Algorithms with More Granular Differential Privacy Guarantees

Badih Ghazi
Google, Mountain View, CA, USA

Ravi Kumar
Google, Mountain View, CA, USA

Pasin Manurangsi
Google, Mountain View, CA, USA

Thomas Steinke
Google, Mountain View, CA, USA

Abstract
Differential privacy is often applied with a privacy parameter that is larger than the theory suggests is ideal; various informal justifications for tolerating large privacy parameters have been proposed. In this work, we consider partial differential privacy (DP), which allows quantifying the privacy guarantee on a per-attribute basis. We study several basic data analysis and learning tasks in this framework, and design algorithms whose per-attribute privacy parameter is smaller than the best possible privacy parameter for the entire record of a person (i.e., all the attributes).

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

Keywords and phrases
Differential Privacy, Algorithms, Per-Attribute Privacy

Digital Object Identifier
10.4230/LIPIcs.ITCS.2023.54

Related Version
Full Version: https://arxiv.org/abs/2209.04053

1 Introduction

Differential Privacy (DP) [35] provides a strict worst-case privacy guarantee – even an adversary that knows the entire dataset except for one bit of information about one individual cannot learn that bit, even when the dataset and the bit in question are arbitrary. Since its inception, researchers have sought to relax the DP definition in order to permit better data analysis while still providing meaningful privacy guarantees [30].

The only approach to relaxing the definition of DP that has gained widespread use – albeit not acceptance – is quantitative relaxation. That is, it is common to set the main privacy parameter (usually denoted by \( \varepsilon \)) to be larger than the theory allows us to easily interpret. More precisely, the privacy loss bound \( \varepsilon \) is used to quantify the tolerable accuracy with which an adversary can learn the unknown bit. Theory would suggest that \( \varepsilon \leq 1 \) provides a good privacy guarantee, and that the guarantee rapidly degrades if we further increase \( \varepsilon \). The setting \( \varepsilon = 10 \) permits a sensitive bit to be revealed with 99.999% accuracy, if we are unlucky enough to be in a truly worst-case setting. Nevertheless, it is common to set \( \varepsilon \geq 10 \) [29, 76, 20]. This raises the question: Can one still provide meaningful privacy guarantees even when the DP parameter \( \varepsilon \) is large?

Informally, we can justify large DP parameters by arguing that, for “realistic” adversaries, “natural” data, and “nice” algorithms, the “real” privacy guarantee is better than the worst-case guarantee summarized by the single parameter \( \varepsilon \). (And, of course, we can also hope that better privacy parameters could be obtained by a more careful analysis of the algorithm.)
In this paper, we seek to understand the above intuitive justification for tolerating large DP parameters. This requires us to formalize what constitutes a “nice” algorithm, which we do with partial DP—a notion based on previous studies in the DP literature that provides a more granular accounting of the privacy loss parameter. In particular, this permits us to quantify a “per-attribute $\varepsilon_0$” in addition to the usual “per-person $\varepsilon$.” (See Section 2 for the formal definition, and Section 3 for an overview of the prior work.) Setting $\varepsilon \geq 10$ may become more palatable if, e.g., we can simultaneously assert that each (sensitive) attribute has $\varepsilon_0 \leq 1$. To interpret such a guarantee, we must also discuss what sort of adversaries and attacks we are and are not protected against.

The focus of our work is on designing and analyzing algorithms that establish a quantitative separation between the attainable per-person $\varepsilon$ and the per-attribute $\varepsilon_0$. The key message is that we demonstrate that, in many circumstances, we can say more about the privacy properties of an algorithm beyond what can be conveyed by the standard single-parameter definition of DP.

### 1.1 Contributions

We investigate a variety of fundamental data analysis and learning tasks through the lens of per-attribute partial DP. That is, we present several algorithms and analyze their fine-grained privacy properties. More specifically, under the partial DP notion, we consider three data analysis tasks and obtain more granular bounds than under standard DP:

(i) We first study algorithms for answering general families of statistical queries (Section 4). We analyze the projection mechanism and a variant of the multiplicative weights exponential mechanism under per-attribute partial DP. These results show a separation between the standard per-person $\varepsilon$ and the per-attribute $\varepsilon_0$ that scales polynomially with the dimension (i.e., the number of attributes in each person’s record).

(ii) We next present a new algorithm for computing histograms (a.k.a., heavy hitters) that gives a per-attribute privacy parameter $\varepsilon_0$ that is exponentially smaller than the standard per-person $\varepsilon$-DP parameter in terms of the number of attributes (Section 5). That is, if each person’s record is $d$ bits, the error of our algorithm grows as $\log \frac{d}{\varepsilon_0}$, while the standard pure DP algorithm would have error $\frac{d}{\varepsilon}$. We also prove a near-matching lower bound.

Note that histograms are an important case study, as the standard algorithms for this closely resemble the kind of worst-case algorithms that we wish to rule out. E.g., if we add Laplace noise to each count in a histogram to achieve $\varepsilon$-DP with $\varepsilon \geq 10$, then $\geq 99\%$ of counts would still round back to the exact value. Depending on the sparsity of the histogram, this would be a weak privacy guarantee in practice. Hence we design a partial DP algorithm that avoids this worst-case behaviour.

(iii) Finally, we present an algorithm for robustly learning halfspaces under per-attribute partial DP (Section 5). This has a per-attribute privacy parameter that does not grow with the dimension, as is the case for the standard per-person privacy parameter.

Our results are summarized in Table 1.

### 2 Formal Definitions & Basic Properties

We briefly recall the definition of differential privacy (DP) [35, 33].

▶ **Definition 1 (DP).** A randomized algorithm $M : \mathcal{X}^n \rightarrow \mathcal{Y}$ is $(\varepsilon, \delta)$-differentially private ($(\varepsilon, \delta)$-DP) if, for all $x, x' \in \mathcal{X}^n$ differing on a single entry (i.e., $\exists i \in [n] \forall j \in [n] \setminus \{i\} \ x_j = x'_j$) and all measurable $S \subset \mathcal{Y}$, we have $\Pr [M(x) \in S] \leq e^\varepsilon \cdot \Pr [M(x') \in S] + \delta$. 

---

54:2 Algorithms with More Granular Differential Privacy Guarantees
We say that a randomized algorithm $\mathcal{A}$ is $(\varepsilon, \delta)$-private if, for all inputs $x, x' \in X^n$ differing on a single entry, $\mathbb{P}[\mathcal{A}(x) \neq \mathcal{A}(x')] \leq \frac{2}{\varepsilon^2}$.

### Definition 3 (CDP).
A randomized algorithm $M : X^n \rightarrow Y$ is $\frac{1}{2}$-zero-Concentrated DP (zCDP) if, for all $x, x' \in X^n$ differing on a single entry, $D_{\diamond}(M(x) \| M(x')) \leq \frac{1}{2}$.

### Partial Differential Privacy
Partial DP is a natural extension of the standard DP definition; we replace the single parameter $\varepsilon$ with a function that measures the dissimilarity of two persons’ records. Similar definitions have appeared in the literature before (Section 3).

### Definition 4 (Partial DP).
Let $\varepsilon : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be symmetric and non-negative (i.e., $\forall x, x' \in \mathcal{X}, \varepsilon(x, x') = \varepsilon(x', x) \geq 0$). We say that a randomized algorithm $M : X^n \rightarrow Y$ is $\varepsilon$-partially DP (\varepsilon-DP) if, for all inputs $x, x' \in X^n$ differing on a single entry and for all measurable $S \subset Y$, $\mathbb{P}[M(x) \in S] \leq \mathbb{P}[M(x') \in S]$.

### Definition 5 (Per-Attribute Partial DP).
For $x, x' \in X = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, we denote the Hamming distance $\|x - x'\|_0 : = |\{i \in [d] : x_i \neq x'_i\}|$. For $\varepsilon \geq 0$, we define $\varepsilon$-per-attribute partial DP ($\varepsilon$-\varepsilon-DP) (which we also call per-attribute DP or Hamming partial DP) to be $\varepsilon$-\varepsilon-DP with $\varepsilon(x, x') : = \varepsilon_0 \cdot \|x - x'\|_0$.

We also consider a partial DP equivalent of CDP.

### Definition 6 (Partial CDP and Per-Attribute Partial CDP).
Let $\varepsilon : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be symmetric and non-negative. We say that a randomized algorithm $M : X^n \rightarrow Y$ is $\varepsilon$-partially CDP (\varepsilon-DP) if, for all inputs $x, x' \in X^n$ differing on a single entry, $D_{\diamond}(M(x) \| M(x')) \leq \frac{1}{2}\varepsilon_i(x_i, x'_i)^2$, where $i \in [n]$ is the index on which $x$ and $x'$ differ (i.e., $\forall i' \neq i$ $x_i = x'_i$).

For $\varepsilon_0 \in \mathbb{R}$, we define $\varepsilon_0$-per-attribute partial CDP ($\varepsilon_0$-\varepsilon-DP) to be $\varepsilon$-\varepsilon-DP with $\varepsilon(x, x') : = \varepsilon_0 \cdot \|x - x'\|_0$. 

---

1 By group privacy, we can assume that $\varepsilon$ satisfies the triangle inequality $\varepsilon(x, x') \leq \varepsilon(x, x') + \varepsilon(x, x')$. 

## Table 1

| Task | Standard $\varepsilon$-DP / $\frac{1}{2}\varepsilon^2$-zCDP | Per-attribute $\varepsilon_0$-\varepsilon-DP |
|------|-------------------------------------------------|----------------------------------|
| $k$-way marginals on $\{0, 1\}^d$, mean error $\alpha$ | $n = O\left(\min\left\{\frac{1}{\sqrt{\alpha}}, \sqrt{\frac{|Q|}{\alpha n}}\right\}\right)$ | $n = O\left(\min\left\{\frac{1}{\sqrt{\alpha}}, \sqrt{\frac{|Q|}{\alpha n}}\right\}\right)$ |
| $k$-way marginals on $\{0, 1\}^d$, max error $\alpha$ | $n = O\left(\frac{2^{k+1} \log d}{n \alpha^2}\right)$ | $n = O\left(\frac{2^{k+1} \log d}{\varepsilon_0 n^2}\right)$ |
| Heavy Hitters on $\{0, 1\}^d$, error $\alpha$ | $n = O\left(\frac{1}{\alpha^2} + \frac{1}{\alpha n} + \frac{1}{\alpha^2 n}\right)$ | $n = O\left(\frac{1}{\varepsilon_0^2 n^2} + \frac{1}{\varepsilon_0 n} + \frac{1}{\alpha^2 n}\right)$ |
| $\gamma$-robust learning halfspaces over $\{\pm 1\}^d$, error $\alpha$ | $n = O\left(\frac{1}{\alpha^2} + \frac{1}{\alpha n} + \frac{1}{\alpha^2 n}\right)$ | $n = O\left(\frac{1}{\varepsilon_0^2 n^2} + \frac{1}{\varepsilon_0 n} + \frac{1}{\alpha^2 n}\right)$ |

The setting with $\delta = 0$ (abbreviated $\varepsilon$-DP) is called pure DP, while $\delta > 0$ is called approximate DP. We also work with zero-concentrated DP (zCDP) [18], which is a refinement of the original definition of concentrated DP [38]. This is formulated via Rényi divergences [72]:

$D_\lambda(P \| Q) := \frac{1}{\lambda - 1} \log \frac{\mathbb{E}_{X \sim P} \left(\frac{P(x)}{Q(x)}\right)^{\lambda - 1}}{\mathbb{E}_{Y \sim Q} \left(\frac{P(y)}{Q(y)}\right)^{\lambda - 1}}$. We define $D_* (P \| Q) := \sup_{\lambda \in (1, \infty)} \frac{1}{\lambda - 1} D_\lambda (P \| Q)$ and $D_\infty (P \| Q) := \sup_{S \subseteq \Omega : |P(S)| > 0} \log(P(S) / Q(S))$. 

### CDP. A randomized algorithm $M : X^n \rightarrow Y$ is $\frac{1}{2}$-zero-Concentrated DP (zCDP) if, for all $x, x' \in X^n$, $D_\infty (M(x) \| M(x')) \leq \frac{1}{2}$. 

### Partial Differential Privacy
Partial DP is a natural extension of the standard DP definition; we replace the single parameter $\varepsilon$ with a function that measures the dissimilarity of two persons’ records. Similar definitions have appeared in the literature before (Section 3).
Definition 6 is a relaxation of Definition 4, i.e., \( \varepsilon \)-\( \nabla \)DP implies \( \varepsilon \)-\( \nabla \)CDP. CDP has better composition properties than pure DP, which makes it more useful in practice.\(^2\)

DP is usually defined in terms of neighboring datasets – i.e., a binary relation, rather than a metric. We can define partial DP in terms of such a graph of neighboring records:

**Lemma 7 (Equivalent Definition of Partial DP).** Let \( G \) be an undirected non-negatively weighted graph on \( X \). Let \( M : X^n \rightarrow Y \) be a randomized algorithm. Suppose, for every pair \( x, x' \in X^n \) differing on a single entry \( i \in [n] \), if \( \{x_i, x'_i\} \) is an edge in the graph \( G \) with weight \( \varepsilon_{0} \), then \( D_{\infty}(M(x)||M(x')) \leq \varepsilon_{0} \) (resp., \( D_{\ast}(M(x)||M(x')) \leq \frac{1}{2} \varepsilon_{0}^2 \)). Let \( \varepsilon : X \times X \rightarrow \mathbb{R} \) be the distance metric on the graph \( G \). Then \( M \) is \( \varepsilon \)-\( \nabla \)DP (resp., \( \varepsilon \)-\( \nabla \)\( \ast \)DP).

In particular, Lemma 7 tells us that per-attribute partial DP is equivalent to changing the neighboring relation of DP to consider changing only a single attribute of a person, rather than an entire person record. That is, an equivalent definition of \( \varepsilon_{0} \)-\( \nabla \)DP (or \( \varepsilon_{0} \)-\( \nabla \)\( \ast \)DP) is to require that for all pairs \( x, x' \in (X_1 \times \cdots \times X_d)^n \) of datasets differing on a single attribute of a person, we have \( \forall S \ P[M(x) \in S] \leq e^{\varepsilon_{0}} \cdot P[M(x') \in S] \) (or, respectively, \( D_{\ast}(M(x)||M(x')) \leq \frac{1}{2} \varepsilon_{0}^2 \)). I.e., the graph in Lemma 7 is the Hamming graph with each edge having the same weight \( \varepsilon_{0} \). We will define such pairs as *neighboring*.

### 2.2 Basic Properties of Partial DP

An essential property of partial DP is that it is directly comparable to standard DP:

**Proposition 8.** If \( M : X^n \rightarrow Y \) satisfies \( \varepsilon_{0} \)-\( \nabla \)DP and \( X = X_1 \times \cdots \times X_d \) consists of \( d \) attributes, then \( M \) satisfies \( (d \cdot \varepsilon_{0}) \)-DP. Similarly, \( \varepsilon_{0} \)-\( \nabla \)\( \ast \)DP implies \( \frac{1}{2}d^2 \varepsilon_{0}^2 \)-\( \ast \)CDP, which in turn implies \( (\varepsilon, \delta) \)-DP for all \( \varepsilon \geq \frac{1}{2}d^2 \varepsilon_{0}^2 \) and \( \delta = \exp\left(-\frac{\varepsilon}{d^2 \varepsilon_{0}^2}\right) \).

More generally, if \( M : X^n \rightarrow Y \) satisfies \( \varepsilon \)-\( \nabla \)DP, then \( M \) satisfies \( \sup_{x, x' \in X} \varepsilon(x, x') \)-\( \nabla \)DP. Conversely, if \( M \) satisfies \( \varepsilon \)-\( \nabla \)DP, then \( M \) satisfies \( \varepsilon \)-\( \nabla \)\( \ast \)DP (where we interpret \( \varepsilon \) as a constant function) and \( \varepsilon \)-\( \nabla \)\( \ast \)DP.

The conversion from per-attribute partial DP to standard DP is an application of the group privacy property (a.k.a. the triangle inequality for Rényi divergences). That is, \( D_{\infty}(P||Q) \leq D_{\infty}(P||R) + D_{\infty}(R||Q) \) and \( D_{\ast}(P||Q) \leq \left(\sqrt{D_{\ast}(P||R)} + \sqrt{D_{\ast}(R||Q)}\right)^2 \) for all appropriate probability distributions \( P, Q, \) and \( R \). More generally, if we have \( \varepsilon_{0} \)-\( \nabla \)DP and the adversary is interested in only \( k \) attributes, then we obtain a privacy guarantee comparable to \( (k \cdot \varepsilon_{0}) \)-DP. This conversion may or may not be tight, but it is important that we can directly relate the partial DP guarantee back to standard DP.

Next we have composition, which is inherited from the standard DP definition:

**Lemma 9 (Sequential composition).** Let \( M_1 : X^n \rightarrow Y_1 \) be \( \varepsilon_1 \)-\( \nabla \)DP (respectively, \( \varepsilon_1 \)-\( \nabla \)\( \ast \)DP). Let \( M_2 : X^n \times Y_1 \rightarrow Y_2 \) be such that the restriction \( M_2(\cdot, y) : X^n \rightarrow Y_2 \) is \( \varepsilon_2 \)-\( \nabla \)DP (resp., \( \varepsilon_2 \)-\( \nabla \)\( \ast \)DP) for all \( y \in Y_1 \). Define \( M_{12} : X^n \rightarrow Y_2 \) by \( M_{12}(x) = M_2(x, M_1(x)) \). Then \( M_{12} \) is \( (\varepsilon_1 + \varepsilon_2) \)-\( \nabla \)DP (resp., \( \sqrt{\varepsilon_1^2 + \varepsilon_2^2} \)-\( \nabla \)\( \ast \)DP).

A simple difference between per-attribute partial DP and standard DP is that if we run multiple DP algorithms on disjoint sets of attributes, then the privacy parameter does not grow with the number of attributes. In contrast, under standard DP, the privacy parameter would grow with the number of attributes following composition.

---

\(^2\) We do not consider approximate DP as a basis for partial DP, as approximate DP has poor group privacy properties, which are essential for our setting. Another option is Gaussian DP [52], which has properties very similar to CDP.
Lemma 10 (Parallel composition, [63]). For $j \in [d]$, let $M_j : X^n \to Y_j$. Let $X = X_1 \times \cdots \times X_d$ and $Y = Y_1 \times \cdots \times Y_d$. Define $M : X^n \to Y$ by $M(x) = M_j((x_{i,j})_{i \in [n]})$ for all $x \in X^n$ and $j \in [d]$, where $x_{i,j} \in X_j$ denotes only the $j$th attribute of the $i$th record. If $M_j$ is $\varepsilon$-DP (resp., $1/\varepsilon^2$-zCDP) for each $j \in [d]$, then $M$ is $\varepsilon \cdot \nabla_{0}DP$ (resp., $\varepsilon \cdot \nabla_{0}CDP$).

For example, if we release independent statistics about the medical records, browsing histories, employment, etc. of some people and each individual release is $\varepsilon$-DP, then the overall release is $\varepsilon \cdot \nabla_{0}DP$, assuming no overlapping attributes. In particular, this example characterizes the setting where non-coordinating entities perform DP analysis on data from the same people.

In the rest of this paper, we provide several algorithms in the per-attribute partial DP framework. We show that, in a variety of settings, there is a separation between partial DP and standard DP, i.e., we can provide $\varepsilon_0 \cdot \nabla_{0}DP$ (or $\varepsilon_0 \cdot \nabla_{0}CDP$) with a small per-attribute privacy parameter $\varepsilon_0$, but it is not possible to provide $\varepsilon$-DP (or $1/\varepsilon^2$-zCDP) with a small per-person privacy parameter $\varepsilon$. We argue that, in such settings, it is more informative to give a small per-attribute guarantee $\varepsilon_0$, in addition to the large per-person guarantee $\varepsilon$.

3 Related Work

Definitions. There is a vast literature on privacy definitions both before and since the introduction of DP [35]; Desfontaines and Pejó [30] catalog 225 DP variants that have been proposed. We only discuss the definitions most closely related to partial DP. We organize these definitions into three categories: (i) more general than partial DP, (ii) the same as or similar to per-attribute partial DP, and (iii) special cases of partial DP, but different from per-attribute partial DP.

(i) Notions more general than partial DP: Chatzikokolakis et al. [24] define a notion of metric DP or $d$-privacy, where the indistinguishability guarantee is determined by a metric $d$ on the space of all input datasets (as opposed to data points in partial DP). An equivalent definition, dubbed Lipschitz privacy was given by Koufogiannis et al. [59]. Similarly, Pufferfish privacy [57] and Blowfish privacy [49] define a generalized notion of neighboring datasets (a.k.a. “secrets”), which yields a generalization of DP by taking the metric to be proportional to the distance between datasets on the graph of neighboring datasets (cf. Lemma 7). Unlike these prior works, we restrict the definition of partial DP to consider pairs of datasets that only differ on the record of a single person, rather than considering pairs of datasets in which the records of multiple people may change. We consider this restriction to be an important feature of our definition, as it ensures that partial DP remains comparable to standard DP (Proposition 8) and thus can still be interpreted as an individual privacy guarantee. Metric DP or context-aware DP has also been studied in the context of local DP [4, 2]; our focus, however, is on central DP.

(ii) Notions comparable to per-attribute partial DP: Our notion of per-attribute partial DP is equivalent to the definition of attribute DP given by Kifer and Machanavajjhala [56]; Kenthapadi et al. [55] and Ahmed et al. [3] also use this definition, but they simply call it “differential privacy” without qualification. Asi et al. [6] define element-level DP, where the distance between data points is determined by the number of “elements” on which they differ; examples of “elements” are whether or not a certain word is included in a person’s message history, or whether or not a domain is in the browsing history.

(iii) Other special cases of partial DP: Andrés et al. [5] define geo-indistinguishability, which is a special case of partial DP in which $\varepsilon(x, x') = \varepsilon_0 \cdot ||x - x'||_2$, i.e., inputs are points in space and the privacy guarantee scales with the Euclidean distance. Another special case of partial DP is edge DP in graphs [48], where a person corresponds to a vertex that may...
have many incident edges, but privacy is only guaranteed on a per-edge basis. Label DP [25] is also a special case of partial DP, where $\varepsilon(x, x') = \infty$ if $x$ and $x'$ differ on any attribute other than the label in the training dataset of a supervised machine learning task.

It is common to assume that each person contributes one record to the dataset, but often a person may contribute multiple records. If we do not account for this, then we have a relaxed version of DP, which has been dubbed item-level DP or record-level DP [50, 66]. A recent line of work on “user-level DP” provides algorithms that ensure standard DP even when each user has multiple records [61, 42, 62, 27].

We remark that most of the related prior work considers definitions based on pure DP, which is rarely used in practice due to its inferior composition properties. Our work also considers concentrated DP; while this extension is straightforward, we believe it is important.

Algorithms. Although many different privacy definitions have been proposed, surprisingly few algorithms have been studied under these notions. To the best of our knowledge, all of the comparable prior algorithmic results are variants of adding Laplace or Gaussian noise scaled to a modified version of sensitivity that fits the definition. Our main technical contribution is a deeper exploration of the algorithmic aspects of a more granular privacy analysis; we provide several algorithms for a variety of standard data analysis tasks and give a more granular privacy analysis for each.

4 Answering Query Workloads

In this section we consider the problem of releasing statistics that depend on overlapping sets of attributes. In particular, we investigate releasing private answers to an arbitrary family $Q$ of queries $\{q_j : \mathcal{X} \rightarrow \mathbb{R}\}_{j=1}^m$. E.g., if $\mathcal{X} = \{0, 1\}^d$, then $Q$ could be all $k$-way parities or $k$-way conjunctions. These examples of low-order marginals are some of the best-studied families of queries in the DP literature; in particular, they are known to be among the “hardest” families of queries [19]. We design partial DP mechanisms for answering such families of queries to contrast the partial DP bounds against the standard DP bounds.

4.1 Warmup: Average Error via Noise Addition

Abusing notation, let $Q(x) \in \mathbb{R}^m$ denote the vector $(\frac{1}{n} \sum_{i=1}^n q_j(x_i))_{j=1}^m$ of answers. Let $\Delta = \sup \{ \|Q(x) - Q(x')\|_2 : x, x' \in \mathcal{X} \}$ be the diameter and let $\Delta_0 = \sup \{ \|Q(x) - Q(x')\|_2 : x, x' \in \mathcal{X}, \|x - x'\|_0 \leq 1 \}$ be the partial diameter of the set of possible answers.

A simple algorithm for answering $Q$ under standard DP with low mean squared error (MSE), is the projection mechanism [70, 37], which adds Gaussian noise $\mathcal{N}(\cdot, \cdot)$ to the vector of all query answers and projects this noisy vector back to the set of answer vectors that are consistent with some valid input. This naturally extends to partial DP. For $\sigma > 0$ and $Q : \mathcal{X} \rightarrow \mathbb{R}^m$, define the projection mechanism $M : \mathcal{X}^n \rightarrow \mathbb{R}^m$ as follows.

On input $x \in \mathcal{X}^n$, compute $Q(x) = \frac{1}{n} \sum_{i=1}^n Q(x_i)$, sample $Y \leftarrow \mathcal{N}(Q(x), \sigma^2 I)$, and output $\hat{Y} := \arg \min_{y \in \text{conv}((Q(z) : z \in \mathcal{X}))} \|y - Y\|_2$, where $\text{conv}(\cdot)$ is the convex hull.

3 The only exception is element-level DP [6], which is based on Rényi DP [67], a relaxation of CDP.
Theorem 11. The projection mechanism $M$ simultaneously satisfies $\frac{1}{2}\varepsilon^2$-$z$CDP and $\varepsilon_0$-$\nabla_0$CDP for $\varepsilon = \frac{\Delta}{\sqrt{m}}$ and $\varepsilon_0 = \frac{\Delta_0}{\sqrt{m}}$. Furthermore, for all $x \in \mathcal{X}^n$,

$$\mathbb{E} \left[ \frac{1}{m} \| M(x) - Q(x) \|_2^2 \right] \leq \min \left\{ \sigma^2 \cdot \frac{\Delta \cdot \sqrt{2 \log |\mathcal{X}|}}{m} \right\}.$$  

(1)

Theorem 11 simply states that the partial DP guarantee of the projection mechanism scales with the partial diameter $\Delta_0$ in place of the diameter $\Delta$; the former could be much smaller, depending on $Q$, as we show next.

Consider the cases of $k$-way (unsigned) conjunctions or parities on $\mathcal{X} = \{0, 1\}^d$, which are families of $m = \binom{d}{k}$ queries each. In both cases, $\Delta = \sqrt{m}$ and $\Delta_0 = \sqrt{\frac{d!}{(d-k)!}} = \sqrt{m \cdot k/d}$. From Theorem 11, for a given noise scale $\sigma$ of the projection mechanism, the ratio of the per-person and per attribute privacy parameters is $\frac{\varepsilon_0}{\varepsilon} = \frac{\Delta_0}{\Delta} = \sqrt{\frac{k}{d}}$. The error guarantee of the projection mechanism under standard DP is near-optimal [19, 14]. Thus we have a separation – in this setting we can report a smaller per-attribute privacy parameter $\varepsilon_0$ than the attainable per-person privacy parameter $\varepsilon$.

Alternatively, if we fix a $\varepsilon_0$-$\nabla_0$CDP guarantee, the MSE is

$$\mathbb{E} \left[ \frac{1}{m} \| M(x) - Q(x) \|_2^2 \right] \leq \min \left\{ \frac{\Delta_0^2}{\varepsilon_0 n^2}, \frac{\Delta_0 \cdot \Delta \cdot \sqrt{2 \log |\mathcal{X}|}}{\varepsilon_0 n m} \right\} = \min \left\{ \frac{m k}{\varepsilon_0 n^2 d}, \frac{\sqrt{k \cdot 2 \log 2}}{\varepsilon_0 n} \right\}.$$  

In contrast, for $\frac{1}{2}\varepsilon^2$-$z$CDP the MSE is

$$\mathbb{E} \left[ \frac{1}{m} \| M(x) - Q(x) \|_2^2 \right] \leq \min \left\{ \frac{\Delta^2}{\varepsilon^2 n^2}, \frac{\Delta \cdot \sqrt{2 \log |\mathcal{X}|}}{\varepsilon n m} \right\} = \min \left\{ \frac{m}{\varepsilon^2 n^2}, \frac{\sqrt{d \cdot 2 \log 2}}{\varepsilon n} \right\}.$$  

That is to say that the MSE under per-attribute partial DP scales with $k$ (the number of attributes that each query can depend on) rather than $d$ (the total number of attributes).

The $\Delta \cdot \sqrt{\log |\mathcal{X}|}$ term in the guarantee of Theorem 11 can, in general, be replaced by the Gaussian width $\mathbb{E}_{\mathcal{X} \sim \mathcal{N}(0, I)} \left[ \sup_{y \in S_x} \langle y, G \rangle \right]$ of the set $S_x := \{ Q(\bar{x}) - Q(x) : \bar{x} \in \mathcal{X} \}$.

### 4.2 Maximum Error via an Iterative Algorithm

Rather than average error, we can obtain bounds on the maximum error over all queries. In the standard DP setting, optimal bounds are given by Private Multiplicative Weights (PMW) [46] and its refinement the Multiplicative Weights Exponential Mechanism (MWEM) [45]. We now look at a per-attribute partial DP version of MWEM, $M_\ell : \mathcal{X}^n \rightarrow [0, 1]^m$, given in Algorithm 1.

In this section we assume that the domain $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ of the queries is finite and the range of the queries is $[0, 1]$. For $x \in \mathcal{X}^n$, let $q(x) = \frac{1}{n} \sum_{i=1}^n q_i(x_i)$ and for a distribution $D$ on $\mathcal{X}$, let $q(D) = \sum_{u \sim D} [q(u)]$.

For each $q \in Q$, let $\text{attr}(q) \subseteq [d]$ denote the attributes that $q$ depends on, i.e., this satisfies the property that, for any $x, x' \in \mathcal{X}$, if $x_i = x'_i$ for all $i \in \text{attr}(q)$, then $q(x) = q(x')$.

Theorem 12. $M_\ell$ in Algorithm 1 satisfies $\varepsilon_0$-$\nabla_0$CDP and simultaneously $\frac{1}{2}(\ell \cdot \varepsilon_0)^2$-$z$CDP and, for all inputs $x \in \mathcal{X}^n$, we have

$$\mathbb{E} \left[ \max_{q_1, \ldots, q_\ell \in Q} \max_{\text{attr}(q_i) \cap \text{attr}(q_j) = \emptyset} \frac{1}{\ell} \sum_{i=1}^\ell |q_i(A) - q_i(x)| \right] \leq \sqrt{\frac{2T}{n^2 \varepsilon_0^2}} + \frac{4 \log |\mathcal{X}|}{T \cdot \ell} + \frac{\sqrt{2T}}{\varepsilon_0 n} \log m.$$  

(2)
If \( T \) we sample versions of the projection mechanism [36].

prove an impossibility result. There has been some limited progress on devising efficient polynomial-time algorithm that comes close to matching the guarantees of MWEM, or to the special case of [80]. This computational hardness is also not specific to synthetic data [60]. However, for the special case of \( k \)-way conjunctions or parities it remains an open problem to devise a polynomial-time algorithm that comes close to matching the guarantees of MWEM, or to prove an impossibility result. There has been some limited progress on devising efficient versions of the projection mechanism [36].

\[ \text{Algorithm 1 Partial DP MWEM, } M_\ell : \mathcal{X}^n \to [0,1]^m. \]

1. **Input:** Private dataset \( x \in \mathcal{X}^n \).
2. **Parameters:** Set \( Q \) of \( m \) queries defined by functions \( q : \mathcal{X} \to [0,1] \), privacy parameter \( \varepsilon_0 > 0 \), iterations \( T \in \mathbb{N} \), and queries per round \( \ell \in \mathbb{N} \).
3. Set \( \varepsilon_T \leftarrow \varepsilon_0 / \sqrt{2T} \) and \( A_1 \leftarrow \text{uniform distribution on } \mathcal{X} \).
4. for \( t = 1, \ldots, T \) do

5. Select \( \{q_{t,1}, \ldots, q_{t,\ell}\} \subseteq Q \) with \( \text{attr}(q_{t,i}) \cap \text{attr}(q_{t,j}) = \emptyset \) for all \( 1 \leq i < j \leq \ell \) in a \( \varepsilon_T - \nabla_0 \text{CDP} \)

manner using the exponential mechanism, i.e.,

\[
P[\{q_{t,1}, \ldots, q_{t,\ell}\} = (q_1, \ldots, q_\ell)] \propto \exp \left( \varepsilon_T \frac{n}{\ell} \sum_{i=1}^\ell |q_i(A_t) - q_i(x)| \right).
\]

6. Answer \( q_{t,1}, \ldots, q_{t,\ell} \) with \( \varepsilon_T - \nabla_0 \text{CDP} \) by sampling \( a_{t,i} \leftarrow \mathcal{N} \left( q_{t,i}(x), \frac{1}{\varepsilon_T} \right) \) for all \( i \in [\ell] \).

(But, if \( a_{t,i} > 1 \), set \( a_{t,i} \leftarrow 1 \) and if \( a_{t,i} < 0 \), set \( a_{t,i} \leftarrow 0 \).)

7. Define the distribution \( A_{t+1} \) on \( \mathcal{X} \) by

\[
A_{t+1}(u) \propto A_t(u) \cdot \exp \left( \frac{1}{2} \sum_{i=1}^\ell q_{t,i}(u) \cdot (a_{t,i} - q_{t,i}(A_t)) \right).
\]

8. \( A \leftarrow \frac{1}{\ell} \sum_{t=1}^T A_t \).

9. **Output:** \( \frac{\mathcal{E}}{X \sim A} \{Q(X)\} \).

If \( T = \Theta \left( \frac{\sqrt{\log |\mathcal{X}|} \varepsilon n}{\sqrt{\ell} \log m} \right) \), then the RHS of (2) is \( O \left( \sqrt{\frac{\log |\mathcal{X}| \log m}{\sqrt{\ell} \varepsilon n}} \right) \).

Note that the LHS of (2) refers to the maximum average error over an arbitrary set of \( \ell \) attribute-disjoint queries; while this is a weaker bound than the maximum error over all individual queries, it is stronger than MSE. We leave it as an open problem to close this gap and obtain a standard maximum error bound.

The main difference between Algorithm 1 and standard MWEM is that in each iteration we sample \( \ell \) attribute-disjoint queries in parallel, rather than a single query; setting \( \ell = 1 \) recovers the standard MWEM algorithm and bound. For \( \ell > 1 \), the RHS of (2) gets smaller, but the LHS also changes.

For the case of \( k \)-way conjunctions on \( \mathcal{X} = \{0,1\}^d \), setting \( \ell = \lfloor d/k \rfloor \) obtains the error bound of \( \alpha = O \left( k^2 \sqrt{\frac{\log d}{\varepsilon n}} \right) \); in contrast, the corresponding bound under standard DP is \( \alpha = O \left( k^2 d^2 \sqrt{\frac{\log d}{\varepsilon n}} \right) \). Thus, the partial DP bound is near-independent of \( d \) and instead depends mainly on \( k \).

We remark that both the projection mechanism and the multiplicative weights exponential mechanism are not polynomial time algorithms in general. This limitation is not specific to partial DP. Under standard cryptographic assumptions it is known to be computationally infeasible to generate synthetic data (as Algorithm 1 does) even for 2-way conjunctions [80]. This computational hardness is also not specific to synthetic data [60]. However, for the special case of \( k \)-way conjunctions or parities it remains an open problem to devise a polynomial-time algorithm that comes close to matching the guarantees of MWEM, or to prove an impossibility result.
5 Histograms, Heavy Hitters & Applications

We consider the fundamental problem of computing a histogram or, equivalently, of computing the heavy hitters, which is well-studied in DP in various models and settings [35, 69, 58, 47, 51, 40, 10, 17, 16, 7, 8].

Definition 13 (Histogram Problem). In the histogram problem, the input dataset consists of $x_1, \ldots, x_n \in \{0, 1\}^d$; the frequency of an element $y \in \{0, 1\}$ is defined as $f_y := |\{i \in [n] : x_i = y\}|$. An algorithm is said to solve the histogram problem with (normalized $\ell_\infty$) error $\nu \in (0, 1)$ if it outputs $\{\hat{f}_y\}_{y \in \{0, 1\}^d}$ such that $\max_y |\hat{f}_y - f_y| \leq \nu n$.

Computing histograms is generally an easy task as far as privacy is concerned. (Although it can be challenging when we combine privacy with computational or communication constraints in a distributed setting.) The canonical algorithm is to add independent noise to each count $f_y$ and apply parallel composition – each individual only contributes to one count. To attain pure DP we would add Laplace noise; for CDP, Gaussian noise; and, for approximate DP, truncated Laplace or Gaussian noise.

However, this canonical algorithm very closely resembles the worst-case algorithm envisaged by the DP definition. Suppose an adversary wishes to determine whether or not $x_i = y$. That is, the adversary seeks to learn one bit about individual $i$. Often the histogram is sparse, so no other individual $j \in [n] \setminus \{i\}$ has $x_j = y$. Thus it suffices for the adversary to figure out whether $f_y = 1$ or whether $f_y = 0$. If we add Laplace noise to attain $\varepsilon$-DP with large $\varepsilon$, then the adversary can easily distinguish between these two cases. Namely, we would add $\xi_y \leftarrow \text{Laplace}(1/\varepsilon)$ to $f_y$ and $P[|\xi_y| \geq \frac{1}{2}] = e^{-\varepsilon/2}$. So, with probability $1 - e^{-\varepsilon/2}$, rounding the private value $f_y + \xi_y$ to the nearest integer returns the non-private value $f_y$. In particular, if $\varepsilon = 10$, rounding returns the true count with probability $> 99\%$.

Given that histograms are a well-studied problem and the canonical algorithm yields the kind of worst-case privacy outcomes that we want to avoid, it is natural to ask whether we can design a per-attribute partial DP algorithm for histograms that avoids this worst-case behaviour. In terms of our intuitive justification for tolerating large $\varepsilon$, the canonical algorithm is not “nice” and the question is whether we can devise a “nice” algorithm for histograms.

We restrict our attention to pure DP, both for simplicity and because this highlights the strength of our results – we are able to obtain these results under the most stringent form of DP. Naturally, our methods can be extended to CDP.

We provide the following result and a nearly-matching lower bound (Theorem 15).

Theorem 14. Let $n \geq O \left( \frac{1}{\varepsilon^2} \cdot \log(d/\eta) \cdot \log(1/\nu) \right)$. Then, there exists an $\varepsilon$-\(\nabla \theta\)DP algorithm that with probability $1 - \eta$ solves the histogram problem with error $\nu$. Moreover, the algorithm runs in expected time $\text{poly}(nd/(\eta \nu))$.

This sample complexity bound should be compared with the one in the standard $\varepsilon$-DP setting, which is $n = \Theta \left( \frac{d}{\varepsilon^2} \right)$. In other words, there is an exponential separation (in terms of the dimension) between the standard DP parameter and the per-attribute partial DP parameter.

\[ \text{The standard } \frac{1}{2} \varepsilon^2 \text{-zCDP sample complexity is } n = \Theta \left( \frac{d}{\varepsilon^2} \right) \text{ and under } (\varepsilon, \delta)\text{-DP it is } n = \Theta \left( \min \{d, \log(1/\delta)\} \right). \]
5.1 Our Algorithm

**Algorithm 2** PrivTree-Based Heavy Hitters.

**PrivHeavyHitter**
1: **Inputs:** $x^1, \ldots, x^n$.
2: **Parameters:** $\lambda, \tau, \mu > 0$.
3: for $j \in [\log d]$ do
4: \quad $L_{(j)} \leftarrow \{0, 1\}$.
5: for $\ell \in [\log d]$ do
6: \quad $\tau_\ell \leftarrow \tau + (\ell - 1)\mu$. \hfill \{Threshold\}
7: for $i \in \mathcal{I}$ do
8: \quad $L_i \leftarrow \emptyset$.
9: for $s \in L_{\text{left}} \cup L_{\text{right}}$ do
10: \quad $f_s^\ell \leftarrow \max\{f_s^\ell, \tau_\ell - \mu\} + \text{Lap}(\lambda)$.
11: if $f_s^\ell > \tau_\ell$ then
12: \quad Add $s$ to $L_i$.
13: **Output:** $L_{[d]}$.

In this section we design a novel algorithm for the histogram problem under per-attribute partial DP. Our algorithm will in fact output a succinct representation of $\{f_y\}_{y \in \{0, 1\}^d}$ by outputting a list $L \subset \{0, 1\}^d$ and $\{f_y\}_{y \in L}$, where $f_y = 0$ for every $y \notin L$ (this is needed for efficiency).

The main component of our algorithm is an $\varepsilon$-$\nabla_0$DP algorithm for finding heavy hitters – i.e., a list $L$ that is “not too large” and contains all $y$ with $f_y \geq \nu n$. The algorithm builds a binary tree over attributes, i.e., each node of the tree corresponds to an interval $I \subseteq [d]$ whose length is a power of two. Each node stores a list $L_I$ of heavy hitters among $x_1, \ldots, x_n$ restricted to the attributes indexed by $I$. Each leaf stores heavy hitters of a single attribute; then the next level nodes store heavy hitters among pairs of attributes; and the root has the final list of heavy hitters. The list $L_I$ can be constructed by estimating the substring frequencies of $y_1 \circ y_2$ for a shortlist of all pairs $y_1, y_2$ that appears as heavy hitters of its children. The key is that each attribute $j \in [d]$ only appears in $\log d$ intervals – one in each level of the tree. This means that we only have to divide the privacy budget over $\log d$ levels, resulting in a noise of roughly $O((\log d)/\varepsilon)$ per level. (Under standard DP, we would need to apply composition over all $2d - 1$ nodes, yielding noise scale $O(d/\varepsilon)$.)

Unfortunately, implementing the binary tree directly requires sample complexity $n \geq O_{\eta, \nu}((\log d)^2/\varepsilon)$ because we need to take a union bound over all $2d - 1$ nodes to bound the probability that a heavy hitter is erroneously dropped, which contributes another factor of $\log d$ in addition to the one we get from composition over levels. To get from here to $O_{\eta, \nu}(\log(d)/\varepsilon)$ as claimed in Theorem 14, we employ yet another technique to save on privacy loss introduced by Zhang et al. [82] for their PrivTree algorithm. The idea is roughly that when a count is far above the threshold, the privacy loss is actually much smaller than usual. Therefore, they introduce “biasing” and capping techniques, which can be thought of as lowering the threshold by a certain amount $\mu$ at each level and clipping the count to be at least the threshold minus $\mu$, respectively. We employ these ideas to shave a $\log d$ factor.

To formally describe our algorithm, we assume, without loss of generality, that $d$ is a power of two and define several additional notations:

- For any $I := \{a, \ldots, b\}$, we let $I_{\text{left}} := \{a, \ldots, \lfloor \frac{a + b}{2} \rfloor \}$ and $I_{\text{right}} := \{\lfloor \frac{a + b}{2} \rfloor + 1, \ldots, b\}$.
- For every $I := \{a, \ldots, b\}$ and $s \in \{0, 1\}^{|I|}$, we define the frequency of $s$ w.r.t. position $I$ as $f_s^I := |\{i \in [n] | x^i|_I = s\}|$. 

For every \( \ell \in [\log d] \), let \( \mathcal{I} \) denote the collection of all sets \( \{2^j(t-1) + 1, \ldots, 2^j t\} \) where \( t \in [d/2^\ell] \). Furthermore, we let \( \mathcal{I} := \bigcup_{\ell \in [\log d]} \mathcal{I}_\ell \).

For two sets \( S_1, S_2 \) of strings, let \( S_1 \circ S_2 \) denote the set of strings resulting from concatenating an element of \( S_1 \) and an element of \( S_2 \), i.e., \( S_1 \circ S_2 := \{ s_1 s_2 \mid s_1 \in S_1, s_2 \in S_2 \} \).

Algorithm 2 contains the complete description. Here, \( \text{Lap}(\cdot) \) is the Laplace noise. It is worth noting that we will eventually choose \( \lambda = O(1/\varepsilon) \) and \( \mu = O(\lambda \cdot \log(1/\nu)) \) where each big-O notation hides a sufficiently large constant. We next study the guarantees of this algorithm.

5.2 Analysis

We provide an overview of the privacy and utility analysis of Algorithm 2. The complete analysis is in the full version.

Privacy. For clarity, below we write the frequencies and lists as functions of the input datasets \( x \) or \( x' \). We first show the following: suppose \( \lambda, \mu \) are such that \( \mu > 1 \) and \( \frac{2}{\lambda} \left(1 + \frac{1}{1 - e^{-\lambda}}\right) \leq \varepsilon \), then \( \text{PrivHeavyHitter} \) is \( \varepsilon \)-\text{\text{DP}}. In fact, we will prove that even outputting all the elements of \( \mathcal{I} \) is \( \varepsilon \)-\text{\text{DP}}, i.e., we show that for any neighboring datasets \( x, x' \) and any values of \( (S_1) \in \mathcal{I} \), it holds that

\[
\Pr[\hat{f}(x) > \tau] \leq \varepsilon \cdot \Pr[\hat{f}(x') > \tau].
\]

To prove this statement, it suffices to consider the case where the differing elements in \( x \) and \( x' \) are 0 and 10_{\mu-1} \) respectively. We let \( I_\ell := \{1, \ldots, 2^\ell\} \) and write \( \hat{f}_s \) and \( f_s \) to mean \( \hat{f}_{I_\ell} \) and \( f_{I_\ell} \), for notational ease. We show that

\[
\prod_{\ell \in [\log d]} \frac{\Pr[\hat{f}(x) > \tau]}{\Pr[\hat{f}(x') > \tau]} \prod_{\ell \in [\log d]} \frac{\Pr[\hat{f}(x) > \tau]}{\Pr[\hat{f}(x') > \tau]}
\]

and bound each of the RHS terms by \( e^{\varepsilon/2} \). To bound the first term, since \( f_{0_{\ell-1}}(x) < f_{0_{\ell-1}}(x') \), we have \( \Pr[\hat{f}_{0_{\ell-1}}(x) > \tau] \leq \Pr[\hat{f}_{0_{\ell-1}}(x') > \tau] \). Therefore, it suffices to bound

\[
\prod_{\ell \in [\log d]} \Pr[\hat{f}_{0_{\ell-1}}(x) > \tau] = \prod_{\ell \in [\log d]} \Pr[\hat{f}_{0_{\ell-1}}(x') > \tau].
\]

Let \( L_0 \) be the smallest integer such that \( f_{0_{\ell-1}}(x) > \tau L_0 - \mu \). (i) For all \( \ell < L_0 \) we have \( \min \{ f_{0_{\ell-1}}(x), \tau - \mu \} = \tau - \mu = \min \{ f_{0_{\ell-1}}(x'), \tau - \mu \} \), and hence \( \forall \ell < L_0 \), \( \Pr[\hat{f}_{0_{\ell-1}}(x) > \tau] = 1 \). (ii) For \( \ell = L_0 \), notice that \( \min \{ f_{0_{\ell-1}}(x), \tau - \mu \} - \min \{ f_{0_{\ell-1}}(x'), \tau - \mu \} \leq 1 \). Hence, by the DP property of the Laplace mechanism,

\[
\frac{\Pr[\hat{f}_{0_{\ell-1}}(x) > \tau L_0]}{\Pr[\hat{f}_{0_{\ell-1}}(x') > \tau L_0]} \leq e^{1/\lambda}.
\]

For \( \ell > L_0 \), \( f_{0_{\ell-1}}(x') > \tau + (\ell - L_0) \mu - \tau \). Together,

\[
\frac{\Pr[\hat{f}_{0_{\ell-1}}(x) > \tau]}{\Pr[\hat{f}_{0_{\ell-1}}(x') > \tau]} \leq \frac{\Pr[(\ell - L_0) \mu + \text{Lap}(\lambda) > 0]}{\Pr[(\ell - L_0) \mu + \text{Lap}(\lambda) > 0]} \leq \exp \left( \frac{1}{\lambda} \cdot \exp \left( \frac{1 - (\ell - L_0) \mu}{\lambda} \right) \right),
\]

where the last step uses [82, Lemma 2.1]. From these inequalities, the bound of \( e^{\varepsilon/2} \) on the first term follows from our choice of \( \lambda, \mu \). The bound for the second term follows similarly.

Utility. We need to show that we discover all heavy hitters and that the expected list size is small; the latter will also imply that the expected running time of the algorithm is small. Let \( \eta, \nu \in (0, 0.1] \). Suppose that \( \tau = 0.5\eta \), \( \tau \geq 8\mu \log d + 8\lambda \log(d/(\eta \nu)) \) and \( \mu \geq \lambda \log(16/\nu) \). We show

(i) Heavy hitters discovered: W.p. \( 1 - 0.5\eta \), \( L_{|d|} \) contains all \( s \in \{0, 1\}^d \) such that \( f_s \geq 2\tau \).

(ii) Expected list size: \( \mathbb{E}[|L_{|d|}|] \leq 8/\nu \).
The expected running time of the algorithm is poly\( (d/\nu) \); this will follow from (ii).

To show (i), we fix any \( s \) such that \( f_s \geq 2\tau \). We will prove that \( \Pr[s \notin L] \leq 0.5\nu \); since there are at most \( 1/\nu \) such \( s \)'s, a union bound yields the claimed result. Indeed,

\[
\Pr[s \notin L[d]] = \Pr[3I \in \mathcal{I}, s \notin L] \leq \sum_{I \in \mathcal{I}} \Pr[s \notin L \mid s|_{I_{\text{left}}} \in L_{I_{\text{left}}} \land s|_{I_{\text{right}}} \in L_{I_{\text{right}}}] \leq \sum_{I \in \mathcal{I}} \Pr[2\tau + \operatorname{Lap}(\lambda) \leq \tau + \mu \log d] \leq \sum_{I \in \mathcal{I}} \Pr[0.5\tau + \operatorname{Lap}(\lambda) \leq 0] \leq \sum_{I \in \mathcal{I}} \exp(-0.5\tau/\lambda)/2 \leq 2d \cdot \exp(-0.5\tau/\lambda)/2 \leq 0.5\nu.
\]

We show (ii), by an induction on \( \ell \) that \( \mathbb{E}[|L_\ell|] \leq 8/\nu \) for all \( \ell \in \{0, \ldots, \log d\} \) and \( I \in \mathcal{I}_\ell \). Consider any \( \ell \in [\log d] \) and assume that the inductive hypothesis holds for \( \ell - 1 \) and any \( I' \in \mathcal{I}_{\ell - 1} \). Consider any \( I \in \mathcal{I}_\ell \) and let \( S := \{ s \in \{0, 1\}^{2\ell} \mid f_s^I > 0.5\tau \} \). Note that \( |S| < n/(0.5\tau) = 4/\nu \) and for any \( s \notin S \), we have \( f_s^I \leq \tau_\ell - \mu \). Hence, for any \( s \in (S_{I_{\text{left}}} \circ S_{I_{\text{right}}}) \setminus S \), we have \( \Pr[s \in L_I] \leq \nu/16. \)

\[
\mathbb{E}[|L_\ell|] = \mathbb{E}[|S \cap L_\ell|] + \mathbb{E}[|L_\ell \setminus S|] \leq |S| + \sum_{s \in (0, 1)^{2\ell} \setminus S} \Pr[s \in L_\ell] \\
\leq 4/\nu + \sum_{s \in (0, 1)^{2\ell} \setminus S} \nu/16 \cdot \Pr[s \in S_{I_{\text{left}}} \circ S_{I_{\text{right}}}] \\
\leq 4/\nu + \nu/16 \cdot \mathbb{E}[|L_{I_{\text{left}}} \circ L_{I_{\text{right}}}|] \leq 4/\nu + \nu/16 \cdot (8/\nu)(8/\nu) = 8/\nu,
\]

where the last inequality follows from induction.

## 5.3 Lower Bound

In this section we show that the \( \log d \) dependence in Theorem 14 necessary; our lower bound matches the upper bound up to the dependence on \( \eta \) and \( \log(1/\nu) \):

\[\textbf{Theorem 15.} \text{ Assume that } d \geq 10e^{1.1c}. \text{ If there exists an } \epsilon \cdot \nabla_0 \text{DP algorithm that with probability 0.1 solves the histogram problem with error } \nu, \text{ then } n \geq \Omega \left( \frac{1}{\epsilon^2 \nu} \log d \right).\]

The constant 0.1 in Theorem 15 was chosen for concreteness, but a similar statement holds for any positive constant. To prove Theorem 15, we follow the same packing-based approach that was used for the standard DP lower bound [47]. The difference is that our packing consists of only \( d \) one-hot vectors (instead of all of \( \{0, 1\}^d \)); since the one-hot vectors are at Hamming distance only 2 apart, the rest of the proof proceeds as before.

## 5.4 Applications of Histograms

Algorithms for histograms are often used as subroutines for other algorithms. As a concrete application of Theorem 14, we obtain partial DP algorithms for the problems of PAC learning point functions and threshold functions, and discrete distribution estimation with \( \ell_2 \)-error.

\[\textbf{Theorem 16.} \text{ For every } \epsilon, \alpha > 0, \text{ there exists an } \epsilon \cdot \nabla_0 \text{DP proper PAC learner with error at most } \alpha \text{ and with sample complexity } n = \tilde{O} \left( \frac{1}{\alpha \epsilon^2} \log d \right) \text{ for point functions and threshold functions.}\]

Theorem 16 should be contrasted with the sample complexity of proper PAC learning of point functions and threshold functions in the standard \( \epsilon \)-DP setting, both of which are \( n = \Theta \left( \frac{d}{\epsilon^2 \alpha} \right) \) [11, 41].
Theorem 17. For every $\varepsilon > 0$ and $n, d \in \mathbb{N}$, there exists an $\varepsilon$-\(\nabla\)DP algorithm for discrete distribution learning whose $\ell^2$-error is \(O\left(\frac{\log d}{\varepsilon^2} + \frac{1}{\varepsilon}\right)\) with probability at least 0.9.

In contrast, for standard DP, a packing lower bound \cite{47} shows that even getting an \(\ell^2\)-error of 0.1 (with constant probability) requires \(n \geq \Omega(d/\varepsilon)\).

6 Robust Learning of Halfspaces

We next consider the problem of robust learning of halfspaces, under the (normalized) Hamming distance. A halfspace is a function \(h_w : \mathbb{R}^d \rightarrow \{-1, +1\}\) where \(w \in \mathbb{R}^d\) and is defined as \(h_w(x) = \text{sign}((w, x))\). We consider the class \(H_{\text{halfspace}} := \{h_w \mid w \in \mathbb{R}^d\}\) of all halfspaces.

Our input dataset consists of pairs \((x_1, y_1), \ldots, (x_n, y_n)\) in \(X = \{-1, +1\}^d \times \{-1, +1\}\) drawn i.i.d. from a distribution \(D\) and our goal is to output a halfspace \(h_w \in H_{\text{halfspace}}\) that mislabels as small fraction of points w.r.t. \(D\) as possible – i.e., minimizing \(\mathbb{P}_{(x,y) \leftarrow D}[h_w(x) \neq y]\). We are interested in robust learning. A sample \((x, y)\) is \(\gamma\)-robustly classified if the hypothesis assigns all points in a \(\gamma d\)-radius Hamming ball around \(x\) to the label \(y\) – i.e., \(\forall \tilde{x} \in \{0,1\}^d : \|\tilde{x} - x\|_0 \leq \gamma d \implies h_w(\tilde{x}) = y\). More formally, the \(\gamma\)-error robust is defined as follows:

Definition 18 (Robust Error). For a distribution \(D\) on \(X = \{-1, +1\}^d \times \{-1, +1\}\) and a hypothesis \(h : \{-1, +1\}^d \rightarrow \{-1, +1\}\), we define its \(\gamma\)-(Hamming-)robust error to be

\[
\mathcal{R}_\gamma(h, D) := \mathbb{P}_{(x,y) \leftarrow D}[\exists \tilde{x} : h(\tilde{x}) \neq y \land \|\tilde{x} - x\|_0 \leq d\gamma].
\]

The goal is to find a halfspace whose \(\gamma\)'-robust error is not much more than the optimal \(\gamma\)-robust error. Here \(\gamma' < \gamma\) represents a relaxation in the margin that we pay for privacy, along with a relaxation in accuracy. The problem is well understood both in the non-private setting \cite{31} and in the standard DP setting \cite{43}. We give a partial DP algorithm:

Theorem 19. Let \(\varepsilon, \gamma, \gamma' \in (0, 1]\) such that \(\gamma > \gamma'\). There is an \(\varepsilon\)-\(\nabla\)DP algorithm \(M : X^n \rightarrow H_{\text{halfspace}}\) such that the following holds. Let \(D\) be a distribution on \(X\) and let \(S \leftarrow D^n\). Then

\[
\mathbb{E}[\mathcal{R}_\gamma'(M(S), D)] \leq \inf_{h \in H_{\text{halfspace}}} \mathcal{R}_\gamma(h, D) + O\left(\frac{1}{\varepsilon \sqrt{n}(\gamma - \gamma')^2} + \frac{\log(1/(\gamma - \gamma'))}{\varepsilon^2 n \cdot (\gamma - \gamma')}\right).
\]

Note that our error bound is independent of the number of attributes \(d\), while in the standard DP setting, the error must grow linearly with \(d\) \cite{43}.

Our algorithm first privatizes the label \(y \in \{-1, +1\}\) via Randomized Response; then, we can focus on a learner that is private in terms of the features \(x \in \{-1, +1\}^d\). Our learner is in fact an instantiation of the exponential mechanism \cite{65} except we do not apply it directly with respect to the (empirical) robust error, because for \(x\) that is exactly at distance \(\gamma d\) from the decision boundary, changing a single coordinate of \(x\) could make it be considered mislabeled under \(\gamma\)-robust error. Instead, we smoothen the loss based on how far \(x\) is from the decision boundary, similarly to the popular hinge loss, in order to reduce the sensitivity which gives the desired result. To describe our algorithm, it will be most clear to separate the privacy of the labels and the privacy of the samples. In this regards, we say that an algorithm is \(\varepsilon\)-sample-\(\nabla\)DP if the DP guarantee is only enforced on changing a single coordinate of a sample. (There is no privacy guarantee on the labels.)

For \(\gamma > \gamma' > 0\) and \(\alpha > 0\), we also say that a mechanism \(M\) is \((\gamma', \gamma)\)-learner with excess loss \(\alpha\) iff \(\mathbb{E}[\mathcal{R}_\gamma'(M(S), D)] \leq \inf_{h \in H_{\text{halfspace}}} \mathcal{R}_\gamma(h, D) + \alpha\).
6.1 From Sample-Only Privacy to Sample-and-Label Privacy

A first observation is that by using randomized response on the labels, we can immediately translate an $\varepsilon$-sample-\nDP algorithm to that of $\varepsilon$-\nDP.

Lemma 20. Let $\mathcal{H}$ be any hypothesis class and $\varepsilon \in (0, 1]$. Suppose that there is an $\varepsilon$-sample-\nDP ($\gamma, \gamma'$)-robust learner for $\mathcal{H}$ with excess loss $\alpha$. Then there is an $\varepsilon$-\nDP ($\gamma, \gamma'$)-robust learner for with excess loss $O(\alpha/\varepsilon)$ and the same sample complexity.

The above lemma essentially means that we can focus our attention to sample-\nDP learners for the rest of the section.

6.2 Robust Empirical Risk Minimization

For a set $S$ of labeled examples, we write $R_\gamma(h, S)$ to denote the $\gamma$-robust error w.r.t. the uniform distribution on $S$ (aka the empirical $\gamma$-robust error). Below we show that, by using a “smoothened” version of the loss similar to the hinge loss, we can get the following sample-\nDP ERM algorithm:

Lemma 21 (ERM for Robust Error). Let $\gamma' < \gamma$. For any finite hypothesis class $\mathcal{H}$, there exists an $\varepsilon$-sample-\nDP algorithm that outputs a hypothesis $h_{\text{priv}}$ such that

$$E[R_{\gamma'}(h_{\text{priv}}; S)] \leq \inf_{h \in \mathcal{H}} R_\gamma(h; S) + O\left(\frac{\log|\mathcal{H}|}{\varepsilon(\gamma - \gamma')d \cdot |S|}\right).$$

6.3 Robust Learning of Halfspaces: Reduction to Nets

We now turn our attention back to halfspaces. A first step is to notice that it suffices consider only halfspaces $w$ where $w$ belongs to some net. For $\nu > 0$, let $N(\nu)$ denote any $\nu$-net $N$ (under $\ell_1$ metric) of the unit $\ell_1$-ball and let $\mathcal{H}_{\text{halfspace}} := \{h_w \mid w \in N\}$. Our formal reduction is stated below.

Lemma 22. Suppose that there exists a $\varepsilon$-sample-\nDP ($\gamma' + \nu, \gamma'$)-robust learner for $\mathcal{H}_{\text{halfspace}}$ with excess loss $\alpha$. Then, there is also an $\varepsilon$-sample-\nDP ($\gamma' + 3\nu, \gamma'$)-robust learner for $\mathcal{H}_{\text{halfspace}}$ with excess loss $\alpha$ (where the sample complexity remains the same).

6.3.1 Private Robust Learner for Halfspaces

We start by providing a private learner for $\mathcal{H}_{\text{halfspace}}^{N(\nu)}$.

Lemma 23. There is an $\varepsilon$-sample-\nDP ($\gamma' + 2\nu, \gamma'$)-robust learner for $\mathcal{H}_{\text{halfspace}}^{N(\nu)}$ with sample complexity $O\left(\frac{1}{\nu^2} + \frac{\log(1/\varepsilon)}{\alpha \varepsilon^2}\right)$.

Combining the above lemma with Lemma 22 with $\nu = (\gamma - \gamma')/5$, we arrive at:

Lemma 24. There is an $\varepsilon$-sample-\nDP ($\gamma, \gamma'$)-robust learner for halfspaces with excess error

$$O\left(\frac{1}{(\gamma - \gamma')^2 \sqrt{n}} + \frac{\log(1/(\gamma - \gamma'))}{\varepsilon(\gamma - \gamma') \cdot n}\right).$$

Combining the above lemma with Lemma 20, we get Theorem 19.
7 Discussion

In this section, we discuss the meaning of partial DP. Ideally, of course, we would provide a standard DP guarantee with a small privacy loss bound (say, \((\varepsilon, \delta)-DP\) with \(\varepsilon \leq 1\) and \(\delta \leq 10^{-6}\)). However, in practice, we are seeing large privacy loss bounds \((\varepsilon \geq 10)\) and we lack a satisfactory way to interpret such guarantees.

Thus the premise of this discussion is that we are in a setting where, in order to provide reasonable utility, we need a large \(\varepsilon\) under the standard definition of \((\varepsilon, \delta)-DP\). The fundamental question is: How can we justify \((\varepsilon, \delta)-DP\) with large \(\varepsilon\)? And, even more importantly, when can we not justify this?

Intuitively, large \(\varepsilon\)s can be justified by informally arguing DP is a worst-case definition and this worst case is not realistic. The goal of partial DP is to provide a framework for formalizing this intuition for justifying large \(\varepsilon\) which is precise enough to also fail to justify large \(\varepsilon\) when the algorithm at hand does indeed exhibit worst-case behaviour. For example, if \(\varepsilon(x, x')\) is large when the only difference between \(x\) and \(x'\) is that the person visited a given website, then we clearly do not have a meaningful privacy guarantee.

Interpretation. Partial DP provides a language to formalize the intuitive notion of a “nice algorithm.” Specifically, it allows us to rule out algorithms that act like performing randomized response on some sensitive feature. For example, if we want to formalize the constraint that the algorithm does not reveal whether or not a given person has a certain disease, we would require that \(\varepsilon(x, x')\) is small whenever the only difference between \(x\) and \(x'\) is that person’s disease status.

To interpret a partial DP guarantee, we must also discuss what constitutes a “realistic adversary.” There are many different ways to restrict the adversary (see §7.1). Per-attribute partial DP naturally corresponds to assuming that the adversary is interested in learning a function of only a few attributes, whereas standard DP protects an arbitrary function of all the attributes of a person.\(^5\) E.g., if the dataset is employment records, we can assume that the adversary wishes to learn the target’s salary, but is not particularly interested in learning their age or whether or not they are an employee. Such assumptions can be justified in a variety of ways, depending on context. In the prior example, age may already be public information and the employer may be willing to disclose who is or is not an employee.

In general, the interpretation of partial DP is context-dependent; the effectiveness of the guarantee depends on what kind of information leakage is concerning.

Limitations. The limitations of (per-attribute) partial DP are the attacks that it does not give good protection against (beyond the baseline standard DP guarantee that partial DP implies). This limitation is inherent – if we want a context-independent guarantee of individual privacy, we cannot do better than the standard DP definition. Membership inference attacks [73, 39] are an example of a worst-case attack – whether a person is included in the dataset or excluded is a function of all the attributes and hence partial DP does not provide a better guarantee than standard DP. Whether membership of the dataset is sensitive depends on the context. If the data selection process itself reveals sensitive information, partial DP is not helpful. For example, if the dataset consists of only HIV patients, then membership inference can reveal that a participant is HIV positive. On the other hand, if

\(^5\) The quantitative guarantee will degrade gracefully with the number of attributes the adversary is interested in.
the dataset consists of all patients of a given hospital system (or a random sample of those patients), then that is potentially less sensitive and membership inference is less of a concern; this is a setting where partial DP may be meaningful.

In any case, providing a partial DP guarantee contains more information than the single parameter of the standard DP definition. Thus we argue that, even in settings where partial DP is not particularly appropriate, it is still no worse than simply providing a standard DP guarantee.

**Correlated Attributes.** To interpret the guarantee of per-attribute partial DP, we must also consider whether sensitive information is reflected in many attributes, as, in this case, the guarantee of per-attribute partial DP would rapidly degrade. For example, suppose each attribute is the person’s location at a given point in time. We would expect that the person is at home for extended periods of time, so the home location would be revealed in many attributes simultaneously. Partial DP would not be particularly useful in such a scenario. Other bad use cases for partial DP include the setting where each attribute is a text message or photo and a person can contribute many text messages and photos and sensitive information may be repeated in many of those messages or photos.

We consider the ideal setting for partial DP to be the case where attributes are heterogeneous (as opposed to the settings discussed in the previous paragraph where the attributes are homogeneous). For example, age, race/ethnicity, gender, home address, income, sexuality, medical status, occupation, criminal history, relationship status, commute length, and immigration status are heterogeneous attributes. These attributes are not independent, but the correlations between them are relatively weak. We can thus hope that partial DP provides meaningful guarantees on a dataset containing these attributes. In particular, it is unlikely that an adversary is interested in some complex function covering all or most of these attributes. A realistic adversary is likely only interested in one of these attributes or maybe a pair of them. While it is possible to, say, guess the income of a person based on their demographics, this is generally not considered to be a privacy violation [64, 15].

In general, privacy should be thought of in terms of causal relationships, not statistical correlations [79]. Indeed the definition of DP is precisely a causal property, as it considers a pair of datasets, which correspond to the real dataset and a hypothetical counterfactual dataset. The definition of DP does not make any distributional assumptions about the data. Correlations are present not only between the attributes of a single person, but also between the data of different people. For example, whether or not a given person has an infectious disease is highly correlated with whether or not the people around them have that disease. Thus revealing the fact that there is an outbreak of an infectious disease reveals information about specific individuals. But this is not a privacy violation. (And if we treated this as a privacy violation, it would prevent us from revealing useful information about disease outbreaks.) However, revealing a specific person’s test result is a potential privacy violation – the key is that there is a direct causal relationship between someone’s data (i.e., their test result) and the information being released. By the same token, a person may have many attributes that are correlated with having a certain disease, but releasing those correlated attributes is fundamentally different from releasing an actual diagnosis.

**Concrete Example: 2020 US Census.** The redistricting data from the 2020 US Census was released in a DP manner. The generally quoted guarantee is \((17.14, 10^{-10})\)-DP plus \((2.47, 10^{-10})\)-DP [1] and applying basic composition to these two releases gives \(\varepsilon = 19.61\). To be precise, the redistricting data satisfies \((2.56 + 0.07)\)-zCDP, which implies \((13.8, 10^{-6})\)-DP.
The Census Bureau provide a detailed breakdown of the privacy allocation [1, 21, 20]. This exactly corresponds to a partial CDP guarantee. Their TopDown algorithm computes multiple histograms across subsets of attributes (which bears some similarity to our heavy hitters algorithm in Section 5). To determine the $\varepsilon$-CDP guarantee, for any $x, x' \in X$, determine the set of attributes on which they differ and then we look at which histograms involve those attributes to determine $\varepsilon(x, x')$. Histograms that only involve attributes on which $x$ and $x'$ agree need not be accounted for under partial DP.

To make this partial CDP guarantee concrete, we can relate it to a specific attack. The Census Bureau conducted a simulated reconstruction and reidentification experiment [28]. The punchline of their simulated attack was learning people’s race and ethnicity from the data that was publicly released from the 2010 US Census. We can calculate a privacy guarantee for just these two attributes in the 2020 release. Specifically, if we allow a person’s race and ethnicity to change, but their other attributes are fixed, then we get a $1.02-z$CDP guarantee, which yields $(7.85, 10^{-6})$-DP. That is, in terms of partial DP $\varepsilon(x, x') \leq 1.02$ when $x$ and $x'$ differ only on the race and ethnicity fields.

Generality of Partial DP Definition. Our algorithmic results (in Sections 4, 5, & 6) focus on per-attribute partial DP. For simplicity, we assume each attribute has the same privacy parameter $\varepsilon_0$. But not all attributes are equally sensitive. Thus it is natural to consider per-attribute guarantees where each attribute has its own privacy parameters. In the 2020 US Census example, the attributes have different privacy parameters. Our definition of partial DP is general enough to capture such non-uniform per-attribute privacy guarantees.

As mentioned in Section 3, it is possible to give an even more general definition than we do, where the $\varepsilon$ metric considers a pair of datasets, not just a pair of individual records. While such added generality may seem like a feature, it makes such a definition harder to interpret. In particular, it becomes hard to relate such a definition back to standard DP. Thus we deliberately chose not to define partial DP so generally.

7.1 Assumptions about the Adversary

To give refined privacy guarantees (e.g., partial DP) meaning, we must give a characterization of what we might consider reasonable restricted adversaries, which we can then use to interpret our definition. In this section, we discuss different types of restricted adversaries.

The definition of DP does not explicitly mention an adversary; it simply states that the output distribution of the algorithm does not change much (as measured by $\varepsilon$) if we arbitrarily change the data of one individual in the input of the algorithm. However, to interpret this definition and give meaning to the privacy guarantee, we must envisage an adversary who sees the output of the algorithm, combines this information with their knowledge, and thereby potentially learns a piece of information about an individual that they should not have been able to learn. The adversary could be a stranger, a close friend or relative, a government entity, or a private entity we do business with and each of these potential adversaries will have different knowledge, resources, and goals.

In effect, standard DP makes minimal assumptions about the adversary – the adversary can have near-complete knowledge of the dataset and can target an arbitrary piece of information about an arbitrary individual.

There are four ways in which we could make assumptions that constrain the adversary:

(i) Assumptions about the Adversary’s Knowledge. DP effectively permits the adversary to know everything about the dataset except for the one bit of private information that they are seeking to extract. Although the adversary may have access to a lot of information from
auxiliary data sources, it is unrealistic to assume that this information is so complete and so accurate. Thus it is natural to assume some uncertainty about the dataset in the eyes of the adversary; this could be formalized by endowing the dataset with randomness and exploiting this randomness in the privacy guarantee [12, 9, 13].

However, such assumptions about the adversary’s knowledge are very brittle [74, 75]. In particular, assumptions about the adversary’s knowledge can be invalidated by future releases of information. That is, we may assume that certain information is unknown to the adversary, but subsequently an auxiliary dataset is made available that contains this information; when that happens, it is too late to retract the output of our algorithm. Such assumptions are also not robust to composition. That is, the output of our algorithm may itself invalidate these assumptions, so, if we run another algorithm subsequently, we cannot make the same assumptions again.

It is also difficult to effectively formulate assumptions about the adversary’s knowledge. For example, it is tempting to assume that the data consists of i.i.d. samples from some nice distribution. However, this corresponds to assuming an entirely naïve adversary with no knowledge of the dataset beyond the general characteristics of the population it came from.

(ii) Assumptions about the Target Individual or Dataset. Distributional assumptions about the data can also encode a different type of privacy guarantee (as opposed to that distribution representing the uncertainty of the adversary). Intuitively, we can encode the assumption that the adversary only targets “typical” individuals in “typical” datasets and the privacy guarantee may fail for individual outliers or abnormal datasets. For example, a basic DP algorithm is to add noise to some statistic that is scaled to its sensitivity; an average-case assumption about the target individual and dataset would allow us to replace the worst-case sensitivity with a notion of average-case sensitivity [44, 78]. However, an individual deviating significantly from the rest of the dataset will have a correspondingly weaker privacy guarantee [81].

This approach has two deficiencies: First, providing unequal privacy protection raises ethical questions. Second, it may be unnecessary to make this compromise. For example, techniques such as clipping can control the worst-case sensitivity or we can use smooth sensitivity [71]. We can also test whether the dataset is typical before performing the analysis and abort if it is not [34]. Thus it is often possible to obtain the benefits of average-case assumptions on the data while still attaining standard DP.

(iii) Assumptions about the Adversary’s Capabilities. Performing a privacy attack generally requires effort. Thus we may make assumptions about the adversary’s ability or willingness to perform the attack [26]. A good example is computational DP [68], where we assume that the adversary’s computational power is limited and thus they cannot, for example, break a cryptographic system.

We can also assume that the adversary will only perform certain types of attacks. For example, k-anonymity and related definitions are tailored to preventing a specific style of record-linkage attacks. In the same vein, the data curator can simulate an attack on the output of their algorithm [23, 52, 22]. On one hand, the success of the simulated attack establishes that the algorithm is not DP – and this can be used to check the privacy analysis [77]. On the other hand, the failure of the simulated attack establishes a privacy guarantee that is meaningful as long as the real adversary is similar to the simulated adversary.
(iv) Assumptions about the Adversary’s Goals. DP protects against an adversary seeking to learn an arbitrary one-bit function of the target individual’s data. Equivalently (up to a factor of two in the privacy parameter), it prevents the adversary from learning whether or not the target individual’s data was included in the dataset. While being included in the dataset may be sensitive depending on how the dataset was collected [73, 39], this often does not correspond to a realistic threat.

Thus we can relax the definition to protect only certain pieces of information from attacks. This corresponds to making an assumption about what function the adversary wants to learn about the target individual. For example, if the dataset corresponds to employment records, we can assume that the adversary wishes to learn the target’s salary or their performance rating, but is not particularly interested in learning their age or whether or not they are an employee. Such assumptions can be justified in a variety of ways. In the prior example, it may be the case that age is already public information and that the employer is willing to disclose who is or is not an employee.

Our partial DP approach corresponds to making assumptions about the adversary’s goals. Specifically, we provide guarantees for adversaries whose goal is to learn a single attribute or a function of a few attributes. Whether such an assumption corresponds to a realistic adversary will depend on the application domain.

7.2 Bayesian & Information-Theoretic Interpretations

Applying Bayes’ law to the standard DP definition gives us a “semantic” interpretation of the guarantee – regardless of the adversary’s prior beliefs, after seeing the DP output, their posterior beliefs cannot change much based on a single person’s data [54]. We can give a similar interpretation for pure partial DP:

Proposition 25. Let $X = X_1 \times \cdots \times X_d$. Let $M : X^n \rightarrow Y$ satisfy $\varepsilon$-DP. Let $P$ be a distribution on $X^n$ representing the adversary’s prior beliefs. Assume that for some $s = \{j_1, \cdots, j_{|s|}\} \subset [d]$, the distribution $P$ can be decomposed as a product distribution over the attributes given by $s$ and the remaining $d - |s|$ attributes. Let $P_{X|M(X)=y}$ denote the conditional distribution of $X$ obtained by drawing $X \leftarrow P$ and conditioning on the event $M(X) = y$ for some fixed $y \in Y$. Similarly, let $P_{X|M(X_{-i})=y}$ denote the conditional distribution of $X$ obtained by drawing $X \leftarrow P$ and conditioning on the event $M(X_{-i}) = y$ for some fixed $y \in Y$ and $i \in [n]$, where $X_{-i}$ denotes $X$ with the $i$th entry removed or blanked. Let $P_{X|y}$ denote the marginal distribution on $X_{j_1} \times \cdots \times X_{j_{|s|}}$ obtained by sampling $X \leftarrow P$ and only revealing the attributes indexed by $s$. Define $P_{X|M(X)=y}$ and $P_{X|M(X_{-i})=y}$ analogously as marginals of the conditional distributions. Then, for all $y \in Y$ and all $i \in [n]$,

$$d_{TV}(P_{X|M(X)=y}, P_{X|M(X_{-i})=y}) \leq e^{2|s|\varepsilon} - 1.$$  

The proof of Proposition 25 directly follows that of [54]. The assumptions of the proposition allow us to essentially ignore all the attributes indexed by $[d] \setminus s$ and, once we discard those irrelevant attributes, we have a $(|s|\varepsilon)$-DP algorithm.

We remark that the product distribution assumption may seem strong, as it implies that there is no connection between the attributes in $s$ and the attributes in $[d] \setminus s$. However, there is one very simple way that this can arise: Suppose the adversary already knows the value of all of the attributes in $[d] \setminus s$. In this case we have a trivial product distribution where the distribution on the attributes in $[d] \setminus s$ is a point mass.

We can also give an information-theoretic interpretation: Suppose $X_1, \cdots, X_n \in X = X_1 \times \cdots \times X_d$ are independent random variables. These correspond to the data of $n$ individuals. The independence assumption is essentially saying that the adversary knows the population,
but knows neither the individuals nor the relationships between the individuals. If $M$ is $\varepsilon$-DP or $\frac{1}{2}\varepsilon^2$-zCDP, then $I(X_i; M(X)) \leq \frac{1}{2}\varepsilon^2$ for all $i \in [n]$ [18]. Here $I(\cdot; \cdot)$ denotes the mutual information in nats. That is to say, DP bounds the amount of information that $M$ reveals about any given record in the dataset. Such a bound can be meaningful even in the large $\varepsilon$ regime; if the individual’s data is high-entropy, then we cannot reconstruct it, even if we can learn some of it [13]. We can give a stronger guarantee under partial DP:

> Proposition 26. Suppose $X_1, \ldots, X_n \in X = X_1 \times \cdots \times X_d$ are independent random variables. Fix $i \in [n]$. Suppose we can partition $[d] = s_1 \cup \cdots \cup s_k$ such that $X_{i,s_1}, \ldots, X_{i,s_k}$ are independent random variables, where $X_{i,s_j}$ denotes the attributes indexed by $s_j$ of individual $i$. Let $M : X^n \to Y$ satisfy $\varepsilon$-$\nabla_0$CDP. Then $I(X_{i,s_j}; M(X)) \leq \frac{1}{2}\varepsilon^2|s_j|^2$ for all $j \in [k]$ and $I(X_i, M(X)) \leq \frac{1}{2}\varepsilon^2 \sum_{j=1}^{k} |s_j|^2$.

8 Conclusion

We have presented several algorithms and analyzed them under partial DP, which gives more granular privacy guarantees than standard DP. Our results demonstrate that there are multiple separations between the achievable per-attribute $\varepsilon_0$-$\nabla_0$DP and the per-person $\varepsilon$-DP parameters – i.e., settings where the achievable per-person parameter is large (say, $\varepsilon \geq 10$), but we can still give more granular guarantees with a smaller parameter (e.g., $\varepsilon_0 \leq 1$). In this case the partial DP guarantee with a small $\varepsilon_0$ may be a meaningful and useful (depending on the context) supplement to standard DP guarantee with a large $\varepsilon$, as the smaller parameter can interpreted more easily, albeit on a per-attribute basis.

We note that interpreting the guarantees of partial DP depends on what constitutes a “realistic” adversary. Per-attribute partial DP naturally corresponds to assuming that the adversary is interested in learning a function of only a few attributes. On the flip side, partial DP may not provide better guarantees than standard DP for membership inference attacks and in cases where sensitive information may be repeated across multiple attributes.

We hope that our work inspires further exploration of more granular privacy guarantees, and further expansion of the DP algorithmic toolkit.

8.1 Further work

We hope that our work inspires further study of refined privacy guarantees. Our partial DP framework can and should be explored further, both in terms of developing and applying algorithms and in terms of further developing the definition. In particular, most of our results are restricted to per-attribute partial DP. A natural extension is to have a different $\varepsilon_0$ for each attribute, as some attributes are more sensitive than others.

A specific open question for algorithms development is to improve Theorem 12 so that we can obtain max error guarantees under per-attribute partial DP (like in the standard DP setting), rather than needing to average over a set of $\ell$ queries.

Going beyond partial DP, there is scope for other refined definitions that capture some limitations on the adversary. We have discussed some possible directions in Section 7.1. To facilitate such work, we propose the following desiderata for other refined definitions:

- The adversary should not be “baked in” to the privacy definition. That is, the definition should be stated in a way that can be interpreted and verified without knowing the specifics of the adversary. Our definition, like the original DP definition [35], is frequentist, rather than Bayesian – i.e., it does not mention the adversary’s beliefs and instead asserts that the output distributions are indistinguishable; this makes it easier to use.
The privacy definition should not be overly tailored to a specific algorithm. It is important to disentangle algorithms from definitions; this is an important conceptual contribution of the original DP definition. In other words, definitions should be re-usable.

The privacy definition should be formally comparable to the standard definition of DP (as we show in Proposition 8) or, at least, it should be possible to ensure that algorithms satisfy both definitions simultaneously. We believe that the goal of such research should not be to replace the standard DP definition, rather the goal should be to supplement it.

In order for a new privacy definition to be useful, we must demonstrate a quantitative separation between it and the standard definition of DP, as we have done with our algorithmic results. A new definition should not be a substitute for designing better algorithms. For example, when computing the mean of unbounded Gaussian data we could either devise an average-case privacy definition to avoid dealing with the infinite global sensitivity of the mean, or we could simply clip the data [53]; we argue that the second option is vastly preferable. Thus, to justify a new privacy definition, new algorithmic results should be matched to a lower bound showing that it is impossible to match the performance under the usual definition of DP.

References

1. John M Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson Garfinkel, Micah Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, et al. The 2020 census disclosure avoidance system topdown algorithm. arXiv, 2022. arXiv:2204.08986.
2. Jayadev Acharya, Kallista Bonawitz, Peter Kairouz, Daniel Ramage, and Ziteng Sun. Context aware local differential privacy. In ICML, pages 52–62, 2020.
3. Faraz Ahmed, Alex X Liu, and Rong Jin. Social graph publishing with privacy guarantees. In ICDCS, pages 447–456, 2016.
4. Márió Alvim, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Anna Pazii. Local differential privacy on metric spaces: optimizing the trade-off with utility. In CSF, pages 262–267, 2018.
5. Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In CCS, pages 901–914, 2013.
6. Hilal Asi, John Duchi, and Omid Javidbakht. Element level differential privacy: The right granularity of privacy. arXiv, 2019. arXiv:1912.04042.
7. Victor Balcer and Albert Cheu. Separating local & shuffled differential privacy via histograms. In ITC, pages 1:1–1:14, 2020.
8. Victor Balcer, Albert Cheu, Matthew Joseph, and Jieming Mao. Connecting robust shuffle privacy and pan-privacy. In SODA, pages 2384–2403, 2021.
9. Raef Bassily, Adam Groce, Jonathan Katz, and Adam Smith. Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy. In FOCS, pages 439–448, 2013.
10. Raef Bassily and Adam D. Smith. Local, private, efficient protocols for succinct histograms. In STOC, pages 127–135, 2015.
11. Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of pure private learners. JMLR, 20:146:1–146:33, 2019.
12. Raghav Bhaskar, Abhishek Bhowmick, Vipul Goyal, Srivatsan Laxman, and Abhradeep Thakurta. Noiseless database privacy. In ASIACRYPT, pages 215–232, 2011.
13. Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. arXiv, 2018. arXiv:1812.00984.
14. Jaroslaw Blasiok, Mark Bun, Aleksandar Nikolov, and Thomas Steinke. Towards instance-optimal private query release. In SODA, pages 2480–2497, 2019.
Mark Bun, Damien Desfontaines, Cynthia Dwork, Moni Naor, Kobbi Nissim, Aaron Roth, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Statistical inference is not a privacy violation. DifferentialPrivacy.org, June 2021. URL: https://differentialprivacy.org/inference-is-not-a-privacy-violation/.

Mark Bun, Jelani Nelson, and Uri Stemmer. Heavy hitters and the structure of local privacy. TALG, 15(4):1–40, 2019.

Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. JMLR, 20:94:1–94:34, 2019.

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In TCC, pages 635–658, 2016.

Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In STOC, pages 1–10, 2014.

US Census Bureau. Census bureau sets key parameters to protect privacy in 2020 census results, 2021. URL: https://www.census.gov/newsroom/press-releases/2021/2020-census-key-parameters.html.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In S & P, pages 1897–1914, 2022.

Nicholas Carlini, Chang Liu, Ulfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In USENIX, pages 267–284, 2019.

Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In PETS, pages 82–102, 2013.

Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In COLT, pages 155–186, 2011.

Kamalika Chaudhuri, Jacob Imola, and Ashwin Machanavajjhala. Capacity bounded differential privacy. In NeurIPS, pages 3469–3478, 2019.

Rachel Cummings, Vitaly Feldman, Audra McMillan, and Kunal Talwar. Mean estimation with user-level privacy under data heterogeneity. In NeurIPS 2021 Workshop Privacy in Machine Learning, 2021.

Damien Desfontaines. Demystifying the US census bureau’s reconstruction attack. https://desfontain.es/privacy/us-census-reconstruction-attack.html, May 2021. Ted is writing things (personal blog).

Damien Desfontaines. A list of real-world uses of differential privacy, 2021. URL: https://desfontain.es/privacy/reall-world-differential-privacy.html.

Ilias Diakonikolas, Daniel M. Kane, and Pasin Manurangsi. The complexity of adversarially robust proper learning of halfspaces with agnostic noise. In NeurIPS, 2020.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. arXiv, 2019. arXiv:1905.02383.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In TCC, pages 486–503, 2006.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In STOC, pages 371–380, 2009.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In TCC, pages 265–284, 2006.
Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Using convex relaxations for efficiently and privately releasing marginals. In SoCG, pages 261–270, 2014.

Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. Discrete & Computational Geometry, 53(3):650–673, 2015.

Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. arXiv, 2016. arXiv:1603.01887.

Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In FOCS, pages 650–669, 2015.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level differentially private learning via correlated sampling. NeurIPS, 2021.

Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Random differential privacy. J. Priv. Confidentiality, 4(2), 2013.

Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. SIAM J. Comput., 44(6):1740–1764, 2015.

Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Thao Nguyen. Robust and private learning of halfspaces. In AISTATS, pages 1603–1611, 2021.

Shiva P Kasiviswanathan and Adam Smith. On the ‘semantics’ of differential privacy: A Bayesian formulation. J. Priv. Confidentiality, 6(1), 2014.
Daniel Levy, Ziteng Sun, Kareem Amin, Satyen Kale, Alex Kulesza, Mehryar Mohri, and Ananda Theertha Suresh. Learning with user-level privacy. In *NeurIPS*, 2021.

Yuhan Liu, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and Michael Riley. Learning discrete distributions: user vs item-level privacy. In *NeurIPS*, 2020.

Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *CACM*, 53(9):89–97, 2010.

Frank McSherry. Lunchtime for data privacy, 2016. URL: https://github.com/frankmcsherry/blog/blob/master/posts/2016-08-16.md.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

Solomon Messing, Christina DeGregorio, Bennett Hillenbrand, Gary King, Saurav Mahanti, Zagreb Mukerjee, Chaya Nayak, Nate Persily, Bogdan State, and Arjun Wilkins. Facebook Privacy-Protected Full URLs Data Set, 2020. doi:10.7910/DVN/TDOAPG.

Ilya Mironov. Rényi differential privacy. In *CSF*, pages 263–275, 2017.

Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational differential privacy. In *CRYPTO*, pages 126–142, 2009.

Nina Mishra and Mark Sandler. Privacy via pseudorandom sketches. In *PODS*, pages 143–152, 2006.

Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the approximate and sparse cases. In *STOC*, 2013.

Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84, 2007.

Alfréd Rényi. On measures of entropy and information. In *4th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561, 1961.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *S & P*, pages 3–18, 2017.

Thomas Steinke and Jonathan Ullman. The pitfalls of average-case differential privacy. DifferentialPrivacy.org, July 2020. URL: https://differentialprivacy.org/average-case-dp/.

Thomas Steinke and Jonathan Ullman. Why privacy needs composition. DifferentialPrivacy.org, August 2020. URL: https://differentialprivacy.org/privacy-composition/.

Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in Apple’s implementation of differential privacy on MacOS 10.12. *arXiv*, 2017. *arXiv*: 1709.02753.

Florian Tramer, Andreas Terzis, Thomas Steinke, Shuang Song, Matthew Jagielski, and Nicholas Carlini. Debugging differential privacy: A case study for privacy auditing. *arXiv*, 2022. *arXiv*:2202.12219.

Aleksei Triastcyn and Boi Faltings. Bayesian differential privacy for machine learning. In *ICML*, pages 9583–9592, 2020.

Michael Carl Tschantz, Shayak Sen, and Anupam Datta. Sok: Differential privacy as a causal property. In *S & P*, pages 354–371, 2020.

Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating synthetic data. In *TCC*, pages 572–587, 2011.

Yu-Xiang Wang. Per-instance differential privacy. *J. Priv. Confidentiality*, 9(1), 2019.

Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *SIGMOD*, pages 155–170, 2016.