Differentially Private Multi-Dimensional Analysis via Double-Sided Clipping and Prefix-Sum Cube

Yufei Wang[1], Xiang Cheng[1*]
[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China*
{wyfs4321,chengxiang}@bupt.edu.cn

ABSTRACT

Multi-dimensional analysis is a powerful tool for online analytical processing (OLAP). In this paper, we focus on the process of answering multi-dimensional analytical queries over data cubes, each of which consists of a collection of cuboids, while satisfying differential privacy (DP). The key technical challenges come from the high sensitivity of sum queries and noise aggregation: constructing a base cuboid requires the data curator to answer a workload of linear sum queries under DP in advance, whose sensitivity will result in a large amount of DP noise, and the DP noise will finally be aggregated when constructing the remaining cuboids. To this end, we present a Differentially **PRI**vate **M**ulti-dimensional **A**nalytic Approach (PRIMA). In PRIMA, we propose a Clipping and Debiasing Sum Query Processing Method (CLASP) which first bounds the sensitivity of sum queries by clipping the data table from both sides and then debiases the data table based on a symmetric sparse vector technology. Moreover, we propose a Hypothesis Testing based Prefix Sum Computing Method (SCOPE) to compute a base prefix-sum cuboid based on hypothesis testing. By employing the base prefix-sum cuboid, any remaining cuboid can be constructed with constant pieces of DP noise aggregated. We conduct experiments on both real-world and synthetic datasets. Experimental results confirm the effectiveness of PRIMA over existing approaches.

1 INTRODUCTION

Multi-dimensional analysis has become an essential element of online analytical processing (OLAP). By constructing a data cube that consists of a collection of cuboids in advance, the multi-dimensional analytical queries can be answered over massive amounts of data, which are stored in a data warehouse, quickly and efficiently. Specifically, the data curator first computes a base cuboid based on the data table via pre-querying. By aggregating the base cuboid in different dimensions, the data curator then constructs the remaining cuboids. Finally, any multi-dimensional analytical queries can be answered over the constructed data cube. However, since the data table usually contains private data, constructing data cubes based on such data may endanger the privacy of individuals. Therefore, the data curator must provide a rigorous privacy guarantee on how multi-dimensional analytical queries are processed.

Differential Privacy (DP) has been increasingly accepted as the state-of-the-art standard for protecting individual privacy in the central setting where a trusted data curator processes the sensitive data of all users, and has been studied extensively in both literature [9, 13, 14, 19, 20, 30, 35, 36] and industry [2, 10, 12, 21, 32]. DP guarantees that the presence or absence of any particular individual's record has a negligible impact on the likelihood that a particular result is returned to a query. Thus, an adversary cannot make meaningful inferences about any individual's record value, or even whether the record is present.

In this paper, we focus on the process of answering multi-dimensional analytical queries over data cubes under DP. Several existing differentially private query processing approaches [5, 11, 17, 18, 37] can be used to give access to the sensitive data with privacy guarantee. However, these approaches focus only on privately releasing count results. When considering releasing sum results over numerical attributes, which is essential for computing the base cuboid, existing approaches will lead to excessive DP noise due to the high sensitivity of the sum query. Although there are pieces of works [5, 18] trying to bound the sensitivity of the sum query by using a truncation operator. These works will inevitably introduce a systematic bias to the final answer. In addition, to limit the DP noise added to the cuboids when constructing the remaining cuboids, Ding et al. [4] propose a general Noise Control Framework, which formulates an optimization problem and minimizes the utility loss. However, this framework cannot cope well with the problem of noise aggregation, i.e., the problem that the DP noise in each cell of the base cuboid will be aggregated when constructing the remaining cuboids. As the domains of attributes increase, the numbers of noisy cells aggregated in the Noise Control Framework will grow in polynomial degrees, resulting in massive DP noise aggregated in the constructed data cube.

To this end, we propose a Differentially **PRI**vate **M**ulti-dimensional **A**nalytic Approach (PRIMA), which can not only support sum queries with both privacy and utility guarantees but also alleviate noise aggregation when constructing data cubes under DP. In PRIMA, we first put forward a **CL**ipping-and-Debi**A**sing **S**um Query **P**rocessing Method (CLASP) for answering sum queries under DP while alleviating the systematic bias. Different from the truncation-based methods that simply delete the records whose measures' values are greater than a threshold, CLASP deliberately chooses a clipping range $[\Phi-\tau, \Phi+\tau]$ and clips the data table such that any record whose measures' values are smaller than $\Phi-\tau$ or larger than $\Phi+\tau$ will be clipped to $\Phi-\tau$ or $\Phi+\tau$. In CLASP,

Proceedings of XXX XXXX XXXXXXXXX, Vol. XX, No. XXXX XXXX-XXXX.
doi:XX.XX/XXX.XX

The corresponding author of this paper is Prof. Xiang Cheng.

the data curator first bounds the sensitivity of sum queries by clipping the data table via a double-sided clipping mechanism. In this mechanism, the data curator first formulates optimization problems for choosing the optimal clipping axis Φ and radio τ, then clips the records whose measures' values are out of the clipping range $[\Phi - \tau, \Phi + \tau]$. If the lower threshold of the clipping range is less than 0, the double-sided clipping mechanism will degrade to a single-sided clipping mechanism and lead to a significant systematic bias. To reduce such systematic bias, the data curator clips the data table again based on a symmetric sparse vector technology, which is a bottom-up mechanism that traverses the data table and outputs a new lower threshold of clipping range τ' for debiasing the clipped data table. Moreover, for constructing the remaining cuboids accurately, the data curator alleviates noise aggregation by constructing the prefix-sum cubes [33], where any cuboids can be constructed through the computation between two pieces of prefix sums. To obtain the prefix sums without violating privacy, the prefix sums can be computed by answering a workload of linear sum queries under DP. However, the queries for computing the prefix sums require multiple accesses to any single record in the data table, which will introduce massive DP noise. Therefore, we design a Hypothe**S**is Testing based Prefix Sum **Co**m**P**uting M**E**thod (SCOPE). Compared with computing the prefix sums directly under DP, SCOPE avoids multiple accesses to the records and computes the prefix sums based on hypothesis testing. Specifically, in SCOPE, the data curator first acquires a noisy sum vector by accessing each record once under DP. The data curator then predicts the distribution of the noisy sum vector based on hypothesis testing. Next, the data curator computes the prefix sums based on both the noisy sum vector and its predicted distribution. Finally, the data curator boosts the accuracy of the prefix sums by enforcing consistency. Based on the prefix sums, the data curator can compute a prefix sum base cuboid and constructs the remaining cuboids without noise aggregation.

The main contributions of this work are summarized as follows:

- We focus on the process of answering multi-dimensional analytical queries over data cubes under DP and present PRIMA which can bound the sensitivity of sum query while minimizing the systematic bias and alleviate noise aggregation when constructing data cubes under DP.
- We put forward a **CL**ipping-and-Debi**A**sing **S**um Query **P**roc-essing Method (CLASP) which consists of a double-sided clipping mechanism and a symmetric sparse vector technology. By clipping the values of records via the double-sided clipping mechanism and reducing the potential systematic bias via the symmetric sparse vector technology, CLASP can answer sum queries under DP accurately.
- We design a Hypothe**S**is Testing based Prefix Sum **CO**m**P**uting M**E**thod (SCOPE). By computing prefix sums based on hypothesis testing and boosting the accuracy of prefix sums by enforcing consistency, SCOPE can improve the utility of the noisy prefix-sum cube and alleviate the problem of noisy aggregation.
- To evaluate the performance of PRIMA, we conduct extensive experiments on both real-world and synthesized datasets. The experimental results demonstrate that PRIMA outperforms the baseline approaches.

2 RELATED WORK

Since DP [8] was introduced, many works have been proposed for preserving individuals' privacy via DP during answering queries. A straightforward way for differentially private query processing is the interactive strategy. In this strategy, when a new query comes, the data curator computes the result directly and returns the noisy version to the analyst after adding DP noise. Adopting this strategy, McSherry et al. [23] firstly propose Privacy Integrated Queries (PINQ) to process counting queries under DP. For supporting general equijoins without scaling up the noise magnitude, Proserpio et al. [27] extend PINQ by assigning a weight to each row in the data table and put forward Weighted PINQ (wPINQ), which scales down the weights of rows in a join to ensure an overall sensitivity of one. However, both PINQ and wPINQ are not compatible with standard databases since these approaches introduce new operators that do not exist in standard SQL. Moreover, the high sensitivity of queries will result in poor utility. To process SQL queries over standard databases under DP, Zhang et al. [37] propose ϵKTELO which is a programming framework and system that aids programmers in developing differentially private programs. To bound the sensitivity of queries, Johnson et al. [17] propose a sound approximation of local sensitivity [7] called elastic sensitivity, which is extended from the idea of smooth sensitivity [25]. Based on the elastic sensitivity, the authors implement an end-to-end DP system for SQL queries based on elastic sensitivity called FLEX. Targeting the same problem, Ge et al. [11] present APEx, which allows data analysts to pose adaptively chosen sequences of queries along with required accuracy bounds.

However, the approaches mentioned above are all designed for processing linear counting queries over a single relation with different predicates, which have a relatively small sensitivity. For answering the queries with high sensitivity, a series of works [5, 18, 29] bound the sensitivity by introducing a truncation operator, which simply deletes the records whose attributes' values are more than a threshold τ before adding DP noise. Specifically, Kotsogiannis et al. [18] present PrivateSQL, an end-to-end differentially private relational database system that learns truncation threshold by applying the sparse vector technique [24]. Dong et al. [5] propose a truncation mechanism called R2T, which can adaptively choose a threshold that is provably close to the optimal one tuned for each particular query/dataset. Qiu et al. [29] present a mechanism for answering a set of sum queries. However, all these truncation-based works will introduce a significant systematic bias into the result.

Since the interactive-based approaches can only process a limited number of count queries under a fixed privacy budget. A series of approaches [4, 15, 22, 28, 34] based on the non-interactive strategy are proposed. In these approaches, the data curator first constructs a differentially private data model that contains certain statistical information about the data table via a set of queries. Any subsequent queries can be then processed over the differentially private data model without spending a privacy budget. In the non-interactive strategy, noise aggregation is a major problem that leads to poor utility. To answer queries with both privacy and utility guarantees,

Hay et al. [15] propose an approach that is able to deal with noise aggregation in single-dimensional range queries via hierarchical intervals. Qardaji et al. [28] optimize the hierarchical intervals by choosing a proper branching factor. Targeting the same problem, Xiao et al. [34] develop an approach called Privelet that provides accurate answers by using Haar wavelet. However, the data models in these approaches are limited to single-dimensional analysis. To tackle this problem, Ding et al. [4] present a general Noise Control Framework which carefully selects a subset of cells in a cuboid to inject Laplace noise and compute the rest cells from them. However, when the domains of attributes are large, the noise aggregation in Noise Control Framework will also cause a poor utility of query results. For answering multi-dimensional range queries under local differential privacy, Wang et al. [33] propose PRISM which alleviates noise aggregation by constructing prefix-sum cube. However, the range based randomized response proposed in PRISM is designed specifically for the local setting and cannot be adapted to the central setting.

3 PRELIMINARIES

3.1 Differential Privacy (DP)

DP requires the outputs of algorithms to be approximately the same even if any individual's record in the database is added or removed. Thus, the presence or absence of any individual's record has a statistically negligible effect on the outputs.

Formally, DP considers an instance data table T of relational schema $R(\mathbb{A}, M)$. The distance between two tables $T, T' \in D^n$ is $d(T, T') = |\{i | t_i \neq t'_i\}|$. Two tables T, T' are neighboring tables if $d(T, T') = 1$.

Definition 1 (Differential Privacy). *A private algorithm $\mathcal{A}$ preserves ϵ-differential privacy is for any pair of neighboring tables T, T', and for all sets O of possible outputs:*

$$Pr[\mathcal{A}(T) \in O] \leq e^{\epsilon} Pr[\mathcal{A}(T') =\in O],$$

where the probability is taken over the randomness of $\mathcal{A}$.

Dwork et al. [8] propose the Laplace mechanism to achieve DP. For a function whose outputs are real, they prove that DP can be achieved by adding noise drawn randomly from Laplace distribution. The Laplace distribution with magnitude γ follows probability density function $Pr[Lap(\gamma) = x] = \frac{1}{2\gamma} e^{-|x|/\gamma}$, where $\gamma = \frac{S}{\epsilon}$ is determined by the desired privacy budget ϵ and the sensitivity S of the function. In particular, the sensitivity S is used to measure the maximum change in the outputs of a function when any individual's record in the data table is changed.

One measure of sensitivity is the global sensitivity [8], which is the maximum difference in the query's result on any two neighboring tables. However, for many functions, the global sensitivity yields high noise, which leads to a poor utility of query results. To this end, a local measure of sensitivity called local sensitivity [7] is proposed, which is the maximum difference between the query's results on a given table and any neighboring table of it.

Definition 2 (Local Sensitivity). *For an algorithm $\mathcal{A}$ and an instance data table T of relational schema $R(\mathbb{A}, M)$, the local sensitivity of $\mathcal{A}$ at T is*

$$LS_{\mathcal{A}} = \max_{y:d(T,T')=1} \left\| \mathcal{A}(T) - \mathcal{A}(T') \right\|.$$

Local sensitivity is often much more practical and lower than global sensitivity since it is a property of the given table rather than the set of all possible tables. In this paper, we will adopt local sensitivity to control the magnitude of noise.

To support multiple differentially private computations, the sequential and parallel compositions of DP are extensively used.

Theorem 1 (Sequential Composition). *Given k random algorithms $\mathcal{A}_i (1 \leq i \leq k)$ that access the same record t, each of which satisfies ϵ_i-DP, then the combination of their outputs satisfies ϵ-LDP, where $\epsilon = \left(\sum_{i=1}^{k} \epsilon_i\right)$.*

Theorem 2 (Parallel Composition). *Given k random algorithms $\mathcal{A}_i (1 \leq i \leq k)$ that access different records $t_i (1 \leq i \leq n)$, each of which satisfies ϵ_i-DP, then the combination of their outputs satisfies ϵ-LDP, where $\epsilon = max\ (\epsilon_1, \cdots, \epsilon_k)$.*

3.2 Data Cube

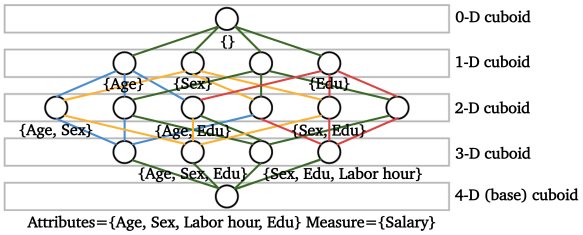


Figure 1: Data Cube

Data cube is a combination of cuboids. For a data table T which is an instance of relational schema $R(\mathbb{A}, M)$, a data cuboid can be viewed as the projection of T on a subset of attributes $\mathbb{B} \subseteq \mathbb{A}$, producing a multi-dimensional array. In the cuboid, each cell aggregates the measure M of records in T that match certain dimensions. Given a data cube among d attributes in T, there are $\binom{d}{\lambda}$ λ-dimensional ('λ-D') cuboids ($0 \leq \lambda \leq d$) in a data cube, representing all possible combinations of attributes in T. In these cuboids, the only d-D cuboid is called a base cuboid. Any other cuboids can be acquired by aggregating the base cuboid. In addition, for any two cuboids C_1 and C_2, if C_1 can be constructed from C_2 by directly aggregating one or more attributes in C_2, then C_1 is said to be an ancestor of C_2, and C_2 is a descendant of C_1. We show an example of constructing a data cube based on a table T in Figure 1.

Example 1. *Suppose there is a measure Salary and a subset of attributes $\mathbb{A} = \{a_1, a_2, a_3, a_4\}$, where a_1, a_2, a_3, and a_4 refer to age, gender, labor hours, and educational level, respectively. To analyze the relationship of workers' salary with their age, gender, labor hours, and educational level, the data curator can construct a data cube among these four attributes which aggregates the Salary. The data cube contains $\binom{4}{0} + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4} = 16$ cuboids, in which the 4-D cuboid $C_{a_1 a_2 a_3 a_4}$ is the base cuboid. In particular, for the 2-D cuboid $C_{a_1 a_2}$, the 3-D cuboid $C_{a_1 a_2 a_4}$ is the descendant of $C_{a_1 a_2}$ while $C_{a_2 a_3 a_4}$ is not, since $C_{a_1 a_2}$ cannot be constructed by aggregating $C_{a_2 a_3 a_4}$.*

3.3 Prefix-Sum Cube

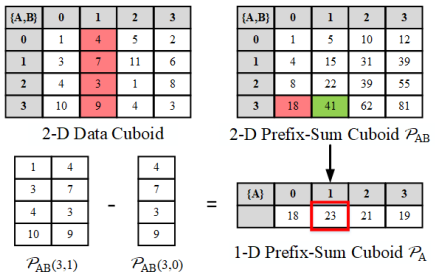


Figure 2: Computing a cell in an ancestor cuboid based on $\mathcal{P}$

Given λ attributes, a prefix-sum cuboid [16] is a λ-D array that has the same structure as the data cuboid among these attributes. The cells in the prefix-sum cuboid contain a kind of auxiliary information called prefix sum [16], which can be computed from the distributions among λ attributes. Similarly, based on the base prefix-sum cuboid, the remaining cuboids can be constructed through the computation between two cells in the base prefix-sum cuboid. Since each cell in the prefix-sum cuboid contains a piece of DP noise, which is introduced when computing the base prefix-sum cuboid under DP, there are only constant pieces of noise being aggregated, irrespective of the domain of the aggregated attributes.

Suppose there is a λ-D cuboid A of size $dom(A_1) \times dom(A_2) \times \cdots \times dom(A_\lambda)$. Prefix-sum cuboid $\mathcal{P}$ is another λ-d array which is used to store various precomputed prefix sums of A. For all $0 \le x_j < dom(A_j)$ and $j \in [0, \lambda - 1]$,

$$\mathcal{P}[x_1, x_2, \cdots, x_\lambda] = Sum(0 : x_1, 0 : x_2, \cdots, 0 : x_\lambda) = \sum_{i_1=0}^{x_1} \cdots \sum_{i_\lambda=0}^{x_\lambda} A[i_1, \cdots, i_\lambda]. \quad (1)$$

Theorem (3) provides how a cell in an ancestor cuboid can be computed from up to 2^λ appropriate cells in the prefix sum base cuboid $\mathcal{P}$. The left-hand side of Equation (2) specifies a cell to be computed. The right-hand side of Equation (2) consists of 2^λ additive terms, each of which is from a cell of $\mathcal{P}$ with a sign '+' or '-' defined by the product of all $s(i)$. For notational convenience, let $\mathcal{P}[x_1, x_2, \cdots, x_\lambda] = 0$ if $x_j = -1$ for some $j \in [0, \lambda - 1]$.

Theorem 3. *For all $j \in [0, \lambda - 1]$, let*

$$s(j) = \begin{cases} 1, & \text{if } x_j = h_j \\ -1, & \text{if } x_j = l_j - 1 \end{cases}.$$

Then, for all $j \in [0, \lambda - 1]$,

$$Sum(l_1 : h_1, \cdots, l_d : h_d) = \sum_{\forall x_j \in \{l_j - 1, h_j\}} \left\{ \left[\prod_{i=1}^{d} s(i) * \mathcal{P}[x_1, x_2, \cdots, x_d] \right] \right\}. \quad (2)$$

We show an example of how to compute a cell in an ancestor cuboid based on a prefix-sum cuboid in Example 2.

Example 2. *Figure 2 shows an example of a prefix-sum cuboid $\mathcal{P}$ with $\lambda = 2$, $dom(M_A) = dom(M_B) = 4$, and the geometrical explanation for the process of computing the value of a certain cell in the 1-D prefix-sum cuboid $\mathcal{P}_A$ based on the 2-D prefix-sum cuboid $\mathcal{P}_{AB}$. Suppose there is a 2-D prefix-sum cuboid $\mathcal{P}_{AB}$ that is computed from the 2-D data cuboid between A and B, a certain cell in the 1-D prefix-sum cuboid $\mathcal{P}_A(1) = Sum(0:1, 0:3) - Sum(0:0, 0:3) = \mathcal{P}_{AB}(3,1) - \mathcal{P}_{AB}(3,0)$.*

3.4 Multi-Dimensional Analysis

Given a table T which is an instance of relational schema $R(\mathbb{A}, M)$, where $\mathbb{A}$ is a set of d attributes and M is a measure, we focus on the following class of multi-dimensional analytical queries:

```
SELECT SUM(M) FROM T WHERE
A1 ∈ [l1, r1] AND A2 ∈ [l2, r2] AND ... AND Ad ∈ [ld, rd].
```

We denote the answer of a multi-dimensional analytical query q over data table T as $\mathcal{A}_q^T$.

3.5 Linear Sum Queries and Query Matrix

The base cuboid can be constructed by answering a set of linear sum queries. For a data table T which is an instance of relational schema $R(\mathbb{A}, M)$, we denote by $dom(\mathbb{A})$ the cross-product of the domains of attributes in $\mathbb{A}$. The data curator chooses a set of attributes $\mathbb{B} \subseteq \mathbb{A}$ relevant to the base cuboid. For example, if the data analyst is interested in a subset of two dimensional multi-dimensional analytic queries over attributes A_1 and A_2, they would set $\mathbb{B} = \{A_1, A_2\}$. The data curator then forms a sum vector $\mathbf{x}$ with one entry for each element of $dom(\mathbb{B})$. For simplicity, we assume $dom(\mathbb{B}) = \{1, 2, ..., n\}$ and for each $i \in dom(\mathbb{B})$, x_i is the sum of records' measures that satisfy the constraints that $|\mathbb{B}| = i$. We represent $\mathbf{x}$ as a column vector of sums: $\mathbf{x} = [x_1, x_2, ..., x_n]^t$. A linear query computes a linear combination of the counts in $\mathbf{x}$.

Definition 3 (Linear Sum Query). *A linear query is a length-n row vector $\boldsymbol{q} = [q_1, ..., q_n]$ with each $q_i \in \mathbb{R}$. The answer to a linear sum query $\boldsymbol{q}$ on $\boldsymbol{x}$ is the vector product $\boldsymbol{qx} = q_1 x_1 + ... + q_n x_n$.*

A series of linear sum queries can be organized into the rows of a pre-query matrix [19, 20] whose definition is given as follows:

Definition 4 (Pre-Query Matrix). *A pre-query matrix is a collection of m linear sum queries, arranged by rows to form an $m \times n$ matrix.*

If $\mathbf{Q}$ is an $m \times n$ pre-query matrix, the query answer for $\mathbf{Q}$ is a length m column vector of query results, which can be computed as the matrix product $\mathbf{Qx}$. The sensitivity of $\mathbf{Q}$ [19, 20] can be computed as follows.

Proposition 4 (Pre-Query Matrix Sensitivity). *The sensitivity of matrix $\boldsymbol{Q}$, denoted as $\Delta_{\boldsymbol{Q}}$, is*

$$\Delta_{\boldsymbol{Q}} = \max_{\|\boldsymbol{x} - \boldsymbol{x'}\|=1} \|\boldsymbol{Qx} - \boldsymbol{Qx'}\|_1 = \max_j \sum_{i=1}^{n} |q_{ij}|.$$

Thus the sensitivity of a pre-query matrix is the maximum L_1 norm of a column.

We show the pre-query matrices for computing a data cuboid and a prefix-sum cuboid in Example 3.

Example 3. *Figure 3 shows two query matrices. Given a subset of attributes $\mathbb{B} = A, B$, where $dom(M_A) = dom(M_B) = 4$, $dom(\mathbb{B}) = dom(M_A) \times dom(M_B) = 16$. $\boldsymbol{I}_{16}$ is the pre-query matrix for computing a 2-D data cuboid, and $\boldsymbol{P}_{16}$ is the pre-query matrix for computing a 2-D prefix-sum cuboid. Specifically, for a data table where the sensitivity of the sum query is LS_f, $\boldsymbol{I}_{16}$ is an identity matrix of size 16×16. This matrix consists of 16 queries and each row of $\boldsymbol{I}_{16}$ represents a linear sum query that asks for the value of a cell in the data cuboid. According to Proposition 4, the sensitivity of each linear sum query in $\boldsymbol{I}_{16}$ is LS_f. $\boldsymbol{P}_{16}$ is a lower triangular matrix of size 16×16, which consists of 16 queries, and each row of $\boldsymbol{P}_{16}$ represents a linear sum query that asks for the value of a cell in the prefix-sum cuboid. According to Proposition 4, the sensitivity of each linear sum query in $\boldsymbol{P}_{16}$ is $16 \cdot LS_f$.*

$$\begin{bmatrix} LS_f & 0 & \dots & 0 \\ 0 & LS_f & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & LS_f \end{bmatrix} \quad \begin{bmatrix} LS_f & 0 & \dots & 0 \\ LS_f & LS_f & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ LS_f & LS_f & \dots & LS_f \end{bmatrix}$$

(a) $\mathbf{I}_{16}$ (b) $\mathbf{P}_{16}$

Figure 3: Two Query Matrices

4 BASELINE APPROACHES

Although there is no work that can be directly used to answer multi-dimensional analytical queries over data cubes under DP, in this section, we provide three baseline approaches, two of which are combinations of related works.

4.1 BASE

For answering multi-dimensional analytical queries over data cubes under DP, one straightforward approach is to compute a base cuboid from the data table under DP and construct the other cuboids from the noisy base cuboid. We denote this approach as BASE.

Limitations. Obviously, BASE will introduce excessive DP noise when answering the linear sum queries and such noise will be aggregated when constructing the remaining cuboids, leading to poor utility of results.

4.2 SVTC

Sparse Vector Technology based Data Cube Constructing (SVTC) is a combination of PrivateSQL [18] and the Noise Control Framework [4]. The former is a differential private SQL query engine that bounds the sensitivity of multi-way join count queries via the Sparse Vector Technology (SVT). The latter is designed for constructing a data cube where an initial set $\mathcal{L}_{pre}$ of cuboids is computed from the data table and the remaining cuboids are aggregated from those in $\mathcal{L}_{pre}$. In SVTC, for a data table $T = \{t_i | 1 \le i \le n\}$, the data curator first truncates the data table T via SVT [24]. For choosing a proper truncation threshold τ, the data curator first assigns a parameter τ_s and a candidate set of truncation threshold $\mathcal{B} = \{b_i = i | 1 \le i \le dom(M)\}$ in advance. Then, the data curator traverses $\mathcal{B}$ in a bottom-up manner. For each b_i, the data curator truncates the data table and outputs a truncated data table T^{b_i} where any record t_i that satisfies $|M^{t_i}| > b_i$ is deleted. Based on T^{b_i}, the data curator answers a stream of queries $Q = \{q_i | 1 \le i \le dom(M)\}$ under DP, where

$$q_i : \texttt{SELECT SUM(M) FROM T}^{\texttt{b}_\texttt{i}} \texttt{ WHERE M = i},$$

and the noisy result is $\mathcal{A}_{q_i}^{T^{b_i}} + Lap(\frac{b_i}{\epsilon})$. During the process of traversing $\mathcal{B}$ in a bottom-up manner, for each b_i, the data curator can acquire a query result $\mathcal{A}_Q^{T^{b_i}} = \sum_{q_i \in Q} (\mathcal{A}_{q_i}^{T^{b_i}} + Lap(\frac{b_i}{\epsilon}))$. Since a larger b_i indicates less records in T are truncated, there is an increment in $\mathcal{A}_Q^{T^{b_i}}$ compared with $\mathcal{A}^{T_Q^{b_{i-1}}}$. For the first $\mathcal{A}_Q^{T^{b_i}}$ whose increment is less than the threshold τ_s, the data curator identifies the corresponding b_i as the optimal truncation threshold τ. After getting τ, the data curator traverses every record in the data table and deletes the records whose measures' values exceed τ. After truncating the data table via SVT, the data curator chooses an initial set $\mathcal{L}_{pre}$ of cuboids according to the Noise Control Framework and computes these cuboids by answering the linear sum queries in the pre-query matrix under DP. Based on $\mathcal{L}_{pre}$, the data curator constructs the remaining cuboids.

Limitations. SVTC attempts to improve the utility of the sum query results by truncating the data table. However, the truncation operator will introduce a systematic error into the result. Moreover, since SVT requires a manually assigned parameter as the metric for choosing the optimal truncation threshold τ, the effectiveness of SVTC is unstable. Furthermore, although the Noise Control Framework can bound the maximal noise over all cuboids, when the domain of attributes $dom(A)$ is large (e.g., $dom(A) > 10$), the utility loss caused by noise aggregation is still enormous.

4.3 R2TC

Race-to-the-Top based Data Cube Constructing (R2TC) is a combination of R2T [5] and the Noise Control Framework [4], where R2T is an instance-optimal truncation mechanism for adaptively choosing the optimal threshold τ. In R2TC, the data curator first chooses an initial set $\mathcal{L}_{pre}$ of cuboids according to the Noise Control Framework. The data curator then initials a candidate set $\mathcal{B} = \{b_i = 2^i | 1 \le i \le \log_2 dom(M)\}$ and a series of truncated data table $\{T^{b_i} | 1 \le i \le \log_2 dom(M)\}$. For a certain linear sum query q in the pre-query matrix, the data curator computes a list of noisy results

$$\mathcal{A}_q^{T^{B_i}} = \mathcal{A}_q^{T^{B_i}} + Lap(B_i, \epsilon') - \ln(\frac{\log(dom(M))}{\beta}) \cdot \frac{B_i}{\epsilon'}, \quad (3)$$

where $\epsilon' = \frac{\epsilon}{\log dom(M)}$ is the privacy budget allocated for q and β is an assigned probability parameter. The data curator chooses $\max(\mathcal{A}_q^{T^{B_i}})$ as the final result of q. After answering all linear sum queries in the pre-query matrix under DP, the data curator computes the initial set $\mathcal{L}_{pre}$ of cuboids and constructs remaining cuboids based on $\mathcal{L}_{pre}$.

Limitations. Although R2TC does not need to manually assign τ_s like SVTC, it still requires a probability parameter β given in advance, which has a direct impact on the utility. Moreover, according to Equation (3), for choosing the optimal threshold, R2TC must shift the result down by an amount that equals to the scale of the noise. As $dom(M)$ increases, it will make the effectiveness of

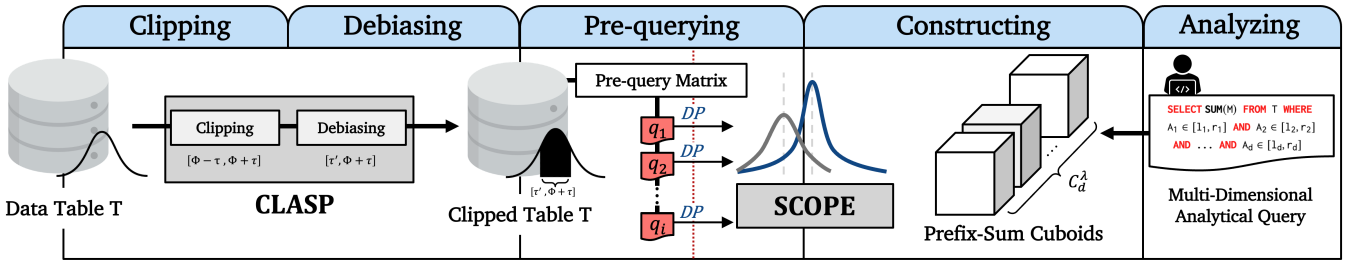


Figure 4: The overview of PRIMA

R2TC drop rapidly. Furthermore, since R2TC is based on the Noise Control Framework, when the domain of attributes is large, the effectiveness of R2TC is limited.

5 SOLUTION

In this section, we present our approach PRIMA for answering multi-dimensional analytic queries over data cubes under DP.

5.1 Overview

As shown in Section 4, the baseline approaches fail to reduce the systematic bias while bounding the high sensitivity of sum queries and alleviate the problem of noise aggregation. To this end, we present a Differentially **PRI**vate **M**ulti-dimensional **A**nalytic Approach (PRIMA). PRIMA contains two key building blocks: **CL**ipping-and-Debi**A**sing **S**um Query **P**rocessing Method (CLASP) and Hypothe**S**is Testing based Prefix Sum **CoP**uting M**E**thod (SCOPE). We show the overview of PRIMA in Figure 4 and introduce the main phases of PRIMA as follows:

1. Data Clipping Phase. Given a class of multi-dimensional analytical queries that associate with d attributes and a measure M. The data curator first determines the optimal clipping axis Φ and clipping radius τ via a double-clipping mechanism, which forms the clipping range $[\Phi-\tau, \Phi+\tau]$. Then, the data curator traverses the data table and clips the measure M of each record whose value is beyond the clipping range $[\Phi-\tau, \Phi+\tau]$. The details of the double-clipping mechanism are shown in Section 5.2.

2. Data Debiasing Phase. After clipping the data table, if the lower threshold of clipping range $\Phi-\tau < 0$, the data curator determines a new lower threshold of clipping range τ' for the clipped values of M via a symmetric sparse vector technology and clips the measure M of each record whose value is beyond the new clipping range $[\tau', \Phi+\tau]$. The details of the symmetric sparse vector technology are described in Section 5.3.

3. Pre-querying Phase. The data curator then organizes a workload of linear sum queries into a pre-query matrix to acquire the noisy sum vectors for constructing the base prefix-sum cuboid. Since the sum queries in the pre-query matrix access different records in the table, according to the parallel composition, a piece of privacy budget ϵ can be shared when answering these queries.

4. Cuboid Constructing Phase. Based on the noisy sum vectors, the data curator computes the base prefix-sum cuboid via SCOPE and constructs the remaining prefix-sum cuboids by aggregating the base prefix-sum cuboid in different dimensions. The details of SCOPE are given in Section 5.4.

Algorithm 1: Data Clipping

Input: The data table $T = \{t_i | 1 \le i \le n\}$, the clipping range $[B_l, B_u]$

Output: Clipped data table $T' = \{t'_i | 1 \le i \le n\}$.

```
for t_i ∈ T do
    if M_{t_i} < B_l then
        M_{t'_i} = B_l
    if M_{t_i} > B_u then
        M_{t'_i} = B_u
Return T'.
```

5. Query Processing Phase. Finally, based on the constructed data cube, the multi-dimensional analytical queries can be processed privately and efficiently.

In the following, we first introduce the double-sided clipping mechanism and the symmetric sparse vector technology in Section 5.2 and Section 5.3, respectively. We then give the details of SCOPE in Section 5.4. Finally, we show the privacy guarantee and utility analysis of PRIMA in Section 5.5.

5.2 Double-sided Clipping Mechanism

The high sensitivity of the sum query is the root cause of the poor result utility. A series of works [5, 18] bound the sensitivity of sum queries by using a truncation operator. Given records $\{t_i | 1 \le i \le n\}$ in a data table T, if each record t_i has a measure whose measure's value is $|M^{t_i}|$, the truncation operator deletes the record t_i that $|M^{t_i}| > \tau$, where τ is a carefully chosen truncation threshold. However, truncation introduces systematic bias to the data table, leading to the poor utility of the query results.

In CLASP, for answering sum queries under DP while alleviating the systematic bias, we put forward a double-sided clipping mechanism. Rather than deleting the records whose measures' values are larger than a specified threshold, the data curator deliberately chooses a clipping range $[\Phi-\tau, \Phi+\tau]$ and clips $|M^{t_i}|$ such that any $|M^{t_i}|$ beyond the clipping range only contributes $\Phi-\tau$ or $\Phi+\tau$ to the query result. In particular, by clipping the records whose measures' values larger than $\Phi+\tau$, the sensitivity of sum queries is bounded and less DP noise is added; by clipping the records whose measures' values smaller than $\Phi-\tau$, the systematic bias in query results is alleviated.

Algorithm 2: Symmetric Sparse Vector Technique

Input: The domain of the measure $dom(M)$, the data table $T = \{t_i | 1 \le i \le n\}$, a stream of difference queries $Q = \{q_i | 1 \le i \le n\}$.

Output: Clipping threshold B.

Initialize a candidate set $\mathcal{T} = \{1, 2, ..., dom(M)\}$;
for $\tau_i \in \mathcal{T}$ **do**
  The clipped data table T'=DataClipping($T, \tau_i, m + \tau$);
  if $\sum_{i=1}^{n} \mathcal{A}_{q_i}^{T} - \sum_{i=1}^{n} \mathcal{A}_{q_i}^{T'} < 0$ **then**
    Return τ_i.

The double-sided clipping mechanism takes a data table T which is an instance of relational schema $R(\mathbb{A}, M)$ as input and outputs a clipped data table T'. In the double-sided clipping mechanism, the data curator first computes the optimal clipping axis Φ by traversing the records in the data table T, then determines the optimal clipping radius τ by formulating an optimization problem. Φ and τ form a clipping range $[\Phi - \tau, \Phi + \tau]$. After determining the clipping range, the data curator clips the data table T by Algorithm 1. Specifically, in Algorithm 1, for the records whose measures' values are large than $\Phi+\tau$, the data curator clips them down to $\Phi+\tau$. For the records whose measures' values are less than $\Phi - \tau$, the data curator clips them up to $\Phi + \tau$. We show the details of how to determine the optimal Φ and τ as follows.

5.2.1 Clipping Axis Φ. Given the original records $\{t_i | 1 \le i \le n\}$ and the clipped records $\{t_i' | 1 \le i \le n\}$, in order to reduce the systematic bias in the clipped data table, we aim to find the optimal Φ that makes the mathematic expectation of the clipped records' measures converge to the mathematic expectation of the original records' measures. We show the optimal Φ in Equation (4).

Theorem 5. *The optimal clipping axis is*

$$\Phi = \frac{\sum_{i=1}^{n} |M^{t_i}|}{n}. \tag{4}$$

Due to space limitation, we put the proof of Theorem 5 in Appendix A.1.

5.2.2 Clipping Radius τ. To determine an optimal clipping radius τ, we consider two types of errors. Specifically, we denote the error caused by adding Laplace noise to the linear sum query results as the DP noise and denote the error caused by the deviation between the original values of the records and the clipped values of the records as the data distortion. A small τ indicates there are more values of records clipped, resulting in a large data distortion. While a large τ indicates the sum query has a large sensitivity, leading to a large DP noise. Therefore, we formalize an optimization problem. Before giving the details of the optimization problem, we first quantify the DP noise and the data distortion.

The magnitude of DP noise can be directly represented by the variance of the Laplace distribution:

$$2(\frac{\Delta f}{\epsilon})^2 = 2[min(\Phi + \tau, dom(M))/\epsilon]^2, \tag{5}$$

where $dom(M)$ is the domain of the measure M. As for the data distortion, we denote the data distortion as

$$\sum_{i=1}^{n} \frac{(|M^{t_i}| - |M^{t_i'}|)^2}{n}. \tag{6}$$

where t_i and t_i' refer to the original and clipped records, respectively.

In Equation (6), the value of $|M^{t_i'}|$ depends on the relationship between the domain of measure $dom(M)$ and the clipping range $[\Phi - \tau, \Phi + \tau]$. We first consider the case where the clipping range is included in the domain of measure: $0 < \Phi - \tau < \Phi + \tau < dom(M)$. For any original record t_i, the value of its measure $|M^{t_i}|$ belongs to either $[0, \Phi - \tau)$, $[\Phi - \tau, \Phi + \tau]$, or $(\Phi + \tau, dom(M)]$. After clipping the data table, the value of the clipped record' measure $|M^{t_i'}|$ is $\Phi - \tau$, $|M^{t_i}|$, or $\Phi + \tau$. Therefore, we can convert Equation (6) to

$$\sum_{i=1}^{\alpha} \frac{(\Phi - \tau - |M^{t_i}|)^2}{n} + \sum_{i=1}^{\beta} \frac{(\Phi + \tau - |M^{t_i}|)^2}{n}. \tag{7}$$

where α is the number of records that satisfy $|M^{t_i}| < \Phi - \tau$, and β is the number of records that satisfy $|M^{t_i}| > \Phi + \tau$. For the case where the clipping range and the domain of measure only partially overlap, we discuss in Section 5.3.

Based on Equation (5) and Equation (7), the objective function is shown as follows:

$$\begin{aligned} &\arg\min \sum_{i=1}^{\alpha} \frac{(\Phi - \tau - |M^{t_i}|)^2}{n} + \sum_{i=1}^{\beta} \frac{(\Phi + \tau - |M^{t_i}|)^2}{n} \\ &+ 2[\frac{min(\Phi + \tau, dom(M))}{\epsilon}]^2. \\ &s.t. \qquad \tau > 0 \end{aligned} \tag{8}$$

The problem is a convex optimization problem. By taking the partial derivative of Equation (8) with respect to τ, the optimal τ is shown in Theorem 6.

Theorem 6. *Suppose $B = \sum_{j=1}^{\beta} |M^{t_j}| - \sum_{i=1}^{\alpha} |M^{t_i}|$. The optimal clipping radius is*

$$\tau = \frac{\epsilon^2 B - [\epsilon^2(\beta - \alpha) + 2n]\Phi}{\epsilon^2(\alpha + \beta) + 2n}. \tag{9}$$

Due to space limitation, we put the proof of Theorem 6 in Appendix A.2.

5.3 Symmetric Sparse Vector Technology

Given a data table T where each record $t_i (1 \le i \le n)$ has a measure whose value is $|M^{t_i}|$, when the distribution of $|M^{t_i}|$ is dispersed and the mathematic expectation of $|M^{t_i}|$ is small, a large clipping radius τ is required to avoid significant data distortion. In this case, the optimal clipping radius τ determined by Theorem 6 may be larger than the clipping axis Φ, making the clipping range and the domain of measure only partially overlap. As a result, the double-sided clipping mechanism will degrade to single-sided clipping mechanism and lead to a significant systematic bias. To alleviate such systematic bias, we propose a Symmetric Sparse Vector Technique (SSVT). In SSVT, when $\Phi - \tau < 0$, the data curator will choose a new lower threshold τ' for the clipping range and clip the records that are beyond $[\tau', \Phi + \tau]$ once more.

Inspired by the Sparse Vector Technique (SVT) [24], SSVT is a bottom-up mechanism that outputs the first candidate value that

satisfies the constraint as the new lower threshold τ' for the clipping range. The details of SSVT are shown in Algorithm 2. For a data table T which is an instance of relational schema $R(\mathbb{A}, M)$. To compute τ', the data curator first initializes a candidate set $\mathcal{T} = \{1, 2, ..., dom(M)\}$, which contains all possible clipping lower bounds. The data curator traverses $\mathcal{T}$ in ascending order. For each candidate $\tau_i \in \mathcal{T}$, the data curator first clips the records in T with the clipping range $[\tau_i, \Phi + \tau]$ and gets a temporary clipped table T'. Based on T and T', the data curator answers a differential query stream $Q = \{q_i | 1 \leq i \leq n\}$ and compares the sum of results $\sum_{i=1}^{n} \mathcal{A}_{q_i}^{T} - \sum_{i=1}^{n} \mathcal{A}_{q_i}^{T'}$. Finally the data curator treats the first τ_i that makes $\sum_{i=1}^{n} \mathcal{A}_{q_i}^{T} - \sum_{i=1}^{n} \mathcal{A}_{q_i}^{T'} < 0$ as the new lower threshold of clipping range.

There are three differences between SVT and SSVT. Firstly, SSVT is customized to alleviate potential systematic bias caused by the double-sided clipping mechanism and cannot be replaced by SVT. Secondly, SSVT can maintain a stable performance since there is no manually assigned threshold required. Thirdly, since the output τ' of SSVT is obtained by tuning it on clipped datasets without accessing the original data table, there is no privacy leakage.

5.4 Hypothesis Testing based Prefix Sum Computing Method

After clipping the data table via CLASP, the data curator can acquire the noisy sum vector by answering the linear sum queries in the pre-query matrix. Based on the noisy sum vector, the data curator first computes the base cuboid and then constructs the remaining cuboids by aggregating the base cuboid in different dimensions. However, constructing the remaining cuboids based on the base cuboid directly will lead to noise aggregation, i.e., the problem that the DP noise in each cell of the base cuboid will be aggregated when constructing the remaining cuboids. As a baseline approach, the Noise Control Framework [4] alleviates noise aggregation by computing a subset of cuboids from the data table in a manner that reduces the overall noise and aggregates the remaining cuboids from the descendant cuboids in the subset. However, noise aggregation is still inevitable in this framework. Since the amount of aggregated DP noise is polynomial in the cardinality of $dom(M)$, a large $dom(M)$ will lead to a massive amount of DP noise aggregated in the constructed cuboids.

We consider alleviating noise aggregation by using the prefix-sum cube, which is a kind of structure where any ancestor cuboids can be constructed by the computation between pieces of prefix sums. However, the prefix sum cannot be applied to construct the base cuboid directly. As shown in Figure 3(b), computing prefix sums by answering a workload of linear sum queries under DP will result in multiple DP noise. Therefore, to take the advantage of the prefix sum without introducing multiple DP noise, we propose a Hypothe**S**is Testing based Prefix Sum **Co**m**P**uting M**E**thod (SCOPE). Rather than directly computing the prefix sums under DP, SCOPE first computes the prefix sums based on the distribution of the noisy sum vector, which is predicted by using hypothesis testing, and then boosts the accuracy of prefix sums by enforcing consistency.

We first describe how to compute prefix sums based on the noisy sum vector. For ease of illustration, we take a 1-D prefix-sum cuboid $\mathcal{P}$ of an attribute A as an example. Suppose the data curator

Algorithm 3: Converting

Input: The noisy sum vector $\mathbf{x}$.
Output: The synthetic sample set $\mathbf{x}'$.

```
Initialize a synthetic sample set x';
for i ∈ {1, 2, ..., dom(A)} do
    for j ∈ range(1, x_i) do
        Append i into x'
Return x'.
```

acquires a noisy sum vector $\mathbf{x} = [x_1, x_2, ..., x_{dom(A)}]$, where $x_j = \sum_{|A^{t_i}|=j, 1 \leq i \leq n} |M^{t_i}| (1 \leq j \leq dom(A))$, by answering a workload of linear sum queries in the pre-query matrix. The data curator first converts the noisy sum vector $\mathbf{x}$ to a synthetic sample set $\mathbf{x}'$. The details of the conversion procedure are shown in Algorithm 3. In Algorithm 3, the data curator takes the noisy sum vector $\mathbf{x}$ as input and initials a synthetic sample set $\mathbf{x}'$. For any element x_i in $\mathbf{x}$, the data curator appends i into $\mathbf{x}'$ for x_i times. For a random variable $\mathcal{X}$ which refers to the source of the unit increment in the noisy sum vector, $\mathbf{x}'$ can be considered as a sample set drawn from the sample space $\Omega = \{1, 2, ..., dom(A)\}$ following a probability distribution P.

To compute the prefix-sum cuboid $\mathcal{P}$, the data curator predicts the probability distribution P of the random variable $\mathcal{X}$ through hypothesis testing. There are various hypothesis testing methods that can be used, such as Komogorov-Smirnov test [3], Anderson-Darling test [1], and Pearson's chi-squared test [26]. Here we adopt the Komogorov-Smirnov test to predict the distribution P, which is one of the most useful and general hypothesis testing methods for quantifying the distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution. Suppose there exist a series of probability distributions in a hypothesis set $\mathbb{P} = P_1, P_2, ..., P_i, ...$ (e.g., normal distribution, Weibull distribution, exponential distribution, etc.). For any distribution P_i in $\mathbb{P}$, the data curator first computes the parameter of P_i based on $\mathbf{x}'$ and gets the corresponding cumulative distribution function $F_{exp}^{i}(x)$ of P_i. After getting $F_{exp}^{i}(x)$, the data curator then computes the empirical distribution function $F_{obv}(x)$:

$$F_{obv}^{i}(x) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}_{(-\infty, x]}(X_j), \tag{10}$$

where X_j is the element in $\mathbf{x}'$ and $\mathbb{1}_{(-\infty, x]}(X_j)$ is the indicator function:

$$\mathbb{1}_{(-\infty, x]}(X_i) = \begin{cases} 1 & \text{if } X_j \leq x \\ 0 & \text{if } X_j > x \end{cases}. \tag{11}$$

Given $F_{exp}^{i}(x)$ and $F_{obv}^{i}(x)$, the Kolmogorov–Smirnov statistic is

$$D_i = \max |F_{exp}^{i}(x) - F_{obv}^{i}(x)|. \tag{12}$$

and the p-value is

$$p_i = 2e^{-2(D_i)^2 \frac{n}{2}}. \tag{13}$$

For all $p_i (1 \leq i \leq |\mathbb{P}|)$, the data curator chooses the largest p-value $p_{max} = max(p_1, p_2, ...)$ and identifies the corresponding probability distribution of random variable $\mathcal{X}$ as P_{max}.

We find that according to the definition, prefix-sum cuboid $\mathcal{P}$ can be regarded as the integral of the sum vector $\mathbf{x}$. If we denote the discretized probability density function and the discretized cumulative distribution function of P_{max} as $f(i)$ and $F(i)$ respectively, we have

$$\mathcal{P}(i) = \frac{\mathbf{x_i}}{f(i)} F(i). \tag{14}$$

Based on the noisy sum vector $\mathbf{x}$ and the predicted distribution P_{max}, the prefix-sum cuboid $\mathcal{P}$ can be computed via Equation (14).

However, the process of computing the prefix sums will inevitably introduce errors: 1) the predicted probability distribution P_{max} may deviate from the actual distribution; 2) the sum vector $\mathbf{x}$ contains DP noise. These errors may make the noisy prefix sum $\mathcal{P}(i_2)$ smaller than $\mathcal{P}(i_1)$ ($0 \leq i_i < i_2 \leq dom|M|$), which leads to inconsistency among cells and violates the prior knowledge that the values in the prefix-sum cube are in ascending order. To improve the utility, the data curator post-processes the noisy prefix-sum cube to remove the inconsistency. Specifically, for any cell $\tilde{P}(i_i)$ in the noisy prefix-sum cube $\tilde{P}$, the data curator makes sure $0 \leq \mathcal{P}(i-1) \leq \mathcal{P}(i) \leq \mathcal{P}(i+1) \leq n$.

5.5 Privacy and Utility Analysis

In this section, we first show the privacy guarantee of PRIMA, then analyze the effectiveness of PRIMA and compare it with baseline approaches from two aspects: pre-query result error, noise aggregation. In addition, we analyze the prediction error in PRIMA, which is caused when predicting the distribution via hypothesis testing.

5.5.1 Privacy Guarantee. In PRIMA, the data table T will be accessed three times. The data curator first traverses the data table twice by using CLASP to determine the clipping threshold. Then the data curator answers the linear sum queries under DP to acquire the noisy sum vectors. In the first two times, all access occurs on the data curator's side, and no sensitive information will be released. When answering the linear sum queries in the pre-query matrix, according to the parallel composition, a piece of privacy budget ϵ is shared by all queries the pre-query matrix. Overall, according to the sequential composition, PRIMA satisfies ϵ-DP.

5.5.2 Pre-query Result Error. We first focus on the pre-query result error while answering the linear sum queries under DP. For the truncation-based approaches, i.e., SVTC and R2TC, the pre-query result error is affected by both the DP noise and the systematic bias. On the one hand, according to Equation (5), the DP noise added to the query result is proportional to the sensitivity. On the other hand, since the truncation operator will delete the records whose measure's values are larger than a threshold, answering the sum query under DP over the truncated data table will lead to and systematic bias. Both SVTC and R2TC face the dilemma that a large truncation threshold cannot alleviate the DP noise, while a small truncation threshold will result in massive systematic bias. In particular, the systematic bias of SVTC is hard to quantify since it is highly impacted by the manually assigned parameter for SVT. While the systematic bias of R2PC is proportional to the logarithm of $dom(M)$. Compared with these baseline approaches, PRIMA resolves the dilemma by clipping the values of records from both sides and minimizes the systematic bias.

5.5.3 Noise Aggregation. Noise aggregation refers to that the DP noise added into the pre-query results will aggregate into the cells when constructing the ancestor cuboids from the descendant cuboids. Both the number of attributes aggregated and the domains of the aggregated attributes will affect the magnitude of noise aggregation. For an attribute A with domain $dom(A)$, the DP noise in the cells of ancestor cuboids will be amplified by $dom(A)$ times. In addition, if there is more than one attribute aggregated, noise aggregation will occur more than once. In BASE, all ancestor cuboids are constructed from the base cuboid directly. The DP noise will be amplified by $dom(A)^{d-\lambda}$ times when constructing λ-D cuboids, which leads to a massive noise aggregation. SVTC and R2TC alleviate noise aggregation by using the Noise Control Framework. By carefully choosing an initial set $\mathcal{L}_{pre}$, the ancestor cuboids can be constructed from arbitrary cuboids in $\mathcal{L}_{pre}$ with a minimum number of attributes aggregated. As a result, in SVTC and R2TC, the DP noise will be amplified by $dom(A)^{\lambda'} \cdot 2|\mathcal{L}_{pre}|^2/\epsilon^2$ times, where λ' is typically set to 1 or 2. However, when $dom(A)$ is large, there is still a large amount of DP noise aggregated. By contrast, PRIMA constructs the cuboids from a base prefix-sum cuboid with an amplification factor $2^{d-\lambda}$, which is a constant regardless of $dom(A)$ and smaller than any baseline approaches and leads to a better utility.

5.5.4 Prediction Error. The prediction error in PRIMA refers to the distance between the true distribution of the noisy sum vector and the distribution that is predicted by hypothesis testing. A major factor that impacts the accuracy of the predicted distribution is the sample size. Larger samples contain more characteristics about the distribution, which can improve the accuracy of the predicted distribution. Since PRIMA is designed for OLAP in a data warehouse, which stores massive amounts of data, it is able to take large enough samples to reduce the prediction error. In addition, the prior knowledge possessed by the data curator, e.g., people's ages typically follow a normal distribution, can also boost the accuracy of the predicted distribution.

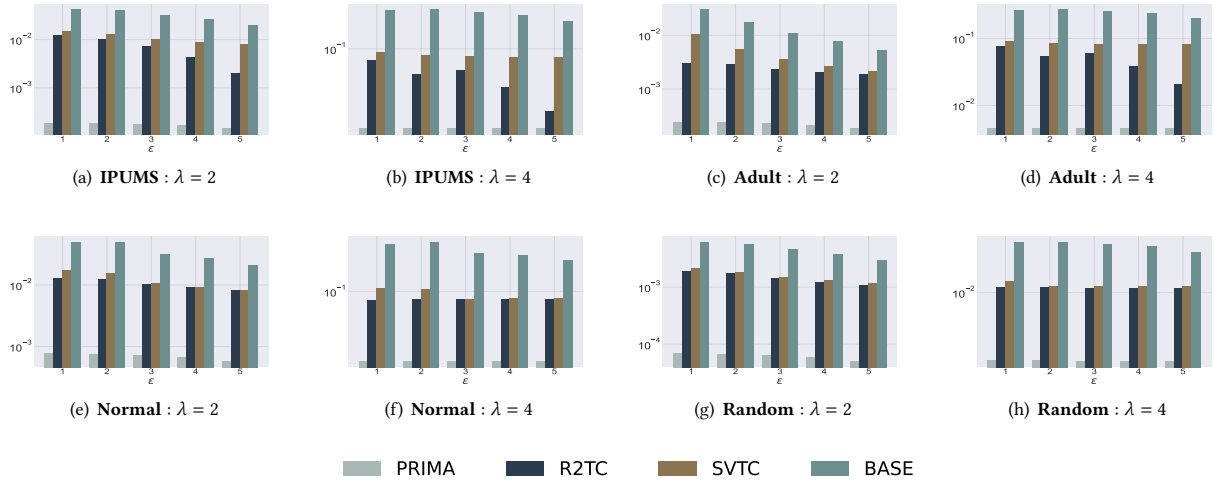
6 EXPERIMENTS

In this section, we first evaluate the performance of PRIMA by varying different parameters. Then we evaluate the advantages of PRIMA's key building blocks.

6.1 Experimental Settings

Datasets. We make use of two real datasets and two synthetic datasets in our experiments:

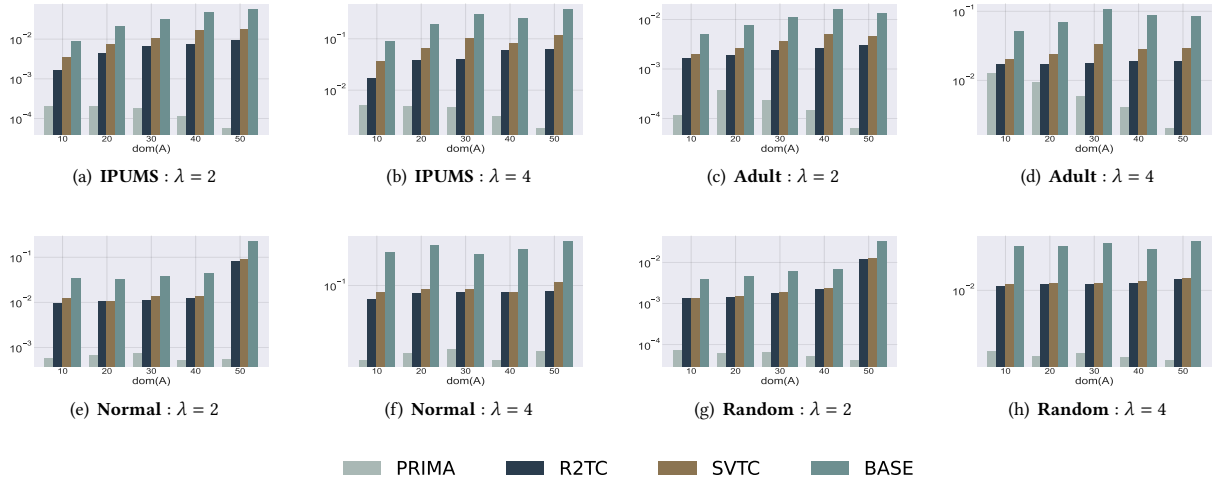
- *IPUMS [31]:* A US census dataset from the IPUMS repository, which contains around 3 million records.
- *Adult [6]:* A dataset from the UCI machine learning repository. After removing missing values, the dataset contains around 50 thousands records.
- *Normal:* A dataset which is synthesized from the multivariate normal distribution with mean 0, standard deviation 1, contains 50 ordinal attributes. The covariance between every two attributes is 0.5.
- *Random:* A dataset which is synthesized from the uniform distribution, contains 50 ordinal attributes.



(a) **IPUMS** : $\lambda = 2$ (b) **IPUMS** : $\lambda = 4$ (c) **Adult** : $\lambda = 2$ (d) **Adult** : $\lambda = 4$

(e) **Normal** : $\lambda = 2$ (f) **Normal** : $\lambda = 4$ (g) **Random** : $\lambda = 2$ (h) **Random** : $\lambda = 4$

Figure 5: Varying ϵ on all datasets under setting of $dom(A) = 30, dom(M) = 30, n = 10^5, d = 4$. MAEs are shown in log scale.



(a) **IPUMS** : $\lambda = 2$ (b) **IPUMS** : $\lambda = 4$ (c) **Adult** : $\lambda = 2$ (d) **Adult** : $\lambda = 4$

(e) **Normal** : $\lambda = 2$ (f) **Normal** : $\lambda = 4$ (g) **Random** : $\lambda = 2$ (h) **Random** : $\lambda = 4$

Figure 6: Varying $dom(A)$ on all datasets under setting of $\epsilon = 2, dom(M) = 30, n = 10^5, d = 4$. MAEs are shown in log scale.

To experiment with different sizes of data table, we generate multiple test datasets from the four datasets with the number of records ranging from 10k to 1M. In addition, for evaluating different numbers of attributes and domain sizes, we generate multiple versions of these four datasets with the domain sizes of attributes ranging from 10 to 50.

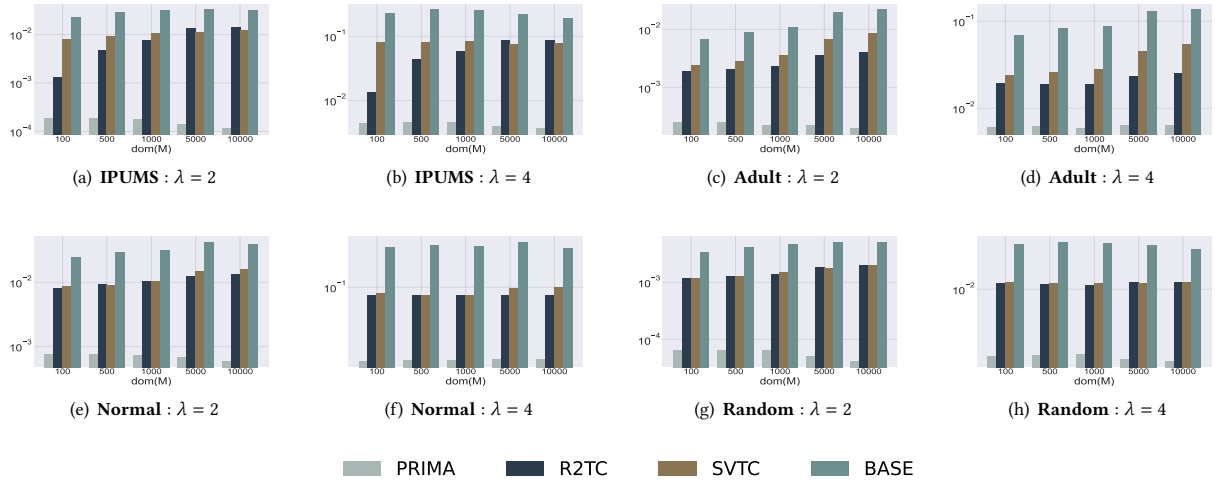
Competitors. To evaluate the overall effectiveness of PRIMA, we compare PRIMA against the baseline approaches including Base, SVTC and R2TC.

In particular, in SVTC and R2TC, we compute an optimal initial set $\mathcal{L}_{pre}$ which contains a 4-D cuboid, two 3-D cuboids, two 2-D cuboids and a 1-D cuboids according to the guideline in [4].

Utility Metric. We use the Mean Absolute Error to measure the accuracy of constructed cuboids: $MAE = AVG\left(\left|\frac{\tilde{c}-c}{n}\right|\right)$, where n is the number of cells in a certain cuboids, $\tilde{c}$ and c are the noisy and true values stored in the cells, respectively.

Methodology. For each dataset and each approach, we construct data cube between any λ attributes and measure the MAE of each cell in the m-D cuboids. We repeated all experiments 10 times to obtain stable results. In all experiments, unless explicitly stated, we use the following default values for other relevant parameters: $\epsilon = 2.0, dom(A) = 30, dom(M) = 30, n = 10^5, d = 4, \lambda = \{2, 4\}$.

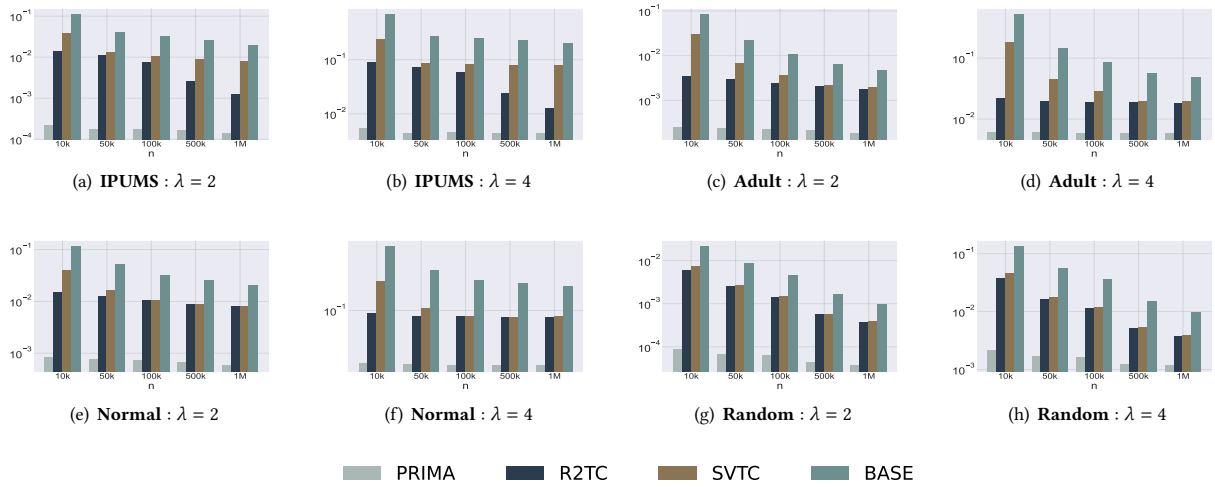
Environment. We implement all the approaches in Python. All experiments are conducted on an Intel Core i5 2.50GHz PC with 8GB RAM.



(a) **IPUMS** : $\lambda = 2$ (b) **IPUMS** : $\lambda = 4$ (c) **Adult** : $\lambda = 2$ (d) **Adult** : $\lambda = 4$

(e) **Normal** : $\lambda = 2$ (f) **Normal** : $\lambda = 4$ (g) **Random** : $\lambda = 2$ (h) **Random** : $\lambda = 4$

Figure 7: Varying $dom(M)$ on all datasets under setting of $\epsilon = 2, dom(A) = 30, n = 10^5, d = 4$. MAEs are shown in log scale.



(a) **IPUMS** : $\lambda = 2$ (b) **IPUMS** : $\lambda = 4$ (c) **Adult** : $\lambda = 2$ (d) **Adult** : $\lambda = 4$

(e) **Normal** : $\lambda = 2$ (f) **Normal** : $\lambda = 4$ (g) **Random** : $\lambda = 2$ (h) **Random** : $\lambda = 4$

Figure 8: Varying n on all datasets under setting of $\epsilon = 2, dom(A) = 30, dom(M) = 30, d = 4$. MAEs are shown in log scale.

6.2 Performance of PRIMA

In this part, we evaluate the performance of different approaches under various parameter settings. In general, these parameters including the privacy budget ϵ, the domains of attributes $dom(A)$, the domain of measure $dom(M)$ and the total number of records n.

Varying privacy budget ϵ. Figure 5 shows the results varying ϵ from 1.0 to 5.0. It is obvious that PRIMA has a clear advantage over other approaches. We also find that, compared with other approaches, the MAEs of PRIMA are robust to the growth of ϵ. The reason can be explained as follows. Since PRIMA reduces the DP noise and the systematic bias via CLASP, these two types of errors have only a minor impact on PRIMA. As a result, the effectiveness of PRIMA is not sensitive to the variations in the privacy budget ϵ.

Varying the domains of attributes $dom(A)$. Figure 6 presents the results varying $dom(A)$ from 10 to 50 on all datasets. We observe that PRIMA consistently outperforms all other approaches. We can find that as $dom(A)$ increases, the MAEs of PRIMA are stable while the MAEs of the baseline approaches increase. The reason can be explained as follows. A larger $dom(A)$ indicates there are more cells in each cuboid. For the baseline approaches, as the number of cells in each cuboid increases, more noise aggregation will take place, resulting in higher MAEs. While PRIMA alleviates noise aggregation by constructing prefix-sum cubes. As a result, the effectiveness of PRIMA is not sensitive to the increase of $dom(A)$.

Varying the domain of attributes $dom(M)$. Figure 7 studies the impact of $dom(M)$. It is obvious that PRIMA performs the best among all approaches. Moreover, we observe that as $dom(M)$

increases, the MAEs of PRIMA are stable while the MAEs of the baseline approaches increase. The reason can be explained as follows. A large $dom(M)$ leads to a higher sensitivity of the sum query. In BASE, a high sensitivity makes the magnitude of DP noise grow. In SVTC and R2TC, massive systematic bias is introduced when truncating the records. While PRIMA can guarantee a good query result utility by clipping the records from double sides.

Varying the number of records n. Figure 8 shows the results varying n from 10K to 1M on all datasets. We can observe that the MAEs of all approaches decrease with the increase in the number of records, and PRIMA always performs the best.

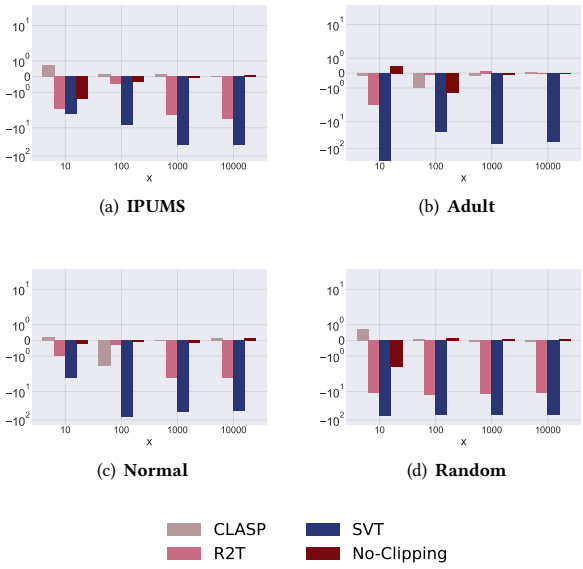


Figure 9: MBs of x queries on all datasets.

6.3 Performance of Key Building Blocks

We first evaluate the advantage of CLASP by comparing CLASP with SVT, R2T and No-Clipping, which is a straightforward method where the data curator answers sum queries under DP without rewriting the data table. For each dataset and each method, we randomly choose x sum queries and measure the Mean Bias $MB = \frac{1}{x}\sum_{i=1}^{x} \tilde{c} - c$. In all experiments, we use the following default values: $\epsilon = 2.0, dom(A) = 30, dom(M) = 30, n = 10^5, d = 4, \lambda = 4$.

The MBs of four methods are shown in Figure 9. For the four methods, the MBs of CLASP and No-Clipping are close to zero as the number of queries x increases. While the MBs of SVT and R2T are negative in all cases. It indicates that SVTC and R2TC will result in a greater systematic bias.

We evaluate the advantages of SCOPE by comparing SCOPE with NCF, which refers to the Noise Control Framework, and CDB, which refers to the method for constructing a data cube in BASE. In this experiment, to construct the m-D cuboids via each method, we first obtain the required noisy sum vectors by answering linear sum queries in the pre-query matrix based on the original data table. We

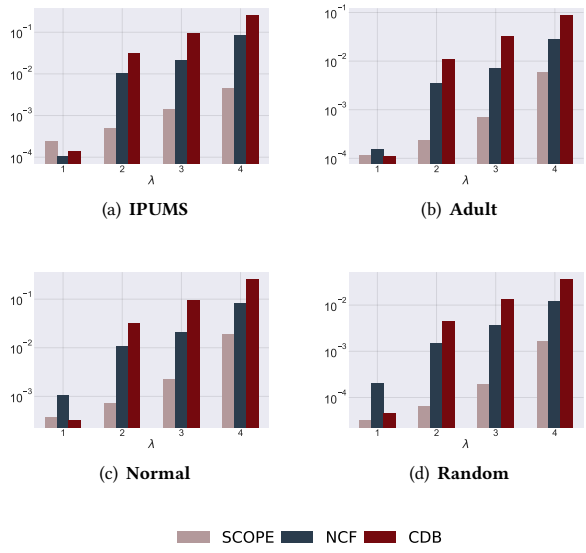


Figure 10: MAEs of λ-D cuboids on all datasets.

evaluate the effectiveness of these methods by measuring the MAE of m-D cuboids. In all experiments, we use the following default values: $\epsilon = 2.0, dom(A) = 30, dom(M) = 30, n = 10^5, d = 4$.

According to the results shown in Figure 10, we find SCOPE outperforms the other methods in all cases except for $m = 1$. The reason can be explained as follows. Since SCOPE alleviates the problem of noise aggregation by constructing prefix-sum cubes, the ancestor prefix-sum cuboids can be constructed without excessive DP noise aggregation. As m increases, the DP noise in the constructed base cuboid generally aggregates in the ancestor cuboids that are constructed via NCF and CDB, while SCOPE can always maintain much lower MAEs. We also observe that when $m = 1$, the MAEs of CDB are smaller than those of SCOPE and NCF. This is because, in SCOPE, the base prefix-sum cuboid contains prediction error; in NCF, there is more than one cuboid in the initial set $\mathcal{L}_{pre}$ required to be computed, leading to a higher sensitivity when answering linear sum queries.

7 CONCLUSION

In this paper, we propose PRIMA for processing multi-dimensional analytical queries over data cubes under DP. To answer sum queries accurately while alleviating systematic bias, we put forward CLASP, which contains a double-sided clipping mechanism for clipping the data table and a symmetric sparse vector technology for debiasing the clipped data table. In addition, to improve the utility of noisy prefix-sum cube and alleviate the problem of noisy aggregation, we design SCOPE, where the distribution of noisy sum vector is predicted via hypothesis testing and the prefix-sum cube is constructed based on the predicted distribution. Sufficient experiments are conducted to illustrate the effectiveness of PRIMA.

REFERENCES

[1] T. W. Anderson and D. A. Darling. 1952. Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics* 23, 2 (1952), 193–212.

[2] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. 2017. Prochlo: Strong Privacy for Analytics in the Crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 441–459.

[3] W.W. Daniel. 1990. *Applied Nonparametric Statistics*. PWS-KENT Pub.

[4] Bolin Ding, Marianne Winslett, Jiawei Han, and Zhenhui Li. 2011. Differentially private data cubes: optimizing noise sources and consistency. In *International Conference on Management of Data*. ACM, 217–228.

[5] Wei Dong, Juanru Fang, Ke Yi, Yuchao Tao, and Ashwin Machanavajjhala. 2022. R2T: Instance-optimal Truncation for Differentially Private Query Evaluation with Foreign Keys. In *SIGMOD '22: International Conference on Management of Data*. ACM, 759–772.

[6] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[7] Cynthia Dwork and Jing Lei. 2009. Differential privacy and robust statistics. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*. ACM, 371–380.

[8] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography Conference*.

[9] Alexander Edmonds, Aleksandar Nikolov, and Jonathan R. Ullman. 2020. The power of factorization mechanisms in local and central differential privacy. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy (Eds.). ACM, 425–438.

[10] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Conference on Computer and Communications Security*. ACM, 1054–1067.

[11] Chang Ge, Xi He, Ihab F. Ilyas, and Ashwin Machanavajjhala. 2019. APEx: Accuracy-Aware Differentially Private Data Exploration. In *Proceedings of the 2019 International Conference on Management of Data*. ACM, 177–194.

[12] 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*. ACM, 1339–1354.

[13] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A Simple and Practical Algorithm for Differentially Private Data Release. In *26th Annual Conference on Neural Information Processing Systems 2012*, Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (Eds.). 2348–2356.

[14] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the Accuracy of Differentially Private Histograms Through Consistency. *Proc. VLDB Endow.* 3, 1 (2010), 1021–1032.

[15] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. 2010. Boosting the Accuracy of Differentially Private Histograms Through Consistency. *Proceedings of the VLDB Endowment* 3, 1 (2010), 1021–1032.

[16] Ching-Tien Ho, Rakesh Agrawal, Nimrod Megiddo, and Ramakrishnan Srikant. 1997. Range Queries in OLAP Data Cubes. In *International Conference on Management of Data*. ACM, 73–88.

[17] Noah M. Johnson, Joseph P. Near, and Dawn Song. 2018. Towards Practical Differential Privacy for SQL Queries. *Proc. VLDB Endow.* 11, 5 (2018), 526–539.

[18] 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 (2019), 1371–1384.

[19] 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, PODS 2010*, Jan Paredaens and Dirk Van Gucht (Eds.). ACM, 123–134.

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

[21] Ashwin Machanavajjhala, Daniel Kifer, John M. Abowd, Johannes Gehrke, and Lars Vilhuber. 2008. Privacy: Theory meets Practice on the Map. In *Proceedings of the 24th International Conference on Data Engineering*. IEEE Computer Society, 277–286.

[22] Ryan McKenna, Gerome Miklau, Michael Hay, and Ashwin Machanavajjhala. 2018. Optimizing error of high-dimensional statistical queries under differential privacy. *Proceedings of the VLDB Endowment* 11, 10 (2018), 1206–1219.

[23] Frank McSherry. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *International Conference on Management of Data*. ACM, 19–30.

[24] Joseph P. Near and Chiké Abuah. 2021. *Programming Differential Privacy*. Vol. 1. https://uvm-plaid.github.io/programming-dp/

[25] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. 2007. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*. ACM, 75–84.

[26] Karl Pearson. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.

[27] Davide Proserpio, Sharon Goldberg, and Frank McSherry. 2014. Calibrating Data to Sensitivity in Private Data Analysis. *Proc. VLDB Endow.* 7, 8 (2014), 637–648.

[28] Wahbeh H. Qardaji, Weining Yang, and Ninghui Li. 2013. Understanding Hierarchical Methods for Differentially Private Histograms. *Proc. VLDB Endow.* 6, 14 (2013), 1954–1965.

[29] Yuan Qiu, Wei Dong, Ke Yi, Bin Wu, and Feifei Li. 2022. Releasing Private Data for Numerical Queries. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 1410–1419.

[30] Vibhor Rastogi and Suman Nath. 2010. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010*, Ahmed K. Elmagarmid and Divyakant Agrawal (Eds.). ACM, 735–746.

[31] Sobek, Matthew, Ruggles, and Steven. 1999. The IPUMS Project. *Historical Methods* 32, 3 (1999), 102–102.

[32] Apple Differential Privacy Team. 2017. Local privacy and statistical minimax rates.

[33] Yufei Wang. 2022. PRISM. https://github.com/wyfs4321/PRISM.

[34] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2010. Differential privacy via wavelet transforms. In *International Conference on Data Engineering*. IEEE, 225–236.

[35] Ganzhao Yuan, Yin Yang, Zhenjie Zhang, and Zhifeng Hao. 2016. Convex Optimization for Linear Query Processing under Approximate Differential Privacy. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 2005–2014.

[36] Ganzhao Yuan, Zhenjie Zhang, Marianne Winslett, Xiaokui Xiao, Yin Yang, and Zhifeng Hao. 2015. Optimizing Batch Linear Queries under Exact and Approximate Differential Privacy. *ACM Trans. Database Syst.* 40, 2 (2015), 11:1–11:47.

[37] Dan Zhang, Ryan McKenna, Ios Kotsogiannis, George Bissias, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. 2020. eKTELO: A Framework for Defining Differentially Private Computations. *ACM Trans. Database Syst.* 45, 1 (2020), 2:1–2:44.

A SUPPLEMENTARY ANALYSIS

A.1 Proof of Theorem 5

Proof. The systematic bias E_b between the original records $\{t_i|1 \le i \le n\}$ and the clipped records $\{t_i'|1 \le i \le n\}$ is

$$E_b = \frac{\sum_{i=1}^{n} |M^{t_i}| - \sum_{i=1}^{n} |M^{t_i'}|}{n}. \tag{15}$$

Suppose there are α records that satisfy $|M^{t_i}| < \Phi - \tau (1 \le i \le \alpha)$, β records that satisfy $|M^{t_j}| > \Phi + \tau (1 \le j \le \beta)$, γ records that satisfy $\Phi - \tau \le |M^{t_k}| \le \Phi + \tau (1 \le k \le \gamma)$, Equation (15) can be converted to

$$\begin{aligned} E_b &= \frac{\sum_{i=1}^{\alpha} |M^{t_i}| - (\Phi - \tau)}{n} + \frac{\sum_{k=1}^{\gamma} |M^{t_k}| - |M^{t_k}|}{n} \\ &+ \frac{\sum_{j=1}^{\beta} |M^{t_j}| - (\Phi + \tau)}{n} \\ &= \frac{\sum_{i=1}^{n} |M^{t_i}|}{n} - \frac{n-\gamma}{n}\Phi + \frac{\alpha - \beta}{n}\tau - \frac{\sum_{i=1}^{\gamma} |M^{t_i}|}{n}. \end{aligned} \tag{16}$$

As shown in Equation (16), the systematic bias is affected by both Φ and τ. For determining the optimal Φ, we ignore the effect of τ by considering the situation where τ approaches 0. Then, we have

$$\lim_{\tau \to +0} E_b = \frac{\sum_{i=1}^{n} |M^{t_i}|}{n} - \Phi. \tag{17}$$

To minimize the systematic bias, we can claim the optimal clipping axis is

$$\Phi = \frac{\sum_{i=1}^{n} |M^{t_i}|}{n}. \tag{18}$$

A.2 Proof of Theorem 6

Proof. We use $Err(\tau)$ to represent the objective function in Equation (8). We can convert $Err(\tau)$ to

$$\begin{aligned} Err(\tau) &= \sum_{i=1}^{\alpha} \frac{[(\Phi - \tau) - |M^{t_i}|]^2}{n} + \sum_{j=1}^{\beta} \frac{[(\Phi + \tau) - |M^{t_j}|]^2}{n} \\ &+ 2[\frac{\Phi + \tau}{\epsilon}]^2 \\ &= \frac{\alpha + \beta}{n}\tau^2 + \frac{2(\beta - \alpha)\Phi + 2(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n}\tau \\ &+ \frac{\alpha\Phi^2 + \beta\Phi^2 - 2\Phi(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n} \\ &+ 2[\frac{\Phi + \tau}{\epsilon}]^2. \end{aligned} \tag{19}$$

The problem is a convex optimization problem. By taking the partial derivative of Equation (19) with respect to τ, the derivative of $Err(\tau)$ is

$$\begin{aligned} Err(\tau)' &= 2\frac{\alpha + \beta}{n}\tau + \frac{2(\beta - \alpha)\Phi + 2(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n} \\ &+ 4[\frac{\Phi + \tau}{\epsilon}]. \end{aligned} \tag{20}$$

Let $Err(\tau)' = 0$, we have

$$\begin{aligned} &Err(\tau)' = 0 \Rightarrow \\ &2\frac{\alpha + \beta}{n}\tau + \frac{2(\beta - \alpha)\Phi + 2(\sum_{i=1}^{\alpha} |M^{t_i}| + \sum_{j=1}^{\beta} |M^{t_i}|)}{n} \\ &+ 4[\frac{\Phi + \tau}{\epsilon}] = 0 \Rightarrow \\ &\tau = \frac{\epsilon^2 B - [\epsilon^2(\beta - \alpha) + 2n]\Phi}{\epsilon^2(\alpha + \beta) + 2n} \end{aligned} \tag{21}$$