisfying $\text{dist}(x, x') \leq \Delta$,

$$\Pr[M(x) \in S] \leq e^{\varepsilon \cdot \text{dist}(x, x')} \Pr[M(x') \in S] + \delta.$$

Note that the original (ε, δ) -GP definition is the special case where $\Delta = \infty$.

LEMMA 3.7. Any mechanism M that is ρ -CGP is also $(\varepsilon, \delta, \Delta)$ -GP, for any $\varepsilon, \delta, \Delta$ such that $\varepsilon \geq \rho\Delta + 2\sqrt{\rho} \log(1/\delta)$.

Plugging in typical values of $\varepsilon = 1, \delta = 10^{-10}$, Lemma 3.7 implies that the privacy of CDP with $\rho \approx 0.01$ is at least as strong as that of $(\varepsilon, \delta, \Delta \approx 10)$ -GP. Note that for $\text{dist}(x, x') \geq 10$, the privacy provided by GP between $M(x)$ and $M(x')$ is already negligible: The adversary can distinguish between x and x' with probability $1 - e^{-10}$. So weakening it further will not introduce any noticeable differences. The proof of Lemma 3.7 uses a similar derivation as that in [3] and can be found in Appendix C, along with an illustration of the relationship in Fig. 6.

An additional note on (ε, δ) -GP vs. (ε, δ) -DP. One of the main advantages of standard $(\varepsilon, \delta > 0)$ -DP is that it allows advanced composition of $(\varepsilon, \delta \geq 0)$ -DP mechanisms, which is not possible for (ε, δ) -GP in general as we've shown in the negative result above. In addition, standard $(\varepsilon, \delta > 0)$ -DP admits natural mechanisms such as the Gaussian mechanism. Below we give a similar construction as in the proof of Lemma 3.4 which shows that adding constant Gaussian noise will not achieve (ε, δ) -GP. Thus, currently we do not know of any mechanism that satisfies $(\varepsilon, \delta > 0)$ -GP but not ε -GP.

LEMMA 3.8. Fix $\varepsilon > 0$ and $\frac{1}{4} > \delta \geq 0$. Let $M : \mathbb{R} \rightarrow \mathbb{R}$ be the mechanism defined by $M(x) := x + \sigma Z$, where $Z \sim \mathcal{N}(0, 1)$ and $\sigma = \sigma(\varepsilon, \delta) > 0$ is any constant which may depend on ε and δ . Fix $x \in \mathbb{R}$. Then there is $x' \in \mathbb{R}$ such that $D_\infty^\delta(\mathcal{M}(x)\|\mathcal{M}(x')) > \varepsilon\|x - x'\|$.

PROOF. Let $x' = x - \Delta$ for some $\Delta > 0$ to be decided, let $S^* := \{y \in \mathbb{R} : y \geq x\}$. Then $\Pr[M(x) \in S^*] = \frac{1}{2}$. Let $Z \sim \mathcal{N}(0, \sigma^2)$, then $\Pr[Z \geq \Delta] \leq e^{-\frac{\Delta^2}{2\sigma^2}}$ by Lemma E.2. Thus, $\Pr[M(x') \in S^*] =$

$\Pr[M(x') - x' \geq \Delta] = \Pr[Z \geq \Delta] \leq e^{-\frac{\Delta^2}{2\sigma^2}}$. Now choose $\Delta > \max(4\varepsilon\sigma^2, \frac{\ln 4}{\varepsilon})$, then

$$\begin{aligned} e^{D_\infty^\delta(\mathcal{M}(x)\|\mathcal{M}(x'))} &\geq \frac{\Pr[M(x) \in S^*] - \delta}{\Pr[M(x') \in S^*]} \\ &\geq \frac{1/2 - \delta}{e^{-\frac{\Delta^2}{2\sigma^2}}} \geq \frac{1}{4} e^{\frac{\Delta^2}{2\sigma^2}} > \frac{1}{4} e^{2\varepsilon\Delta} > e^{\varepsilon\|x-x'\|}, \end{aligned}$$

where the second last inequality is due to $\Delta > 4\varepsilon\sigma^2$ and the last inequality is due to $\Delta > \frac{\ln 4}{\varepsilon}$. \square

4 APPLICATIONS

So far we've only described basic mechanisms for GP and CGP, where both require the function to be Lipschitz. In general, differentiable functions with bounded first-order derivatives are Lipschitz. Some examples are: point functions such as computing the mean, median and general linear maps; functions which deal with distance such as (min or max) distance or projection of a point to a line or collection of points (see also Lemma E.1). However, many functions of broad interest - such as identifying the nearest neighbor and computing the convex hull - cannot be easily posed as such. For these functions, we have to design multi-step algorithms which leverage composition and more advanced technical tools, as will be demonstrated in our applications.

In this section, we develop mechanisms for queries where the input is an n -tuple of points, i.e., the input domain is $U = (\mathbb{R}^d)^n$. For simplicity, we focus on the $d = 2$ case; extension to $d > 2$ dimensions is briefly discussed at the end of each subsection.

Both GP and CGP require a definition of the distance function $\text{dist}(\cdot, \cdot)$. Let $x = (x_1, \dots, x_n)$ and $x' = (x'_1, \dots, x'_n)$ be two tuples of points, where each $x_i, x'_i \in \mathbb{R}^2$ for $i \in [n]$. While the Euclidean distance is often the default distance function in \mathbb{R}^2 , there are different ways to combine n Euclidean distances into one. The most natural ones are: $\text{dist}_\infty(x, x') := \max_i \text{dist}(x_i, x'_i)$, $\text{dist}_1(x, x') := \sum_i \text{dist}(x_i, x'_i)$, and $\text{dist}_2(x, x') := \sqrt{\sum_i \text{dist}(x_i, x'_i)^2}$. Among these, dist_∞ provides the strongest privacy guarantee and is also the distance function adopted by [1]. In the remaining sections, we work mainly with dist_∞ and also briefly discuss dist_1 and dist_2 in Appendix H.

4.1 Technical Lemmas

The utility analyses in the later subsections depend on some technical lemmas, whose proofs are deferred to Appendix E. The first lemma characterizes the magnitude of $\|\tilde{x} - x\|$, where $x \in \mathbb{R}^2$, and \tilde{x} is the privatization of x obtained from either the planar Laplace mechanism or Gaussian mechanism (see Appendix A for the case $d \geq 2$).

LEMMA 4.1 (GENERALIZED GAMMA DISTRIBUTION [35, 36]). Let $G \sim \mathcal{G}(\lambda, k, p)$, which has pdf $g(r) = \frac{p/\lambda^k}{\Gamma(k/p)} r^{k-1} e^{-(r/\lambda)^p}$ for $r \in (0, \infty)$, where $\Gamma : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is the gamma function defined by $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$. Then $\mathbb{E}[G] = \lambda \frac{\Gamma(k/p+1/p)}{\Gamma(k/p)}$ and $\Pr[G \leq r] = \frac{\gamma(k/p, (r/\lambda)^p)}{\Gamma(k/p)}$, where $\gamma : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ defined by $\gamma(z, v) = \int_0^v t^{z-1} e^{-t} dt$ is the lower incomplete gamma function.

Recall the Lambert W function [7], which computes the inverse of the function $f(w) := we^w$. We have the following inequality from [4], which will be used for bounding the growth of $\|\tilde{x} - x\|$ when \tilde{x} is obtained from the planar Laplace mechanism.

LEMMA 4.2 ([4]). *Let $f^{-1}(\cdot)$ be the Lambert W function. For $u > 0$, we have*

$$-1 - \sqrt{2u} - u < f^{-1}(-e^{-u-1}) < -1 - \sqrt{2u} - 2u/3.$$

The analysis in Section 4.3 involves the random variable $Y := Z + W$, where Z, W are i.i.d. $\text{Lap}(b)$ random variables. We will use the following technical results:

LEMMA 4.3. *Let $Z, W \sim \text{Lap}(b)$, and $Y := Z + W$. Then for $1 > \beta > 0$ with probability $1 - \beta$, $|Y| \leq b \left(\sqrt{2 \log(1/\beta)} + \log(1/\beta) \right)$.*

LEMMA 4.4. *Given $y \in \mathbb{R}$, suppose we draw a $V \sim \text{Lap}(2b)$ until $V \leq y$. Let $r(y)$ be the number of draws given y and let $R = r(Y)$, where $Y := Z + W$ and $Z, W \sim \text{Lap}(b)$. Then $\mathbb{E}[R] \leq 4$.*

4.2 The Identity Query

The first problem we consider is the identity query $g(x) := x$, i.e., we wish to privatize the entire tuple of n points.

GP mechanism with dist_∞ . With dist_∞ , Lemma 2.6 requires us to sample a y from the pdf $m(x)(y) \propto e^{-\varepsilon \max_i \|y_i - x_i\|}$, which does not resemble any well-known distribution. Thus, implementing this would be difficult (a weakness typical of the exponential mechanism). To get around this difficulty, one can use privacy composition to privatize each $g_i : (\mathbb{R}^2)^n \rightarrow \mathbb{R}^2$ defined by $g_i(x) = x_i$, which is 1-Lipschitz with respect to dist_∞ , using the 2-dimensional Laplace mechanism [1]. However, [1] did not give any utility analysis of this mechanism. We provide one below.

LEMMA 4.5. *Let $G : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be the ε -GP mechanism which on input x draws a $y \in \mathbb{R}^2$ from a distribution with pdf $\propto e^{-\varepsilon \|x - y\|}$. Then with probability at least $1 - \beta$,*

$$\|G(x) - x\| \leq O\left(\frac{1}{\varepsilon} \log \frac{1}{\beta}\right).$$

PROOF. The following identities of the lower incomplete gamma function can be easily derived:

$$\begin{aligned} \gamma(1, v) &= \int_0^v e^{-t} dt = 1 - e^{-v}, \\ \gamma(z + 1, v) &= z\gamma(z, v) - v^z e^{-v}, z > 0. \end{aligned}$$

Let R denote the random variable $\|G(x) - x\|$. Now $R \sim \mathcal{G}(1/\varepsilon, 2, 1)$ and by Lemma 4.1,

$$\begin{aligned} \Pr[R > r] &= 1 - \Pr[R \leq r] \\ &= 1 - \frac{\gamma(2, r\varepsilon)}{\Gamma(2)} = 1 - \frac{\gamma(1, r\varepsilon) - r\varepsilon e^{-r\varepsilon}}{1} \\ &= 1 - (1 - e^{-r\varepsilon} - r\varepsilon e^{-r\varepsilon}) = (1 + r\varepsilon)e^{-r\varepsilon}. \end{aligned}$$

We want to find r so that the above probability is at most β . We will solve $(1 + y)e^{-y} = \beta$ for $y \geq 0$ via the Lambert W function f^{-1} where $f(w) := we^w$. We can write

$$\begin{aligned} -\beta &= (-1 - y)e^{-y} = (-1 - y)e^{-1-y}e^1 \\ -\frac{\beta}{e} &= (-1 - y)e^{-1-y}. \end{aligned}$$

Then we can set $w = -1 - y \leq -1$, where $w = f^{-1}\left(\frac{-\beta}{e}\right)$. Using the inequality $-1 - \sqrt{2u} - u < f^{-1}(-e^{-u-1})$ from Lemma 4.2 for $u = \log(1/\beta) > 0$, we have $y = -w - 1 < \sqrt{2 \log(1/\beta)} + \log(1/\beta)$. I.e. $r < \frac{1}{\varepsilon} (\sqrt{2 \log(1/\beta)} + \log(1/\beta))$. \square

THEOREM 4.6. *Let $G : (\mathbb{R}^2)^n \rightarrow (\mathbb{R}^2)^n$ be defined by $G(x) = (G_1(x_1), \dots, G_n(x_n))$, where each $G_i(x_i)$ returns a $y_i \in \mathbb{R}^2$ from a distribution with pdf $\propto e^{-\frac{\varepsilon}{n} \|x_i - y_i\|}$, for $i \in [n]$. Then G is ε -GP and with probability at least $1 - \beta$,*

$$\text{dist}_\infty(G(x), x) \leq O\left(\frac{n}{\varepsilon} \log \frac{n}{\beta}\right).$$

PROOF. Follows from Lemmas 2.7, 4.5, and a union bound. \square

CGP mechanism with dist_∞ . Under dist_∞ , we simply apply use CGP composition and apply the Gaussian mechanism on each $g_i(x) = x_i$, i.e., adding a noise drawn from $\sqrt{n/2\rho} \cdot \mathcal{N}(0, 1)$ to each coordinate of each point. Below, we analyze its error in terms of dist_∞ :

LEMMA 4.7. *Let $G : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be the ρ -CGP mechanism defined by $G(x) := x + \frac{1}{\sqrt{2\rho}}Z$ where $Z \sim \mathcal{N}(0, I_{2 \times 2})$. Then with probability at least $1 - \beta$, we have $\|G(x) - x\| \leq \sqrt{\log(1/\beta)/\rho}$.*

PROOF. Let R denote the random variable $\|G(x) - x\|$. Then $R \sim \mathcal{G}(1/\sqrt{\rho}, 2, 2)$ and by Lemma 4.1

$$\begin{aligned} \Pr[R > r] &= 1 - \Pr[P \leq r] \\ &= 1 - \frac{\gamma(1, (\sqrt{\rho}r)^2)}{\Gamma(1)} = 1 - (1 - e^{-\rho r^2}) = e^{-\rho r^2}. \end{aligned}$$

Setting the above probability to be β gives $r = \sqrt{\log(1/\beta)/\rho}$. \square

THEOREM 4.8. *Let $G : (\mathbb{R}^2)^n \rightarrow (\mathbb{R}^2)^n$ be defined by $G(x) = (G_1(x_1), \dots, G_n(x_n))$, where $G_i(x_i) := x_i + \frac{\sqrt{n}}{\sqrt{2\rho}}Z_i$ and $Z_i \sim \mathcal{N}(0, I_{2 \times 2})$, for $i \in [n]$. Then G is ρ -CGP and with probability at least $1 - \beta$,*

$$\text{dist}_\infty(G(x), x) \leq \sqrt{\frac{n \log(n/\beta)}{\rho}}.$$

PROOF. Follows from Lemmas 3.2, 4.7, and a union bound. \square

We see that the CGP mechanism is an $\tilde{O}(\sqrt{n})$ -factor better than the GP mechanism.

Extension to d dimensions. All the mechanisms above extend to d dimensions straightforwardly using the d -dimensional Laplace and Gaussian mechanism, respectively. Correspondingly, the utility gap between GP and CGP enlarges to $\tilde{O}(\sqrt{dn})$.

4.3 k Nearest Neighbors

Given a query point $p \in \mathbb{R}^2$ and a tuple $x = (x_1, \dots, x_n) \in (\mathbb{R}^2)^n$, the k nearest neighbors (k NN) query returns the k indices of the points in x that are the nearest to p . This could be useful in applications where x represents the locations visited by a user, say, in the past l days, and the data collector wishes to know if and when s/he has gotten into the proximity of a point p of interest (or danger). Note that for modularity, we define the problem so that only the indices of the k nearest neighbors are returned; if their locations are also needed (e.g., for computing the distance to p), then one

can reserve part of the privacy budget to privatize these k locations using the mechanism from Section 4.2.

For the k NN query, the canonical GP or CGP mechanism cannot be applied, as it is not Lipschitz. In fact, while we can still use dist_∞ for the input domain, it is not even clear what distance function to use for the output domain. As a simple baseline algorithm, we can privatize the entire x using the mechanisms from Section 4.2. Then finding the k nearest neighbors of p becomes just a post-processing step. However, as shown in Section 4.2, this baseline method has an error of $\tilde{O}(n/\varepsilon)$ and $\tilde{O}(\sqrt{n}/\rho)$ under GP and CGP, respectively. Below we develop an algorithm whose error depends (linearly or square-root) only on k , while the dependency on n will be logarithmic.

The sparse vector technique. As a technical tool, we first show that the *sparse vector technique* (SVT) [13], originally designed under standard DP, can be adapted to satisfy ε -GP, hence ρ -CGP for $\rho = \varepsilon^2/2$, which can be of independent interest.

Under GP or CGP, we are given a (possibly infinite) sequence of K -Lipschitz queries g_1, g_2, \dots with range in \mathbb{R} . The goal of SVT is to find the (index of the) first query whose output exceeds a given threshold T . The details are given in Algorithm 1, in which we flip the goal, i.e., try to find the first query whose output is below T . This does not introduce any real difference but makes our k NN algorithm more natural. The proof that Algorithm 1 satisfies ε -GP is similar to the DP version [13, 26] and is included in Appendix D for completeness.

Algorithm 1 Sparse Vector Technique

Input: $x \in U$; ε ; T ; K ; g_1, g_2, \dots where each g_j is K -Lipschitz
Output: variable-length sequence y_1, y_2, \dots

```

1:  $\varepsilon_1 \leftarrow \varepsilon/2, \varepsilon_2 \leftarrow \varepsilon/2$ 
2: draw  $W \sim \text{Lap}(K/\varepsilon_1)$ 
3: for  $j = 1, \dots$  do
4:   draw  $V_j \sim \text{Lap}(2K/\varepsilon_2)$ 
5:   if  $g_j(x) + V_j \leq T + W$  then
6:     output  $y_j = \top$  and HALT
7:   else
8:     output  $y_j = \perp$ 
9:   end if
10: end for

```

Nearest neighbor. We first present an algorithm for finding the nearest neighbor, i.e., the case $k = 1$. For $k > 1$, we will iteratively invoke it k times to find the k nearest neighbors. For generality, we describe our nearest neighbor algorithm as one that tries to find the nearest neighbor among a given subset of m points in x .

The starting observation is that the function $g_i(x; p) := \|x_i - p\|$ is 1-Lipschitz. Below, we show that shortest distance $h(x; p) := \min_j g_j(x; p)$ is also 1-Lipschitz. This allows us to first obtain a privatized $h(x; p)$. Then with this privatized shortest distance as T , we use SVT to cycle through the points in x , using their distances to p as the queries. When the SVT terminates, a point whose distance close to T will be identified with high probability.

LEMMA 4.9. *Let $g_j : U^n \rightarrow \mathbb{R}$ be K -Lipschitz (using dist_∞ on U^n) for each $j \in [m]$. Then $h(x) = \min_{j \in [m]} g_j(x)$ is also K -Lipschitz.*

PROOF. Let $x, x' \in (U)^n$. Let $j^* := \arg \min_{j \in [m]} g_j(x)$ and $l^* := \arg \min_{j \in [m]} g_j(x')$. Then

$$\begin{aligned} h(x) - h(x') &= g_{j^*}(x) - g_{l^*}(x') \leq g_{l^*}(x) - g_{l^*}(x') \\ &\leq K \|x_{l^*} - x'_{l^*}\| \leq K \cdot \text{dist}_\infty(x, x'), \end{aligned}$$

and

$$\begin{aligned} h(x') - h(x) &= g_{l^*}(x') - g_{j^*}(x) \leq g_{j^*}(x') - g_{j^*}(x) \\ &\leq K \|x'_{j^*} - x_{j^*}\| \leq K \cdot \text{dist}_\infty(x, x'). \end{aligned}$$

Combining the inequalities above, $|h(x) - h(x')| \leq K \cdot \text{dist}_\infty(x, x')$. \square

The private nearest neighbor (PNN) algorithm is described in Algorithm 2. In line 2, we assume the output of SVT is the index corresponding to the position of \top . The notation \bar{l}_m denotes the $(l \bmod m)$ -th index in I , so that it cycles through the m indices in I . We use x_I to denote the subset of points indexed by I . Since line 1 is $\varepsilon/3$ -GP and line 2 is $2\varepsilon/3$ -GP, Algorithm 2 is ε -GP.

Algorithm 2 Private Nearest Neighbor (PNN)

Input: $x \in (\mathbb{R}^2)^n$; $p \in \mathbb{R}^2$; $I \subseteq [n]$, $|I| = m$; ε ; (optional) $\gamma_0 \geq 0$
Output: index of the privatized nearest neighbor in I

```

1:  $T \leftarrow h(x_I; p) + Z + \gamma_0, Z \sim \text{Lap}(\frac{2}{\varepsilon})$ 
2:  $t \leftarrow \text{SVT}(x, 2\varepsilon/3, T, K = 1, g'_1, g'_2, \dots), g'_l := g_{\bar{l}_m}(\cdot; p)$ 
3: return  $\bar{l}_m$ 

```

Now we show that the PNN algorithm has an error of $O(\frac{1}{\varepsilon} \log \frac{m}{\beta})$ with probability $1 - \beta$, i.e., the returned nearest neighbor is at most this much farther away than the true nearest neighbor. Note that this is optimal up to a logarithmic factor, since the PNN query degenerates into the identity query when $m = 1$, which has an error of $O(\frac{1}{\varepsilon} \log \frac{1}{\beta})$.

We first prove a looser but simpler result:

THEOREM 4.10. *Fix any $0 < \beta < 1$, and let $\gamma := \gamma_1 + \gamma_2$, where $\gamma_1 := \frac{3}{\varepsilon} \left(\sqrt{2 \log \left(\frac{m+1}{\beta} \right)} + \log \left(\frac{m+1}{\beta} \right) \right)$ and $\gamma_2 := \frac{6}{\varepsilon} \log \left(\frac{m+1}{\beta} \right)$. Set $\gamma_0 = \gamma$ in Algorithm 2, then it outputs a \bar{l}_m such that $g_{\bar{l}_m}(x) \leq h(x) + 2\gamma$ with probability at least $1 - \beta$. The algorithm terminates in $O(m)$ time with probability $1 - \beta$.*

PROOF. Let W, V_1, V_2, \dots be the sequence of Laplace random variables in the SVT call. Let $t^* := \arg \min_{j \in [m]} g_j(x)$. When SVT visits t^* , it halts if $V_{t^*} \leq W + Z + \gamma_0$. We have $\Pr[|W + Z| \leq \gamma_1] \geq 1 - \beta/(m+1)$ by Lemma 4.3, and $\Pr[\max_{j \in [m]} |V_j| \leq \gamma_2] \geq 1 - m\beta/(m+1)$ by the Laplace tail bound and a union bound, i.e., $V_{t^*} \leq \max_{j \in [m]} |V_j| \leq W + Z + \gamma_0$ with probability $1 - \beta$. Consequently, we have simultaneously that (1) SVT halts the first time it visits g_{t^*} or earlier; (2) $g_t(x) + V_t \leq T + W$, hence

$$\begin{aligned} g_t(x) &\leq T + W - V_t = h(x) + Z + \gamma_0 + W - V_t \\ &\leq h(x) + |Z + W| + |V_t| + \gamma_0 \\ &\leq h(x) + \gamma_1 + \gamma_2 + \gamma_0 = h(x) + 2\gamma. \end{aligned}$$

\square

The optional parameter γ_0 is used to control the trade-off between the error and running time of the algorithm. In Theorem 4.10, we set $\gamma_0 = \gamma$ so that the SVT terminates within the first cycle of x but the error is doubled. Below, we give a more refined analysis, showing that setting $\gamma_0 = 0$ will not increase the asymptotic running time while reducing the error.

THEOREM 4.11. *Fix any $0 < \beta < 1$, and $0 < \beta_1, \beta_2, \beta_3$ such that $\beta_1 + \beta_2 + \beta_3 = \beta$. Let $\gamma := \gamma_1 + \gamma_2$, where $\gamma_1 := \frac{3}{\varepsilon} \left(\sqrt{2 \log \left(\frac{1}{\beta_1} \right)} + \log \left(\frac{1}{\beta_1} \right) \right)$ and $\gamma_2 := \frac{6}{\varepsilon} \log \left(\frac{4m}{\beta_2 \beta_3} \right)$. Set $\gamma_0 = 0$ in Algorithm 2, then it outputs a \bar{t}_m such that $g_{\bar{t}_m}(x) \leq h(x) + \gamma$ with probability at least $1 - \beta$. The algorithm runs in expected $O(m)$ time.*

PROOF. Let $t^* := \arg \min_{j \in [m]} g_j(x)$ and t be the index (which is a random variable) returned by the SVT call. Let k be positive integer. Then when SVT visits kt^* : it halts if $V_{kt^*} \leq W + Z$. By Lemma 4.4, the expected number of times t^* is visited is at most 4, hence the expected running time is $O(m)$. Moreover, by Markov's inequality, with probability at least $1 - \beta_2$ the number of times t^* is visited is at most $4/\beta_2$. I.e. $t \leq 4m/\beta_2$. We also have $\Pr[\max_{j \leq 4m/\beta_2} |V_j| \leq \gamma_2] \leq 1 - \beta_3$ and $\Pr[|Z + W| \leq \gamma_1] \geq 1 - \beta_1$. Thus, by a union bound, we have with probability at least $1 - (\beta_1 + \beta_2 + \beta_3)$:

$$\begin{aligned} g_{\bar{t}_m}(x) &\leq T + W - V_t = h(x) + Z + W - V_t \\ &\leq h(x) + |Z + W| + \max_{j \leq 4m/\beta_2} |V_j| \\ &\leq h(x) + \gamma_1 + \gamma_2 = h(x) + \gamma. \end{aligned}$$

□

Remark: We can in fact even set $\gamma_0 < 0$, which would further reduce the error at the cost of a longer running time. Theoretically, the running time would increase exponentially in $|\gamma_0|$, in the pathological case where all points in x but the nearest neighbor are very far away from p . On most typical instances, however, the running time is still close to linear with a $\gamma_0 < 0$.

k nearest neighbors. Now, to find the k nearest neighbors of p , we simply invoke Algorithm 2 k times, as shown in Algorithm 3. It follows from Theorem 3.2 that Algorithm 3 is ρ -CGP. We can similarly obtain an ε -GP version of the algorithm by replacing line 3 of the algorithm with $t = \text{PNN}(x, p, I_{j-1}, \varepsilon/k)$.

Algorithm 3 k -Private Nearest Neighbors (k -PNN)

Input: $p \in \mathbb{R}^2$; $x = (x_1, \dots, x_n) \in (\mathbb{R}^2)^n$; $k \in \mathbb{Z}_{>0}$; ρ

Output: $J \subseteq [n]$

- 1: $I_0 \leftarrow [n]$, $J_0 \leftarrow \emptyset$
 - 2: **for** $j = 1, \dots, k$ **do**
 - 3: $t \leftarrow \text{PNN}(x, p, I_{j-1}, \sqrt{2\rho/k})$
 - 4: $J_j \leftarrow J_{j-1} \cup \{t\}$
 - 5: $I_j \leftarrow I_{j-1} \setminus \{t\}$
 - 6: **end for**
 - 7: **return** $J = J_k$
-

The error of the k -PNN is $O\left(\sqrt{k/\rho} \log(n/\beta)\right)$, as shown in the following theorem.

THEOREM 4.12. *Fix $1 > \beta > 0$, let $\gamma := \frac{15\sqrt{k}}{\sqrt{2\rho}} \log((4n+2)/\beta) + \frac{3\sqrt{k}}{\sqrt{\rho}} \sqrt{\log((4n+2)/\beta)}$. Fix $1 \leq j \leq k$, let $t_j \in J$ be the index corresponding to the j th nearest neighbor output by Algorithm 3. Let t_j^* be the true j th nearest neighbor. Then with probability at least $1 - \beta$,*

$$\|x_{t_j} - p\| \leq \|x_{t_j^*} - p\| + \gamma.$$

PROOF. By Theorem 4.11, with $\varepsilon_j = \sqrt{2\rho/k}$, $\beta_1 = \beta_3 = \beta/(4n+2)$ and $\beta_2 = 4n\beta/(4n+2)$, we have with probability at least $1 - \beta$, $g_{t_j} = \|x_{t_j} - p\| \leq \min_{i \in [n] \setminus J_{j-1}} \|x_i - p\| + \gamma$. It remains to show $\min_{i \in [n] \setminus J_{j-1}} \|x_i - p\| \leq \|x_{t_j^*} - p\|$. Let $J_{j-1}^* = \{t_1^*, \dots, t_{j-1}^*\}$ be the set of indices containing the true $j-1$ nearest neighbors. If $J_{j-1} = J_{j-1}^*$ then $\min_{i \in [n] \setminus J_{j-1}} \|x_i - p\| = \|x_{t_j^*} - p\|$; otherwise there is t_l^* where $1 \leq l \leq j-1$ which is not in J_{j-1} , then $\min_{i \in [n] \setminus J_{j-1}} \|x_i - p\| \leq \|x_{t_l^*} - p\| \leq \|x_{t_j^*} - p\|$. □

On the other hand, the error of the ε -GP version of the algorithm has an error of $O(k/\varepsilon \cdot \log(n/\beta))$. The proof is similar and omitted.

Extension to d dimensions. Both the GP and CGP algorithms for the k NN problem extend to d dimensions verbatim, as the function $g_j(x) = \|x_i - p\|_2$ is 1-Lipschitz in any dimensions, so the noise required does not depend on d . Consequently, the error guarantee remains the same for any d .

4.4 Convex Hull

In the last application, we are interested in releasing a privatized convex hull of a collection of points. Given a tuple $x = (x_1, \dots, x_n) \in (\mathbb{R}^2)^n$, let $\text{CONV}(x)$ denote the convex hull of the points in x . As in the k NN algorithm, we first privately find a set of k indices $A \subseteq [n]$. Let x_A be the subset of points indexed by A . Then we release a privatized x_A , denoted \tilde{x}_A , and compute $\text{CONV}(\tilde{x}_A)$ as a post-processing step. However, unlike the k NN problem where k is given, here k is an internal parameter that controls the balance of two sources of errors: A large k will make $\text{CONV}(x_A)$ close to $\text{CONV}(x)$ but enlarge the gap between $\text{CONV}(x_A)$ and $\text{CONV}(\tilde{x}_A)$ as the noise injected in \tilde{x}_A is $\tilde{O}(\sqrt{k})$ by Theorem 4.8. These two sources of errors must be quantified in order to find an optimal k .

Finding A privately. We first consider the problem of, for a given k , how to privately find a subset of k points, indexed by A , so that $\text{CONV}(x_A)$ is as close to $\text{CONV}(x)$ as possible. The intuition is that we want to include in A those points that are near the boundary of $\text{CONV}(x)$. Meanwhile, the points selected to A should not be too close to each other, so as to maximize their coverage. Thus, the idea is to find “equally” spaced points near the boundary of $\text{CONV}(x)$. We do so in three steps:

- (1) Let $c(x)$ be the center of x , whose coordinates are the mid-points between the smallest and largest coordinates in each dimension. In Appendix F we show that $c(\cdot)$ is $\sqrt{2}$ -Lipschitz, so we can invoke the Gaussian mechanism to obtain a privatized \tilde{c} .
- (2) Next, we identify a privatized and large enough radius \tilde{R} so that a circle of radius \tilde{R} around \tilde{c} encloses x . This is done by applying the Gaussian mechanism on the function $g(x) := \max_{i \in [n]} \|x_i - \tilde{c}\| = -\min_{i \in [n]} (-\|x_i - \tilde{c}\|)$, which

is 1-Lipschitz by Lemmas E.1 and 4.9. Then we enlarge it by $\tilde{O}(1)$ so that it encloses all points in x with high probability.

- (3) Finally, we place k points equally spaced on the boundary of the circle, and find the nearest neighbor in x to each of the k points, using the PNN algorithm from the previous section.

The detailed private convex hull (PCH) algorithm is given in Algorithm 4. We use a privacy budget of $\frac{2}{3}\rho_0$ for line 2, and $\frac{1}{3}\rho_0$ for line 3. The k calls of PNN at line 8 consume a total of $(\rho - \rho_0)$ privacy budget. So the whole algorithm satisfies ρ -CGP.

Algorithm 4 Private Convex Hull (PCH)

Input: $x = (x_1, \dots, x_n) \in (\mathbb{R}^2)^n$; k ; ρ ; β

Output: $A \subseteq [n]$

```

1:  $\rho_0 \leftarrow \rho/20, \rho_1 \leftarrow (\rho - \rho_0)/k$ 
2:  $\tilde{c} \leftarrow c(x) + Z_c, Z_c \sim \mathcal{N}(0, \frac{3}{2\rho_0} I_{2 \times 2})$ 
3:  $\tilde{R} \leftarrow \max_i (\|x_i - \tilde{c}\|) + \sqrt{3 \log(2/\beta)/\rho_0} + Z_R, Z_R \sim \mathcal{N}(0, \frac{3}{2\rho_0})$ 
4:  $A \leftarrow \emptyset$ ;
5: for  $j = 1, \dots, k$  do
6:    $\theta_j \leftarrow \frac{2\pi(j-1)}{k}$ 
7:    $P_j \leftarrow \tilde{c} + [\tilde{R} \cos(\theta_j), \tilde{R} \sin(\theta_j)]^T$ 
8:    $a_j \leftarrow \text{PNN}(x, P_j, \{1, \dots, n\}, \sqrt{2\rho_1})$ 
9:    $A \leftarrow A \cup \{a_j\}$ 
10: end for
11: return  $A$ 

```

To bound the difference between $\text{CONV}(x_A)$ and $\text{CONV}(x)$, we will show that a small expansion of $\text{CONV}(x_A)$ will enclose $\text{CONV}(x)$. This is formalized using the Minkowski sum. Recall for two sets of vectors $V, V' \subseteq \mathbb{R}^d$, their Minkowski sum is: $V + V' := \{v + v' : v \in V, v' \in V'\}$, while $V - V'$ is defined such that $(V - V') + V' = V$. Let \mathcal{B}_r denote the ball of radius r centered at the origin, and let $\omega(x) := \max_{i,j} \|x_i - x_j\|$ be the diameter of x .

LEMMA 4.13. *Let $A = \text{PCH}(x, k, \rho, \beta)$ be the output of Algorithm 4 and $x_A := \{x_{a_j}\}_{a_j \in A}$. Then with probability at least $1 - \beta$,*

$$\text{CONV}(x_A) \subseteq \text{CONV}(x) \subseteq \text{CONV}(x_A) + \mathcal{B}_\gamma,$$

$$\text{where } \gamma = O\left(\frac{\sqrt{k} \log(n/\beta)}{\sqrt{\rho}} + \frac{\omega(x)}{k}\right).$$

We first show that, given a large enough circle which encloses all points in x , the set of nearest neighbors in x of k equally spaced points on the boundary of the circle provides a good approximation to $\text{CONV}(x)$. Let V^* denote the indices of the set of vertices of $\text{CONV}(x)$, and let $p : [0, 2\pi] \rightarrow \mathbb{R}^2$ be a function defined by $p(\theta; r, o) := o + [r \cos(\theta), r \sin(\theta)]^T$.

LEMMA 4.14. *Let $C = (o, R)$ be a circle of radius R centered at o which encloses all points in x . Let k be a positive integer, and let P_1, \dots, P_k be k equally spaced points on the boundary of C , where $P_j := p(2\pi(j-1)/k; R, o)$ for $j \in [k]$. Let $V_k := \{\arg \min_{i \in [n]} \|x_i - P_j\| : j \in [k]\}$ be the set of nearest neighbors to the k points. Then for any $v \in V^* \setminus V_k$, there is $u \in V_k$ such that x_u is at a distance of at most $\gamma_2 := \frac{2\pi}{k}R$ from x_v .*

PROOF. Let $v \in V^* \setminus V_k$. Recall a point x_i in x is a vertex of $\text{CONV}(x)$ if there is a hyperplane (a line in \mathbb{R}^2) that separates x_j

from the rest of x ; or equivalently, there is a hyperplane that just touches x_i such that all of x are on the same side of the hyperplane. Let $h(x_v)$ denote such a hyperplane for x_v . By construction, there are no points between $h(x_v)$ and the boundary of C ; if there were such a point, it would be crossed by $h(x_v)$ before x_v , contradicting the definition of $h(x_v)$. Consider a line perpendicular to $h(x_v)$ extended from x_v until it meets the boundary of C , let $p_v^* = p(\theta_v^*; R, c)$ denote this point of intersection, for some $\theta_v^* \in [0, 2\pi]$. Then p_v^* is on the boundary between the points P_l and P_{l+1} for some $l \in [k]$. Without loss of generality, assume p_v^* lies closer to P_l on the boundary. Then $\|P_l - p_v^*\| \leq \frac{1}{2} \cdot \frac{2\pi\tilde{R}}{k}$. Let $l^* := \arg \min_{i \in [n]} \|P_l - x_i\|$. Next, we will show that x_{l^*} is at most a distance of γ_2 from x_v , and since $x_{l^*} \in V_k$, it is the x_u desired in the lemma.

To this end, consider the triangle with vertices (x_v, P_l, x_{l^*}) . Let $a := \|P_l - x_v\|$, $b := \|P_l - x_{l^*}\|$ and $c := \|x_v - x_{l^*}\|$. Let $\varphi \in [0, \pi]$ be the angle between the edges $x_v P_l$ and $x_{l^*} P_l$. By the law of cosines, $c^2 = a^2 + b^2 - 2ab \cos(\varphi)$, which is maximized when $\cos(\varphi)$ is minimized and b is maximized. Since $\cos(\varphi)$ is strictly decreasing for $\varphi \in [0, \pi]$, c is maximized when φ is as large as possible without crossing $h(x_v)$. Also, $b = \|P_l - x_{l^*}\| = \min_{i \in [n]} \|P_l - x_i\| \leq \|P_l - x_v\| = a$. Thus, c is at most equal to the distance between x_v and the other point where a circle of radius a centered at P_l intersects $h(x_v)$, which can be obtained by mirroring x_v about a line that is perpendicular to $h(x_v)$ and crosses P_l (see Fig. 1 for an illustration). Thus c is at most $2 \cdot \|P_l - p_v^*\| \leq \frac{2\pi}{k}R = \gamma_2$. \square

Now, to prove Lemma 4.13, it remains to show that the set of indices A returned by Algorithm 4 is a good approximation to V_k .

PROOF OF LEMMA 4.13. By the Gaussian tail bound with probability at least $1 - \beta/2$, $\max_i \|x_i - \tilde{c}\| \leq \tilde{R} \leq \omega(x) + O\left(\sqrt{\log(1/\beta)/\rho}\right)$.

Then, a circle \tilde{C} centered at \tilde{c} with radius \tilde{R} encloses all of the points in x . The analysis below is conditioned upon this happening. By Lemma 4.14, with $P_j := p(2\pi(j-1)/k; \tilde{R}, \tilde{c})$ for $j \in [k]$ and $V_k := \{\arg \min_{i \in [n]} \|x_i - P_j\| : j \in [k]\}$, then for each $v \in V_k \setminus V^*$ we can find $u \in V_k$ such that x_u is at a distance of at most $\gamma_2 = \frac{2\pi}{k}\tilde{R}$ from x_v .

$$\text{Let } \gamma_1 := \frac{15 \log(2(4n+2)k/\beta) + 3\sqrt{2 \log(2(4n+2)k/\beta)}}{\sqrt{2\rho_1}} = O\left(\frac{\sqrt{k} \log(n/\beta)}{\sqrt{\rho}}\right).$$

Note that $\max_{j \in [k]} \|x_{a_j} - x_{j^*}\| \leq \gamma_1$ with probability at least $1 - \beta/2$, which follows from Theorem 4.12 and a union bound, where $j^* := \arg \min_{i \in [n]} \|P_j - x_i\|$ corresponds to the true nearest neighbor of P_j . Thus, for each $v \in V^*$, there is $a_u \in A$ such that $\|x_{a_u} - x_v\| \leq \|x_{a_u} - x_u\| + \|x_u - x_v\| \leq \gamma_1 + \gamma_2 = O\left(\frac{\sqrt{k} \log(n/\beta)}{\sqrt{\rho}} + \frac{\omega(x)}{k}\right)$. \square

Finding the convex hull privately. We first run Algorithm 4 with, say, half of the privacy budget. Then, we use the other half of budget to privatize the points x_A using the algorithm from Section 4.2, denoted \tilde{x}_A . More precisely, for each $a \in A$, we release $\tilde{x}_a := x_a + \frac{1}{\sqrt{\rho/k}} \cdot \mathcal{N}(0, I_{2 \times 2})$. Finally, we return $\text{CONV}(\tilde{x}_A)$.

We can also bound the error of $\text{CONV}(\tilde{x}_A)$ in terms of the Minkowski sum:

LEMMA 4.15. *With probability at least $1 - \beta$,*

$$\text{CONV}(\tilde{x}_A) - \mathcal{B}_{\gamma_3} \subseteq \text{CONV}(x) \subseteq \text{CONV}(\tilde{x}_A) + \mathcal{B}_\gamma,$$

$$\text{where } \gamma = O\left(\frac{\sqrt{k} \log(n/\beta)}{\sqrt{\rho}} + \frac{\omega(x)}{k}\right) \text{ and } \gamma_3 = O\left(\frac{\sqrt{k} \log(k/\beta)}{\sqrt{\rho}}\right).$$

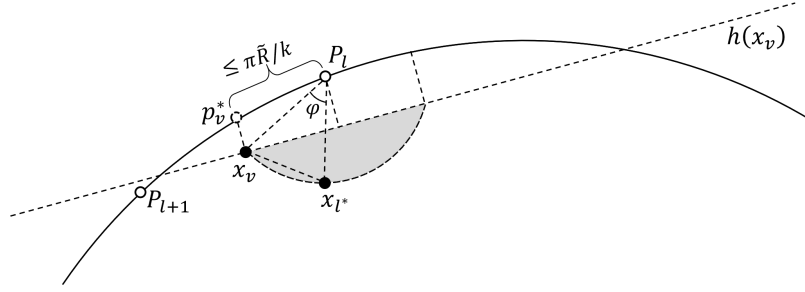


Figure 1: Illustration of the position of x_{l^*} in Lemma 4.14.

PROOF. Let $\gamma_1 := \frac{15 \log(4(4n+2)k/\beta) + 3\sqrt{2 \log(4(4n+2)k/\beta)}}{\sqrt{\rho_1}}$ and $\gamma_2 := 2\pi\tilde{R}/k$. Let $x_A \subseteq x$ be the collection of "anchor" points corresponding to the indices in A . Following the arguments in the proof of Theorem 4.13, with probability $1 - \beta/2$, $\text{CONV}(x_A) \subseteq \text{CONV}(x) \subseteq \text{CONV}(x_A) + \mathcal{B}_{(\gamma_1 + \gamma_2)}$. Let $\gamma_3 := \frac{\sqrt{2k \log(2k/\beta)}}{\sqrt{\rho}}$, then $\max_{j \in [k]} \|\tilde{x}_{a_j} - x_{a_j}\| \leq \gamma_3$ with probability at least $1 - \beta/2$. Thus,

$$\text{CONV}(\tilde{x}_A) - \mathcal{B}_{\gamma_3} \subseteq \text{CONV}(x_A) \subseteq \text{CONV}(\tilde{x}_A) + \mathcal{B}_{\gamma_3}.$$

By a union bound, together we have with probability $1 - \beta$,

$$\text{CONV}(\tilde{x}_A) - \mathcal{B}_{\gamma} \subseteq \text{CONV}(x_A) \subseteq \text{CONV}(x) \subseteq \text{CONV}(\tilde{x}_A) + \mathcal{B}_{\gamma},$$

where $\gamma = \gamma_1 + \gamma_2 + \gamma_3$. \square

Setting k . To minimize the error, we can choose k so as to balance the two terms $\frac{\sqrt{k} \log(n/\beta)}{\sqrt{\rho}}$ and $\frac{\omega(x)}{k}$, yielding $k = \left(\frac{\omega(x)\sqrt{\rho}}{\log(n/\beta)} \right)^{2/3}$. One caveat is that we cannot use $\omega(x)$ directly as it is private information. So we replace it with \tilde{R} , which is larger than $\omega(x)$ by an additive $O\left(\sqrt{\log(1/\beta)/\rho}\right)$ term. Plugging this value of k into Lemma 4.15, we obtain:

THEOREM 4.16. *The private convex hull algorithm above satisfies ρ -CGP. It returns an \tilde{x}_A such that with probability at least $1 - \beta$,*

$$\text{CONV}(\tilde{x}_A) - \mathcal{B}_{\gamma} \subseteq \text{CONV}(x) \subseteq \text{CONV}(\tilde{x}_A) + \mathcal{B}_{\gamma},$$

$$\text{where } \gamma = O\left(\frac{\left(\omega(x) + \sqrt{\log(1/\beta)/\rho}\right)^{1/3} \log^{2/3}(n/\beta)}{\rho^{1/3}}\right) = \tilde{O}\left(\omega(x)^{1/3} + 1\right).$$

Note that the extra $\tilde{O}(1)$ term, resulting from the technical replacement of $\omega(x)$ with \tilde{R} , is actually unavoidable: In the extreme case where x consists of just one point, we have $\omega(x) = 0$ while the convex hull problem degenerates into the identity query, which must have $\tilde{O}(1)$ error.

The algorithm can also be made to satisfy ε -GP. Similar to the k NN problem, the ε -GP version of the algorithm can only use basic composition to allocate the privacy to the k PNN queries in Algorithm 4, as well as for privatizing the k points in x_A . This leads to an overall error of $O\left(\frac{k \log(n/\beta)}{\varepsilon} + \frac{\omega(x)}{k}\right)$, which is $\tilde{O}\left(\sqrt{\omega(x)} + 1\right)$ after balancing these two terms similarly as above.

Remark 1: If $\omega(x) > \tilde{\Omega}(n^{3/2})$, i.e., the point set is very sparse (relative to the unit distance), then the optimal value of k would be greater than n . In this case, the algorithm essentially degenerates

into the naive algorithm that privatizes all n points and then computes the convex hull. Then the $O(\omega(x)/k)$ terms goes away, and the error just becomes $\tilde{O}(\sqrt{n})$, same as that for the identity query. For the GP version, the error is then $\tilde{O}(n)$.

Remark 2: In the analysis above, we simply balanced the two error terms for finding a good k . This gets the asymptotic result right. In our implementation, we derive the precise constant coefficients in the error expression of Lemma 4.15, and minimize it by setting its derivative to 0 to find an optimal k .

Extending to d dimensions. Algorithm 4 can be generalized to work in d dimensions as follows: (1) find a privatized center \tilde{c} , by computing the midpoint of each dimension, which is \sqrt{d} -Lipschitz (alternatively, the d -dimensional mean can be used, and computing the mean is 1-Lipshitz); (2) find a privatized and large enough radius \tilde{R} , by computing $\max_{i \in [n]} \|x_i - \tilde{c}\|$, which is 1-Lipschitz; (3) place k "evenly spaced" points on the hypersphere of radius \tilde{R} centered at \tilde{c} and find their private nearest neighbors; and (4) release the k nearest neighbors by adding Gaussian noise of scale $\tilde{O}(\sqrt{dk})$ and compute their convex hull. Omitting the details, we can show that the total error becomes $\tilde{O}\left(\sqrt{dk} + \omega(x)/k^{1/(d-1)}\right)$, which minimizes to $\tilde{O}\left(\omega(x)^{\frac{d-1}{d+1}} d^{\frac{1}{d+1}} + 1\right)$. The error of the GP version is $\tilde{O}\left(\omega(x)^{\frac{d-1}{d}} d^{\frac{1}{d}} + 1\right)$.

5 EXPERIMENTS

We perform experiments on a dataset containing mobility traces of taxi cabs [33]. The dataset contains approximately 500 trajectories which were collected over 30 days in the Bay Area of San Francisco. Most ($> 95\%$) of these trajectories contain 5000 – 30000 points with an average of 20000 points. Each trajectory corresponds to one collection of points in $(\mathbb{R}^2)^n$. We convert the GPS coordinates into \mathbb{R}^2 coordinates via the Mercator projection, with meters (m) being the units of distance. As argued in [1, 5], the choice of the distance unit is unimportant as all it matters is $\varepsilon \cdot \text{dist}(x, x')$ ($\rho \cdot \text{dist}(x, x')^2$ for CGP). If for example kilometers are used, we can just scale up ε by a factor of 10^3 (resp. scale up ρ by a factor of 10^6). Also, mechanisms employing different distance units can be composed after converting to the same distance unit. The values of ρ used in these experiments are in the range $[5 \cdot 10^{-8}, 5 \cdot 10^{-4}]$, corresponding to ε values in the range of approximately 0.002 (i.e. 2 per 1000m) to 0.2 (i.e. 2 per 10m). We also include the same set of experiments for larger privacy budgets (ρ ranging from 10^{-4} to 1)

in Appendix G. The code for all the experiments can be found at: <https://github.com/hkustDB/ConcentratedGeoPrivacy>.

We examine the error with respect to various parameter settings. The parameters of interest are the number of input points (n), privacy level (ρ) and the number of neighbors (k). When we compare ρ -CGP algorithms with ϵ -GP algorithms, we advantage the latter with a larger privacy budget using the relationship described in Lemma 3.7 with $\delta = 10^{-10}$ and $\epsilon\Delta \geq 10$. Following the discussion after Lemma 3.7, the differences in the privacy guarantees between GP and CGP are thus negligible. In each experiment, we randomly select 50 collections of points and we sample n points from each collection to be used as inputs to the algorithms. If a collection has less than n points, all of its points will be used. Each experiment is repeated 25 times and we report the mean, the 25th and 75th percentiles. We group the experiments by application as follows:

The identity query (Fig. 2). We evaluate the algorithms which privatize each location with (ϵ/n) -GP (GP Basic) and (ρ/n) -CGP (CGP Basic), respectively. We report both the max error on a single point and the ℓ_2 error across the entire tuple.

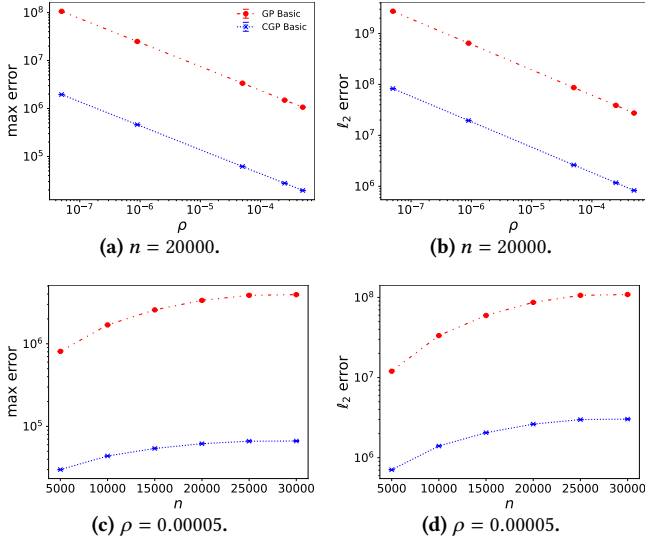


Figure 2: Releasing a collection of points : error with respect to privacy level (top) and to tuple size (bottom).

Both algorithms have smaller error as the privacy budget ρ increases and have larger error as the number of points n increases. When n is held fixed, the error of GP Basic is close to \sqrt{n} times that of CGP Basic, as can be seen in Fig. 2. This is consistent with our utility analysis in Section 4.2.

k nearest neighbors (Fig. 3, 4). We compare Algorithm 3 (CGP PNN) and its ϵ -GP variant (GP PNN) against the baseline algorithms GP Basic and CGP Basic. In each experiment, a query point p is generated uniformly at random from a list of coordinates composed of the centers of all $1m \times 1m$ squares that are crossed by some trajectory. The error is computed as the sum of distances of the output neighbors (to the query point), normalized by that of the true (non-private) nearest neighbors.

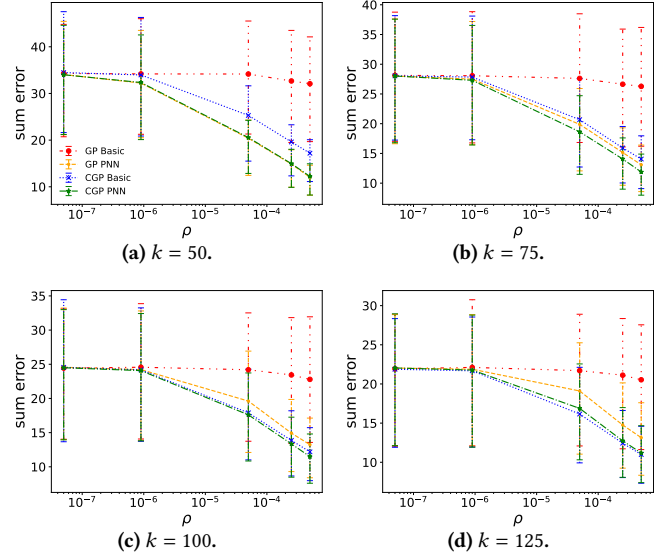


Figure 3: Nearest neighbor: error with respect to privacy level, fixing $n = 20000$.

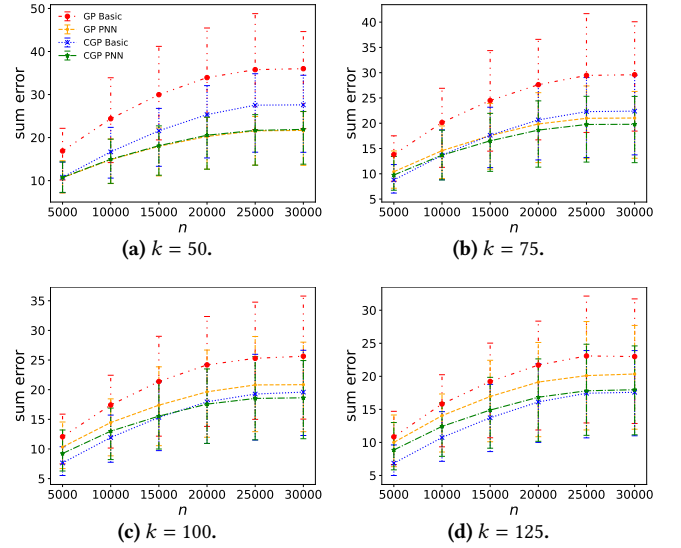


Figure 4: Nearest neighbor: error with respect to tuple size, fixing $\rho = 0.00005$.

Across all of the plots in Fig. 3 and 4, we see that for small k , the PNN algorithms perform better, consistent with our analysis. As k increases, CGP Basic starts to outperform GP PNN and eventually CGP PNN, since their errors scale with k and \sqrt{k} , respectively, and logarithmically in n . On the other hand, CGP Basic scales with \sqrt{n} and is independent of k . GP Basic has the worst error, since it scales with n .

Convex hull (Fig. 5). We compare Algorithm 4 (CGP PCH) and its ϵ -GP variant (GP PCH) against the baseline algorithms GP Basic and CGP Basic. We compute the convex hulls from the privatized

points returned by these algorithms. The PCH algorithms return k points, while the baseline algorithms return n points. The parameter k is set as described in Section 4.4, which has a different value in CGP PCH and GP PCH. We also round it to the range [16, 128] for efficiency reasons. The utility is measured by the Jaccard similarity index between the privatized convex hull and the true convex hull (higher is better):

$$\begin{aligned} \text{Jacc}(\text{CONVH}(M(x)), \text{CONVH}(x)) \\ = \frac{\mu(\text{CONVH}(M(x)) \cap \text{CONVH}(x))}{\mu(\text{CONVH}(M(x)) \cup \text{CONVH}(x))}, \end{aligned}$$

where $\mu(C)$ is the area enclosed by the edges of C .

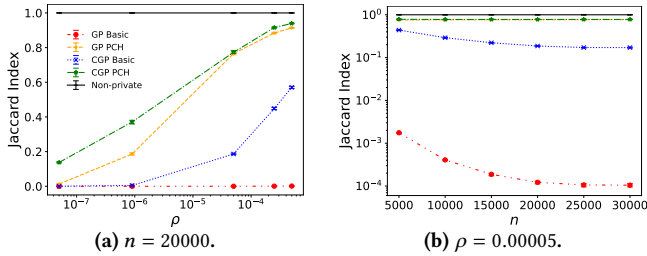


Figure 5: Convex hull : error with respect to privacy level (left) and to tuple size (right).

All four algorithms have better performance as ρ increases; in the high privacy regime (lower privacy budget), the baseline algorithms have little utility, and GP Basic has essentially no utility even for the highest privacy budget considered. The baseline algorithms, especially GP Basic, lose utility as the number of points n increases, since the area enclosed by the noisy convex hull becomes larger due to a smaller privacy budget being allotted to each point. The PCH algorithms do not suffer from this issue and maintain consistently high utility as n increases. While the CGP version of the PCH algorithm has some advantage over the GP version, the gap is not large. Indeed, while the difference is a polynomial $\tilde{O}(\omega(x)^{1/6})$ theoretically, it is not as significant on real datasets.

6 CONCLUDING REMARKS

We have shown that CGP offers many benefits over GP, including simplicity of the mechanism, lower noise scale in high dimensions, and better composability. This potentially opens a door for more problems on private geometric data to be studied. One particularly interesting direction is when the input x is an (unordered) set of n points as opposed to an (ordered) n -tuple. Between point sets, popular metrics include the earth mover distance or the Hausdorff distance. Other metric spaces, such as curves using the Fréchet distance, or even non-geometric data such as strings using the edit distance, can also be studied under the CGP framework, and are of potential interest.

ACKNOWLEDGMENTS

This work has been supported by HKRGC under grants 16201819, 16205420, and 16205422. We thank the anonymous reviewers of CCS '23 for their valuable comments.

REFERENCES

- [1] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 901–914.
- [2] Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. 2019. Private center points and learning of halfspaces. In *Conference on Learning Theory*. PMLR, 269–282.
- [3] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.
- [4] Ioannis Chatzigeorgiou. 2013. Bounds on the Lambert function and their application to the outage analysis of user cooperation. *IEEE Communications Letters* 17, 8 (2013), 1505–1508.
- [5] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10–12, 2013. Proceedings 13*. Springer, 82–102.
- [6] Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length n-grams. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 638–649.
- [7] Robert M Corless, Gaston H Gonnet, David EG Hare, David J Jeffrey, and Donald E Knuth. 1996. On the LambertW function. *Advances in Computational mathematics* 5, 1 (1996), 329–359.
- [8] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. 2012. Differentially private spatial decompositions. In *2012 IEEE 28th International Conference on Data Engineering*. IEEE, 20–31.
- [9] Teddy Cunningham, Graham Cormode, Hakan Ferhatosmanoglu, and Divesh Srivastava. 2021. Real-World Trajectory Sharing with Local Differential Privacy. *Proc. VLDB Endow.* 14, 11 (jul 2021), 2283–2295. <https://doi.org/10.14778/3476249.3476280>
- [10] Damien Desfontaines and Balázs Pejó. 2020. SoK: Differential privacies. In *Proceedings on Privacy Enhancing Technologies Symposium*. 288–311.
- [11] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.
- [12] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 381–390.
- [13] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
- [14] Cynthia Dwork and Guy N Rothblum. 2016. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887* (2016).
- [15] Úlfar Erlingsson, Vasil Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 1054–1067.
- [16] Yue Gao and Or Sheffet. 2021. Differentially Private Approximations of a Convex Hull in Low Dimensions. In *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [17] Mehmet Emre Gursay, Ling Liu, Stacey Truex, Lei Yu, and Wenqi Wei. 2018. Utility-aware synthesis of differentially private and attack-resilient location traces. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 196–211.
- [18] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M Procopiuc, and Divesh Srivastava. 2015. DPT: differentially private trajectory synthesis using hierarchical reference systems. *Proceedings of the VLDB Endowment* 8, 11 (2015), 1154–1165.
- [19] Ziyue Huang, Yuting Liang, and Ke Yi. 2021. Instance-optimal Mean Estimation Under Differential Privacy. *Advances in Neural Information Processing Systems* 34 (2021).
- [20] Ziyue Huang and Ke Yi. 2021. Approximate Range Counting Under Differential Privacy. In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [21] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In *International conference on machine learning*. PMLR, 1376–1385.
- [22] Haim Kaplan, Micha Sharir, and Uri Stemmer. 2020. How to find a point in the convex hull privately. In *International Symposium on Computational Geometry*.
- [23] Shiva P Kasiviswanathan and Adam Smith. 2014. On the ‘semantics’ of differential privacy: A bayesian formulation. *Journal of Privacy and Confidentiality* 6, 1 (2014).
- [24] Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. 2021. Learning with user-level privacy. *Advances in Neural Information Processing Systems* 34 (2021), 12466–12479.
- [25] Fang Liu, Dong Wang, and Zhengquan Xu. 2021. Privacy-preserving travel time prediction with uncertainty using GPS trace data. *IEEE Transactions on Mobile Computing* 22, 1 (2021), 417–428.

- [26] Min Lyu, Dong Su, and Ninghui Li. 2017. Understanding the Sparse Vector Technique for Differential Privacy. *Proc. VLDB Endow.* 10, 6 (feb 2017), 637–648. <https://doi.org/10.14778/3055330.3055331>
- [27] George Marsaglia and Wai Wan Tsang. 2000. A simple method for generating gamma variables. *ACM Transactions on Mathematical Software (TOMS)* 26, 3 (2000), 363–372.
- [28] Frank McSherry and Kunal Talwar. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. IEEE, 94–103.
- [29] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 263–275.
- [30] Jack Murtagh and Salil Vadhan. 2015. The complexity of computing the optimal composition of differential privacy. In *Theory of Cryptography Conference*. Springer, 157–175.
- [31] Shanmugavelayutham Muthukrishnan and Aleksandar Nikolov. 2012. Optimal private halfspace counting via discrepancy. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. 1285–1292.
- [32] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. 2013. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. 351–360.
- [33] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. 2009. CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from <https://crawdad.org/epfl/mobility/20090224>. <https://doi.org/10.15783/C7J010>
- [34] Alfréd Rényi. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, Vol. 4. University of California Press, 547–562.
- [35] Edney W Stacy. 1962. A generalization of the gamma distribution. *The Annals of mathematical statistics* (1962), 1187–1192.
- [36] E Webb Stacy and G Arthur Mihram. 1965. Parameter estimation for a generalized gamma distribution. *Technometrics* 7, 3 (1965), 349–358.
- [37] Tim Van Erven and Peter Harremo. 2014. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* 60, 7 (2014), 3797–3820.
- [38] Yonghui Xiao and Li Xiong. 2015. Protecting locations with differential privacy under temporal correlations. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 1298–1309.
- [39] Mengmeng Yang, Ivan Tjuawinata, and Kwok-Yan Lam. 2022. K-Means Clustering With Local d_χ -Privacy for Privacy-Preserving Data Analysis. *IEEE Transactions on Information Forensics and Security* 17 (2022), 2524–2537.
- [40] Jun Zhao, Teng Wang, Tao Bai, Kwok-Yan Lam, Zhiying Xu, Shuyu Shi, Xuebin Ren, Xinyu Yang, Yang Liu, and Han Yu. 2019. Reviewing and improving the Gaussian mechanism for differential privacy. *arXiv preprint arXiv:1911.12060* (2019).
- [41] Lu Zhou, Le Yu, Suguo Du, Haojin Zhu, and Cailian Chen. 2018. Achieving differentially private location privacy in edge-assistant connected vehicles. *IEEE Internet of Things Journal* 6, 3 (2018), 4472–4481.

A THE d -DIMENSIONAL LAPLACE MECHANISM

The magnitude R can be simulated by rejection sampling [27]. See full version at: <https://arxiv.org/abs/2305.19756>.

B PROOF FOR CGP COMPOSITION

See full version at: <https://arxiv.org/abs/2305.19756>.

C PROOF FOR RELATIONSHIP BETWEEN CGP AND $(\epsilon, \delta, \Delta)$ -GP

See full version at: <https://arxiv.org/abs/2305.19756>.

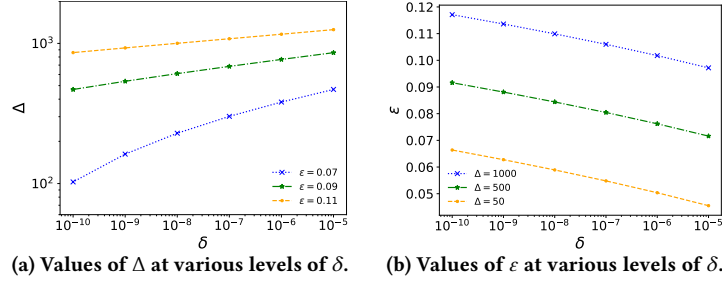


Figure 6: $(\epsilon, \delta, \Delta)$ -GP implied by ρ -CGP, fixing $\rho = 0.00005$.

D PROOF FOR ϵ -GP VARIANT OF SPARSE VECTOR TECHNIQUE

See full version at: <https://arxiv.org/abs/2305.19756>.

E TECHNICAL LEMMAS AND MISSING PROOFS

LEMMA E.1. Let (U_0, dist_{U_0}) , (U_1, dist_{U_1}) and (U_2, dist_{U_2}) be metric spaces. Let $g_1 : U_0 \rightarrow U_1$ and $g_2 : U_1 \rightarrow U_2$ be Lipschitz with constants K_1 and K_2 respectively.

- (1) [Composition] $h : U_0 \rightarrow U_2$ defined by $h(x) := g_2(g_1(x))$ is $(K_1 K_2)$ -Lipschitz;
- (2) [Linear maps] $h : U_0 \rightarrow U_1$ defined by $h(x) = a \cdot g_1(x) + b$ is $(K_1 \cdot |a|)$ -Lipschitz if U_1 is a real vector space, for any $a \in \mathbb{R}$ and $b \in U_1$.

LEMMA E.2 (UNIVARIATE GAUSSIAN TAIL BOUND). Let $Z \sim \mathcal{N}(0, \sigma^2)$. Then $\Pr[Z > t] \leq e^{-\frac{t^2}{2\sigma^2}}$ for all $t \geq 0$.

LEMMA E.3. Let $l(\mu, b)(\cdot)$ denote the pdf of a Laplace random variable with mean μ and scale b . For any shift $s \in \mathbb{R}$ in the mean and for any $y \in \mathbb{R}$: $l(\mu, b)(y) \leq e^{|s|/b} \cdot l(\mu + s, b)(y)$.

LEMMA E.4. Let Z, W be i.i.d. $\text{Lap}(b)$ random variables. Then the pdf of $Y := Z + W$ is given by $\frac{1}{4b^2}(be^{-|y|/b} + |y|e^{-|y|/b})$ for $y \in \mathbb{R}$.

LEMMA E.5. Let Y be a random variable with pdf $\frac{1}{4b^2}(be^{-|y|/b} + |y|e^{-|y|/b})$ for $y \in \mathbb{R}$, where $b > 0$. Then $|Y| \leq b \left(\sqrt{2 \log(1/\beta)} + \log\left(\frac{1}{\beta}\right) \right)$ with probability $1 - \beta$.

LEMMA E.6 (LEMMA 4.3). Let $Z, W \sim \text{Lap}(b)$, and $Y := Z + W$. Then for $1 > \beta > 0$, with probability $1 - \beta$, $|Y| \leq b \left(\sqrt{2 \log(1/\beta)} + \log\left(\frac{1}{\beta}\right) \right)$.

LEMMA E.7 (LEMMA 4.4). Given $y \in \mathbb{R}$, suppose we draw a $V \sim \text{Lap}(2b)$ until $V \leq y$. Let $r(y)$ be the number of draws given y and let $R = r(Y)$, where $Y := Z + W$ and $Z, W \sim \text{Lap}(b)$. Then $\mathbb{E}[R] \leq 4$.

For missing proofs, see full version at: <https://arxiv.org/abs/2305.19756>.

F PROOF FOR LIPSCHITZNESS OF COMPUTING THE CENTER

LEMMA F.1. Let $h_l : (\mathbb{R}^2)^n \rightarrow \mathbb{R}$ be defined by $h_l(x) = \max_{i \in [n]} x_{i,l}$. Let $g_l : (\mathbb{R}^2)^n \rightarrow \mathbb{R}$ be defined by $g_l(x) = \max_{i \in [n]} (-x_{i,l})$, where $x_{i,l}$ denotes the l th coordinate of x_i for $l = 1, 2$. Let $c : (\mathbb{R}^2)^n \rightarrow \mathbb{R}^2$ be defined by $c(x) = \frac{1}{2}(h_1(x) - g_1(x), h_2(x) - g_2(x))$. Then c is $\sqrt{2}$ -Lipschitz.

For missing proofs, see full version at: <https://arxiv.org/abs/2305.19756>.

G EXPERIMENTS IN THE LOW PRIVACY REGIME

See full version at: <https://arxiv.org/abs/2305.19756>.

H METRICS FOR THE CENTRAL MODEL

See full version at: <https://arxiv.org/abs/2305.19756>.