An Uncertainty Principle is a Price of Privacy-Preserving Microdata

Anonymous Author(s)
Affiliation
Address
email

Abstract

Privacy-protected microdata is often the desired output of a differentially private algorithm since microdata is a familiar and convenient format for downstream users. However, there is a statistical price for this kind of convenience. We show that there is a kind of uncertainty principle that governs the trade-off between accuracy of a population of interest ("sum query") vs. accuracy of its component sub-populations ("point queries"). Compared to differentially private query answering systems that are not required to produce microdata, accuracy can degrade by a logarithmic factor. For example, in the case of pure differential privacy, without the microdata requirement, one can provide noisy answers to the sum query and all point queries while guaranteeing that each answer has squared error $O(1/\epsilon^2)$. With the microdata requirement, one must choose between allowing an additional $\log^2(d)$ factor (d is the number of point queries) for some point queries or allowing an extra $O(d^2)$ factor for the sum query. We present lower bounds for pure, approximate, and concentrated differential privacy. We propose mitigation strategies and create a collection of benchmark datasets that can be used for public study of this problem.

1 Introduction

Differential Privacy [14] is a mathematical theory of information leakage that allows organizations to publish noisy statistics about their datasets while protecting the confidentiality of user information. Its state-of-the-art guarantees have resulted in adoption by data collectors such as the U.S. Census Bureau [27, 8, 20, 1], Google [17, 6], Apple [32], Microsoft [11], Uber [23], and Facebook [29].

In many cases, downstream users desire the output of disclosure avoidance systems to take the form of microdata. For example, this has historically been the case for Census data, and is expected to be the case for the upcoming 2020 Decennial Census of Population and Housing [18]. However, an end-user study of the data demonstration products released by an early prototype of the Census Bureau's disclosure avoidance system showed significant anomalies in the privacy-preserving microdata [30]. They noted the following: the system first produced differentially private noisy query answers, called *measurements*, and then synthesized privacy-preserving microdata so that query answers computed from the privacy-protected microdata matched the noisy measurements as closely as possible (based on some objective function). However, after the privacy-protected microdata were created, they compared (1) the original measurement query noisy answers and (2) the values of the same queries computed from the privacy-protected microdata. They noted that in some cases, the query error from the privacy-protected microdata was "much larger" than the measurement query error [30].

In this paper, we show that such kinds of anomalies are an inherent and unavoidable consequence of privacy-protected microdata. We further show that the additional errors caused by privacy-protected microdata also satisfy a *new* uncertainty principle that trades off error between accuracy on populations and accuracy on sub-populations. We next explain this principle.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

First, our criterion is *per-query* expected squared error. That is, if Q is a collection of queries, $\mathfrak{D}$ is the true data, and $\widetilde{\mathfrak{D}}$ is the privacy-protected microdata, we are interested in $\max_{q \in Q} E_{\widetilde{\mathfrak{D}}}[(q(\mathfrak{D}) - q(\widetilde{\mathfrak{D}}))^2]$, where the randomness comes from the algorithm that ingests $\mathfrak{D}$ and outputs privacy protected $\widetilde{\mathfrak{D}}$. This metric measures whether there exist "bad" queries that have systematically large errors *on average*. It is *not* to be confused with outlier noise error: $E_{\widetilde{\mathfrak{D}}}[\max_{q \in Q}(q(\mathfrak{D}) - q(\widetilde{\mathfrak{D}}))^2]$.

Next, consider a collection of d disjoint[1] counting queries $q_1, \ldots, q_d$ and a special query q_* that is equal to their sum ($q_*(\mathfrak{D}) = \sum_i q_i(\mathfrak{D})$). We call $q_1, \ldots, q_d$ the *point queries* and q_* the *sum* query. Examples include (1) $q_*(\mathfrak{D})$ = "# of Black or African Americans in the data living in California" and $q_i(\mathfrak{D})$ = "# of Black of African Americans in the data living in county i in California" and (2) $q_*(\mathfrak{D})$ = "population of a given county" (which can be used in federal and state-level funding allocations) and $q_i(\mathfrak{D})$ = "population in Census block i in that county" (useful for redistricting). Thus, for different use-cases, accuracies at these local and aggregate scales are important.

It is well-known that queries $q_1, \ldots, q_d, q_*$ can be answered using ϵ-differential privacy by adding Laplace$(2/\epsilon)$ noise to each query [16], thus guaranteeing that each query answer has expected squared error $8/\epsilon^2$. However, in this paper, we show that it is not possible to guarantee this kind of error if one is required to produce differentially private microdata $\widetilde{\mathfrak{D}}$ and answer queries using it (i.e., computing $q_1(\widetilde{\mathfrak{D}}), \ldots, q_d(\widetilde{\mathfrak{D}}), q_*(\widetilde{\mathfrak{D}})$). Specifically, suppose an ϵ-differentially private microdata-producing algorithm can guarantee that, for all datasets $\mathfrak{D}$, $E_{\widetilde{\mathfrak{D}}}[(q_*(\mathfrak{D}) - q_*(\widetilde{\mathfrak{D}}))^2] \leq D^2$ and $\max_i E_{\widetilde{\mathfrak{D}}}[(q_i(\mathfrak{D}) - q_i(\widetilde{\mathfrak{D}}))^2] \leq C^2$ for some constants C and D. Then one has to choose:

- If $D^2 \in O(1/\epsilon^2)$ then $C^2 \in \Omega(\frac{1}{\epsilon^2}\log^2(d))$. That is, making the sum query accurate may force us to take a $\log^2(d)$ penalty in the expected squared error some of the point queries, or
- If $C^2 \in O(1/\epsilon^2)$ then $D^2 \in \Omega(\frac{d^2}{\epsilon^2})$. That is, a low per-query error guarantee for point queries may increase expected squared error of the sum query by a factor of d^2.

We present such lower bound results for pure differential privacy [14], approximate differential privacy [13], and concentrated differential privacy [7], with nearly matching upper bounds.

We note that this uncertainty principle affects some, but not all, possible datasets. That is, there are datasets for which the error penalties do not exist. Thus, the goal in practical privacy-protected microdata generation should be to minimize the occurrence of this uncertainty principle (since eliminating it entirely is impossible). To this end, we propose a benchmark suite of real and synthetic datasets that can be used by the wider community for further study of this problem. We also propose some algorithms, inspired by our lower and upper bound proofs, for mitigating the effects of this uncertainty principle. Limitations: empirically, these algorithms perform well on the benchmarks but we do not have theoretical proofs of performance.

2 Preliminaries

Let $\mathfrak{D}$ denote a dataset, M a differentially private algorithm, and let $\widetilde{\mathfrak{D}}$ be a privacy-preserving dataset (e.g., $M(\mathfrak{D}) = \widetilde{\mathfrak{D}}$). A counting query q is associated with a predicate ψ, and the query answer $q(\mathfrak{D})$ is the number of records in $\mathfrak{D}$ that satisfy ψ. We let $q_1, \ldots, q_d$ represent a set of d counting queries whose corresponding predicates $\psi_1, \ldots, \psi_d$ are **disjoint** (no record can satisfy more than one of the predicates). We also let q_* denote their sum: $q_*(\mathfrak{D}) = \sum_{i=1}^{d} q_i(\mathfrak{D})$.

2.1 Differential Privacy

Differential privacy is currently considered the gold standard in privacy protections. It relies on the concept of neighboring datasets, defined as follows.

Definition 1 (Neighbors)**.** *Two datasets $\mathfrak{D}_1$ and $\mathfrak{D}_2$ are neighbors, denoted by $\mathfrak{D}_1 \sim \mathfrak{D}_2$, if $\mathfrak{D}_1$ can be obtained from $\mathfrak{D}_2$ by adding or removing one record.*

Using this concept of neighbors, differential privacy ensures that adding or removing one record from a dataset has little effect on the probabilistic outcomes of an algorithm:

[1] That is, adding/removing a record into the data can only affect the answer to one of the queries.

Definition 2 (Differential Privacy [14]). *Given privacy parameters $\epsilon > 0$ and $\delta \geq 0$, a randomized algorithm M satisfies (ϵ, δ)-DP if for all pairs of datasets $\mathfrak{D}_1, \mathfrak{D}_2$ that are neighbors of each other, and for all $S \subseteq \text{range}(M)$, the following equation holds:*

$$P(M(\mathfrak{D}_1) \in S) \leq e^{\epsilon} P(M(\mathfrak{D}_2) \in S) + \delta$$

Where the probability is only over the randomness in M (not the randomness in the data). When $\delta = 0$, we say that M satisfies pure differential privacy *(also known as ϵ-differential privacy or ϵ-DP) and when $\delta > 0$ we say that M satisfies* approximate differential privacy.

Another important version of differential privacy, is ρ-zCDP (concentrated differential privacy):

Definition 3 (zCDP [7]). *Given a privacy parameter ρ, a randomized algorithm M satisfies ρ-zCDP if for all pairs of datasets $\mathfrak{D}_1, \mathfrak{D}_2$ that are neighbors of each other and all numbers $\alpha > 1$,*

$$\mathcal{D}_\alpha(M(D_1)||M(D_2)) \leq \rho\alpha$$

where $\mathcal{D}_\alpha(P||Q) \equiv \frac{1}{\alpha-1} \log\left(E_{x\sim P}\left[\frac{P(x)^{\alpha-1}}{Q(x)^{\alpha-1}}\right]\right)$ is the Renyi divergence of order α between probability distributions P and Q.

Although zCDP is difficult to interpret, there are useful results that help provide intuition. First, any M that satisfies ϵ-differential privacy also satisfies ρ-zCDP with $\rho = \frac{\epsilon^2}{2}$ [7]. In general a ρ-zCDP algorithm does not satisfy pure differential privacy but does satisfy (ϵ, δ)-DP for infinitely many pairs of ϵ and δ that lie along a curve (see [9] and [2] for conversions between ρ-zCDP and (ϵ, δ)-DP).

2.2 Algorithm Design with Differential Privacy

A few basic principles underlie the construction of many algorithms for differential privacy. The first is sensitivity, which measures the maximum impact that one record can have on a set of queries (regardless of input data):

Definition 4 (Sensitivity [14]). *The L_p global sensitivity of a set Q of queries, denoted by $\Delta_p(Q)$, is defined as* $\sup_{\mathfrak{D}_1 \sim \mathfrak{D}_2} \left(\sum_{q\in Q} |q(\mathfrak{D}_1) - q(\mathfrak{D}_2)|^p\right)^{1/p}$.

Global sensitivity can be used with the Laplace and Gaussian distributions to form basic mechanisms. Let Lap(α) represent a draw from the Laplace distribution with density $f(x) = \frac{1}{2\alpha} e^{-|x|/\alpha}$ and $N(0, \sigma^2)$ represent the zero-mean Gaussian distribution with variance σ^2. Each appearance of $Lap(\alpha)$ or $N(0, \sigma^2)$ represents an independent sample from the corresponding distribution.

Theorem 1 (Laplace Mechanism [14]). *Given a privacy parameter $\epsilon > 0$, a set Q of queries, and an input dataset $\mathfrak{D}$, the mechanism M that returns the set of noisy answers $\{q(\mathfrak{D}) + Lap(\Delta_1(Q)/\epsilon)\}_{q\in Q}$ satisfies ϵ-differential privacy.*

Theorem 2 (Gaussian Mechanism [7]). *Given a privacy parameter $\epsilon > 0$, a set Q of queries, and an input dataset $\mathfrak{D}$, the mechanism M that returns the set of noisy answers $\{q(\mathfrak{D}) + N(0, \Delta_2(Q)^2/(2\rho))\}_{q\in Q}$ satisfies ρ-zCDP.*

All of these privacy definitions are postprocessing invariant [16]. That is, let A be an arbitrary algorithm. Then $A \circ M$ (i.e., the algorithm that outputs $A(M(\mathfrak{D}))$) satisfies (ϵ, δ)-DP (resp., ρ-zCDP) if M satisfies (ϵ, δ)-DP (resp., ρ-zCDP); in other words, the privacy parameters do not degrade.

They also have useful sequential composition properties. Let $M_1, \ldots, M_k$ be algorithms that satisfy pure differential privacy with corresponding parameters $\epsilon_1, \ldots, \epsilon_k$ (resp., zCDP with corresponding privacy parameters $\rho_1, \ldots, \rho_k$), then the algorithm that M that releases all of their outputs (i.e., releases $M_1(\mathfrak{D}), \ldots, M_k(\mathfrak{D})$) satisfies $\sum_i \epsilon_i-$differential privacy [16] (resp., $\sum_i \rho_i$-zCDP [7]).

3 The Uncertainty Principle

The setting of d disjoint queries $q_1, \ldots, q_d$ and their sum q_* are some of the most important types of query sets. As discussed earlier, population counts in small geographic regions such as Census blocks (examples of q_i) are important for redistricting while population counts in larger regions such as

counties (examples of q_*) are used for federal and state funding formulas. Thus any tension between the q_i and q_* can have significant impact on the entire U. S. population. While this is just one example of a query set, almost every table produced in previous censuses is a query set with disjoint queries and their sums [35]. Thus this is an important collection of queries to study.

3.1 Lower Bounds

We strengthen our results by removing some restrictions on M. While its input is a dataset, its output can be a positively weighted dataset – a collection of records in which each record r has a nonnegative weight w. A query q with predicate ψ can be evaluated over a weighted dataset by adding up the weights of the records that satisfy ψ (hence an ordinary dataset is a weighted dataset where all weights are 1). The lower bound is the following (see supplementary material for proofs).

Theorem 3. *Let $q_1 \ldots, q_d$ be a set collection of disjoint queries and let q_* be their sum. Let M be a randomized algorithm whose input is a dataset and whose output is a positively weighted dataset. Suppose M guarantees that for each query q_i and dataset $\mathfrak{D}$, $E[(q_i(\mathfrak{D}) - q_i(M(\mathfrak{D})))^2] \leq C^2$ and $E[(q_*(\mathfrak{D}) - q_*(M(\mathfrak{D})))^2] \leq D^2$ for some values C and D, where the expectation is **only** over the randomness in M.*

- *If M satisfies ϵ-differential privacy then for any $k > 0$, we have $e^{2\epsilon(2C+k)} \geq \frac{k(d-1)}{16C+8D+4k}$ which implies (a) if $D^2 \leq \lambda/\epsilon^2$ for some constant λ, then $C^2 \in \Omega(\frac{1}{\epsilon^2}\log^2(d))$, and (b) if $C \leq \lambda/\epsilon^2$ then $D \in \Omega(d^2/\epsilon^2)$.*
- *If M satisfies (ϵ, δ)-DP then for any $k > 0$, we have $\left(\frac{\delta}{\epsilon} + \frac{4C+2D+k}{k(d-1)}\right) e^{4\epsilon C + 2k\epsilon} \geq 1/4$, which implies (a) if $D^2 \leq \lambda/\epsilon^2$ for some constant λ, then $C^2 \in \Omega\left(\min(\frac{1}{\epsilon^2}\log^2(d),\ \frac{1}{\epsilon^2}\log^2\frac{\epsilon}{\delta})\right)$; (b) if $C \leq \lambda/\epsilon^2$ then either $\epsilon \in O(\delta)$ or $D^2 \in \Omega(d^2/\epsilon^2)$.*
- *If M satisfies ρ-zCDP, then the tradeoff function between C and D (which is more complex and omitted due to space constraints) implies: (a) if $D^2 \leq \lambda/\rho$ for some λ, then $C^2 \in \Omega(\log(d)/\rho)$, and (b) if $C^2 \leq \lambda/\rho$, then for any $\gamma \in (0,1)$, we must have $D^2 \in \Omega(d^{2\gamma}/\rho)$.*

Balcer and Vadhan [3] recently showed a statistical price of privacy-preserving release of the top-k counts in a histogram. They proved an analogous $O(\log^2(d/k))$ penalty for point queries under ϵ-DP (and also results for approximate DP). Interestingly, although they did not consider tradeoffs with the sum query (since its value was assumed to be public in their work), the results in our Theorem 3 (for ϵ-DP and approximate DP, but not zCDP) can be proved as corollaries of their Theorem 7.2.

We also note that the tradeoff functions between C and D in Theorem 3 show a much stronger result than items (a) and (b) in Theorem 3. For example, they rule out the possibility that both C^2 and D^2 can simultaneously be just slightly larger than $O(1/\epsilon^2)$. To understand and interpret Theorem 3, let us compare to the Laplace and Gaussian mechanisms, which can produce negative query answers, hence are not equivalent to producing positively weighted datasets (hence not covered by Theorem 3).

It is easy to see that $\Delta_1(q_1, \ldots, q_d, q_*) = 2$ and $\Delta_2(q_1, \ldots, q_d, q_*) = \sqrt{2}$. Hence, an algorithm M'_ϵ can add independent Lap$(2/\epsilon)$ noise to each query to satisfy ϵ-DP, and an algorithm M'_ρ can add independent $N(0, 1/\rho)$ noise to each query to satisfy ρ-zCDP. Thus M'_ϵ achieves expected squared error of $8/\epsilon^2$ for q_* and each q_i (i.e., $C^2 = D^2 = 8/\epsilon^2$). Meanwhile M'_ρ achieves $1/\rho$ expected squared error ($C^2 = D^2 = 1/\rho$). These expected error guarantees hold for all datasets $\mathfrak{D}$.

Theorem 3 says that privacy-preserving algorithms M that are required to produce positively weighed datasets cannot guarantee the same low error – there are input datasets $\mathfrak{D}$ for which the expected errors can be significantly larger. In the case of M that satisfy ϵ-DP, if we want low error for the sum query (e.g., $D^2 = O(1/\epsilon^2)$, matching the Laplace mechanism), on some datasets we may need to pay a $\log^2(d)$ penalty for some point queries (i.e., there will be specific point queries with consistently large error). On the other hand, if we want low error for the point queries (e.g., $C^2 = O(1/\epsilon^2)$) then on some datasets we will pay a d^2 penalty on the sum query.

In the case of ρ-zCDP, the penalties are smaller. If we want to match the error of the Gaussian mechanism on the sum query, we may need to pay a penalty of $\log(d)$ on point queries; if we want $O(1/\rho)$ expected squared error on each point query, we may need to pay a penalty of nearly d^2 on q_*.

For approximate DP, the weakest privacy definition here, the degradation factor can be roughly $\log^2(\epsilon/\delta)$ no matter how large d is.

Remark 1. The result implies that if privacy-preserving microdata is generated by obtaining noisy measurement query answers (e.g., with the Laplace or Gaussian mechanisms) and then postprocessing the noisy answers (e.g., [25, 21]), some of the measurement queries computed directly from the privacy-preserving microdata will have errors that are larger than their original noisy answers.

Remark 2. Not all is lost, as the proofs are based on packing arguments that show there exist difficult datasets for which these errors are unavoidable. The properties of these datasets are that there are $O(d)$ point queries that have true answer 0, and a few point queries whose true answers are $O(\log(d)/\epsilon)$ for pure DP, $O(\sqrt{\log(d)/\rho})$ for zCDP, and $O(\log(1/\delta)/\epsilon)$ for approximate DP. The latter point queries are ones with high error when q_* is required to be accurate. For other datasets (such as those for which every point query answer is large compared to those quantities), the penalties effectively disappear. Thus, if operational requirements mandate that M must output microdata, it is necessary to find algorithms with good instance-near-optimality properties (i.e., for nice datasets they should match the error profile of the Laplace/Gaussian mechanisms; for extremely sparse datasets their performance should degrade as little as is possible mathematically). In Section 4 we propose some initial algorithms that perform well empirically, but they do not come with theoretical guarantees.

3.2 Upper Bounds

These lower bounds are nearly tight, as shown by the upper bounds in Theorem 4. The proofs construct postprocessing algorithms that first obtain noisy answers $a_1, \ldots, a_d, a_*$ to the queries $q_1, \ldots, q_d, q_*$. A postprocessing step converts the a_i and a_* into consistent noisy answers $a'_1, \ldots, a'_d, a'_*$ (i.e., they are nonnegative and $\sum_i a'_i = a'_*$). Weighted datasets are constructed from the latter quantities. To get weighted datasets with higher accuracy on point queries, the postprocessing ignores a_* and sets $a'_i = \max\{0, a_i\}$. To obtain synthetic data with higher accuracy on the sum query, a'_* is set to a_* and the a'_i are obtained by minimizing squared distance to the a_i subject to the a'_i being nonnegative and adding up to a_*. The full proofs are in the supplementary material.

Theorem 4 (Upper bound for pure DP and zCDP)**.** *Let $q_1, \ldots, q_d$ be a set of disjoint queries and let q_* be their sum. Given privacy parameters $\epsilon > 0$ and $\rho > 0$ There exist algorithms $M_\epsilon, M_\rho, M'_\epsilon, M'_\rho$, $M'_{\epsilon,\delta}$ that output a positively weighted dataset and have the following properties:*

1. *M_ϵ satisfies ϵ-DP, and for all $\mathfrak{D}$ and i, $E\left[(q_i(M_\epsilon(\mathfrak{D})) - q_i(\mathfrak{D}))^2\right] \leq 2/\epsilon^2$ and $E\left[(q_*(M_\epsilon(\mathfrak{D})) - q_*(\mathfrak{D}))^2\right] \leq 2d^2/\epsilon^2$.*
2. *M_ρ satisfies ρ-zCDP, and for all $\mathfrak{D}$ and i, $E\left[(q_i(M_\rho(\mathfrak{D})) - q_i(\mathfrak{D}))^2\right] \leq 1/(2\rho)$ and $E\left[(q_*(M_\rho(\mathfrak{D})) - q_*(\mathfrak{D}))^2\right] \leq d^2/(2\rho)$.*
3. *M'_ϵ satisfies ϵ-DP, and for all $\mathfrak{D}$ and i, $E\left[(q_i(M'_\epsilon(\mathfrak{D})) - q_i(\mathfrak{D}))^2\right] \in O(\log^2(d)/\epsilon^2)$ and $E\left[(q_*(M'_\epsilon(\mathfrak{D})) - q_*(\mathfrak{D}))^2\right] \in O(1/\epsilon^2)$*
4. *M'_ρ satisfies ρ-zCDP, and for all $\mathfrak{D}$ and i, $E\left[(q_i(M'_\rho(\mathfrak{D})) - q_i(\mathfrak{D}))^2\right] \in O(\log(d)/\rho)$ and $E\left[(q_*(M'_\rho(\mathfrak{D})) - q_*(\mathfrak{D}))^2\right] \in O(1/\rho)$*
5. *$M'_{\epsilon,\delta}$ satisfies (ϵ, δ)-DP and for all $\mathfrak{D}$ and i, $E\left[(q_i(M'_{\epsilon,\delta}(\mathfrak{D})) - q_i(\mathfrak{D}))^2\right] \in O(\log^2(1/\delta)/\epsilon^2 + 1)$ and $E\left[(q_*(M'_{\epsilon,\delta}(\mathfrak{D})) - q_*(\mathfrak{D}))^2\right] \in O(1/\epsilon^2)$. Also note M_ϵ and M'_ϵ satisfy ϵ, δ-DP.*

4 Algorithms

For tabular data, typically end-users are interested in multiple marginals of the data. Examples include the gender by age marginals at the national, state, and county levels (for constructing age pyramids); the marginal on race at the national, state, county, tract, and block levels both for demographic

research and for enforcement of voting rights; total populations in each state, county, etc. (for various funding formulas). Thus these query sets have many different point query/sum query collections embedded in them. Examples include: female population in a county (sum query) and number of females of each age in the county (point queries); or total Asian population (sum query) and Asian population in each county (point queries). Thus algorithms designed to minimize the appearance of the uncertainty principle should not be designed for a *single* collection of sum/point queries; instead, they should support *many* counting queries.

To describe algorithms, it is helpful to view the dataset $\mathfrak{D}$ as a vector $\mathbf{x}$, where each element i corresponds to a possible record r_i. Then $\mathbf{x}[i]$ is the number of times r_i appears in $\mathfrak{D}$. The goal is to produce a privacy-protected version $\widetilde{\mathbf{x}}$ whose entries are nonnegative real numbers, which can be converted to a positively weighted dataset $\widetilde{\mathfrak{D}}$ ($\widetilde{\mathbf{x}}[i]$ is the weight of record r_i in $\widetilde{\mathfrak{D}}$). In this setting, a counting query q is just a vector of 1s and 0s with the same dimensionality as $\mathbf{x}$, and the query answer is computed as the dot product $q \cdot \mathbf{x}$.

The algorithms we present here (2 baselines and 2 proposed algorithms) are all based on the idea of first computing noisy query answers and then postprocessing them to obtain $\widetilde{\mathbf{x}}$. This setup allows an organization to release both $\widetilde{\mathbf{x}}$ and the noisy answers (for more statistically-oriented end-users). Thus, given a set Q of counting queries, for each $q \in Q$, the data collector computes a noisy answer a_q by adding noise with distribution F_q to the true answer. We assume the data collector chooses the noise distributions to achieve their desired privacy definition (e.g., ϵ-DP, ρ-zCDP).

Baseline: NNLS Postprocessing. The first baseline we consider is the commonly used nonnegative least squares (NNLS), in which $\widetilde{\mathbf{x}}$ is produced as the solution to the following optimization problem:

$$\widetilde{\mathbf{x}} \leftarrow \arg\min_{\widetilde{\mathbf{x}}} \sum_{q \in Q} \frac{(a_q - q \cdot \widetilde{\mathbf{x}})^2}{variance(F_q)} \text{ s.t. } \widetilde{\mathbf{x}}[i] \geq 0 \text{ for all } i$$

Baseline: Max Fitting Postprocessing. The next baseline is an adaptation of a bilevel optimization approach [28] that was originally used for optimization problems whose parameters are sensitive. The idea here is to find the positively weighted datasets whose query answers minimize the L_∞ distance to the noisy query answers, breaking ties using least squares error:

$$dist \leftarrow \min_{\widetilde{\mathbf{x}}} \max_{q \in Q} \frac{|a_q - q \cdot \widetilde{\mathbf{x}}|}{std(F_q)} \text{ s.t. } \widetilde{\mathbf{x}}[i] \geq 0 \text{ for all } i$$

$$\widetilde{\mathbf{x}} \leftarrow \arg\min_{\widetilde{\mathbf{x}}} \sum_{q \in Q} \frac{(a_q - q \cdot \widetilde{\mathbf{x}})^2}{variance(F_q)} \text{ s.t. } \max_{q \in Q} \frac{|a_q - q \cdot \widetilde{\mathbf{x}}|}{std(F_q)} \leq dist \text{ and } \widetilde{\mathbf{x}}[i] \geq 0 \text{ for all } i$$

Sequential Fitting Postprocessing. Since it is provably not always possible to output microdata that fits the noisy answers well, we propose an approach that prioritize queries. Thus the query set Q is partitioned by the user into query sets $Q_1, \ldots, Q_k$. We use the above NNLS approach to fit a vector $\widetilde{\mathbf{x}}_1$ to the noisy answers of queries in Q_1 (highest priority). We then fit $\widetilde{\mathbf{x}}_2$ to the noisy answers for queries in Q_2 (next highest priority) subject to the constraints that $\widetilde{\mathbf{x}}_2$ matches $\widetilde{\mathbf{x}}_1$ on queries in Q_1. Then we fit $\widetilde{\mathbf{x}}_3$ using noisy answers to queries in Q_3 while forcing $\widetilde{\mathbf{x}}_3$ to match $\widetilde{\mathbf{x}}_2$ on queries in Q_1 and Q_2, and so on and return the final $\widetilde{\mathbf{x}}_k$ at the end. The pseudocode is shown in Algorithm 1. This algorithm is the one that matches the upper bounds in Theorem 4 (referred to as M'_ϵ when the noisy answers a_q use Laplace noise, and M'_ρ for Gaussian noise).

Remark. The constrained optimizations in max fitting and sequential fitting are difficult for quadratic program optimizers, often resulting in numerical errors, slow convergence, and infeasibility errors (due to occasional insufficient solution quality in earlier stages of the multistage optimization). They require significant engineering effort, tuning of slack parameters (slightly relaxing equality and inequality constraints) and optimizer-specific parameters. So, an ideal solution would also avoid constraints other than nonnegativity for point queries. This is a rationale for our next method.

ReWeighted Fitting Postprocessing. This method (shown in Algorithm 2) avoids constraints as much as possible in an eventual NNLS solve (Line 14) but is limited to query sets of the form $Q = \bigcup_{i=1}^{k} Q_i$, where the queries inside each Q_i are disjoint and have the same noise distribution. One example is when Q is a collection of marginal queries (e.g., Q_1 = marginal on age, Q_2 = marginal on age by race, Q_3 = marginal on gender by race), which are arguably the most important

Algorithm 1: Sequential Fitting (Postprocessing)

Input: Query set Q, noisy answers a_q for $q \in Q$ and noise distributions F_q for $q \in Q$.
Input: $Q_1, \ldots, Q_k$: partition of Q based on query priority.
$\widetilde{\mathbf{x}}_1 \leftarrow \arg\min_{\widetilde{\mathbf{x}}} \sum_{q \in Q_1} \frac{(a_q - q \cdot \widetilde{\mathbf{x}})^2}{variance(F_q)}$ s.t. $\widetilde{\mathbf{x}}[i] \geq 0$ for all i
Fit $\leftarrow Q_1$
for $\ell = 2, \ldots, k$ **do**
  $\widetilde{\mathbf{x}}_\ell \leftarrow \arg\min_{\widetilde{\mathbf{x}}} \sum_{q \in Q_\ell} \frac{(a_q - q \cdot \widetilde{\mathbf{x}})^2}{variance(F_q)}$ s.t. $\widetilde{\mathbf{x}}[i] \geq 0$ for all i and $q \cdot \widetilde{\mathbf{x}} = q \cdot \widetilde{\mathbf{x}}_{\ell-1}$ for all $q \in$ Fit
  Fit $\leftarrow$ Fit $\cup Q_\ell$
Return: $\widetilde{\mathbf{x}}_k$

types of queries. Within each Q_i, the algorithm tries to find a cutoff value so that queries with noisy answers above it are likely to have true value that is non-zero (Lines 5-6). The idea is that if $n_\dagger$ is the number of queries below the threshold, and if they truly had value 0, then their largest noisy value (i.e., the max of $n_\dagger$ 0-mean Laplace or Gaussian random variables) should not be near the cutoff with high probability (controlled by the confidence parameter γ). The "low" queries are the ones with noisy answers below the cutoff. The algorithm uses the existing noisy answers to estimate the sum of these "low" queries (Lines 11-12) and adds that "low query sum" (Line 13) to the nonnegative least squares optimization while downweighting the individual low queries (Line 9, the downweight depends on the extreme value distribution of the max of $n_\dagger$ 0-mean Laplace or Gaussian random variables, Line 7). To avoid double counting, both places where a "low" query is used (individually and as part of a sum) have their weights cut in half. Note the algorithm only uses existing noisy answers and has no access to the true data.

Algorithm 2: ReWeighted Fitting (Postprocessing)

1 **Input:** Query set $Q = \bigcup_{i=1}^{k}$; Within a Q_i, the queries are disjoint. F_i is the noise distribution of each query in Q_i. Given noisy answers a_q for $q \in Q$ that satisfy the chosen privacy definition.
2 **Input:** Confidence parameter γ close to 1 (e.g., 0.99, the setting used in experiments)
3 $S \leftarrow \emptyset$ **for** $i = 1, \ldots, k$ **do**
4 $a_{(1)}, a_{(2)}, \ldots$ are the given noisy answers (to queries in Q_i) arranged in sorted order
5 $j^* \leftarrow$ smallest j s.t. $P(\max(j$ fresh random variable with distribution$F_i) \geq a_{(j)}) \leq 1 - \gamma$
6 $cutoff \leftarrow a_{(j^*)}$.
7 $downweight \leftarrow$ median of distribution of max of j random variables sampled from F_i
8 For each query $q \in Q_i$ whose noisy answer a_q is $\geq$ cutoff, add $(q, a, 1/var(F_i))$ to S.
9 For each query $q \in Q_i$ whose a_q is $<$ cutoff, add $(q, a, \frac{1}{2 * var(F_i) * downweight^2})$ to S.
10 $n_i^\dagger \leftarrow$ number of queries selected in Line 9 (i.e., their noisy answers were $< cutoff$)
11 $q_\dagger \leftarrow$ sum of queries selected in Line 9
12 $a_\dagger \leftarrow$ sum of their existing noisy answers
13 Add $(q_\dagger, a_\dagger, \frac{1}{2 * n_i^\dagger var(F_i)})$ to S
14 $\widetilde{\mathbf{x}} \leftarrow \arg\min_{\widetilde{\mathbf{x}}} \sum_{(q', a', w') \in S} w'(q'(\widetilde{\mathbf{x}}) - a')^2$ s.t., $\widetilde{\mathbf{x}}[i] \geq 0$ for all i.
15 **Return:** $\widetilde{\mathbf{x}}$

5 Experiments

To make our code fully open source, we wrote it in Julia [5] and after trying several open-source optimizers, we settled on COSMO [19]. We created a collection of benchmark datasets that were small enough so that running the postprocessing algorithms thousands of times on each dataset (to estimate expected errors) was feasible, but large enough to demonstrate the uncertainty principle. The full benchmark of 15 real datasets and 16 synthetic datasets is described (and provided) in the supplementary material. Here we only present results for an interesting subset. The only synthetic dataset we present here, called Level00-2d is a 10×10 histogram where one element is large (10000) and the others are 0. The other 15 datasets we present here were taken from the 2016 ACS Microdata

Dataset Nickname	Dataset	OLS	NNLS	MaxFit	Seq	ReWeight
$\mathfrak{D}01$	Level00-2d	101.3	461.9	533.9	149.2	108.5
$\mathfrak{D}02$	PUMA0101301	107.2	547.2	500.3	106.7	112.5
$\mathfrak{D}03$	PUMA0800803	107.2	446.1	571.7	120.3	107.2
$\mathfrak{D}04$	PUMA1304600	107.2	408.1	426.3	120.8	109.8
$\mathfrak{D}05$	PUMA1703529	107.2	435.3	426.3	134.9	110.9
$\mathfrak{D}06$	PUMA1703531	107.2	584.0	677.4	111.4	108.1
$\mathfrak{D}07$	PUMA1901700	107.2	395.1	443.6	119.1	110.4
$\mathfrak{D}08$	PUMA2401004	107.2	369.3	329.0	109.6	107.5
$\mathfrak{D}09$	PUMA2602702	107.2	467.8	472.0	146.0	109.2
$\mathfrak{D}10$	PUMA2801100	107.2	543.7	558.2	117.8	110.8
$\mathfrak{D}11$	PUMA2901901	107.2	485.2	464.4	126.5	110.8
$\mathfrak{D}12$	PUMA3200405	107.2	329.1	301.0	122.9	108.4
$\mathfrak{D}13$	PUMA3603710	107.2	300.3	293.3	85.7	108.8
$\mathfrak{D}14$	PUMA3604010	107.2	399.9	386.5	129.8	111.3
$\mathfrak{D}15$	PUMA5101301	107.2	396.1	369.5	139.2	107.2
$\mathfrak{D}16$	PUMA5151255	107.2	330.7	280.3	139.1	107.8

Table 1: Squared Error for Sum Query (overall $\epsilon = 0.5$))

Sample [34]. Each represents a 9×24 "race by Hispanic origin" histogram from Public Use Microdata Areas that were considered outliers in their states in terms of racial composition.

For these datasets, we applied the Laplace mechanism with $\epsilon = 0.5$ to answer the sum query, both 1-way marginal queries, and identity queries (for each cell, how many people are in it); this is also the priority order used by Sequential Fitting. Error results for the marginals, other privacy parameters and zCDP can be found in the supplementary material. We ran the Laplace mechanism + different postprocessing strategies (described in Section 4) 1,000 times for each dataset to estimate expected squared error of each query. We added an ordinary least squares (OLS) optimization for comparison purposes (it is NNLS without nonnegativity constraints). OLS is free from the uncertainty principle because it does not produce positively weighted microdata. Thus, to minimize the effect of the uncertainty principle, the other postprocessing methods should try to achieve errors that are not much worse than OLS. We note that the multi-stage optimization in Max and Sequential fitting are generally very difficult for optimization software, so we only kept those runs in which the optimizer succeeded (thus results for Max and Sequential Fitting are slightly optimistically biased).

In Table 1, we show the squared error of these postprocessing methods for the sum query. The NNLS and MaxFitting baselines perform poorly for this query, with errors typically 4-5x those of the OLS method (which is close to the variance of the original noisy answer to the sum query). Meanwhile Sequential and ReWeighted fitting perform much better. Standard errors were roughly 2-6% of the reported metrics (omitted for space, but shown in the supplementary materials).

For Table 2 we examine the expected errors of each cell query (i.e., q_i is the number of people in cell i). We find the cell with the largest expected error and report it (the "Max" column). We also find the total squared error of the cell queries and report them in the "Total" column. Again, the standard errors are roughly 2-6% of the reported metrics, except that they are sometimes higher for Max and Sequential fitting since averages were only computing on the subset of runs for which the optimizer did not fail.

Generally, we see NNLS performing slightly better in terms of the max expected error compared to ReWeight, although their total errors are comparable and ReWeight significantly outperforms NNLS on the sum query.

Overall, from these experiments and our supplementary material, both ReWeight and Sequential fitting (though not perfect) avoid incidents where there are extremely high errors (unlike NNLS and Max Fitting for sum queries), and this is important in practice. ReWeight and Sequential fitting have similar performance. ReWeight is faster while Sequential needs significant tuning of optimizers in order to succeed. However, one advantage of Sequential is its algorithmic transparency – it can directly prioritize queries for the tradeoffs caused by the uncertainty principle (in our experiments, the sum query had highest priority for Sequential Fitting).

	OLS		NNLS		MaxFit		Seq		ReWeight	
Data	Total	Max	Total	Max	Total	Max	Total	Max	Total	Max
$\mathfrak{D}01$	10516.5	124.0	344.2	147.4	443.7	173.1	437.3	283.6	159.2	78.4
$\mathfrak{D}02$	23906.2	142.9	809.0	135.6	910.7	144.6	782.9	179.7	731.3	209.8
$\mathfrak{D}03$	23906.2	142.9	1179.8	107.5	1235.3	125.4	1171.7	189.8	1123.8	141.9
$\mathfrak{D}04$	23906.2	142.9	1313.0	111.9	1385.5	142.3	1049.1	126.3	1264.4	136.0
$\mathfrak{D}05$	23906.2	142.9	1243.8	105.3	1257.2	96.7	1019.1	114.3	1285.7	160.5
$\mathfrak{D}06$	23906.2	142.9	562.2	94.9	599.0	72.1	429.9	112.9	409.8	78.8
$\mathfrak{D}07$	23906.2	142.9	1516.1	115.9	1665.9	129.7	1312.1	156.9	1617.1	205.0
$\mathfrak{D}08$	23906.2	142.9	1954.4	130.0	1971.8	147.8	1983.4	311.3	1760.1	168.9
$\mathfrak{D}09$	23906.2	142.9	977.2	100.0	956.4	109.4	843.4	121.7	930.1	156.2
$\mathfrak{D}10$	23906.2	142.9	686.9	97.5	705.7	79.0	534.2	92.7	516.0	78.7
$\mathfrak{D}11$	23906.2	142.9	944.4	100.4	919.2	103.2	809.4	131.6	888.2	138.2
$\mathfrak{D}12$	23906.2	142.9	2189.2	119.6	2191.5	134.7	1918.5	142.3	2336.1	259.1
$\mathfrak{D}13$	23906.2	142.9	2884.1	119.2	3088.6	149.1	2484.2	140.7	2870.4	166.1
$\mathfrak{D}14$	23906.2	142.9	1432.5	105.9	1442.3	120.6	1262.1	122.7	1448.6	194.0
$\mathfrak{D}15$	23906.2	142.9	1474.7	108.3	1498.6	101.8	1394.5	203.4	1392.9	153.2
$\mathfrak{D}16$	23906.2	142.9	2239.7	130.3	2274.1	124.3	2079.0	178.5	2123.0	172.8

Table 2: Squared Errors Id Query (overall $\epsilon = 0.5$).

6 Related Work

The requirement to produce microdata is an example of the notion of consistency in privacy-preserving query answering. A variety of work [4, 22, 31, 25, 10, 12, 26, 24, 21] has shown that the creation of a privacy-preserving data synopsis, and then answering queries from it, can improve query accuracy under a variety of metrics such as maximum simultaneous error and total error. It is known, however, that the production of privacy-preserving microdata comes at the expense of increased computational cost [36, 15, 33]. For example, under standard complexity assumptions [33], there is no polynomial-time algorithm for generating privacy-protected synthetic data whose two-way marginals are all accurate.

Aside from the computational price, Balcer and Vadhan [3] also recently showed a statistical price of privacy-protected synthetic data. They considered releasing different kinds of privacy-protected representations of nonnegative noisy histograms (for example, releasing the top-k noisy cells under ϵ-DP had a $\log^2(d/k)$ penalty term for squared error), but assumed the value of the sum query was publicly known in their work. Our constructions are based on their proof techniques (see discussion after Theorem 3).

7 Conclusions, Future Work, and Broader Impact

Public use data has many different end-users, so a single aggregated performance measure, such as total error across all queries, is not a reliable measure of data quality. The accuracy of each query is important and this sets up multiple conflicting quality criteria for public-use data. Thus an important direction for future work is to identify all tradeoffs in privacy-preserving microdata as well as algorithms with provable guarantees on instance-optimality (i.e., improve performance on datasets that do not trigger the uncertainty principles).

Broader Impact

The uncertainty principle presented in this paper (as well as [3]) along with the known computational price of generating microdata suggests that organizations should also consider alternative forms of producing privacy-protected data products, such as releasing noisy query answers. This kind of a shift may also require educating and providing training materials to end-users in such a data release format. It also suggests that microdata-generating algorithms should be designed as postprocessing algorithms that convert unbiased noisy measurements into microdata (so that the "statistics-friendly" noisy measurements can also be released and studied by data scientists).

References

[1] John M. Abowd. The U.S. Census Bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, pages 2867–2867, New York, NY, USA, 2018. ACM.

[2] Shahab Asoodeh, Jiachun Liao, Flavio P. Calmon, Oliver Kosut, and Lalitha Sankar. A better bound gives a hundred rounds: Enhanced privacy guarantees via f-divergences. In *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020.

[3] Victor Balcer and Salil Vadhan. Differential privacy on finite computers. *Journal of Privacy and Confidentiality*, 9(2), Sep. 2019.

[4] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. Privacy, accuracy and consistency too: A holistic solution to contingency table release. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, 2007.

[5] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

[6] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong privacy for analytics in the crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pages 441–459, New York, NY, USA, 2017. ACM.

[7] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography*, 2016.

[8] U. S. Census Bureau. On the map: Longitudinal employer-household dynamics, 2019.

[9] Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. In *NeurIPS*, 2020.

[10] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. Differentially private spatial decompositions. In *ICDE*, 2012.

[11] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 3574–3583, USA, 2017. Curran Associates Inc.

[12] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. Differentially private data cubes: Optimizing noise sources and consistency. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, 2011.

[13] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, 2006.

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[15] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete Comput. Geom.*, 53(3):650–673, 2015.

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

[17] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA, 2014. ACM.

[18] Simson L. Garfinkel, John M. Abowd, and Sarah Powazek. Issues encountered deploying differential privacy. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, 2018.

[19] Michael Garstka, Mark Cannon, and Paul Goulart. COSMO: A conic operator splitting method for large convex problems. In *European Control Conference*, 2019.

[20] Samuel Haney, Ashwin Machanavajjhala, John M. Abowd, Matthew Graham, Mark Kutzbach, and Lars Vilhuber. Utility cost of formal privacy for releasing national employer-employee statistics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1339–1354, New York, NY, USA, 2017. ACM.

[21] Moritz Hardt, Katrina Ligett, and Frank McSherry. A simple and practical algorithm for differentially private data release. In *NIPS*, 2012.

[22] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1–2):1021–1032, September 2010.

[23] Noah Johnson, Joseph P Near, and Dawn Song. Towards practical differential privacy for sql queries. *Proceedings of the VLDB Endowment*, 11(5):526–539, 2018.

[24] Jaewoo Lee, Yue Wang, and Daniel Kifer. Maximum likelihood postprocessing for differential privacy under consistency constraints. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

[25] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *PODS*, 2010.

[26] Chao Li, Gerome Miklau, Michael Hay, Andrew Mcgregor, and Vibhor Rastogi. The matrix mechanism: Optimizing linear counting queries under differential privacy. *The VLDB Journal*, 24(6):757–781, 2015.

[27] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *2008 IEEE 24th International Conference on Data Engineering*, pages 277–286, Piscataway, NJ, USA, April 2008. IEEE.

[28] T. W. Mak, Ferdinando Fioretto, and P. V. Hentenryck. Bilevel optimization for differentially private optimization. *ArXiv*, abs/2001.09508, 2020.

[29] 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.

[30] National Academies of Sciences, Engineering, and Medicine. Chapter 11: Census bureau's responses and own analyses of 2010 demonstration data products. In *Proceedings of the 2020 CNSTAT Workhop on Census Data Products: Data Needs and Privacy Considerations*. The National Academies Press, 2020.

[31] Wahbeh Qardaji, Weining Yang, and Ninghui Li. Understanding hierarchical methods for differentially private histograms. 2013.

[32] Apple Differential Privacy Team. Learning with privacy at scale, 2017.

[33] Jonathan Ullman and Salil Vadhan. Pcps and the hardness of generating private synthetic data. In *Proceedings of the 8th Conference on Theory of Cryptography*, TCC'11, page 400–416, Berlin, Heidelberg, 2011. Springer-Verlag.

[34] U.S. Census Bureau. 2016 american community surve public use microdata samples (pums). `https://www.census.gov/programs-surveys/acs/microdata/access.2016.html`.

[35] U.S. Census Bureau. Decennial census: 2010 summary files. `https://www.census.gov/mp/www/cat/decennial_census_2010/`.

[36] Salil Vadhan. *The Complexity of Differential Privacy*, pages 347–450. Springer International Publishing, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] **Our claims are that we present an uncertainty principle (for which we prove upper and lower bounds in Section 3 with credit to [3] for important heavy lifting), some algorithms that try to mitigate its effects (Section 4) and creation of benchmark datasets (briefly described in Section 5 and more fully described in the supplementary material. The supplementary material contains 3 items: our source code (which includes the benchmark data), a version of this paper with appendix containing full proofs and data description, and a third document containing all of the experimental results we generated)**
   (b) Did you describe the limitations of your work? [Yes] **We discussed the limitations of the uncertainty principle in Remark 2 in Section 3 which explains that the principle only occurs in some (worst-case) datasets, although real-world datasets seem to be closer to the worst-case. We also explain in the introduction that we do not have theoretical guarantees for our mitigation algorithms and experiments show that they are good but not perfect.**
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] **We believe that privacy research has positive impacts. However, we note that the uncertainty principle shows that the requirement that privacy-preserving algorithms should produce microdata is a requirement that can lead to further accuracy loss. In the conclusions we recommend that algorithms should be based on postprocessing noisy query answers so that the noisy query answers can also be released. We believe that this is good theoretically-motivated guidance to data publishers.**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] **We also note that the datasets we use are public and our code can be fully open-sourced.**
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] **Our Theorems 3 and 4 are self-contained and include their assumptions**
   (b) Did you include complete proofs of all theoretical results? [Yes] **In the supplementary materials.**
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] **In the supplementary materials.**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] **We explained our experimental settings, and more details are added in the separate supplementary document that contains all of our experiments.**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] **Due to space constraints, we report a range on the standard errors of our experiments. The actual standard errors appear in the supplementary files in the document that has the full experiments**
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] **We made the benchmarks small enough to avoid the need for clusters and specialized hardware. All of the experiments were run on a desktop computer**
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] **We cited the Julia language, the COSMO optimizer and the 2016 ACS Public Use Microdata Samples used to create our dataset histograms.**
   (b) Did you mention the license of the assets? [N/A] **ACS is public domain, and we do not distribute COSMO or Julia (users need to install it themselves).**

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] **Our code is in the supplementary material**

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] **We only use public data (2016 ACS Public Microdata Samples (PUMS))**

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] **In general, the 2016 ACS PUMS may contain PII despite the U.S. Census Bureau's efforts to provide confidentiality protections. However, we only use the Hispanic origins and race variables of people in Public Use Microata Areas which contain at least 100,000 people.**

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]