Budget Sharing for Multi-Analyst Differential Privacy

David Pujol
Duke University
Durham, NC, United States
dpujol@cs.duke.edu

Yikai Wu
Duke University
Durham, NC, United States
yikai.wu@duke.edu

Brandon Fain
Duke University
Durham, NC, United States
btfain@cs.duke.edu

Ashwin Machanavajjhala
Duke University
Durham, NC, United States
ashwin@cs.duke.edu

ABSTRACT

Large organizations that collect data about populations (like the US Census Bureau) release summary statistics that are used by multiple stakeholders for resource allocation and policy making problems. These organizations are also legally required to protect the privacy of individuals from whom they collect data. Differential Privacy (DP) provides a solution to release useful summary data while preserving privacy. Most DP mechanisms are designed to answer a single set of queries. In reality, there are often multiple stakeholders that use a given data release and have overlapping but not-identical queries. This introduces a novel joint optimization problem in DP where the privacy budget must be shared among different analysts.

We initiate study into the problem of DP query answering across multiple analysts. To capture the competing goals and priorities of multiple analysts, we formulate three desiderata that any mechanism should satisfy in this setting – The Sharing Incentive, Non-Interference, and Adaptivity – while still optimizing for overall error. We demonstrate how existing DP query answering mechanisms in the multi-analyst settings fail to satisfy at least one of the desiderata. We present novel DP algorithms that provably satisfy all our desiderata and empirically show that they incur low error on realistic tasks.

1 INTRODUCTION

Large data collecting organizations like Facebook, Google, The U.S. Census Bureau, and Medicare often release summary statistics about individuals and populations. Access to such data is incredibly useful for multiple resource allocation, policy-making and scientific endeavors. Decisions like congressional seat apportionment, school funding and emergency response plans all depend on census data [18]. Facebook's trove of user interaction data was found to be valuable in studying the impact of social media on elections and democracy [28].

While these data releases are very useful, they may reveal sensitive information about individuals [14, 26, 34]. Differential Privacy (DP) [10, 11] is the gold standard of privacy protection through the addition of randomized noise. However, due to the fundamental law of information recovery [9], making an unbounded number of releases from a dataset (even if each satisfies DP) will eventually allow an attacker to accurately reconstruct the underlying dataset. Because of this, data curators must bound the amount of information released using a parameter known as the privacy loss budget ϵ. Traditional privacy mechanisms focus on minimizing the error introduced by differential privacy, where error trades off with ϵ.

1.1 Multi-analyst DP data release problem

We study the common real-world situation where multiple stakeholders or analysts are interested in a particular data release and the data curator must decide how the stakeholders should share the limited privacy budget. Consider the role of Facebook in its partnership with Social Science One [1]. Facebook wanted to aid research on the effect of social media on democracy and elections by sharing some social network data. In order to participate and receive the privacy protected data each analyst had to submit their specific tasks and queries ahead of time. With the given set of queries from each analyst and a fixed privacy budget, Facebook created a single data release to be used by all analysts. Using existing DP techniques, Facebook had three options: (a) split the privacy budget and answer each analyst's queries individually, (b) join all analysts' queries together and answer them all at once using a workload answering mechanism [4, 7, 8, 16, 23, 24, 27, 29, 32, 33, 35, 37], or (c) generate a single set of synthetic data [36] for all analysts to use.

Option (a) is inefficient as the same query can be answered multiple times, each time using some of the privacy budget. Option (b) may be efficient with respect to overall error but does not differentiate between the queries of different analysts. Some analysts may receive drastically more error than others, perhaps much more than they would have under (a). Option (b) therefore lacks much in the way of guarantees to an individual analyst. Option (c) is agnostic to any analyst's particular queries and may incur inefficiencies due to its inability to adapt to the specific queries being asked.

Though all of these techniques have their uses, they all have some undesirable properties in the multi-analyst setting. This is because almost all of the work in differential privacy up until now has focused (often implicitly) on the single analyst case. We are interested in designing effective shared systems for multi-analyst differentially private data release that simultaneously provide guarantees to individual analysts and ensure good overall performance. We call this the multi-analyst differentially private data release problem.

1.2 Contributions

Our work introduces the multi-analyst differentially private data release problem. In this context we ask: *"How should one design a privacy mechanism when multiple analysts may be in competition over the limited privacy budget"*. Our main contributions in this work are as follows.

- We study (for the first time) differentially private query answering across multiple analysts. We consider a realistic setting where multiple analysts pose query workloads and the data owner makes a single private release to answer all analyst queries.
- We define three minimum desiderata that that we will argue any differentially private mechanism should satisfy in a multi-agent setting – The Sharing Incentive, Non-Interference and Adaptivity.
- We show empirically that existing mechanisms for answering large sets of queries either violate at least one of the desiderata described or are inefficient.
- We introduce mechanisms which provably satisfy all of the desiderata while maintaining efficiency.

2 BACKGROUND

Data Representation We consider databases where each individual corresponds to exactly one tuple. The algorithms considered use a vector representation of the database denoted x. More specifically, given a set of predicates $\mathcal{B} = \{\phi_1 \dots \phi_n\}$, the original database D is transformed into a vector of counts x^D where x_i^D is the number of records in D which satisfy ϕ_i. For simplicity, we denote the length of the data vector as n and we will use the notation x in order to refer to the vector form of database D

Predicate counting queries are a versatile class of queries that count the number of tuples satisfying a logical predicate. A predicate corresponds to a condition in the **WHERE** clause of an SQL query. So a predicate counting query is one of the form **SELECT Count (*) FROM R WHERE** ϕ. Workloads of counting queries can express queries such as histograms, high dimensional range queries, marginals, and datacubes among others.

Like databases, a predicate counting query can be represented as a n-length vector w such that the answer to the query is $w^T x$. A workload is a set of m predicate counting queries arranged in a $m \times n$ matrix W, where each row is the vector form of a single query. Many common queries can be represented as workloads in this form. For example, a histogram query is simply represented by an $n \times n$ identity matrix.

Differential Privacy [10, 11] is a formal model of privacy that grantees each individual that any query computed from sensitive data would have been almost as likely as if the individual had

opted out. More formally, Differential Privacy is a property of a randomized algorithm which bounds the ratio of output probabilities induced by changes in a single record.

DEFINITION 1 (DIFFERENTIAL PRIVACY). *A randomized mechanism* $\mathcal{M}$ *is* (ϵ, δ)-*differentially private if for two neighboring databases* D, *and* D' *which differ in at most one row, and any outputs* $O \subseteq Range(\mathcal{M})$:

$$\Pr[\mathcal{M}(D) \in O] \leq \exp(\epsilon) \times \Pr[\mathcal{M}(D') \in O] + \delta$$

The parameter ϵ often called the privacy budget quantifies the privacy loss. Here we focus exclusively on ϵ-Differential Privacy, i.e when $\delta = 0$.

The Laplace Mechanism is a differentially private primitive which underlines the algorithms used here. We describe the vector version of the Laplace Mechanism below.

DEFINITION 2 (LAPLACE MECHANISM, VECTOR FORM). *Given an* $m \times N$ *workload matrix* W, *the randomized algorithm which outputs the following vector is* ϵ-*differentially private* [11].

$$W x + \text{Lap}\left(\frac{\|W\|_1}{\epsilon}\right)^m$$

Where $\|W\|_1$ is the maximum L1 column norm of W and $\text{Lap}(\sigma)^m$ denotes the m-length vector of m independent samples from a Laplace distribution with mean 0 and scale σ.

Differentially private releases compose with each-other in that if there are two private releases of the same data with two different privacy budgets the amount of privacy lost is equivalent to the sum of their privacy budgets. More formally we have the following.

THEOREM 1 (DP COMPOSITION [11]). *Let* $\mathcal{M}_1$ *be an* ϵ_1-*differentially private algorithm and* $\mathcal{M}_2$ *be an* ϵ_2-*differentially private algorithm. Then their combination defined to be* $\mathcal{M}_{1,2}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ *is* $\epsilon_1 + \epsilon_2$-*differentially private*

Of the many algorithms proposed in the literature, we will consider a class of measures that invoke the **Select, Measure, Reconstruct** paradigm, where instead of directly answering the queries, they first **select** a new set of strategy queries. They then **measure** the strategy queries using a privacy protecting mechanism (in this case the Laplace Mechanism [11]) and finally **reconstruct** the answers to the original input queries from the noisy measurements. Examples of mechanisms that follow this paradigm are the Matrix Mechanism [25] and it's derivatives such as HDMM[27]. The Matrix Mechanism answers a workload of queries W by first selecting a strategy workload A to answer. It then measures the queries in A using the Laplace Mechanism and then reconstructs the answers to W from the noisy answers of A. Given a workload matrix W and a strategy matrix A, the expected total square error of the Matrix mechanism is as follows.

$$\text{Error}(W, A, \epsilon) = \frac{2}{\epsilon^2} \|A\|_1^2 \|W A^+\|_F^2 \tag{1}$$

Where $\|A\|_1$ is the L1 column norm of A and the norm considered here is the frobenius norm. For concreteness in this work we consider only mechanisms which answer workloads of linear queries. These mechanisms can be extended to answer non-linear queries by adding post-processing steps which reconstruct non-linear queries from answers to several linear queries.

3 PROBLEM FORMULATION

3.1 Setting

We consider the setting where there are k analysts with associated positive weights $s_1, s_2 \ldots s_k \in (0, 1)$ such that $s_1 + s_2 \ldots s_k = 1$. These weights represent the share of the total privacy budget to which each analyst is entitled and can be interpreted as the relative importance of each analysts' queries; the natural default is to use proportional weights of $1/k$ for every analyst.

Each analyst submits a workload of queries $W_1, W_2 \ldots W_k \in \mathcal{W}$ The data curator then answers all of the queries using a multi-analyst differentially private mechanism. We define a multi analyst differentially private mechanism $\mathcal{M}$ as a function that takes as input each analysts' set of queries, their respective shares of the privacy budget and the overall privacy budget and outputs a single data release containing the answers to all of the queries.

We can describe the mean squared error experienced by a particular analyst in a multi-analyst Matrix Mechanism as follows.

$$\mathrm{Err}_i(\mathcal{M}, \mathcal{W}, \epsilon) = \frac{2}{\epsilon^2} \|A\|_1^2 \|W_i A^+\|_F^2, \qquad (2)$$

where W_i is the matrix form of the workload W_i given by the ith analyst, A is the strategy matrix produced by mechanism $\mathcal{M}$ with input $\mathcal{W}$. This formula is only for linear queries. For non-linear queries, we must use real datasets to get query answers and estimate expected errors.

3.2 Desiderata

For ease of exposition, imagine that each analyst is given the choice to either have their queries answered independently with their share of the privacy budget or to join the collective, a group of analysts whose queries are answered with a multi-analyst DP mechanism using the sum of all of the collective analysts' privacy budget. We argue that any multi-analyst differentially private mechanism should satisfy three desiderata. First, the mechanism should incentivize a rational agent to participate in the collective by guaranteeing no worse expected error than if their queries were answered independently. Second, the mechanism should never cause any analyst to regret that another analyst is participating in the collective and increasing the former's expected error. Third, the mechanism should be able to adapt to and optimize for the particular queries being asked by all analysts. In this section we formalize these criteria through three separate desiderata: the Sharing Incentive, Non-Interference, and Adaptivity. We introduce each of the desiderata as well as current common practice through a rolling example which demonstrates the importance of these desiderata even in a simple case.

EXAMPLE 1. *Alice, Bob, and Carol are analysts working on a private dataset of US COVID-19 deaths by age provided by the Center for Disease Control [3]. The populations are split into 11 buckets by age. The data curator decides to use a privacy budget of $\epsilon = 1$. Each of the analysts are entitled to an equal share of the privacy budget (that is, each has weight $1/3$). Alice and Bob both want to ask the a histogram of the counts by age (we call this the identity workload on age). Carol wants to ask for the total of all counts in the database.*

The first desideratum, the **Sharing Incentive** requires that each analyst, in expectation, receives at most as much error as if they had computed their query answers independently using the same mechanism and their fraction of the privacy budget. This captures the idea that each analyst should always benefit from joining the collective.

DEFINITION 3 (SHARING INCENTIVE). *A mechanism $\mathcal{M}$ satisfies the Sharing Incentive if for every analyst i the following holds.*

$$\mathrm{Err}_i(\mathcal{M}, \mathcal{W}, \epsilon) \leq \mathrm{Err}_i(\mathcal{M}, \{W_i\}, s_i \epsilon)$$

EXAMPLE 2. *The data curator decides to split each analyst off and give them each $\epsilon/3$ of the privacy budget in order to answer their queries independently using HDMM. In this case Alice and Bob both receive a total expected error of ± 198 people while Carol receives an expected error of ± 18 people.*

Suppose the data curator decides to pool the queries and jointly answer them using HDMM. Alice and Bob receive ± 22 as expected their error which is less than their error using the independent mechanism. Carol received ± 22 as her expected error which is more error than in the independent case where her expected error was ± 18 thus violating the Sharing Incentive.

In this case Carol would prefer her workload to be answered independently while Alice and Bob would join together. If the mechanism were to satisfy the Sharing Incentive, Carol would be guaranteed no worse error by joining Alice and Bob and as such should always make that choice.

The second desiderata is **Non-Interference**, which states that adding an additional analyst to the collective group, with their associated share of the privacy budget, should not increase the error experienced by any of the analysts already in the collective. This desiderata ensures that no analyst in the collective can ask (intentionally or unintentionally) a malicious set of queries which would increase the error of any of the other analysts more than if they had never joined the collective. Likewise, Non-interference ensures that adding more analysts to the collective (and with them more privacy budget) can only improve the accuracy of all agents.

DEFINITION 4 (NON-INTERFERENCE). *A mechanism $\mathcal{M}$ satisfies Non-Interference if for all analysts $i \neq j$, for all workloads W_i, W_j*

$$\mathrm{Err}_i(\mathcal{M}, \mathcal{W}, \epsilon) \leq \mathrm{Err}_i(\mathcal{M}, \mathcal{W} \setminus W_j, (1 - s_j)\epsilon)$$

EXAMPLE 3. *Alice and Bob have decided to join the collective and answer their queries together since they have the same queries. They run the joint mechanism on their queries using $\frac{2}{3}\epsilon$ of the budget. Here they both receive ± 22 people as expected error. Carol then joins the collective. They then rerun the same mechanism using the entire budget. In this case, Alice and Bob receive an expected error of ± 24 people, which is more than their original ± 22 people expected error therefore violating Non-Interference.*

In this case, Carol joining the collective makes both Alice and Bob worse off. If the mechanism were to satisfy Non-Interference, Alice and Bob would be guaranteed that no matter what workload Carol asks they can be to be no worse off for allowing Carol into the collective.

Our third desideratum is **Adaptivity**, which states that a mechanism should be able to adapt to the inputs given. We say that a mechanism is adaptive if it changes its query answering strategy

based off all the inputs given. This ensures that a mechanism can adapt to the specific queries being asked by analysts in order to avoid high error for particular sets of queries.

EXAMPLE 4. *The data curator chooses to use a non-adaptive mechanism which always releases data by answering the Identity workload. Alice and Bob are happy since this is their exact workload. Carol is punished since her query workload cannot be efficiently reconstructed using the identity workload and receives an expected error of ± 24 people, which is worse than her independent expected error of ± 18 people.*

An adaptive mechanism would be able to adapt it's query answering strategy in order to account for Carol's queries therefore reducing her error. The concept of Adaptivity highlights the flaws of various trivial mechanisms which satisfy the Sharing Incentive and Non-Interference by intentionally ignoring the inputs or interactions between analysts inputs.

Tradeoffs Between Desiderata and Accuracy. Both the Sharing Incentive and Non-interference add additional constraints to mechanisms in the multi-analyst setting. As such, we expect that mechanisms which satisfy these desiderata will suffer some accuracy loss. In contrast, adaptivity is not in conflict with accuracy. Rather adaptivity is a requirement that a mechanism should optimize it's query strategy to be more efficient for a given workload. Overall, we expect a mechanism that is adaptive to perform better over a wide range of queries as opposed to its non-adaptive counterpart.

3.3 Problem Statement

Our goal is to design multi-agent differentially private mechanisms that answer the workloads submitted by the analysts with low error while satisfying the three desiderata – sharing incentive, non-interference and workload adaptivity. More formally:

PROBLEM 1. *Given any k workloads $W_1, \ldots, W_k$ of queries on a database D with weights $s_1, \ldots, s_k \in (0,1)$ s.t. $s_1 + \ldots + s_k = 1$, design an adaptive mechanism $\mathcal{M}$ such that:*

- *$\mathcal{M}$ satisfies ϵ-differential privacy, and*
- *$\mathcal{M}$ satisfies sharing incentive (Definition 3), non-interference (Definition 4) .*

4 DESIGN PARADIGMS

Here we introduce 4 design paradigms which we use to guide our design of multi analyst differentially private mechanisms. We call the first two classes Independent and Workload Agnostic. These classes use existing mechanisms without explicitly considering the group structure of the problem. We consider them as baselines for comparison; it is easy to see in theory and we show empirically that these mechanisms lead to poor performance with respect to total error. We call the other two classes of mechanisms Collect First and Select First. These mechanisms adapt the Select Measure Reconstruct Paradigm by aggregating all analysts' queries either before or after the selection step respectively. Each of the paradigms are depicted in Figure 1.

Independent Mechanisms give each analyst their share of the overall privacy budget proportional to their weights $s_1, \ldots s_k$ and answers each analyst's queries independently of one another using some workload answering mechanism. Mechanisms of this class by definition satisfy both the Sharing Incentive and Non-Interference since analysts always have the same expected error regardless of how many analysts are in the collective or what their queries are.

LEMMA 1. *Any Independent Mechanism satisfies both the Sharing Incentive and Non-Interference*

These mechanisms are not efficient as they typically answer each individual query with less privacy budget than other mechanisms and may answer the same or similar queries multiple times. In Section 6 we will show a mechanism that satisfies all the desiderata and can achieve up to $\sqrt{k}$ times better error than its independent counterpart.

Workload Agnostic Mechanisms always answer the same set of queries with the entire budget regardless of the analysts' workloads. Mechanisms of this class also trivially satisfy both the Sharing Incentive and Non-Interference since the same workload is always answered regardless of the preferences of the analysts. Joining the collective only increases the overall privacy budget leading to an overall decrease in error, satisfying the sharing incentive. Likewise, whenever a new analyst joins the collective the workload remains the same and the privacy budget increases causing an overall decrease in error for all analysts, therefore satisfying Non-Interference.

LEMMA 2. *Any Workload Agnostic mechanism satisfies both the Sharing Incentive and Non-Interference*

Workload Agnostic Mechanisms are not adaptive and this causes them to be inefficient with respect to total error, even for a single analyst. For example, if a noisy count was released for people of ages $\{0, 1, 2 \ldots 99\}$ but an analyst asks for the total count of all people then the answer to the total query, reconstructed by adding together all of the noisy counts, has at least 10 times larger error than if the total query was answered directly using all of the privacy budget.

Collect First Mechanisms collect all analysts' queries together before the selection step. These mechanisms combine all of the workloads of each analyst into some weighted query set, then run the selection step to select a single strategy workload for all the analysts' workloads.

Select First Mechanisms collect all the analysts' queries after the selection step in a Select Measure Reconstruct mechanism. Mechanisms of this class allow an individual strategy for each analyst's queries. After a strategy is selected for each analyst's queries they are all aggregated into a joint strategy workload which is answered directly.

The fundamental problem with designing a Collect-First Mechanism that satisfies the sharing incentive and Non-Interference is that it is difficult to reason about or enforce the properties on any useful selection step (such as HDMM) optimizing on the joint workload. Select-First Mechanisms are easier to work with because we do not need to reason over the multi-analyst properties of the selection step when it is applied to each analyst independently first.

5 ADAPTING EXISTING MECHANISMS

We introduce and analyze four mechanisms for multi-analyst query answering. Each of these mechanisms directly invokes one of the
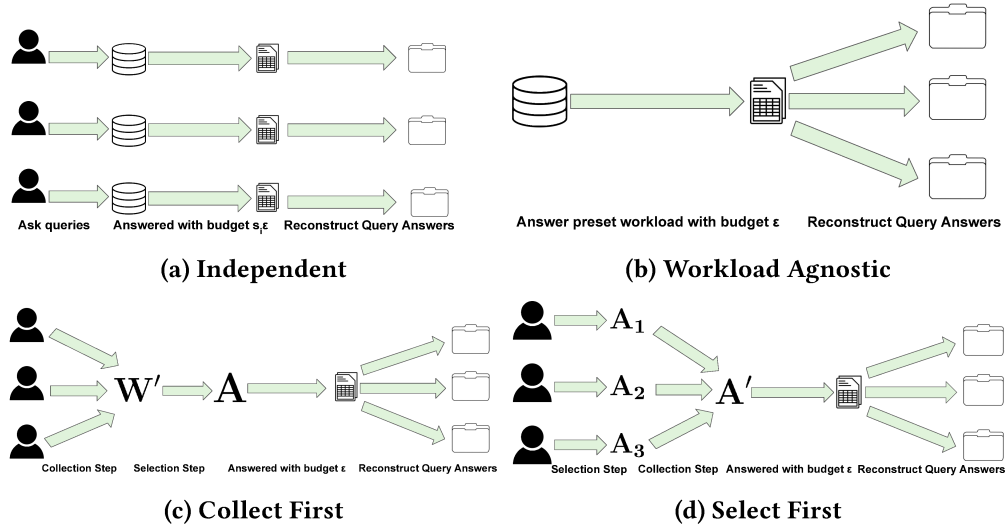
(a) Independent   **(b) Workload Agnostic**

(c) Collect First   **(d) Select First**

Figure 1: Design Paradigms for Multi-Analyst DP Query Answering

Table 1: Desiderata satisfied by algorithms

Desiderata / Mechanism	Sharing Incentive	Non-interference	Adaptivity
Independent	✓	✓	✓
Identity	✓	✓	✗
Utilitarian	✗	✗	✓
Weighted Utilitarian	?	✗	✓
0-Waterfilling	✓	✓	✓

design paradigms listed in Section 4. Of these mechanisms, the Independent Mechanism and Identity Mechanism are both direct applications of existing mechanisms to the multi-analyst setting. The Utilitarian Mechanism and Weighted Utilitarian Mechanism are adaptations of HDMM[27] to the multi-analyst setting. The desiderata satisfied by algorithms are shown in Table 1

All of the properties and proofs given hold for linear queries which are answered directly by the mechanisms. These properties do not inherently hold for any non-linear queries reconstructed from linear query answers. In Section 7.4 we empirically study our mechanisms answering non-linear queries such as median and percentiles.

Independent HDMM invokes the Independent Mechanism paradigm and simply runs HDMM[27] for each analyst using their share of the privacy budget. We use this mechanism as a baseline to compare other mechanisms to in Section 7. As an example of an Independent Mechanism, it satisfies all three of the desiderata.

Identity Mechanism, as shown in algorithm 1 answers the identity strategy regardless of the preferences of the individual analysts. As an example of a workload agnostic mechanism it satisfies the Sharing Incentive and Non-Interference but is non-adaptive.

The Utilitarian Mechanism is an example of a collect first mechanism. The Utilitarian Mechanism first aggregates each analysts' queries by creating a multi-set of queries which contains all the

analysts' queries with multiplicity equal to the number of analysts asking the query. We then run a selection step on this joint query set. In general, we would expect this mechanism to achieve the minimum expected total error across all analysts, but the mechanism can easily violate both the Sharing Incentive and Non-Interference.

Algorithm 1: Independent Mechanism

input : Set of k workloads $\mathcal{W} \leftarrow \{W_1, W_2, \ldots, W_k\}$,
         Set of k budget weights $S \leftarrow \{s_1, s_2, \ldots, s_k\}$,
         Data vector $\boldsymbol{x}$,
         privacy budget ϵ,
         Selection Mechanism $\mathcal{M}$

  Selection Step

1   $\mathcal{A} \leftarrow \{\mathcal{M}(W_i) \mid W_i \in \mathcal{W}\}$

  Measure step

2   $Y \leftarrow \{A_i \boldsymbol{x} + \text{Lap}(\frac{1}{s_i \epsilon} \|A_i\|_1) \mid A_i \in \mathcal{A}, s_i \in S\}$

  Reconstruct step

3   $\bar{X} \leftarrow \{A_i^+ \boldsymbol{y}_i \mid A_i \in \mathcal{A}, \boldsymbol{y}_i \in \boldsymbol{y}\}$

     ▷ A^+ is the Moore–Penrose inverse of A and $A^+A = I$.

4   $\text{ans} \leftarrow \{W_i(\bar{\boldsymbol{x}}_i) \mid W_i \in \mathcal{W}, \bar{\boldsymbol{x}}_i \in \bar{X}\}$

5   **return** ans

THEOREM 2. *The Utilitarian Mechanism does not satisfy the sharing incentive*

PROOF. Take the general case from Example 2 where we have k analysts. $k - 1$ of those analysts ask the Identity workload (a histogram of all counts). The last analyst asks the Total workload (a sum of all counts). Each analyst receives an equal share $\frac{\epsilon}{k}$ of the privacy budget. In the independent case each analyst asking the Identity workload would receive identity as a strategy and would experience $\left(\frac{2k}{\epsilon}\right)^2 n$ error. The one analyst asking total would

Algorithm 2: Identity Mechanism

> **input** : $\mathcal{W}, S, x, \epsilon, \mathcal{M}$ ▷ defined in Algorithm 1
> **Selection Step**
> 1 $A \leftarrow \text{Identity}(n)$
>   **Measure step**
> 2 $y \leftarrow Ax + \text{Lap}(\frac{1}{\epsilon})$
>   **Reconstruct step**
> 3 $\bar{x} \leftarrow A^{+}y$
> 4 $\text{ans} \leftarrow \{W_i(\bar{x}) \mid W_i \in \mathcal{W}\}$
> 5   **return** ans

receive the Total workload as their strategy and experience $\left(\frac{2k}{\epsilon}\right)^2$ error. If all the analysts join the collective an optimal utilitarian mechanism would chose the Identity workload as the workload that optimizes on total error. In this case (now using the entire privacy budget) each analyst would receive $\left(\frac{2}{\epsilon}\right)^2 n$ error. In this case all the analysts asking the identity workload would benefit while the analyst asking the Total workload will get increasingly worse error as n (the size of the database) increases and will violate sharing incentive when $k^2 < n$.   □

THEOREM 3. *The Utilitarian Mechanism does not satisfy non-interference*

PROOF. Consider the case where there are k analysts each with an equal $\frac{\epsilon}{k}$ share of the privacy budget. $k-1$ of these analysts ask the Total workload and the last analyst asks the identity workload. If the $k-1$ analysts asking the Total workload are in the collective the strategy used would directly answer the Total workload and receive $\left(\frac{2k}{(k-1)\epsilon}\right)^2$ expected error. If the last analyst were to join an optimal utilitarian mechanism would answer the queries using the identity strategy which optimizes on overall error. This would result in the $k-1$ analysts each receiving $\left(\frac{2}{\epsilon}\right)^2 n$ expected error which violates non interference when $\left(\frac{k}{k-1}\right)^2 < n$ □

The Weighted Utilitarian Mechanism is a variant of the Utilitarian Mechanism that attempts to directly optimize for the Sharing Incentive. This is achieved by weighting the queries prior to the collection step. This requires an additional set of k parameters which we call workload weights $\Omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$, where ω_i is the weight for workload W_i. After weighting each of the queries, the Utilitarian Mechanism is run on the weighted query sets. The Utilitarian Mechanism is a special case of Weighted Utilitarian Mechanism where $\omega_1 = \omega_2 = \cdots = \omega_k = 1$.

In an attempt to satisfy the Sharing Incentive we set the weights as the inverse of the expected error of the mechanism in the independent case.

$$\omega_i = \text{Err}_i(\mathcal{M}, W_i, s_i\epsilon)^{-1} \tag{3}$$

These weights incentivize an optimizer to satisfy the sharing incentive as an analyst's utility is above 1 only if they have less error than required to satisfy the sharing incentive. We see in Section 7 that these weights allow for the utilitarian mechanism to satisfy the sharing incentive in practical settings and we have not been able to create settings where the sharing incentive is violated. It is unclear if it satisfies the Sharing Incentive in all settings.

CONJECTURE 1. *The weighted utilitarian mechanism satisfies the sharing incentive*

In Section 7.4 we are able to show empirically that the weighted utilitarian mechanism does violate non interference.

THEOREM 4. *The weighted utilitarian mechanism does not satisfy non-interference*

Algorithm 3: Weighted Utilitarian Mechanism

> **input** : $\mathcal{W}, S, x, \epsilon, \mathcal{M},$ ▷ defined in Algorithm 1
>   Set of k workload weights $\Omega \leftarrow \{\omega_1, \omega_2, \ldots, \omega_k\}$
>   **Collection Step**
> 1 $\mathcal{W}' \leftarrow \biguplus_{i=1}^{k} \omega_i W_i$ ▷ $\uplus$ is multi-set union
>   **Selection Step**
> 2 $A \leftarrow \mathcal{M}(\mathcal{W}')$
>   **Measure step**
> 3 $y \leftarrow Ax + \text{Lap}(\frac{1}{\epsilon} \|A\|_1)$
>   **Reconstruct step**
> 4 $\bar{x} \leftarrow A^{+}y$
> 5 $\text{ans} \leftarrow \{W_i(\bar{x}) \mid W_i \in \mathcal{W}\}$
> 6   **return** ans

6 THE WATERFILLING MECHANISM

The Waterfilling Mechanism is an example of a select first mechanism which satisfies all three of the desiderata. We first start with a simplified example of the Waterfilling Mechanism seen in Figure 2 and then discuss the full Waterfilling Mechanism.

In this example there are three analysts Alice, Bob, and Carol each given the same share of the budget, $\frac{1}{3}$. Alice asks only the blue query and assigns all of her share to that query. Bob asks the red, blue, and green queries and assigns each query equal amounts of his share of the privacy budget. Carol asks the blue and green queries and like Bob assigns his share of the budget equally across all her queries. The Waterfilling mechanism then buckets similar queries (in this example by bucketing red blue and green queries) and their associated shares of privacy budget together. Once all the queries are assigned to buckets the mechanism answers a single query for each bucket using the entire privacy budget in each bucket. The mechanism then uses those answered queries to reconstruct the analysts original queries. In Figure 2, we can see that since the red query was only asked by one analyst it receives the same amount of privacy budget as if were asked independently. Meanwhile since each analyst asked the blue query it is answered once using the pooled contribution of privacy budget from each analyst, resulting in a more accurate estimate than if each analyst had independently answered the blue query, even if they subsequently shared their results with one another.

The example shown in Figure 2 is a simplified version of the Waterfilling Mechanism. The Waterfilling Mechanism as defined in Algorithm 4 has three key differences. The first key difference is the selection step. In the simplified Waterfilling Mechanism analyst's queries are bucketed directly. However in practice a selection step
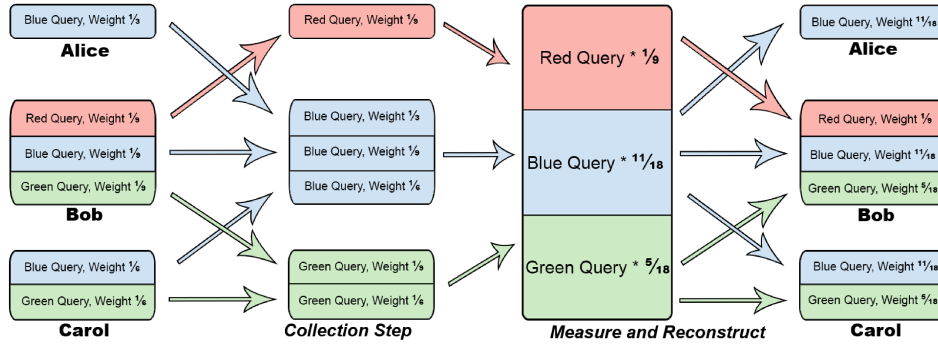
Figure 2: Simplified Waterfilling Mechanism

Algorithm 4: τ - Waterfilling Mechanism

input : $\mathcal{W}, S, \boldsymbol{x}, \epsilon, \mathcal{M},$ ▷ defined in Algorithm 1
        tolerance parameter τ

Selection Step

1   $\mathcal{A} \leftarrow \{\mathcal{M}(W_i) \mid W_i \in \mathcal{W}\}$

Collection step

2   buckets $\mathcal{B} \leftarrow \{\}$

3   **for** $A_i \in \mathcal{A}$ **do**

4   **for** $v \in \text{Rows}(s_i A_i / \|A\|_1)$ **do**

5   **if** exists $B \in \mathcal{B}$ **s.t.** $\text{sim}(v, \sum_{\boldsymbol{u} \in B} \boldsymbol{u}) \geq 1 - \tau$ **then**
                     ▷ sim is the cosine similarity

6         $B \leftarrow B \cup \{v\}$

7   **else**

8   **new** $B \leftarrow \{v\}$

9         $\mathcal{B} \leftarrow \mathcal{B} \cup \{B\}$

10   $A \leftarrow \text{Mat}\left(\{\sum_{\boldsymbol{u} \in B} \boldsymbol{u} \mid B \in \mathcal{B}\}\right)$
    ▷ Mat converts a set of vectors into a matrix, each row of A
      is the sum of vectors in a bucket

Measure step

11   $\boldsymbol{y} \leftarrow A\boldsymbol{x} + \text{Lap}(\frac{1}{\epsilon}\|A\|_1)$

Reconstruct step

12   $\bar{\boldsymbol{x}} \leftarrow A^+\boldsymbol{y}$

13   ans $\leftarrow \{W_i(\bar{\boldsymbol{x}}) \mid W_i \in \mathcal{W}\}$

14   **return** ans

---

is done first. This selection step takes in the analyst's workload and outputs a strategy workload that may be more efficient to answer directly. The second key difference is sensitivity scaling. The simplified example assumes that the sensitivity of each query is 1 and that all three queries overlap somewhat causing Alice's sensitivity to be 1 Bob's sensitivity to be 3 and Carols sensitivity to be 2. In order to avoid sensitivity scaling issues, the Waterfilling Mechanism scales each analyst's strategy workload to have a sensitivity of 1 prior to the bucketing step. The third key difference is in the bucketing step. In the simplified example we only bucketed identical queries. Since the selection step introduces some numerical instability we allow for queries which are approximately equal to be added to the same bucket. We introduce an additional parameter τ which

determines how much two queries are allowed to deviate to be assigned to the same bucket. In Algorithm 4 we allow two queries with cosine similarity greater than $1 - \tau$ to be assigned to the same bucket. Once the buckets are filled the query answered is the unit vector representing the average query in the bucket.

All of the proofs below assume that $\tau = 0$ and may not hold for higher values of τ. We set τ to be 10^{-3} in experiments and we empirically evaluate the performance of the Waterfilling mechanism as τ changes in Section 7.4.

Here we prove a stronger property than either the Sharing Incentive or Non-Interference. We show that adding an additional analyst to an arbitrary collective increase the error experienced by any analyst, A property we call Analyst Monotonicity.

THEOREM 5. *Let $\mathcal{W}$ be the set of all workloads of the analysts in an arbitrary collective. For all analysts $i \neq j$, for all workloads $W_i \in \mathcal{W}, W_j \notin \mathcal{W}$ the 0-Waterfilling mechanism satisfies both of the following*

$$\text{Err}_i\left(\mathcal{M}, \mathcal{W} \cup W_j, \left[s_j + \sum_{l:W_l \in \mathcal{W}} s_l\right]\epsilon\right) \leq \text{Err}_i\left(\mathcal{M}, \mathcal{W}, \left[\sum_{l:W_l \in \mathcal{W}} s_l\right]\epsilon\right)$$
$$(4)$$

$$\text{Err}_j\left(\mathcal{M}, \mathcal{W} \cup W_j, \left[s_j + \sum_{l:W_l \in \mathcal{W}} s_l\right]\epsilon\right) \leq \text{Err}_j\left(\mathcal{M}, W_j, s_j\epsilon\right) \quad (5)$$

We first show that regardless of the number of analysts in the collective the scale of the noise added to the queries remains the same. We then show that the error introduced by reconstructing the original query answers (frobenius norm term of Equation (1)) can only decrease as more analysts are added the collective therefore resulting in error that either decreases or remains the same for each analyst.

LEMMA 3. *Consider adding an analyst to the collective with strategy matrix A_i and weight s_i. If the L1 norm of every column of A_i is 1, the sensitivity of the resultant strategy queries will increase by s_i, formally*

$$\|A'\|_1 = \|A\|_1 + s_i,$$

where A and A' are the resultant strategy matrix before and after adding this analyst respectively.

PROOF. For any matrix M, We define cnorm(M) as a vector where the ith entry is the L1 norm of the ith column of M, formally

$$\text{cnorm}(M) = \sum_{v \in M} |v|,$$

where v's are the row vectors of M and $|v|$ is the vector which takes entry-wise absolute value of v.

In Alg. 4, each row of A corresponds to a bucket $B \in \mathcal{B}$. Thus, particularly for A,

$$\text{cnorm}(A) = \sum_{v \in A} |v| = \sum_{B \in \mathcal{B}} \left| \sum_{u \in B} u \right|. \tag{6}$$

Consider adding a query v' to buckets $\mathcal{B}$ and let the new buckets be $\mathcal{B}'$. Let $e' = v'/\|v'\|$. If $e' \cdot e_B < 1$ for all buckets $B \in \mathcal{B}$, v' will be put in a new bucket B' and thus $|\sum_{u \in B'} u| = |v'|$. Also, $\mathcal{B}' = \mathcal{B} \cup \{B'\}$.

Otherwise, there exists a bucket $B^* \in \mathcal{B}$ and $e' \cdot e_{B^*} = 1$. In this case, v' will be put in the bucket B^* and $\mathcal{B}' = \mathcal{B}$ with updated $B^{*'}$. Since e' and e_{B^*} are both unit vector, $e' \cdot e_{B^*} = 1$ means $v'/\|v'\| = e' = e_{B^*} = \sum_{u \in B^*} u / \| \sum_{u \in B^*} u \|$. Thus,

$$\left| \sum_{u \in B^{*'}} u \right| = \left| \sum_{u \in B^*} u + v' \right| = \left| \sum_{u \in B^*} u \right| + |v'|.$$

In both cases, we have

$$\sum_{B \in \mathcal{B}'} \left| \sum_{u \in B} u \right| = \sum_{B \in \mathcal{B}} \left| \sum_{u \in B} u \right| + |v'|. \tag{7}$$

In this process, we add $s_i A_i$ to B resulting in B'. From Equation (6) and Equation (7) we get,

$$\text{cnorm}(A') = \sum_{B \in \mathcal{B}'} \left| \sum_{u \in B} u \right| = \sum_{B \in \mathcal{B}} \left| \sum_{u \in B} u \right| + \sum_{v \in s_i A_i} |v|$$
$$= \text{cnorm}(A) + \text{cnorm}(s_i A_i).$$

Given the L1 norm of every column of $A_i \in \mathcal{A}$ is 1, we have cnorm($s_i A_i$) = $s_i \mathbf{1}$, where $\mathbf{1}$ is a all-one vector. Since the L1 norm of a matrix is the maximum of all L1 column norms, we have

$$\|A'\|_1 = \max(\text{cnorm}(A')) = \max(\text{cnorm}(A) + s_i \mathbf{1})$$
$$= \max(\text{cnorm}(A)) + s_i = \|A\|_1 + s_i$$

$\square$

Since we can consider the strategy matrix with no analysts as the zero matrix, and adding an additional analyst adds their weight to the sensitivity, the L1 norm for the strategy matrix for k analysts is

$$\|A\|_1 = \sum_{i=1}^{k} s_i$$

Since the ith analyst is entitled to $s_i \epsilon$ of the budget and the sensitivity of the strategy query set is equal to the sum of each analysts' weights, the scale of the noise term in Equation (1) is the same regardless of the number of analysts. Let $z \leq k$ be any arbitrary number of analysts. The scale of the noise term in Equation (1) is as follows.

$$\frac{2 \|A\|_1^2}{\epsilon^2} = \frac{2 \left(\sum_{i=1}^{z} s_i \right)^2}{\left(\sum_{i=1}^{z} s_i \epsilon \right)^2} = \frac{2}{\epsilon^2} \tag{8}$$

Since the amount of noise being added to each query in the final strategy is the same, the amount of error experienced by each analyst is only dependent on the frobenius norm term of Equation (1).

We first note that adding a new analyst to the collective results in a change to the overall strategy matrix that can either be expressed by multiplying it by some diagonal matrix with all entries greater than 1 (adding weight to a bucket) or by adding additional rows (creating new buckets). We show below that either of these operations results in a frobenius norm term that is no greater than the term with the original strategy matrix.

LEMMA 4. *For any workload matrix W and any strategy A*

$$\left\| W(DA)^+ \right\|_F \leq \left\| WA^+ \right\|_F$$

where D is a diagonal matrix with all diagonal entries greater than or equal to 1 and A is a full rank matrix.

PROOF. We first note that since D is a diagonal matrix with all entries greater than or equal to 1 then D^{-1} is a diagonal matrix with all values less than or equal to 1. Since this matrix cannot increase the value of any entry of any matrix multiplied by it the following holds.

$$\left\| WA^+ D^{-1} \right\|_F \leq \left\| WA^+ \right\|_F$$

We then note that $WA^+ D^{-1}$ is a solution to the linear system of equations $B(DA) = W$. Since $WA^+ D^+$ is a solution to the linear system of equations then it is the least squares solution to the set of linear equations [5] and as such the following holds.

$$\left\| W(DA)^+ \right\|_F \leq \left\| WA^+ D^{-1} \right\|_F \leq \left\| WA^+ \right\|_F$$

$\square$

LEMMA 5. *Let $\tilde{A}$ be the original strategy matrix A with additional queries (rows) added to it. We can write this as a block matrix as $\tilde{A} = \begin{bmatrix} A \\ C \end{bmatrix}$ Where C are the additional queries. For any workload W and any strategy A*

$$\left\| W\tilde{A}^+ \right\|_F \leq \left\| WA^+ \right\|_F$$

PROOF. Let $\hat{A}$ be the original matrix A padded with additional rows of zeros in order to be the same size as $\tilde{A}$ written in block matrix form as $\hat{A} = \begin{bmatrix} A \\ 0 \end{bmatrix}$. We note that by the formula for block matrix pseudo-inverse, the pseudo-inverse of $\hat{A}$ is as follows. $\hat{A}^+ = \begin{bmatrix} A^+ & 0 \end{bmatrix}$ We then note that $W\hat{A}^+$ is a solution to the linear system of equations as follows.

$$W\hat{A}^+ \tilde{A} = W \begin{bmatrix} A^+ & 0 \end{bmatrix} \begin{bmatrix} A \\ C \end{bmatrix} = WA^+ A = W$$

Therefore since $W\hat{A}^+$ is a solution to the linear system of equations and since $W\tilde{A}^+$ is the least squares solution to the linear set of equations [5] we get the following.

$$\left\| W\tilde{A}^+ \right\|_F \leq \left\| W\hat{A}^+ \right\|_F = \left\| WA^+ \right\|_F$$

$\square$

Proof of Theorem 5. Let A be the strategy matrix produced by the Waterfilling Mechanism without analyst j. Let $\tilde{A}$ be A with additional rows appended to it and let D be a diagonal matrix with all entries 1 or greater.

$$
\mathrm{Err}_i\left(\mathcal{M}, \mathcal{W} \cup W_j, \left[s_j + \sum_{l:W_l \in \mathcal{W}} s_l\right]\epsilon\right)
$$

$$
= \quad \frac{2}{\epsilon^2} \left\|W_i(D\tilde{A}^+)\right\|_F^2 \quad \text{(from Equation (8))}
$$

$$
\leq \quad \frac{2}{\epsilon^2} \left\|W_i\tilde{A}^+\right\|_F^2 \quad \text{(from Lemma 4)}
$$

$$
\leq \quad \frac{2}{\epsilon^2} \left\|W_iA^+\right\|_F^2 \quad \text{(from Lemma 5)}
$$

$$
= \quad \mathrm{Err}_i\left(\mathcal{M}, \mathcal{W}, \left[\sum_{l:W_l \in \mathcal{W}} s_l\right]\epsilon\right)
$$

If we instead assume A is the strategy matrix produced by the Waterfilling Mechanism with only analyst j then the same process satisfies Equation (5). □

Since adding an additional analyst to the collective can only decrease the amount of expected error experienced by any analyst, we have the following as corollaries for Theorem 5.

COROLLARY 1. *Waterfilling Mechanism satisfies sharing incentive*

COROLLARY 2. *Waterfilling Mechanism satisfies non-interference*

Unlike Independent Mechanisms, Waterfilling Mechanisms satisfy all the desiderata while being efficient with respect to error.

THEOREM 6. *The Waterfilling Mechanism can achieve as much as k times better error than the Independent Mechanism and always achieves no more error than the Independent Mechanism.*

PROOF. Consider the pathological example of k analysts each of whom ask the same single linear counting query to be answered with the Laplace Mechanism. In this case the overall expected error using the Waterfilling mechanism is that of answering the single query once using the entire privacy budget using the Laplace mechanism. This results in an expected error of $\frac{2}{\epsilon^2}$. If each analyst were to independently answer their queries using $\frac{\epsilon}{k}$ of the budget each and then post process the k results by taking the sample median it would result in a mean squared error of $\frac{2k}{\epsilon^2}$. By Corollary 1 the Waterfilling Mechanism always achieves at most as much error as the Independent Mechanism satisfying the second statement. □

7 EXPERIMENTS

We designed experiments to both test if the mechanisms proposed satisfy the desiderata as well as how they perform in practice. We show 4 different experiments using different inputs and data sets.

- **Practical Settings**: We show that the Waterfilling Mechanism maintains high efficiency while still satisfying all three desiderata. We also show that mechanisms that optimize for overall error such as the Utilitarian mechanism fail to satisfy both the Sharing Incentive and Non-Interference.

- **Marginals**: Here we show that non-adaptive mechanisms such as the Identity mechanism may incur high error on particular classes of queries such as marginal queries, while adaptive mechanism can perform well on wide ranges of queries.
- **Data-Dependent Non-linear Queries**: We show that the Waterfilling Mechanism retains it's properties when used to reconstruct non-linear queries from a set of linear strategy queries.
- **Tolerance for Waterfilling**: We evaluate the efficacy and properties of the mechanism using various levels of τ and show that $\tau = 10^{-3}$ performs well and does not result in any violations of the sharing incentive.

7.1 Experimental Setup

For the following experiments we use HDMM [27] as the selection step, but any selection step can be used in practice. In addition, we can consider the Identity Mechanism a variant of matrix mechanism with a fixed identity strategy matrix I, MM(I).

For all experiments we used $\epsilon = 1$ for our total privacy budget. In addition, The Waterfilling Mechanism has a tolerance parameter τ. We experimented with several values of τ. Results shown in Section 7.4 found $\tau = 0.001$ is a value that achieves good overall accuracy. As such we set it to be 0.001 in all our experiments.

For the figures, each workload is given an abbreviations as follows: Ind (Independent HDMM), Iden (Identity mechanism), Util (Utilitarian HDMM), WUtil (Weighted Utilitarian HDMM), and Water (HDMM Waterfilling Mechanism). For each experiment we run the optimization 10 times and pick the strategy with the minimum loss.

7.2 Empirical Measures

We design several empirical measures based on our desiderata to provide an overall understanding of the mechanisms. All measures are with respect to a single mechanism and a single set of workloads. **Total Error** is the sum of expected errors of all analysts. This is a common measure found in the literature to show the efficiency of the algorithm.

Maximum Ratio Error of a mechanism $\mathcal{M}$ for a given analyst is the expected error of $\mathcal{M}$ divided by the expected error of the independent version. For non-independent adaptive algorithms, it is a measure of the Sharing Incentive as it measures to what extent one analyst gets better or worse off compared to asking the query on their own. We present the maximum of the ratio errors among all analysts. The maximum ratio error amongst all analysts is

$$
\max_i\left(\frac{\mathrm{Err}_i(\mathcal{M}, \mathcal{W}, \epsilon)}{\mathrm{Err}_i(\mathcal{M}, W_i, s_i\epsilon)}\right).
$$

If the value is larger than 1, the mechanism violates the Sharing Incentive as the error in the joint case is greater than the error experienced in the independent case.

Empirical Interference is a quantifiable measure to show the extent which a mechanism violates Non-Interference or the distance from violating it. For each analyst i, we define the interference with respect to another analysts j as the ratio of the expected error for analyst j when all analysts are included to the case when excluding analyst i. If this ratio is larger than 1, analyst j can be worse off

when analyst i joins the workload set. We define the interference of analyst i on analyst j to be

$$I_i(j) = \frac{\text{Err}_j(\mathcal{M}, \mathcal{W}, \epsilon)}{\text{Err}_j(\mathcal{M}, \mathcal{W}_i^c, (1 - s_i)\epsilon)}$$

This represents the relative change in error experienced by analyst j when analyst i joins the collective. We then define the interference of mechanism $\mathcal{M}$ on the set $\mathcal{W}$ as the maximum of interference among all analysts, as

$$I_{\mathcal{M}}(\mathcal{W}) = \max_{1 \le i, j \le k, i \ne j} I_i(j).$$

Intuitively, it represents the maximum ratio increase of the expected error of any analyst when another analyst joins the workload set. If $I_{\mathcal{M}}(\mathcal{W}) \le 1$, mechanism $\mathcal{M}$ satisfies Non-Interference on $\mathcal{W}$. Since $\mathcal{M}$ is usually a non-deterministic mechanism, rerunning the mechanism with $\mathcal{W}_i^c$ may give different strategy matrices to other analysts. Thus, we fix strategy matrices for Select First Mechanisms to ensure a more reasonable comparison. Since the strategies used by Collect First Mechanisms are dependent on each analysts input it is not possible to fix the strategy matrix.

7.3 Workloads and Datasets

Here we describe the methods used to generate workloads for each analyst as well as the data-sets used. When considering only linear queries all of our mechanisms are data independent and as such do not require a dataset in order to be evaluated. We only use a dataset when we extend our evaluation to non-linear queries and data dependent queries.

Practical settings: We generate practical settings using a series of random steps using the census example workloads provided in [27]. We tested on the race workloads with domain size $n = 64$.

(1) We first fix the domain size n. We then generate the number of analysts by picking an integer k uniformly random from $[2, k_{\max}]$. We let the number of analysts be k. Each analyst is given equal weight.
(2) Each analyst then pick a workload uniformly random from the set of 8 workloads, including 3 race workloads, Identity, Total, Prefix Sum, H2 workload, and custom workload.
(3) If they get custom workload, we chose their matrix size by picking an integer uniformly random from $[1, 2n]$.
(4) For each query in the matrix we chose a class of query uniformly sampled from the set including range queries (0-1 vector with contiguous entries), singleton queries, sum queries (random 0-1 vector) and random queries (random vector). The query is thus a random query within its class.
(5) The custom workload is thus a vertical stack of the queries.
(6) We repeat this procedure t times to get t randomly chosen sets of workloads. We call them t instances.

Marginals: We also experiment on another common type of workloads, marginals. For a dataset with d attributes with domain size n_i for the ith attribute, we can define a m-way marginal as the follows. Let S be a size m subset of $\{1, 2, \ldots, d\}$, we can express the workload as the Kronecker product $A_1 \otimes A_2 \otimes \ldots \otimes A_d$, where $A_i = I_{n_i}$ if $i \in S$ and $A_i = T_{n_i}$ otherwise. Here I_{n_i} is the identity workload matrix and T_{n_i} is the total workload matrix. Specifically, a 0-way marginal is the Total workload and a d-way marginal is the Identity

workload. Also, since there are $\binom{d}{m}$ size-m subset of $\{1, 2, \ldots, d\}$, there are $\binom{d}{m}$ different m-way marginals. In our experiments for simplicity, we use d attributes all with domain size 2. We repeat the process for generating analyst workloads from the practical settings in this case each individual analyst chooses a workload uniformly at random from the set of set of $\binom{d}{m}$ m-way marginals.

Data-dependent Non-linear Queries: In previous experiments, all workloads are linear and the expected error can thus be calculated without data. Our mechanisms can also be used for non-linear queries. We experiment on some common non-linear queries including *mean*, *medium*, and *percentiles* based on a histogram.

Error in this case is data-dependent and needs to be empirically calculated using real datasets. We use the Census Population Projections [2]. The dataset is Population Changes by Race. We choose year 2020 and Projected Migration for Two or more races. The domain size of data is $n = 86$, representing ages from 0 to 85.

As in the previous 2 experiments we use the procedure from practical settings in order to generate each analyst's workloads except the set of workloads to select from only contains 4 queries, *mean*, *medium*, *25-percentile*, and *75-percentile*. *Mean* is reconstructed from the workload containing the Total query T_n and the weighted sum query, a vector representing the attribute values (0 to 85 in our case). *Medium* and *percentiles* are reconstructed from the Prefix Sum workload P_n.

Tolerance for Water-filling: To examine the effect of tolerance in practice, we experimented on different values of tolerance τ for the HDMM Water-Filling mechanism. Figure 5 shows the case when $\tau \in [0.1, 0]$. We experimented with greater value of τ those values resulted in greater error and have been omitted from the figures. The workloads used are 1-way marginals as defined in Section 7.3.

7.4 Results

Practical settings: Figure 3a gives an overall view of the efficiency of different mechanisms. As expected, Utilitarian HDMM, a mechanism optimized for overall error, performs the best. Meanwhile Independent HDMM, a mechanism which does not utilize the group structure of the problem at all performs the worst. We note that the Weighted Utilitarian Mechanism in exchange for satisfying the sharing incentive performs slightly worse than the Utilitarian but performs better than the Waterfilling Mechanism which satisfies all three desiderata. The Waterfilling Mechanism performs as well as the Identity Mechanism while still satisfying adaptivity. This shows as stated in Section 3.2 that while there is a small cost in order to satisfy the sharing incentive and Non-Interference, satisfying adaptivity comes at no accuracy cost.

We present the results for $k_{\max} = 20$ as a representative in Figure 3. The figure is a box plot of $t = 100$ instances is generated randomly using the procedure in Section 7.3. The green line represents the median and the green triangle represents the mean. The box represents the interquartile range.

Figure 3b shows how other mechanisms compared with Independent HDMM in terms of maximum ratio error. Utilitarian HDMM violates the Sharing Incentive in a small number of instances as there are some outliers with maximum ratio error larger than 1. Weighted Utilitarian and The Waterfilling Mechanism satisfied the Sharing Incentive. Although Identity also has some outliers larger

(a) Practical Total Errors (log scale)   (b) Practical Max Ratio Errors   (c) Practical Empirical Interference

(d) Marginal Total Errors (log scale)   (e) Marginal Max Ratio Errors   (f) Marginal Empirical Interference
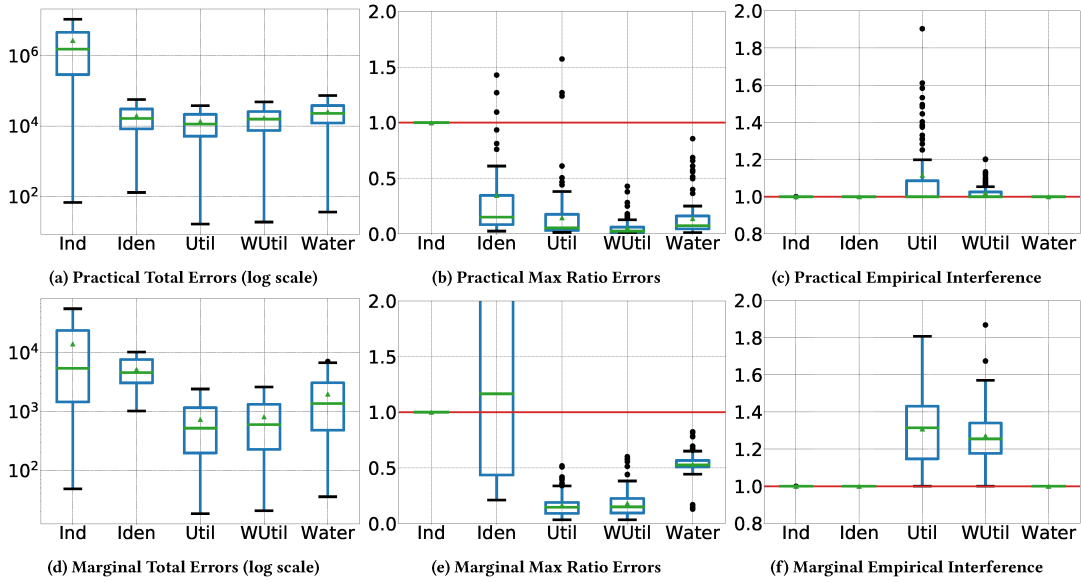
Figure 3: Empirical Measures for practical settings (above) and 1-way marginals (below). Values of maximum ratio error and empirical interference above 1 signify a violation of the Sharing Incentive and Non-Interference respectively.

than 1, since independent HDMM is not the independent form of this mechanism it does not violate the Sharing Incentive.

Figure 3c gives an empirical indication on whether a mechanism satisfies Non-Interference. It can be seen that both Utilitarian and Weighted Utilitarian HDMM violate Non-Interference in some cases. Weighted Utilitarian has fewer instances which violate Non-Interference than Utilitarian. The Weighted Utilitarian mechanism also violates Non-Interference to a smaller extent than the Utilitarian Mechanism. The other three mechanisms do not violate Non-Interference as we expect.

Marginal Workloads: In Figure 3 we show the results for 1-way marginal with $d = 8$, $k_{\max} = 20$, and $n = 256$. This figure also contains 100 instances. In particular, there are d 1-way marginals each corresponds to an attribute. Figure 3d shows Identity mechanism performs worse than the Waterfilling Mechanism and both Utilitarian mechanisms. The addition of the 1-way marginals drastically increases the error of identity compared to that of the other mechanisms. This is an example where the Identity Mechanism performs poorly with regard to total error for a common type of workloads. This is also observed for 1-way marginals with $d = 6, 7, 9, 10$. Figure 3e and Figure 3f are qualitatively similar to those in the practical settings. The Waterfilling Mechanism continues to satisfy all the desiderata while maintaining lower error than the Independent and Identity Mechanisms. Both Utilitarian mechanisms achieve lower overall error but at the cost of violating non interference.

Data-dependent Non-linear Queries: Figure 4a shows that the Independent Mechanism performs much worse than all other mechanisms in terms of total error. Figure 4b is the zoomed in version of Figure 4a, removing Independent. Since the answer of a non-linear query is reconstructed using the result of a different linear workload, Utilitarian is not guaranteed to have the lowest total

errors. We can see that Weighted Utilitarian outperforms Utilitarian here. The other two mechanism have higher total errors, and the Waterfilling Mechanism has a better median total errors than Identity.

Figure 4c and Figure 4d shows the max ratio errors and empirical interference. Since Independent and Identity mechanism satisfy the Sharing Incentive and Non-Interference by definition, we omit them here. We can see that all 3 other mechanisms satisfy the Sharing Incentive as they all have max ratio errors smaller than 1. Both Utilitarian mechanisms violate Non-Interference as shown in Figure 4d. Waterfilling Mechanisms satisfies Non-Interference. The outliers are due to numerical errors since we are using empirical expected errors instead of analytical ones.

These results show that our mechanisms also perform well for non-linear queries and have similar properties as the instances with linear queries. The results are qualitatively similar for $k_{\max} = 10$.

Tolerance for Water-filling Mechanism: Figure 5b shows that the total error is large at both ends, $\tau = 0.1$ and $\tau = 0$. The total error is the smallest for $\tau = 0.01$ and is also small for $\tau = 10^{-3}$ and $\tau = 10^{-4}$. This shows that there is no simple relation between the value of tolerance and total errors and we should not set $\tau = 0$ exactly in practice. Figure 5b shows the violation of Sharing Incentive when $\tau = 0.1$ and $\tau = 0.01$. From this result, we see that $\tau = 0.01$ is too large and $\tau = 10^{-3}$ (our default setting) is reasonable. We do not observe violation of Non-Interference any value of τ.

8 RELATED WORK

There has already been significant work on answering sets of queries in a differentially private manner, including theoretical lower bounds on error [6, 15] and many practical algorithms [4, 7, 8, 16, 23, 24, 27, 29, 32, 35, 37]. Each of these mechanisms primarily attempts to optimize the total error (or utilitarian social welfare)

(a) Total Errors (log scale)   (b) Total Errors (zoomed in)   (c) Max Ratio Errors   (d) Empirical Interference
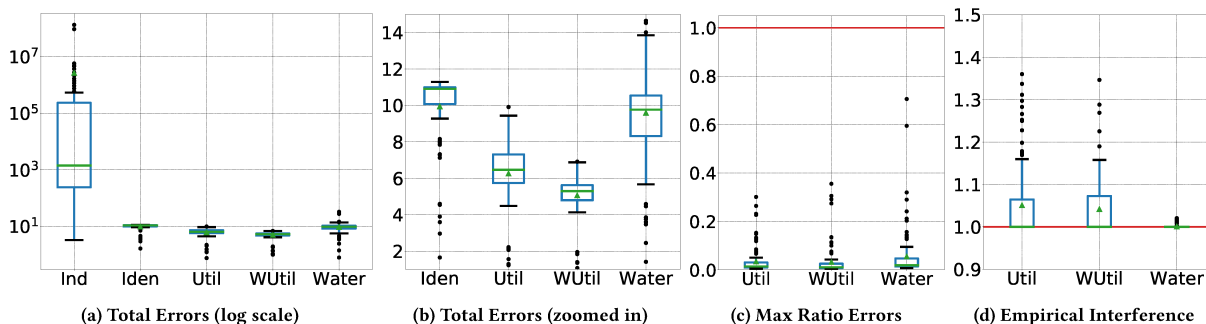
Figure 4: Empirical measures for non-linear queries. Errors shown are empirical expected errors calculated using real data. Values of maximum ratio error and empirical interference above 1 signify a violation of the Sharing Incentive and Non-Interference respectively.
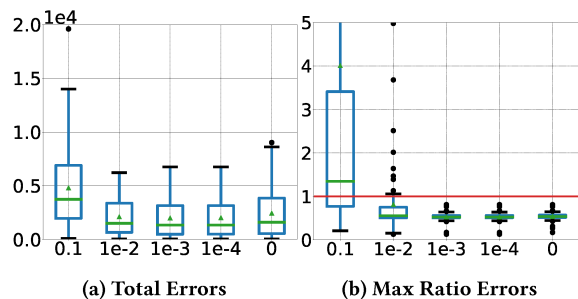

(a) Total Errors   (b) Max Ratio Errors

Figure 5: Total and Maximum Ratio Errors for 1-way marginals using HDMM Water-filling mechanism with different values of τ (x-axis). Values of maximum ratio error above 1 signify a violation of Sharing Incentive.

instead of distributing error in some manner. Likewise these mechanisms are not intended for any group answering setting but are instead designed for single analyst use.

Sharing computational resources such as memory and network has been considered in the context of resource allocation for data centers and networking [12, 13, 17, 19, 22, 30, 31]. For example, the influential work on dominant resource fairness [13] studies the allocation of several heterogeneous computational resources among agents (the owners of various jobs in a data center) and designs protocols that are simultaneously efficient and ensure good treatment of all agents through the Sharing Incentive and strategy-proof guarantees. In a sense, our work considers the same questions of how to design an effective shared system from the perspective of differential privacy and data release, recognizing that in the common case where there are multiple analysts, privacy budget is indeed a shared resource.

9 FUTURE WORK

There remain many technical problems in Differential Privacy which remain unanswered and may serve as powerful tools in the multi analyst setting. In this work we consider the offline setting where analysts submit their entire workload in advance and receive all of their answers at once. However most query answering settings are done in an online setting where analysts may adaptively chose their next query in response to a previous query answer. While there is some work on online differentially private query answering [20, 21] there are still significant hurdles to be overcome. To the best of our knowledge there is no differentially private mechanism which which answers queries with an adaptive strategy optimized to account for arbitrary prior knowledge. Such a mechanism would be essential to the online multi analyst problem as it would allow for prior query answers to inform future query answers and budget use.

10 CONCLUSION

We see as in Figure 3a that the traditional method of independently answering using fractional budgets results in an enormous increase in overall error when compared to joint mechanisms. In our practical cases we see over an order of magnitude difference between independent HDMM and HDMM waterfilling. We show in Figure 3b that a naively implemented joint mechanism (utilitarian HDMM) can result in violation of the Sharing Incentive resulting in some analysts gaining their extra utility at the expense of other analysts who are worse off than in the independent case. Likewise Figure 3c shows that naively implemented joint mechanisms can allow analysts to interfere with other analysts by asking vastly different query sets. In Figure 3d we show that mechanisms which are non-adaptive may suffer great losses in utility based off the queries being asked. When compared to the Utilitarian mechanism, which directly optimizes on overall error, the Waterfilling mechanism performs slightly worse while still satisfying all the desiderata.

ACKNOWLEDGMENTS

REFERENCES

[1] 2018. Our Facebook Partnership. https://socialscience.one/our-facebook-partnership
[2] 2020. Census Population Projections. https://wonder.cdc.gov/population.html
[3] 2020. COVID-19 Provisional Counts - Weekly Updates by Select Demographic and Geographic Characteristics. https://www.cdc.gov/nchs/nvss/vsrr/covid_weekly/index.htm
[4] Gergely Acs, Claude Castelluccia, and Rui Chen. 2012. Differentially private histogram publishing through lossy compression. *Proceedings - IEEE International Conference on Data Mining, ICDM* (2012), 1–10. https://doi.org/10.1109/ICDM.2012.80
[5] Adi Ben-israel and Thomas N. E. Greville. 2001. Generalized Inverses: Theory and Applications.
[6] Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. 2012. Unconditional differentially private mechanisms for linear queries. *Proceedings of the 44th symposium on Theory of Computing - STOC 12* (2012). https://doi.org/10.1145/2213977.2214089
[7] Shixi Chen and Shuigeng Zhou. 2013. Recursive Mechanism: Towards Node Differential Privacy and Unrestricted Joins. In *ACM SIGMOD*.
[8] Bolin Ding and Marianne Winslett. 2011. Differentially Private Data Cubes : Optimizing Noise Sources and Consistency. (2011).
[9] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS 03* (2003). https://doi.org/10.1145/773153.773173
[10] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*, Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener (Eds.). Springer Berlin Heidelberg.
[11] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* (2014).
[12] Rupert Freeman, Seyed Majid Zahedi, Vincent Conitzer, and Benjamin C. Lee. 2018. Dynamic Proportional Sharing: A Game-Theoretic Approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 1, Article 3 (2018), 36 pages. https://doi.org/10.1145/3179406
[13] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. 2011. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. In *Proceedings of the 8th USENIX Symposium on Networked System Design and Implementation (NSDI)* (Boston, MA). USENIX Association, 323–336. http://dl.acm.org/citation.cfm?id=1972457.1972490
[14] Samuel Haney, Ashwin Machanavajjhala, John M. Abowd, Matthew Graham, Mark Kutzbach, and Lars Vilhuber. 2017. Utility Cost of Formal Privacy for Releasing National Employer-Employee Statistics. In *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, Illinois, USA) *(SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1339–1354. https://doi.org/10.1145/3035918.3035940
[15] Moritz Hardt and Kunal Talwar. 2010. On the geometry of differential privacy. *Proceedings of the 42nd ACM symposium on Theory of computing - STOC 10* (2010). https://doi.org/10.1145/1806689.1806786
[16] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment* (2010).
[17] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. 2011. Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation* (Boston, MA) *(NSDI'11)*. USENIX Association, USA, 295–308.
[18] Marisa Hotchkiss and Jessica Phelan. 2017. Uses of Census Bureau Data in Federal Funds Distribution.
[19] Carlee Joe-Wong, Soumya Sen, Tian Lan, and Mung Chiang. 2013. Multiresource Allocation: Fairness-efficiency Tradeoffs in a Unifying Framework. *IEEE/ACM Transactions on Networking* 21, 6 (2013), 1785–1798. https://doi.org/10.1109/TNET.2012.2233213
[20] Noah M. Johnson, Joseph P. Near, and Dawn Xiaodong Song. 2017. Practical Differential Privacy for SQL Queries Using Elastic Sensitivity. *CoRR* abs/1706.09479

[21] Ios Kotsogiannis, Yuchao Tao, Xi He, Maryam Fanaeepour, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2019. PrivateSQL: A Differentially Private SQL Query Engine. *Proc. VLDB Endow.* 12, 11 (July 2019), 1371–1384. https://doi.org/10.14778/3342263.3342274
[22] Mayuresh Kunjir, Brandon Fain, Kamesh Munagala, and Shivnath Babu. 2017. ROBUS: Fair Cache Allocation for Data-parallel Workloads. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD)* (Chicago, Illinois, USA). ACM, 219–234. https://doi.org/10.1145/3035918.3064018
[23] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. 2014. A Data- and Workload-Aware Algorithm for Range Queries Under Differential Privacy. *PVLDB* 7, 5 (2014).
[24] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. 2010. Optimizing Linear Counting Queries Under Differential Privacy. In *Proceedings of the Twenty-ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Indianapolis, Indiana, USA) *(PODS '10)*. ACM, New York, NY, USA, 123–134. https://doi.org/10.1145/1807085.1807104
[25] Chao Li and Gerome Miklau. 2013. Optimal Error of Query Sets Under the Differentially-private Matrix Mechanism. In *Proceedings of the 16th International Conference on Database Theory (ICDT '13)*. ACM.
[26] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber. 2008. Privacy: Theory meets Practice on the Map. In *2008 IEEE 24th International Conference on Data Engineering*. 277–286. https://doi.org/10.1109/ICDE.2008.4497436
[27] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. 2018. Optimizing Error of High-dimensional Statistical Queries Under Differential Privacy. *PVLDB* 11, 10 (2018).
[28] Solomon Messing, Christina DeGregorio, Bennett Hillenbrand, Gary King, Saurav Mahanti, Zagreb Mukerjee, Chaya Nayak, Nate Persily, Bogdan State, and Arjun Wilkins. 2020. Facebook Privacy-Protected Full URLs Data Set. https://doi.org/10.7910/DVN/TDOAPG
[29] Arjun Narayan and Andreas Haeberlen. 2012. DJoin: Differentially Private Join Queries over Distributed Databases *(OSDI'12)*. USENIX Association, USA, 149–162.
[30] David C. Parkes, Ariel D. Procaccia, and Nisarg Shah. 2015. Beyond Dominant Resource Fairness: Extensions, Limitations, and Indivisibilities. *ACM Transactions Economics and Computation* 3, 1, Article 3 (2015), 22 pages. https://doi.org/10.1145/2739040
[31] Lucian Popa, Gautam Kumar, Mosharaf Chowdhury, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. 2012. FairCloud: Sharing the Network in Cloud Computing. In *Proceedings of the 2012 ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)* (Helsinki, Finland). ACM, 187–198. https://doi.org/10.1145/2342356.2342396
[32] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2013. Understanding Hierarchical Methods for Differentially Private Histograms. *Proc. VLDB Endow.* 6, 14 (Sept. 2013), 1954–1965. https://doi.org/10.14778/2556549.2556576
[33] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. 2020. DP-CGAN: Differentially Private Synthetic Data and Label Generation. arXiv:2001.09700 [cs.LG]
[34] Jaideep Vaidya, Basit Shafiq, Xiaoqian Jiang, and Lucila Ohno-Machado. 2013. Identifying inference attacks against healthcare data repositories. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science* (Mar 2013). https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3845790/
[35] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. 2013. Differentially private histogram publication. *The VLDB Journal* 22, 6 (apr 2013), 797–822. https://doi.org/10.1007/s00778-013-0309-y
[36] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2014. PrivBayes: private data release via bayesian networks. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu (Eds.). ACM, 1423–1434. https://doi.org/10.1145/2588555.2588573
[37] Xiaojian Zhang, Rui Chen, Jianliang Xu, Xiaofeng Meng, and Yingtao Xie. 2014. Towards Accurate Histogram Publication under Differential Privacy. *Proc. SIAM SDM Workshop on Data Mining for Medicine and Healthcare* (2014).

(2017). arXiv:1706.09479 http://arxiv.org/abs/1706.09479