Differentially-Private Publication of Origin-Destination Matrices with Intermediate Stops

Sina Shaham
University of Southern California
Los Angeles, California, USA
sshaham@usc.edu

Gabriel Ghinita
University of Massachusetts
Boston, Massachusetts, USA
gghinita@cs.umb.edu

Cyrus Shahabi
University of Southern California
Los Angeles, California, USA
shahabi@usc.edu

ABSTRACT

Conventional origin-destination (OD) matrices record the count of trips between pairs of start and end locations, and have been extensively used in transportation, traffic planning, etc. More recently, due to use case scenarios such as COVID-19 pandemic spread modeling, it is increasingly important to also record intermediate points along an individual's path, rather than only the trip start and end points. This can be achieved by using a multi-dimensional *frequency matrix* over a data space partitioning at the desired level of granularity. However, serious privacy constraints occur when releasing OD matrix data, and especially when adding multiple intermediate points, which makes individual trajectories more distinguishable to an attacker. To address this threat, we propose a technique for privacy-preserving publication of multi-dimensional OD matrices that achieves differential privacy (DP), the de-facto standard in private data release. We propose a family of approaches that factor in important data properties such as data density and homogeneity in order to build OD matrices that provide provable protection guarantees while preserving query accuracy. Extensive experiments on real and synthetic datasets show that the proposed approaches clearly outperform existing state-of-the-art.

1 INTRODUCTION

Origin-destination (OD) matrices have been extensively used to characterize the demand for transportation between pairs of start and end trip points. Using OD matrices, one can provision appropriate capacity for a transportation infrastructure, by determining what is the demand (or trip *frequency*) for each source-destination pair. However, novel applications require more level of detail, for which conventional OD matrices are insufficient, due to the fact that they have a 2D structure, and intermediate points along a trajectory cannot be captured. Consider, for instance, the study of COVID-19 spread patterns in the ongoing pandemic, where an analyst needs to determine not only the end points of a trajectory, but also the intermediate points that a certain individual has visited, and where possible exposure to the virus occurred. In this case, it is necessary to record several distinct points across a trajectory, which leads to an increase in the dimensionality of OD matrices. We denote such enhanced data structures as OD matrices *with intermediate stops*.

While such detailed OD matrices capture additional information, they also pose a more serious privacy threat for the individuals included in the data, since the finer level of granularity of trajectory representation allows an adversary to pinpoint a user with better accuracy. For instance, there may be a large number of users that travel between a suburban neighborhood and the city center. However, when intermediate stops are also included, e.g., a specific type of store that sells ethnic products, a gym specializing on a certain type of yoga, and a fertility clinic, there are far fewer individuals who follow such a path (and sometimes, perhaps just one individual), which may lead to serious privacy breaches related to that individual's gender, race and lifestyle details. It is thus essential to protect the privacy of individuals whose trajectories are aggregated to build detailed OD matrices, and *differential privacy (DP)* [7] is the model of choice to achieve an appropriate level of protection.

Specifically, DP bounds the ability of an adversary such that s/he cannot determine with significant probability whether the trajectory data of a target individual is present in the released OD matrix or not. The OD matrix with intermediate stop points is equivalent to a *multi-dimensional frequency matrix*, in which an element represents the number of individuals who took a trip that includes that specific sequence of start, intermediate and end points. According to DP, carefully calibrated noise is added to each count to bound the identification probability of any single individual.

Several approaches tackled the problem of protecting frequency matrices for location data, but they do have serious limitations. For instance, solutions for DP-compliant location data histograms [4, 15, 20, 21] build data-independent structures that do not adapt well to data density, and assume a fixed dimensionality of the indexing structure, typically 2D only. As we show in Section 6, they do not handle well skewed datasets, which are the most typical ones in the case of geospatial data. Another category of approaches attempts to capture trajectories using prefix trees or n-grams [1, 2], but those approaches transform cells in the data domain into a sequence of abstract string labels, and lose the proximity semantics that are so important when querying location-based data.

We propose a novel technique for sanitization of OD matrices with intermediate stops such that location proximity semantics are preserved, and at the same time the characteristics of the data are carefully factored in to boost query accuracy. Specifically, we build custom data structures that tune important characteristics like index fan-out and split points to account for data properties. This way, we are able to achieve superior accuracy while at the same time enforcing the strong protection guarantees of DP.

Our specific contributions are:

- We identify important properties of indexing data structures that have a high impact on query accuracy when representing location frequency matrices;
- We design customized approaches that are guided by intrinsic data properties and automatically tune structure parameters, such as fan-out, split points and index height;
- We perform a detailed analysis of the obtained data structures that allows us to allocate differential privacy budget in a manner that is conducive to preserving as much data accuracy as possible under a given privacy constraint;

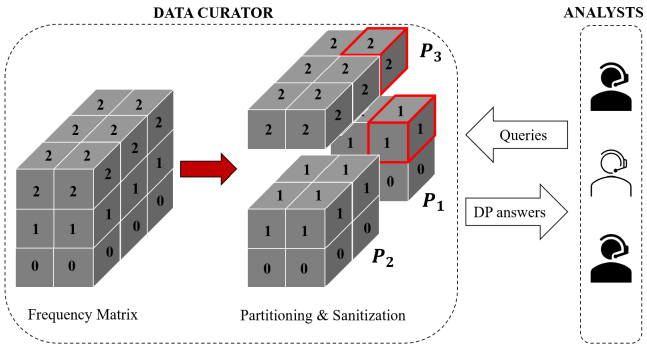


Figure 1: System model for private frequency matrices.

- We perform an extensive experimental evaluation on both real and synthetic datasets which shows that our proposed techniques adapt well to data characteristics and outperform existing state-of-the-art in terms of query accuracy.

Section 2 introduces necessary background concepts and definitions. Section 3 outlines data-independent techniques, followed by data-adaptive approaches in Section 4. Section 5 surveys related work. We present experimental evaluation results in Section 6, followed by conclusions in Section 7.

2 BACKGROUND AND DEFINITIONS

We assume the two-party system model shown in Fig. 1: a trusted data curator/owner collects the frequency matrix directly from individuals and sanitizes the data. Untrusted data analysts are interested in querying the private frequency matrix.

Let $F_1 \times F_2 \times ... \times F_d$ be a d-dimensional array representing a frequency matrix F. Each entry $f_i \in F$ is a number denoting a *frequency* or *count*. For example, a two-dimensional frequency matrix can model a map with each entry indicating the number of individuals located in a particular area. The frequency matrix corresponds to a d-dimensional finite space hyper-rectangle, or d-orthotope. According to the differential privacy model, a protection mechanism adds to each matrix element noise from a carefully selected random distribution to prevent an adversary from learning with significant probability whether a particular individual's data was used or not when creating the matrix.

2.1 Differential Privacy

Differential privacy (DP) [7] is a popular privacy model which provides strong protection guarantees. It presents an aggregate query interface (i.e., count queries) and ensures that the presence or absence of an individual in the data does not significantly change the results of a query. Consider two frequency matrices F and F' that differ in a single record t, i.e., $F' = F \bigcup \{t\}$ or $F' = F \setminus \{t\}$. F and F' are commonly referred to as *neighboring* or *sibling* datasets.

Definition 1 (ϵ-Differential Privacy)**.** A randomized mechanism $\mathcal{A}$ provides ϵ-DP if for any pair of neighboring frequency matrices F and F', and any output value $S \in Range(\mathcal{A})$,

$$\frac{Pr(F = S)}{Pr(F' = S)} \leq e^{\epsilon} \tag{1}$$

Parameter ϵ is the *privacy budget*: lower values result in stricter privacy, but also require addition of noise with larger magnitude, decreasing query accuracy. The *sequential composability* property of DP states that running in succession multiple mechanisms that each satisfy DP with privacy budgets $\epsilon_1, \epsilon_2, ..., \epsilon_n$ respectively, is equivalent to the execution of a single mechanism with $\epsilon = \sum_{i=1}^{n} \epsilon_i$.

An essential concept of DP is the *sensitivity* of queries, which measures the maximal difference that can be achieved by the addition or removal of a single individual's record in the database.

Definition 2 (L_1-Sensitivity)**.** Given two sibling datasets F, F' and a set of real-valued functions $\mathcal{G} = \{g_1, g_2, ..., g_m\}$, the L_1-sensitivity of F is measured as

$$s = \max_{\forall F, F'} \sum_{i=1}^{m} |g_i(F) - g_i(F')|$$

The Laplace mechanism is a widely used technique to achieve ϵ-DP. It adds to the output of a query function g noise drawn from a Laplace distribution with scale b, where b depends on two factors: sensitivity and privacy budget.

$$Pr(x|b) = \frac{1}{2b} e^{|x|/b} \text{ where } b = \frac{s}{\epsilon} \tag{2}$$

In the rest of the paper, we denote Laplace noise by $Lap(\frac{s}{\epsilon})$. In the case of query functions that are modeled through a partitioning of the dataspace (e.g., a set of non-overlapping histogram bins), sensitivity is equal to 1, since a record can fall in exactly one partition.

Table 1: Summary of notations.

Symbol	Description
F	Frequency matrix
F_i	Dimension cardinality
N	Total count of F
$\overline{N}$	Sanitized total count of F
m	Partitioning constant
s	Sensitivity
ϵ_{tot}	Total privacy budget
ϵ_{prt}	Partitioning budget
ϵ_{data}	Data perturbation budget
$H(F)$	Entropy of F
$Lap(s/\epsilon)$	Laplace noise with sensitivity s and budget ϵ

2.2 Problem Statement

Starting with an input frequency matrix, we create a set of non-overlapping partitions of the matrix and then publish a set of noisy counts for each of these partitions, according to the Laplace mechanism. The *sanitized*, DP-compliant frequency matrix consists of the *boundaries* of all partitions and their *noisy counts*. Since partitions are non-overlapping, we keep sensitivity low (i.e., 1). We refer to each input cell in the original frequency matrix as an *entry*, hence a *partition* is a group of matrix entries. Analysts (i.e., users of the sanitized matrix) ask multi-dimensional *range queries*.

Definition 3. *(Range Query)* A range query on the frequency matrix F is a d-orthotope with dimensions denoted as $d_1 \times d_2 \times ... \times d_n$, where d_i represents a continuous interval in dimension i.

For example, consider the $3 \times 2 \times 3$ frequency matrix shown in Fig. 1. The generation of partitions is referred to as *partitioning* and the addition of noise to total sums is referred to as *sanitization*. The example shows a sample partitioning of the matrix

generating three partitions P_1, P_2 and P_3 with total counts of 2, 4 and 12, respectively. In a simplified setup, the sanitization follows by adding Laplace noise to the partitions' total count and answering queries based on the resulting private frequency matrix. Moreover, a *uniformity assumption* [4] is made within each partition to answer queries with varying shapes and sizes. For example, if the sanitized counts are $2+n_1$, $4+n_2$, and $12+n_3$, where n_i denotes Laplace noise added for sanitization, and an analyst asks a query including two cells whose borders are shown in bold red color, the answer is $\frac{12+n_3}{6} + \frac{2+n_1}{4}$.

Suppose that the total count of a partition entailing q entries is p, and its noisy count is denoted by $\overline{p}$. One can see that there are two sources of error while answering a query. The first type of error is referred to as *noise error*, which is due to Laplace noise added to the partition counts. The second source of error is referred to as *uniformity error*. The uniformity error arises as the assumption of uniformity is made within each partition so that the noisy count of a cell in the partition can be calculated as $\overline{p}/q$.

To evaluate accuracy of query results, we use the *mean relative error (MRE)*. For a query q with the true count p and noisy count $\overline{p}$, MRE is calculated as

$$MRE(q) = \frac{|p - \overline{p}|}{p} \times 100 \tag{3}$$

Problem 1. *Given a frequency matrix F, generate an ϵ-differentially private version of F such that the expected value of relative error (MRE) is minimized.*

In the design of methods for the publication of private frequency matrices, we make extensive use of *entropy* to understand the amount of information contained in the frequency matrix and the effect that partitioning has on information loss.

Definition 4. *(Entropy)* Given a frequency matrix F and a set of partitions $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ with the total counts $p_1, p_2, ..., p_n$, the entropy of F is defined as:

$$H(F|\mathcal{P}) = -\sum_{i=1}^{n} \frac{p_i}{\sum_{j=1}^{n} p_j} \log_2 \frac{p_i}{\sum_{j=1}^{n} p_j} \tag{4}$$

Table 1 summarizes notations used throughout the paper.

2.3 Trajectory Modeling with OD Matrices

Conventional OD matrices allow analysts to determine how many individuals traveled between pairs of locations, e.g., between the central business district (CBD) and a suburb. Increasing availability of mobile data and their use in complex planning problems makes it important to expand the expressiveness of OD matrices, by allowing one to include intermediate stops, which essentially amounts to supporting queries on trajectories. Furthermore, conventional OD matrices tend to use abstract representations of locations, where the spatial information may be lost, e.g., by tabulating counts of individuals traveling between pairs of zipcodes. Proximity of zipcodes may be lost in the process, and if one wishes to change the representation granularity, or perform range-based queries (e.g., find how many users traveled from a $1km$ circle centered at point A to a $1km$ circle centered at B), such functionality is not possible.

Our proposed multi-dimensional histograms produce a hierarchical partitioning of the data domain that preserves locality and proximity information. It allows flexible queries, and captures intermediate points along a trajectory, as shown in Figure 2. Assume a trajectory representation where one wishes to capture

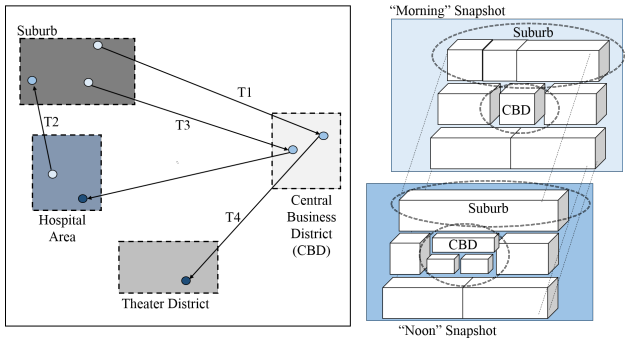


Figure 2: Capturing trajectory data using OD matrices.

daily activities across several time frames, e.g., *morning->noon->evening*. Trajectory T_1 corresponds to a person who lives in a suburb, works in CBD and goes to see a play in the evening. This can be captured using a multi-dimensional histogram where the first pair of spatial coordinates corresponds to the morning location (suburb), followed by another pair in the CBD, and finally the evening in the theater district. Each of the time frames can be partitioned independently, resulting in the structure on the right half of Figure 2 (due to space constraints, we do not represent the evening time frame). Each trajectory corresponds to a single entry in this multi-dimensional matrix, according to each location at each time frame.

An important advantage of this representation is that the specific partitioning used for a particular dimension is customized to the data corresponding to that time frame. For instance, the same part of the space can be present in different frames, but with different granularities. In this example, the CBD area has low granularity for the morning time frame, since few people live there, but high granularity in the noon frame. Similarly, a theater district will not present interest in queries for the first or second time frame, but will likely be of high interest in the evening frame. Conventional OD matrices cannot accommodate such scenarios.

3 DATA-INDEPENDENT APPROACHES

In this section, we introduce two data-independent approaches for the sanitization of frequency matrices with arbitrary dimensionality. These are extensions of existing work, particularly the technique in [15]. In Section 4 we will introduce more advanced data-dependent techniques that account for data distribution.

3.1 Extended Uniform Grid (EUG)

We extend the work in [15], originally proposed for two-dimensional frequency matrices. We refer to that algorithm as Uniform Grid (UG). The main idea of UG was to sanitize the total count of the frequency matrix and substitute it in a formula that results in a constant value m that represents the granularity of dividing each dimension of a 2D frequency matrix. After partitioning, the count in each of the partitions is sanitized using the Laplace mechanism.

While the approach in [15] only works for two-dimensional data, EUG provides a detailed analytical model that finds the optimal m value for uniform partitioning in any number of dimensions. EUG is formally presented in Algorithm 1. Suppose that the frequency matrix F has d dimensions represented by a $F_1 \times F_2 \times ... \times F_d$ array, and let N denote the total count of

F. The objective is to find a value of m such that, by updating the granularity of F to m^d and applying the Laplace mechanism, the utility of the published private histogram is maximized. The algorithm starts by utilizing a small amount of budget denoted as ϵ_0 to obtain a noisy count of the total number of entries in the frequency matrix.

$$\overline{N} = N + Lap(s/\epsilon_0), \tag{5}$$

where $\overline{N}$ denotes the sanitized count. The sanitized count is used for the estimation of m by formulating an optimization problem.

The value of m can be estimated by considering the existing error sources, i.e., noise error and non-uniformity error. The former is used for sanitization of counts, and the latter is due to the assumption that data in each partition are uniform. Consider a query that selects r portions of F, calculated by dividing its covered entries over the total number of entries. Hence, the query entails rm^d entries of F. On the one hand, given that the noise added to each partition has a variance of $2/\epsilon^2$, the total additive noise variance sums up to $\frac{2rm^d}{\epsilon^2}$, or equivalently standard deviation of $\frac{\sqrt{2r}m^{d/2}}{\epsilon}$.

On the other hand, the query can be seen as a d-orthotope where the side length is proportional to $\sqrt[d]{r}$. Thus, each side of the orthotope spans $\sqrt[d]{r} \times m$ cells, and the number of points located inside the query is on average $\sqrt[d]{r} \times m \times \frac{\overline{N}}{m^d}$. The term $\overline{N}/m^d$ comes from the assumption of data uniformity in F. By further assuming that the non-uniformity error on average is some portion of the total density of the cells on the query border, we have the non-uniformity error as $\sqrt[d]{r} \times \frac{\overline{N}}{c_0 m^{d-1}}$ for some constant c_0. Therefore, the aim is to find the optimal value of m that minimizes the summation of two errors, i.e.,

$$\min_m \frac{\sqrt{2r}m^{d/2}}{\epsilon} + \sqrt[d]{r} \times \frac{\overline{N}}{c_0 m^{d-1}} \tag{6}$$

Solving based on stationary conditions of the above convex problem results in the optimal m given by:

$$\frac{d\sqrt{2r}m^{d-1}}{\epsilon} - (d-1)\sqrt[d]{r} \times \frac{\overline{N}}{c_0 m^{2d}} = 0 \tag{7}$$

$$\rightarrow m = \left(\frac{2(d-1)}{d} \times r^{(1/d-1/2)} \times \frac{\overline{N}\epsilon}{\sqrt{2}c_0}\right)^{2/(3d-2)} \tag{8}$$

The base case of the problem occurs when the frequency matrix has two dimensions and results in the same equation proposed by [15]:

$$m = \sqrt{\frac{\overline{N}\epsilon}{\sqrt{2}c_0}} \tag{9}$$

For higher dimensions, if query size is known in advance, Equation (8) can be used with the given r to estimate the value of m; otherwise, by assuming that all query sizes are equally likely, integration over r leads to Equation (13). For derivation, let us define an auxiliary variable α as

$$\alpha = \left(\frac{2(d-1)}{d} \times \frac{\overline{N}\epsilon}{\sqrt{2}c_0}\right)^{2/(3d-2)} \tag{10}$$

Algorithm 1 Extended Uniform Grid (EUG)

Input: F, ϵ_{tot}, ϵ_0, s;
$\overline{N} \leftarrow SUM(F) + Lap(s/\epsilon_0)$
$\epsilon_{\text{tot}} \leftarrow \epsilon_{\text{tot}} - \epsilon_0$
$d \leftarrow$ Number of dims in F
$m \leftarrow \left(\frac{2(d-1)}{d} \times r^{(1/d-1/2)} \times \frac{\overline{N}\epsilon}{\sqrt{2}c_0}\right)^{2/(3d-2)}$
// UPDATE GRANULARITY
Divide each dimension by m
for each new partition i **do**
    $N' \leftarrow SUM(i)$
    $\overline{N}' \leftarrow N' + Lap(s/\epsilon_{\text{tot}})$
    for each entry j in i **do**
        $j \leftarrow \overline{N}'/|i|$
return F

Integration over r leads to

$$\int_0^1 \alpha \times r^{\frac{2-d}{d(3d-2)}}\, dr = \frac{\alpha}{\frac{2-d}{d(3d-2)}+1} r^{\frac{2-d}{d(3d-2)}+1}\Bigg|_0^1 \tag{11}$$

$$= \alpha \times \left(\frac{d(3d-2)}{3d^2-3d+2}\right), \tag{12}$$

and ultimately, results in:

$$m = \left(\frac{2(d-1)}{d} \times \frac{\overline{N}\epsilon}{\sqrt{2}c_0}\right)^{2/(3d-2)} \times \left(\frac{d(3d-2)}{3d^2-3d+2}\right). \tag{13}$$

Once the value of m is calculated, each dimension of matrix F is divided into m equal intervals generating m^d partitions. The entries in each partition are set to the partition's sanitized total count divided by the number of entries it contains. The sanitized total count of a partition is generated by adding its entries and using Laplace mechanism with the privacy budget of $\epsilon_{\text{tot}} - \epsilon_0$.

3.2 Entropy-based Partitioning (EBP)

A critical point in the EUG algorithm is how to determine the value of m. We propose Entropy-based Partitioning (EBP), a method for estimating a good value of m based on the concept of entropy. In addition to providing better accuracy, EBP also addresses the issue with EUG's arbitrary choice of constant c_0 which is empirically set to $10/\sqrt{2}$. EBP proposes a more informed parameter selection process that does not require arbitrary value settings.

Consider a d-dimensional frequency matrix F with dimensions $F_1 \times F_2 \times \ldots \times F_d$, and let N represent the total count of F. Moreover, denote the privacy budget allocated for the calculation of m by ϵ. As in the case of Algorithm 1, the objective is to find a value of m that, by updating the granularity of F to m^d, and applying the Laplace mechanism, the utility of the published private histogram is maximized. We look at the problem from an information theory perspective. Once the granularity of F is updated, the variance of total Laplace noise used to sanitize partitions adds up to $\frac{2m^d}{\epsilon^2}$, leading to total standard deviation of $\frac{\sqrt{2}m^{d/2}}{\epsilon}$. The entropy of the noise imposed on the frequency matrix is therefore,

$$H\left(\frac{\sqrt{2}m^{d/2}}{\epsilon}\right) = -\log_2 \frac{\epsilon}{\sqrt{2}m^{d/2}}. \tag{14}$$

On the other hand, consider the amount of information loss that occurs due to the change in granularity. To calculate the information loss, the amount of information before and after changing the granularity F is required. The information contained in F before change of granularity can be calculated as $H(F)$, denoting the entropy of F. After partitioning is conducted, the entropy is reduced to $H(F|m)$, denoting entropy calculated based on the updated frequency matrix with the granularity of m^d. Thus, the amount of information loss incurred due to change in granularity is:

$$\text{Information Loss} = H(F) - H(F|m). \tag{15}$$

An optimization problem can be formulated to find the optimal value of m that minimizes the average query error.

$$\min_m H(\frac{\sqrt{2}m^{d/2}}{\epsilon}) + H(F) - H(F|m). \tag{16}$$

By increasing the value of m, information loss becomes smaller, but the induced noise grows larger. The optimal value of m is reached when the noise is equal to information loss. Unfortunately, entropy cannot be directly calculated due to privacy concerns; however, an approximation can be employed as follows. We assume that the number of entries is in the order of the number of data points, and data points are uniformly distributed over the m^d partitions. Entropy before/after changing granularity can be approximated as

$$H(F) \approx -\log_2(1/N),\ H(F|M) \approx -\log_2(1/m^d) \tag{17}$$

To preserve the privacy of users, the value of N is sanitized beforehand based on the Laplace mechanism. The value of m minimizing the optimization problem is derived as

$$-\log_2 \frac{\epsilon}{\sqrt{2}m^{d/2}} = -\log_2(1/N) + \log_2(1/m^d) \tag{18}$$

$$\rightarrow \log_2 \frac{\epsilon}{\sqrt{2}m^{d/2}} = \log_2(m^d/N) \rightarrow m = \sqrt[3d/2]{\frac{N\epsilon}{\sqrt{2}}} \tag{19}$$

The derived formula in Equation (19) is an alternative method to calculate the value of m in the EUG algorithm. Therefore, the pseudocode in Algorithm 1 applies to EBP by replacing the formula in line 4 with Equation (19).

4 DATA-DEPENDENT APPROACHES

4.1 Overview

Data-independent algorithms overlook critical information about the distribution of data points, as they always assume uniform distribution. This is particularly problematic for higher dimensional frequency matrices, due to their tendency to be sparse.

To improve accuracy when publishing higher dimensional frequency matrices, we propose a tree-based approach called Density-Aware Framework (DAF) that takes into account density variation across different regions of the space. In addition, DAF introduces a key feature that enables custom stop conditions for partitioning. Intuitively, denser parts of the space should be split in more granular fashion, while for sparse areas the partitioning can stop earlier, since most likely large regions of the space are empty. The decision of when to stop partitioning the frequency matrix is made privately, and avoids over-partitioning which can lead to large errors in higher dimensional frequency matrices.

DAF is a hierarchical partitioning approach that resembles a tree index. Each node covers a portion of the frequency matrix, with the root node covering all entries. Descendants of a node are generated by a non-overlapping split of the parent node's entries.

Algorithm 2 DAF-Entropy

1: **Global Constants:** ϵ_{tot}, m_0
2: **function** DAF-Entropy(x, acc)
3: $d \leftarrow$ Number of dimensions
4: $d' \leftarrow x.depth$
5: **if** $d' = d$ **then**
6: x.ncount $\leftarrow x$.count + $Lap(1/(\epsilon_{\text{tot}} - acc))$
7: **return** TRUE
8: **if** $d' = 0$ **then**
9: x.ncount = x.count + $Lap(1/(\epsilon_{tot}/100))$
10: $acc \leftarrow acc + \epsilon_{tot}/100$
11: $m_0, m \leftarrow \sqrt[3(d-d')/2]{\frac{(x.ncount) \times (\epsilon_{\text{tot}} - acc)}{\sqrt{2}}}$
12: **else**
13: $mem \leftarrow \frac{\epsilon_{\text{tot}} \times m_0^{d'/3} \times (1 - m_0^{1/3})}{m_0^{1/3}(1 - m_0^{d/3})}$
14: x.ncount $\leftarrow x$.count + $Lap(1/mem)$
15: $acc \leftarrow acc + mem$
16: $m \leftarrow \sqrt[3(d-d')/2]{\frac{(x.ncount) \times (\epsilon_{\text{tot}} - acc)}{\sqrt{2}}}$
17: **if** Stop Conditions= TRUE **then**
18: $mem \leftarrow \epsilon_{\text{tot}} - acc$
19: x.ncount $\leftarrow x$.count + $Lap(1/mem)$
20: **return** TRUE
21: $M \leftarrow$ Split $(d'+1)$th dimension into m intervals
22: **for** i=1 to m **do**
23: create a new node x'
24: $x'.F \leftarrow x.F$ with ith dimension set to $M[i]$
25: $x'.depth \leftarrow d' + 1$
26: $x'.count \leftarrow SUM(x'.F)$
27: DAF-Entropy(x', acc)

The split is conducted based on the depth of the node, such that nodes at depth i are created by dividing dimension i of their parent node's partition. The maximum index height is $d + 1$. The fanout and the split point are customized at each node based on sanitized local information about the data. We propose two DAF alternatives based on different split objective functions: (i) *DAF-Entropy* (Section 4.2) which uses entropy information to estimate good split parameters, and (ii) *DAF-Homogeneity* (Section 4.3) which focuses on creating partitions with high intra-region homogeneity. Section 4.4 introduces privacy budget allocation considerations that are relevant to both approaches.

4.2 DAF-Entropy

DAF-Entropy has the recursive structure presented in Algorithm 2. It receives as inputs the current node to split denoted by x, privacy budget ϵ_{tot}, variable acc tracking the budget spent so far (initially set to zero), and a constant m_0 set in the first round of the recursion which is used for budget allocation purposes at all levels of the tree (more details are provided in Section 4.4). Each tree node x is an object with four attributes: (i) $x.F$; the node's associated entries in the frequency matrix, (ii) $x.count$; the actual sum of entries in $x.F$, (iii) $x.ncount$; the sanitized (or noisy) count, and (iv) $x.depth$; the node's depth in the tree. The initial run of the function is performed for the root node, representing the whole frequency matrix.

DAF-Entropy sanitizes the total count of the root node and utilizes Equation (19) to partition the first dimension of the frequency matrix. New nodes are generated for each new partition assigned as one of the node's children. The algorithm recursively visits children and repeats the same process with the key difference that the split is done based on the second dimension. More generally, upon reaching a node at depth i, the split is conducted in the $(i+1)$-th dimension.

Once a new node is visited, its count is sanitized, and stop conditions are tested on the sanitized count. If satisfied, the tree is pruned, and the node turns into a leaf. A special technique is used in such a scenario to enhance accuracy. The algorithm uses the remaining amount of budget that was supposed to be used while visiting children to update the sanitized count. This technique improves accuracy as budget allocation is such that lower levels of the tree are allocated more budget. Thus, it is worth updating the sanitized count based on the remaining amount of budget. Note that, stop conditions can be selected based on application-specific details; however, the most prominent stop condition that can help avoid over-partitioning is to stop when the sanitized count is below a certain threshold. The algorithm continues until reaching depth d, indicating that partitioning on all d dimensions has been implemented successfully or a stop condition is reached. Finally, the sanitized counts of the leaves are published, representing the private frequency matrix.

4.3 DAF-Homogeneity

The partitioning process plays a critical role in the private publication of frequency matrices. Hence, several attempts have been made in prior work [8, 19] to find an efficient splitting mechanism, including partitioning independent of data, based on medians or using the frequency matrix's total count to estimate a viable partitioning granularity. Our earlier work in [16] shows that partitioning based on homogeneity can significantly improve the utility of private frequency matrices in 2D. The principal idea is to have mechanisms that can cluster the entries such that data density is homogeneous within each cluster. Recall that partitioning needs to follow the DP constraint as with any other part of the algorithm. Here, we extend the approach in [16] to higher dimensional frequency matrices. The approach is built on top of Algorithm 2, with a key difference that once fanout is calculated for a node, an alternative method is used to partition the space based on homogeneity.

Suppose that while executing Algorithm 2, a node with depth i is visited. DAF-Homogeneity starts by dividing the calculated amount of budget into two parts: sanitization budget (ϵ_{data}), and partitioning budget (ϵ_{prt}).

$$\epsilon_{\text{prt}} = q\epsilon_i,\ \epsilon_{\text{data}} = (1-q)\epsilon_i \tag{20}$$

Constant q denotes the ratio of the budget assigned for partitioning. This value is experimentally set to 0.3. Next, the node's count is sanitized based on the Laplace mechanism with the privacy budget ϵ_{data}, and substituted in Equation (19) to calculate the fanout m.

Suppose that $m = 3$ and recall that for nodes with depth i, the split is conducted on dimension $i+1$. Let us denote the interval corresponding to the $(i+1)$-th dimension by $[k_{\text{start}}, k_{\text{end}}]$. In the case of DAF-Homogeneity, given that the fanout is calculated to be 3, the generated intervals for the $i+1$ dimension of children would be $[k_{\text{start}}, k_1)$, $[k_1, k_2)$, and $[k_2, k_{\text{end}}]$, where

$$k_1 = \lfloor k_0 + \frac{k_{\text{end}} - k_{\text{start}}}{3} \rfloor, k_2 = \lfloor k_0 + 2 \times \frac{k_{\text{end}} - k_{\text{start}}}{3} \rfloor \tag{21}$$

Algorithm 3 DAF-Homogeneity

1: **Global Constants:** p, q, ϵ_{tot}, m_0
2: **function** DAF-Homogeneity(x, acc)
3: $d \leftarrow$ Number of dimensions
4: $d' \leftarrow x.depth$
5: **if** $d' = d$ **then**
6: x.ncount $\leftarrow x$.count + $Lap(1/(\epsilon_{\text{tot}} - acc))$
7: **return** TRUE
8: **if** $d' = 0$ **then**
9: x.ncount = x.count + $Lap(1/(\epsilon_{tot}/100))$
10: $acc \leftarrow acc + \epsilon_{tot}/100$
11: $m_0, m \leftarrow \sqrt[3(d-d')/2]{\frac{(x.ncount) \times (\epsilon_{\text{tot}} - acc)}{\sqrt{2}}}$
12: $\mathcal{K}_1, ...\mathcal{K}_p \leftarrow$ Use m to generate candidate sets
13: Compute $O(\mathcal{K}_1)$, $O(\mathcal{K}_2)$, .., $O(\mathcal{K}_p)$
14: $\overline{O(\mathcal{K}_i)} \leftarrow O(\mathcal{K}_i) + Lap(2/(p \times \epsilon_{\text{prt}})), \forall i = 1...p$
15: $\mathcal{K} \leftarrow \underset{i}{Minimize}\ \overline{O(\mathcal{K}_i)}\quad \forall i$=1...$p$
16: **else**
17: $\epsilon \leftarrow \frac{\epsilon_{\text{tot}} \times m_0^{d'/3} \times (1 - m_0^{1/3})}{m_0^{1/3}(1 - m_0^{d/3})}$
18: $acc \leftarrow acc + \epsilon$
19: $\epsilon_{\text{prt}} \leftarrow q\epsilon$
20: $\epsilon_{\text{data}} \leftarrow (1-q)\epsilon$
21: x.ncount $\leftarrow x$.count + $Lap(1/\epsilon_{\text{data}})$
22: $m \leftarrow \sqrt[3(d-d')/2]{\frac{(x.ncount) \times (\epsilon_{\text{tot}} - acc)}{\sqrt{2}}}$
23: Execute lines 12 to 15
24: **if** Stop Conditions= TRUE **then**
25: x.ncount $\leftarrow x$.count + $Lap(1/(\epsilon_{\text{tot}} - acc))$
26: **return** TRUE
27: **else**
28: $M \leftarrow$ Split $(d'+1)$th dimension based on $\mathcal{K}$
29: **for** i=1 to m **do**
30: create a new node x'
31: $x'.F \leftarrow x.F$ with ith dimension set to $M[i]$
32: $x'.depth \leftarrow d' + 1$
33: $x'.count \leftarrow SUM(x'.F)$
34: DAF-Homogeneity(x', acc)

Instead of simply selecting k_1 and k_2 as splitting points, DAF-Homogeneity follows an alternative method: it generates p sets of candidate partitioning sets $\mathcal{K}_1$, $\mathcal{K}_2$, ...$\mathcal{K}_p$, where p is an input to the algorithm. Each set $\mathcal{K}_j$ has a cardinality equal to the desired fanout, and is generated by drawing uniformly random split positions from every partition. For example, consider the first candidate set to be $\mathcal{K}_1 = \{k'_1, k'_2, k'_3\}$, where k'_1, k'_2, and k'_3 are uniformly random coordinates drawn from intervals $[k_{\text{start}}, k_1)$, $[k_1, k_2)$, and $[k_2, k_{\text{end}}]$, respectively. Furthermore, let us denote the frequency matrix generated by setting the $i+1$ dimension into the jth interval by F^j. Next, the algorithm computes the homogeneity objective function for candidate sets, resulting in $O(\mathcal{K}_1)$, $O(\mathcal{K}_2)$, .., $O(\mathcal{K}_p)$, where

$$O(\mathcal{K}) = \sum_{i=1}^{|\mathcal{K}|+1} \sum_{f_j \in F^i} |f_j - \mu_{F^i}|, \tag{22}$$

In the above equation, μ_{F^i} denotes the average of entries in F^i.

$$\mu_{F^i} = \frac{\sum_{f_j \in F^i} f_j}{|F^i|} \tag{23}$$

Then, the output values are sanitized based on the Laplace mechanism with the reserved privacy budget for partitioning.

$$\overline{O(\mathcal{K}_i)} = O(\mathcal{K}_i) + Lap(s/\epsilon_{\text{prt}}), \forall i = 1...p \tag{24}$$

The optimal candidate set is chosen as the one that results in the minimum sanitized output.

$$\underset{i}{Minimize} \ \ \overline{O(\mathcal{K}_i)} \quad \forall i{=}1...p \tag{25}$$

Lemma 4.1. *Sensitivity of the homogeneity objective function is* 2.

Proof. In the calculation of objective function $O(\mathcal{K})$ for a given split index set $\mathcal{K}$, a data entry's existence or absence only affects one cell and the corresponding cluster. Let us denote the objective function after addition or removal of one data record by $O(\mathcal{K})'$.

$$O(\mathcal{K})' = \sum_{i=1}^{|\mathcal{K}|+1} \sum_{f'_j \in F^i} |f'_j - \mu'_{F_i}|, \tag{26}$$

Without loss of generality assume that the additional record is located in the first cluster which results in $\mu'_{F^1} = \mu_{F^1} + 1/|F^1|$, and $\mu'_{F^i} = \mu_{F^i}$ for all $i = 2, ..., k+1$. Similarly, the counts are equal ($f'_i = f_i$) for all entries except a single entry denoted by x for which we have we have $f'_x = f_x + 1$. Writing triangle inequality results in

$$\Big||f_i - \mu_1 - \frac{1}{|F^1|}| - |f_i - \mu_1|\Big| \leq \frac{1}{|F^1|} \quad \forall i{=}1...|F^1| - \{x\} \tag{27}$$

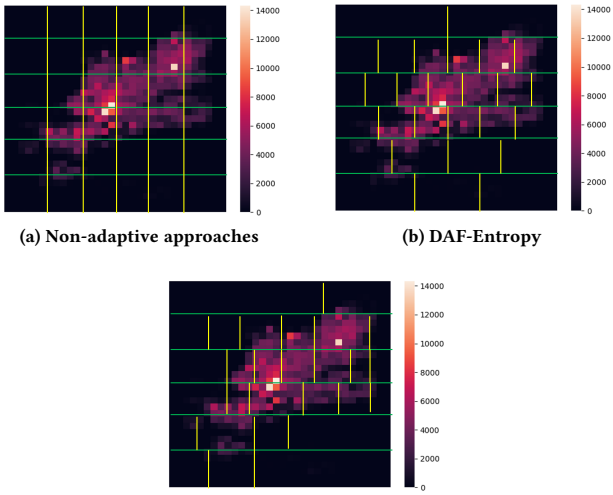
The sensitivity of the objective function can have the maximum value of two, as proven by the following inequality:

$$\Big||f_x + 1 - \mu_1 - \frac{1}{|F^1|}| - |f_x - \mu_1|\Big| \leq \frac{|F^1| - 1}{|F^1|} \tag{28}$$

□

The DAF-Homogeneity pseudocode is shown in Algorithm 3.

To better understand the reason why the proposed DAF approaches outperform competitor techniques, Fig. 3 provides a heatmap representation of Los Angeles city with 500,000 sampled from the Veraset dataset (experimental setup details are provided in Section 6.1). The partitioning conducted in the first and second dimensions are shown by green and yellow lines, respectively. For non-adaptive approaches, only the sanitized total population count is used for partitioning, and therefore, both dimensions are divided equally without considering user distribution (Fig. 3a). Conversely, the DAF-Entropy approach is adaptively adjusting the number of partitioning as the dimension changes (Fig. 3b). The DAF-Homogeneity technique goes one step further, and adjusts the number of partitions generated in each dimension, by selecting the split point such that resulting areas will exhibit homogeneous intra-bin density, hence reducing the negative effects of uniformity assumption and increasing query accuracy.



(a) Non-adaptive approaches (b) DAF-Entropy

(c) DAF-Homogeneity

Figure 3: Intuition behind DAF sanitization approaches.

4.4 Budget Allocation

The derivation of the optimal amount of privacy budget allocated for different levels of the hierarchy is a challenging task as nodes have varying fanouts. We formulate an optimization problem to achieve a good quality budget allocation. Denote the fanout of the root node by m_0. We assume that the progression of fanout is geometric. At depth i, there exist approximately m_0^i nodes. Furthermore, we denote the budget allocated to depth i of the tree by ϵ_i. The goal is to minimize the variance of the noise added to each level:

$$\underset{\epsilon_1...\epsilon_d}{min} \quad \sum_{i=1}^{d} m_0^i/\epsilon_i^2, \sum_{i=1}^{d} \epsilon_i = \epsilon'_{\text{tot}}, \quad \epsilon_i > 0 \ \ \forall i = 1...d \tag{29}$$

where, $\epsilon'_{\text{tot}} = \epsilon_{\text{tot}} - \epsilon_0$. We have intentionally separated the root node's budget, as it will be used to calculate m_0. The optimization problem can be solved by writing Lagrangian and KKT conditions.

$$L(\epsilon_1, ..., \epsilon_d, \lambda) = \sum_{i=1}^{d} m_0^i/\epsilon_i^2 + \lambda(\sum_{i=1}^{d} \epsilon_i - \epsilon) \tag{30}$$

$$\Rightarrow \frac{\partial L}{\partial \epsilon_i} = \frac{-2m_0^i}{\epsilon_i^3} + \lambda = 0 \Rightarrow \epsilon_i = \frac{(2m_0^i)^{1/3}}{\lambda^{1/3}}, \tag{31}$$

which leads to

$$\epsilon_i = \frac{\epsilon'_{\text{tot}} \times m_0^{i/3}}{\sum_{i=1}^{d} m_0^{i/3}} = \frac{\epsilon'_{\text{tot}} \times m_0^{i/3} \times (1 - m_0^{1/3})}{m_0^{1/3}(1 - m_0^{d/3})}. \tag{32}$$

A question arises on how to calculate the value m_0 upon which the above optimization problem is formulated. Note that the formulation only considers depths 1 to d, and the root node is excluded from the equation. The value of m_0 is calculated in the first run of the recursive algorithm 2, and we set the budget to:

$$\epsilon_0 = \frac{\epsilon_{\text{tot}}}{100} \tag{33}$$

Therefore, a comparably small amount of budget is allocated to the root node to derive m_0. Based on the above formulation, one can see that lower levels of the tree benefit from significantly higher levels of budget. This helps to improve the utility of the

published private histogram, as the sanitized leaf set of the tree represents the counts published by our approach.

5 RELATED WORK

Prior works on private publication of frequency matrices can be classified into three categories: data independent, partially data dependent, and data dependent algorithms. The algorithms in the first category are independent of the underlying dataset. The partial data dependent algorithms are the category of algorithms where the number of data points is used to generate the private FMs, but no consideration is made for the data distribution. The algorithms in the last category take the distribution of data points into consideration to improve the utility. Most algorithms are developed to address only the publication of 1D and 2D FMs.

In the category of data-independent approaches, two baseline algorithms that stand out are called *singular* and *identity*. The singular algorithm [8] considers the frequency matrix as a single partition and adds Laplace noise to the total count. The queries are answered based on the sanitized total count only, considering the assumption of data uniformity. The identity algorithm [7] on the other hand, adds Laplace noise to each entry of the frequency matrix. The number of partitions in this algorithm is equal to the total number of entries. The *Privlet* algorithm [18] enhances the performance of the identity algorithm by transforming the frequency matrix based on wavelets and by adding noise in the new domain. Then, the algorithm converts back to the noisy matrix and releases the DP counts. The authors in [4] build a quadtree on top of the FM: a tree-based spatial indexing structure that splits each node into four equal quarters, regardless of data placement. The so-called binning or partitioning of space without observing the histograms is studied in [3]. The authors consider the amount of overlap between bins and propose an algorithm called 'varywidth' that provides improved performance in terms of the trade-off between the spatial precision and the accumulated variance over differentially private queries. The use of summaries for private publication of histograms is explored in [5]. The authors show it is possible to reduce the two-step approach of generating private summaries, in which first the private histogram is generated and then the summaries are released, to a one-step approach. The one-step method prevents the data owner and data user from getting overwhelmed with the large computational complexity overhead.

In contrast to the data independent algorithms, data dependent approaches exploit the distribution of data in the FM to answer queries with higher accuracy. General purpose mechanisms [13, 14] and their workload-aware counterpart DAWA [12] operate over a discrete 1D domain; however, they can be applied to the 2D domain by dimensional reduction transformations such as Hilbert curves [8]. Unfortunately, dimensionality reduction can prevent range queries from being answered accurately, and also increases computational complexity. This significantly limits their practicality, particularly for higher-dimensional data. Data-aware tree-based algorithms such as k-d-trees [19] allocate a portion of the budget to partitioning, and generate split points based on density. Hybrid approaches between data-independent and data-dependent algorithms have also been proposed, e.g., UG and AG [15]. We refer to these approaches as partially data-dependent. Only the sanitized total count of the FM is used in the partitioning process. The UG algorithm and its extension [15] sanitize the total count of FMs and use it to alter the granularity of FM such that the utility of the published private FM is improved.

Table 2: Summary of Compared Approaches

Strategy	Symbol
Baseline Algorithms	IDENTITY [7] UNIFORM [7]
Non-adaptive Sanitization Approaches	EUG EBP MKM [11]
With partitioning budget	DAF-Entropy
Without partitioning budget	DAF-Homogeneity

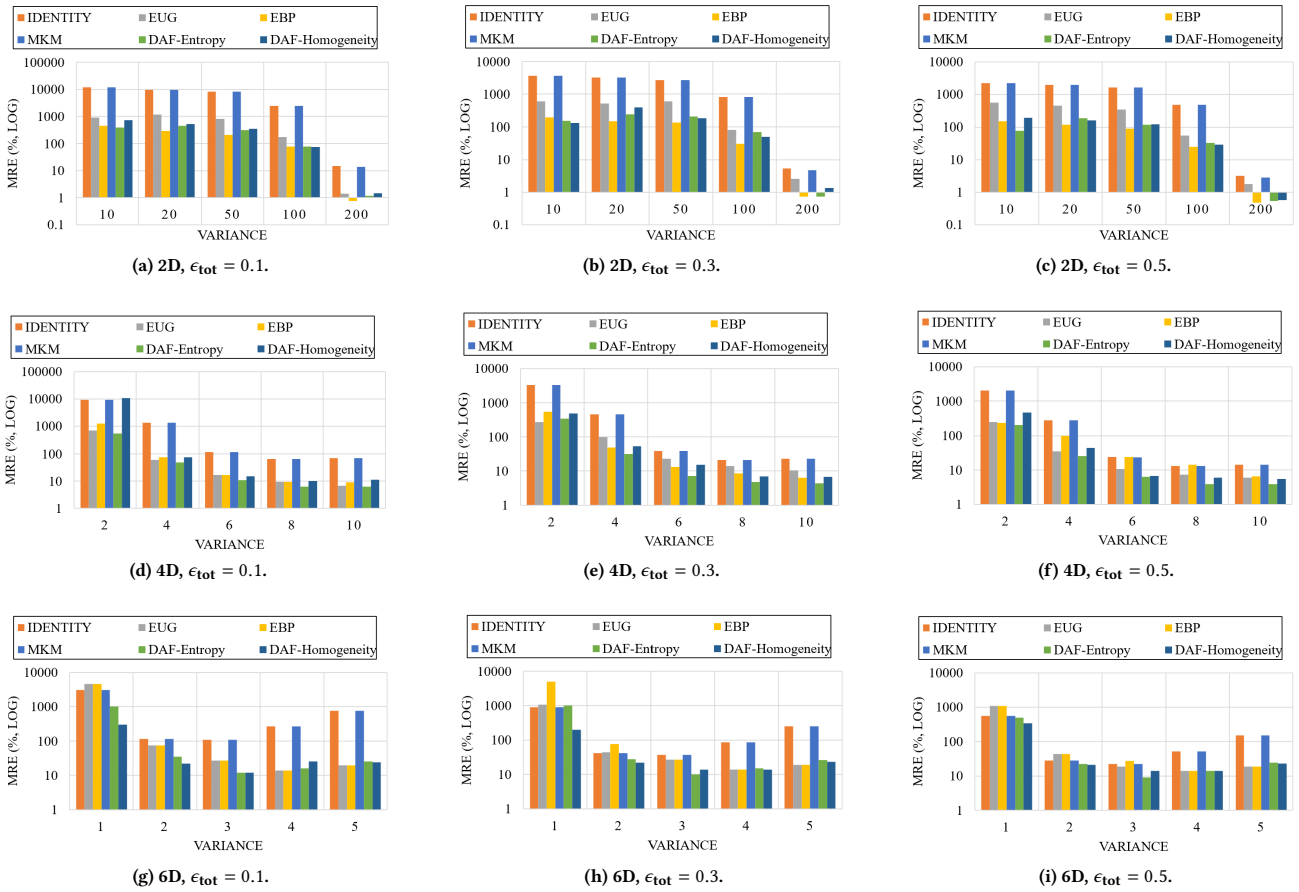
The MKM approach proposed in [11] provides an alternative formula to partition FM considering its dimensionality. As is the case in UG, the formula only takes as input the total count of the frequency matrix and determines the granularity of FM based on the sanitized total count. In some cases, such approaches have been shown to provide superior performance to more complex methods [8].

There is prior work in storage, processing, and compression of histograms, but without considerations for privacy. The authors in [9] focus on lowering the computational complexity of matrix multiplication and storage. The proposed approach generates an execution plan for the multiplication of dense and sparse matrices. A cost model is also proposed to understand the sparsity of matrices and the estimation of density. The execution plan tends to optimize the overall cost overhead. An adaptive tile matrix representation is proposed in [10] for large matrix multiplication. An operator called ATMULT with the capability of shared memory parallel matrix multiplication is proposed for dynamic tile-granular optimizations, conducted based on the density estimation. The work in [6] studies the problem of density estimation for higher dimensional histograms. The main idea is to estimate the distribution of data for a given set of samples. The algorithm provides near-optimal sample complexity, i.e. close to theoretical information limit, and runs in polynomial time.

6 EXPERIMENTAL EVALUATION

6.1 Experimental Setup

Synthetic Datasets. We generate synthetic frequency matrices according to both Gaussian and Zipf distribution. To generate a d-dimensional Gaussian frequency matrix F with dimensions $F_1 \times F_2 \times ... \times F_d$, a uniformly random integer is sampled in each dimension: $c_i \sim \text{Uniform}(1, F_i),\ \forall i = 1...d$. The generated point $(c_1, c_2, ..., c_d)$ is selected as the cluster center and 1 million datapoints are generated with respect to the cluster center according to a normal distribution. Specifically, each data point $(x_1, x_2, ..., x_d) \in \mathcal{Z}^d$ is sampled from a multivariate Gaussian distribution $(X_1, X_2, ..., X_d)$, where $X_i \sim \mathcal{N}(c_i, var)$. Changing the variance var allows us to adjust the degree of data skewness (lower values of var will correspond to more skewed data). Zipfian data are generated by sampling each datapoint from a multivariate Zipf distribution $(Y_1, Y_2, ..., Y_d)$, where $Y_i \sim \frac{x^{-a}}{\zeta(a)}$. $\zeta(.)$ denotes the Riemann Zeta function and parameter a controls the skew in the frequency matrix. As opposed to variance in Gaussian distribution, a higher value of a results in a more skewed distribution for the Zipf distribution.



(a) 2D, $\epsilon_{tot} = 0.1$. (b) 2D, $\epsilon_{tot} = 0.3$. (c) 2D, $\epsilon_{tot} = 0.5$.

(d) 4D, $\epsilon_{tot} = 0.1$. (e) 4D, $\epsilon_{tot} = 0.3$. (f) 4D, $\epsilon_{tot} = 0.5$.

(g) 6D, $\epsilon_{tot} = 0.1$. (h) 6D, $\epsilon_{tot} = 0.3$. (i) 6D, $\epsilon_{tot} = 0.5$.

Figure 4: Synthetic dataset results, Gaussian distribution, random shape and size queries.

Real-world datasets. We use a subset of the Veraset[1] dataset [17], including location measurements of cell phones in three US cities: New York, Denver and Detroit. For each city, we consider a large geographical region covering a 70×70 km^2 area centered at the city's central latitude and longitude. These are chosen to represent cities with high, moderate and low densities, respectively. Cities are modeled by a 1000×1000 frequency matrix where each entry represents the number of data points in the corresponding region of the city. The selected data generates a frequency matrix of 1 million data points during the time period March 1-7, 2020.

Based on the real location data, we construct origin-destination matrices: in each city, 300, 000 trajectories are sampled, and their origin, destination and intermediate points are included in the OD matrix. The data are stored as a multi-dimensional frequency matrix generated as follows: the map of each city is discretized to a 1000×1000 grid, and for every trajectory with the origin coordinates of (x_o, y_o) and destination coordinates of (x_d, y_d), the element $F[x_o, y_o, x_d, y_d]$ in the frequency matrix is incremented by one. A similar process is conducted for intermediate points, with the distinction that the matrix dimension count increases.

The evaluation metric used to compare the results is Mean Relative Error (MRE), formally defined in Section 2, Eq. (3). We evaluate the accuracy of considered approaches on the basis of:

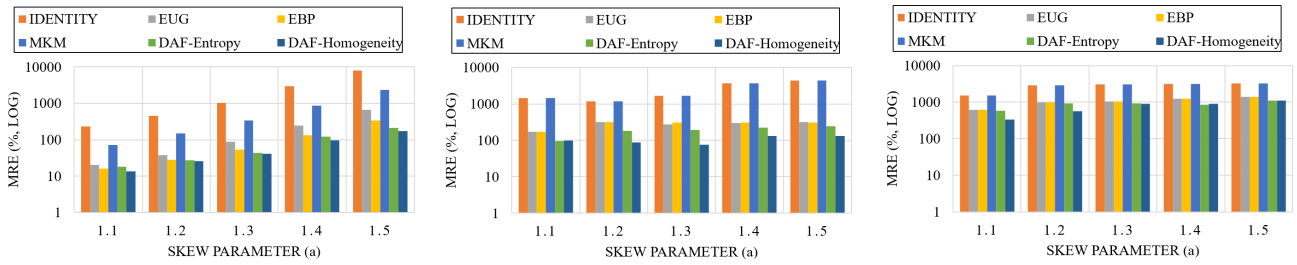
- *Varying Data Skewness/Distribution.* The generation of synthetic datasets is conducted for Gaussian and Zipfian random variables with distinct variances; for real-world datasets we select cities with a wide range of skewness properties.
- *Varying Query Shape/Size.* Each data point in our experiments is the average MRE result of 1000 queries generated based on random shapes and sizes. Additionally, the impact of small, medium and large queries is evaluated.
- *Varying Privacy Budget.* The experiments consider three privacy budget values of 0.1, 0.3, 0.5 modeling high, moderate, and low privacy constraints.
- *Varying dimensionality.* We run experiments on frequency matrices with dimensionality from two to six.

Compared Approaches. Table 2 provides a summary and corresponding references for each of the algorithms used in our evaluation. More details about each of the baselines are provided in Section 5. In total, we consider six techniques: IDENTITY, EUG, EBP, MKM, DAF-Entropy, and DAF-Homogeneity.

6.2 Results on Synthetic Datasets

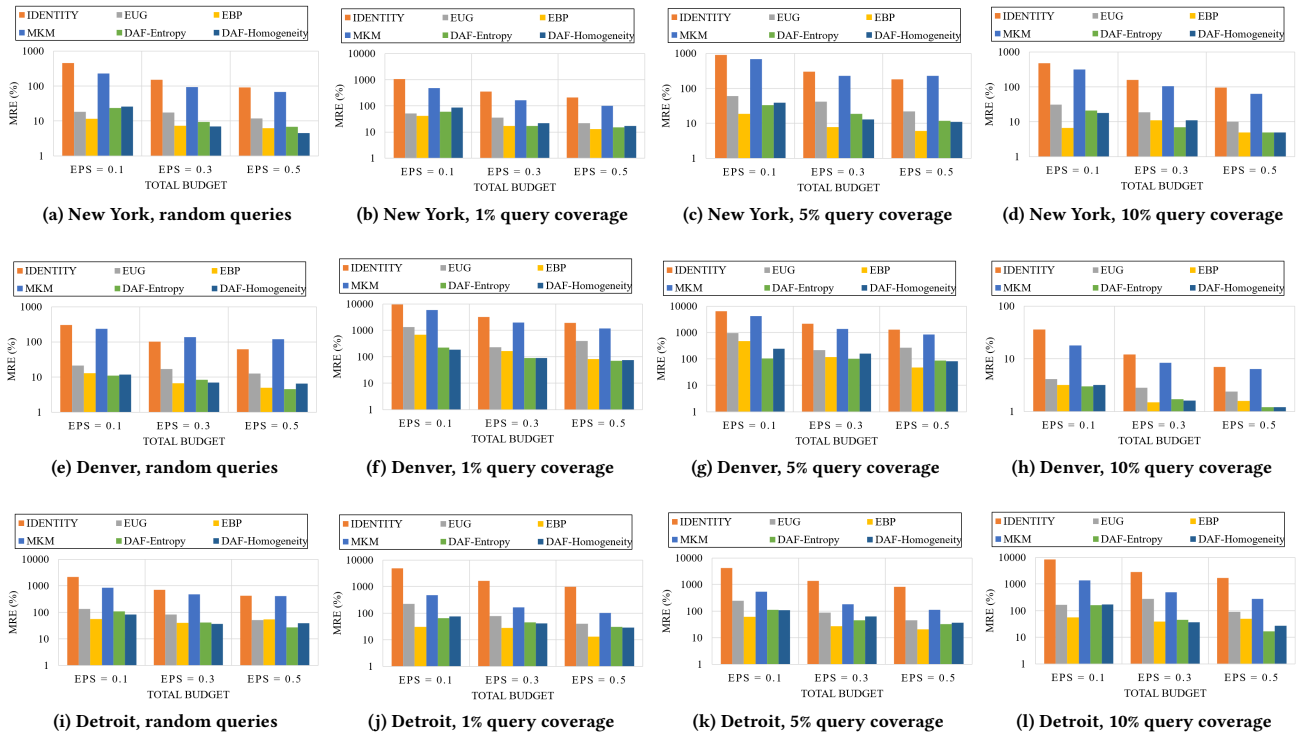
Figure 4 presents evaluation results on synthetic datasets. For each distinct dimensionality (i.e., row), we consider low, medium and high privacy budget settings. The width of frequency matrices in each dimension is set to $\sqrt[d]{N}$.

[1]Veraset is a data-as-a-service company that provides anonymized population movement data collected through signals of cell phones across the USA.



(a) 2D. **(b) 4D.** **(c) 6D.**

Figure 5: Synthetic dataset results, Zipf distribution, random shape and size queries, $\epsilon_{tot} = 0.1$.



(a) New York, random queries **(b) New York, 1% query coverage** **(c) New York, 5% query coverage** **(d) New York, 10% query coverage**

(e) Denver, random queries **(f) Denver, 1% query coverage** **(g) Denver, 5% query coverage** **(h) Denver, 10% query coverage**

(i) Detroit, random queries **(j) Detroit, 1% query coverage** **(k) Detroit, 5% query coverage** **(l) Detroit, 10% query coverage**

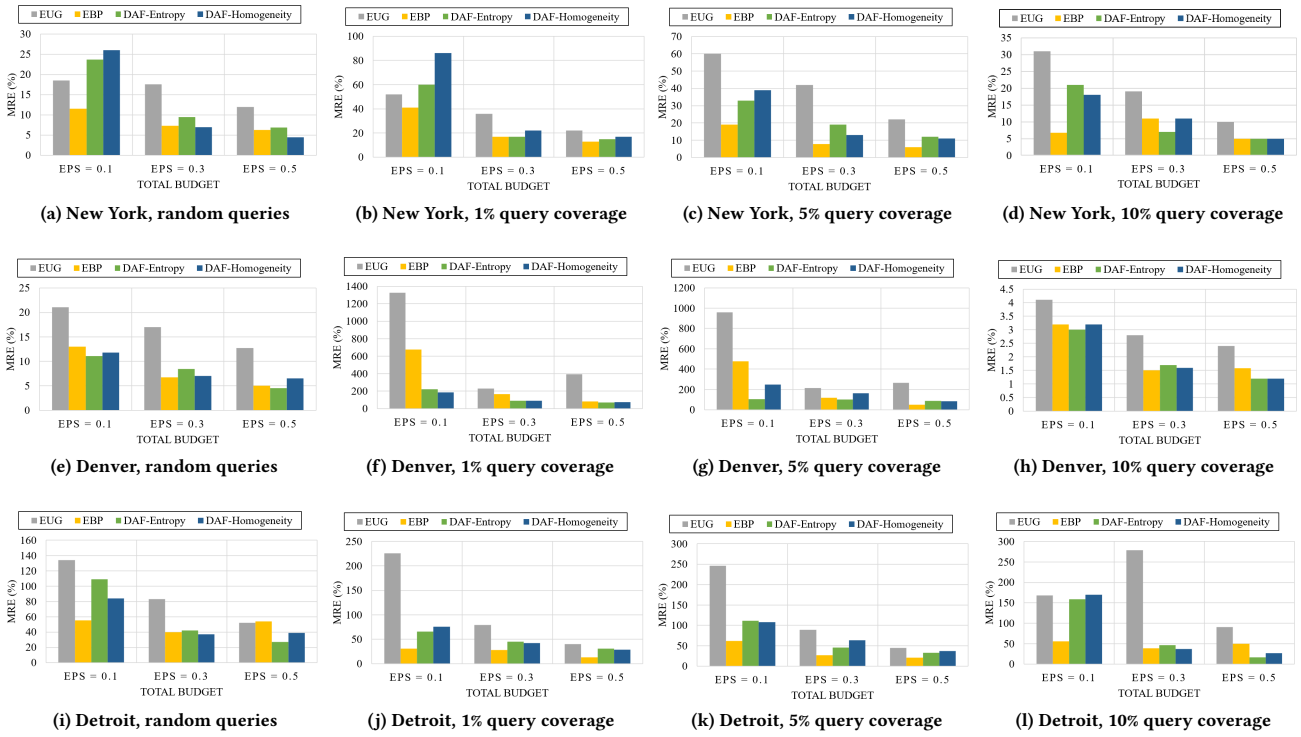
Figure 6: Population histograms in 2D for real datasets.

For the 2D case, results are shown in Figures 4a-4c. EBP and DAF-Entropy provide superior accuracy compared to other techniques, followed by DAF-Homogeneity and EUG. The MKM and IDENTITY algorithms exhibit similar performance, and we observed that MKM is reaching the maximum granularity for the frequency matrix. This is justified by the fact that the MKM approach does not follow the *epsilon-scale* exchangeability principle identified in [8]. In general, there exist two scenarios in which data-independent algorithms perform better: (i) the data points are distributed almost uniformly, (i.e., high variance) and (ii) the data points are densely populated in the cluster center in a handful of matrix entries (i.e., low variance). The superior performance of the DAF framework becomes more evident in higher dimensions. In almost all experiments conducted, the DAF framework outperformed the data-independent sanitization approaches. Among the two objective functions that we developed for DAF, DAF-Entropy generally outperforms DAF-Homogeneity.

We also evaluate the studied approaches for Zipf synthetic distribution of data. Figure 5 shows similar relative trends, with the proposed approaches outperforming existing work by an order of magnitude. The error increases as the skew parameter *a* increases.

6.3 Results on Real-World Datasets

Figure 6 shows the accuracy of all studied methods on 2D data, for various query workloads: a mix of random queries, as well as fixed coverage queries with range from 1% to 10% of dataspace side. As in the case of synthetic data, the IDENTITY and MKM benchmarks underperform by an order of magnitude. For all methods, the error decreases when the query range increases, which is expected, since coarser queries can be accurately answered using most methods. However, the more challenging case is that of small query ranges, which provide more detailed information to the analyst.



(a) New York, random queries (b) New York, 1% query coverage (c) New York, 5% query coverage (d) New York, 10% query coverage

(e) Denver, random queries (f) Denver, 1% query coverage (g) Denver, 5% query coverage (h) Denver, 10% query coverage

(i) Detroit, random queries (j) Detroit, 1% query coverage (k) Detroit, 5% query coverage (l) Detroit, 10% query coverage

Figure 7: Population histograms in 2D on real datasets, no baselines.

Due to their poor performance, we exclude IDENTITY and MKM from the rest of the results, and focus on studying the relative performance of the proposed approaches, illustrated in Figure 7 on linear scale. The EUG algorithm results in poorer accuracy overall. For Detroit and New York EBP has performed better than competing techniques. The EBP and DAF results are comparable for the Denver datasets, with DAF-Homogeneity providing the highest accuracy. The EBP algorithm performs better in cities where the entropy of the population histogram is higher. This aligns with our expectations, as greater entropy can be an indicator of higher skewness, where EUG performs worse. When increasing the privacy budget, the error of all algorithms decreases consistently, since the noise required to satisfy the privacy bound becomes lower. Fig. 8 presents the results for higher-dimensionality matrices. Similar to the results observed for synthetic datasets, DAF-Entropy has superior accuracy on average compared to the other techniques. The relative accuracy gain achieved by DAF is observed to increase as the number of dimensions increases.

Table 3 shows the execution time for all techniques. The DAF methods have faster execution time, because they adapt to data and do not perform unnecessary splits. In all cases, the proposed techniques complete execution in less than five minutes.

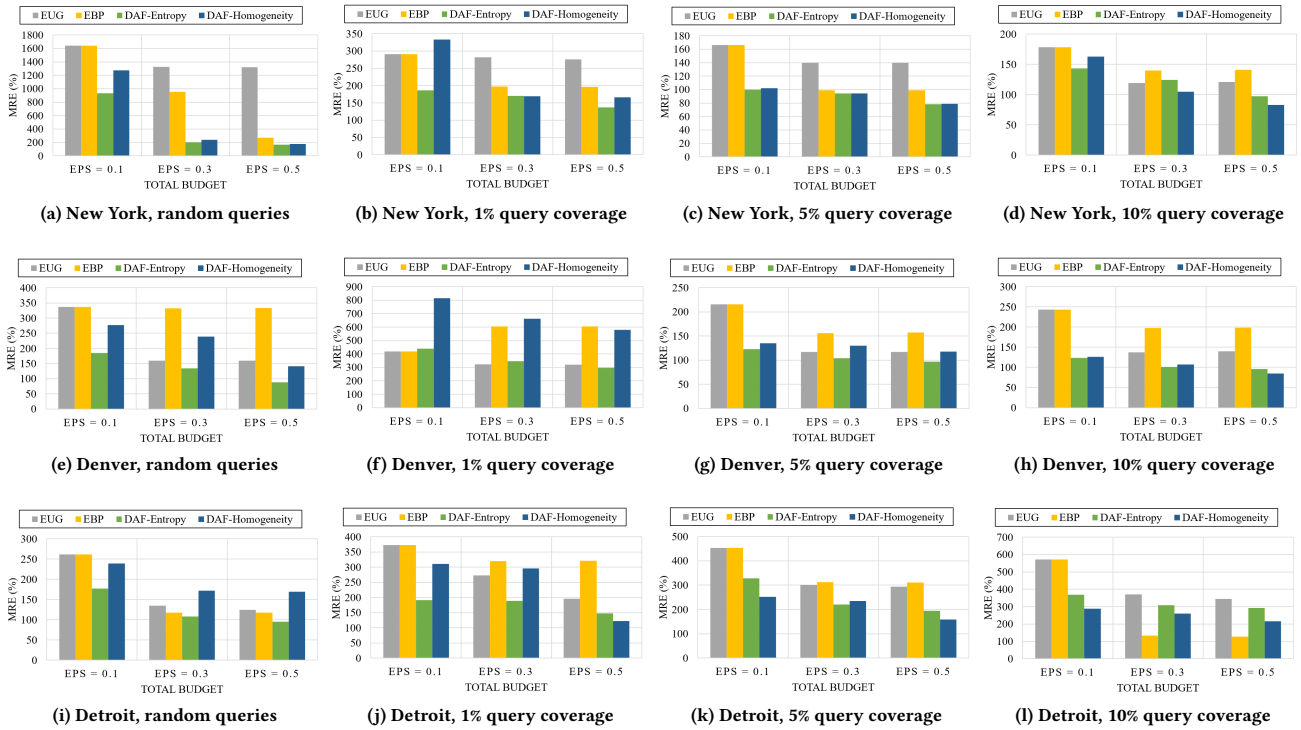
Discussion. Data-independent methods perform better when data are highly uniform or highly concentrated around the cluster center. However, most location datasets do not fall in either of these cases, hence there is need for carefully-designed density-aware approaches, like the ones we proposed. In lower dimensions, the EBP algorithm outperforms competitor approaches on both real-world and synthetic datasets. In higher dimensions, the density-aware algorithms outperform data-independent algorithms. The improvement margin increases as the number of dimensions grows. On average, DAF-Entropy outperforms its homogeneity-based counterpart due to the additional budget required for evaluating homogeneity metrics of candidate splits in the latter.

Table 3: Running time of algorithms (seconds), 2D, $\epsilon = 0.1$

	IDENTITY	EUG	EBP	MKM	DAF-Entropy	DAF-Homogeneity
New York	89	87	87	177	.47	0.5
Denver	91	91	94	182	0.38	0.46
Detroit	111	111	110	272	0.34	0.48

7 CONCLUSION

We proposed a customized privacy-preserving approach for the publication of origin-destination matrices with intermediate stops in the context of differential privacy. Our proposed approaches provide the strong formal protection guarantees of differential privacy, while achieving superior accuracy to existing techniques that are designed for low-dimensionality location data and do not adapt well to data properties such as density variation. In future work, we plan to further improve accuracy by considering more sophisticated mechanisms in addition to Laplace noise addition. We will also investigate the correlation between location and semantic features of the geographical dataspace, which can provide additional accuracy in the case of semantic-centric queries (e.g., an analyst may be interested in trajectories that satisfy some semantic constraint, like *workplace-entertainment-sports sequences*, where the type of feature visited is more important than the actual geographical placement).



(a) New York, random queries (b) New York, 1% query coverage (c) New York, 5% query coverage (d) New York, 10% query coverage

(e) Denver, random queries (f) Denver, 1% query coverage (g) Denver, 5% query coverage (h) Denver, 10% query coverage

(i) Detroit, random queries (j) Detroit, 1% query coverage (k) Detroit, 5% query coverage (l) Detroit, 10% query coverage

Figure 8: Origin-Destination matrices in 4D, real datasets.

ACKNOWLEDGEMENT

This research has been funded in part by NSF grants IIS-1910950, IIS-1909806, CNS-2027794, IIS-2128661 and CNS-2125530, the USC Integrated Media Systems Center (IMSC), and an unrestricted cash gift from Microsoft Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of any of the sponsors such as the NSF.

REFERENCES

[1] Gergely Acs, Claude Castelluccia, and Rui Chen. 2012. Differentially private histogram publishing through lossy compression. In *Intl. Conference on Data Mining*. 1–10.

[2] Rui Chen, Gergely Acs, and Claude Castelluccia. 2012. Differentially private sequential data publication via variable-length n-grams. In *ACM CCS*. 638–649.

[3] Graham Cormode, Minos Garofalakis, and Michael Shekelyan. 2021. Data-Independent Space Partitionings for Summaries. In *Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 285–298.

[4] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. 2012. Differentially private spatial decompositions. In *IEEE Conf. on Data Engineering*. 20–31.

[5] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, and Thanh TL Tran. 2012. Differentially private summaries for sparse data. In *Proceedings of the 15th International Conference on Database Theory*. 299–311.

[6] Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. 2018. Fast and sample near-optimal algorithms for learning multidimensional histograms. In *Conference On Learning Theory*. PMLR, 819–842.

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

[8] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, and Dan Zhang. 2016. Principled evaluation of differentially private algorithms using dpbench. In *Proceedings of the 2016 International Conference on Management of Data*. 139–154.

[9] David Kernert, Frank Köhler, and Wolfgang Lehner. 2015. SpMacho-Optimizing Sparse Linear Algebra Expressions with Probabilistic Density Estimation.. In *EDBT*. 289–300.

[10] David Kernert, Wolfgang Lehner, and Frank Köhler. 2016. Topology-aware optimization of big sparse matrices and matrix multiplications on main-memory systems. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 823–834.

[11] Jing Lei. 2011. Differentially private m-estimators. *Advances in Neural Information Processing Systems* 24 (2011), 361–369.

[12] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. 2014. A data-and workload-aware algorithm for range queries under differential privacy. *Proceedings of the VLDB Endowment* 7, 5 (2014), 341–352.

[13] Chao Li and Gerome Miklau. 2012. An Adaptive Mechanism for Accurate Query Answering under Differential Privacy. *Proc. VLDB Endow.* 5, 6 (2012), 514–525.

[14] Chao Li and Gerome Miklau. 2013. Optimal error of query sets under the differentially-private matrix mechanism. In *Intl. Conference on Database Theory*. 272–283.

[15] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2013. Differentially private grids for geospatial data. In *IEEE International conference on data engineering*. IEEE, 757–768.

[16] Sina Shaham, Gabriel Ghinita, Ritesh Ahuja, John Krumm, and Cyrus Shahabi. 2021. HTF: Homogeneous Tree Framework for Differentially-Private Release of Location Data. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 184–194.

[17] Veraset. [n.d.]. Veraset Movement data for the USA, The largest, deepest and broadest available movement dataset (anonymized GPS signals).

[18] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2010. Differential privacy via wavelet transforms. *IEEE Transactions on knowledge and data engineering* 23, 8 (2010), 1200–1214.

[19] Yonghui Xiao, Li Xiong, and Chun Yuan. 2010. Differentially private data release through multidimensional partitioning. In *Workshop on Secure Data Management*. Springer, 150–168.

[20] Jun Zhang, Xiaokui Xiao, and Xing Xie. 2016. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proceedings of the 2016 International Conference on Management of Data*. 155–170.

[21] Xiaojian Zhang, Rui Chen, Jianliang Xu, Xiaofeng Meng, and Yingtao Xie. 2014. Towards accurate histogram publication under differential privacy. In *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 587–595.