World Wide Web (2020) 23:549–572
https://doi.org/10.1007/s11280-019-00769-8

A low cost and un-cancelled laplace noise based differential privacy algorithm for spatial decompositions

Xiaocui Li[1] · Yangtao Wang[1] · Jingkuan Song[2] · Yu Liu[1] · Xinyu Zhang[3] · Ke Zhou[1] · Chunhua Li[1]

Received: 30 May 2019 / Revised: 5 September 2019 / Accepted: 5 December 2019 /
Published online: 11 January 2020

Abstract
Differentially private spatial decompositions split the whole domain into sub-domains recursively to generate a hierarchical private tree and add Laplace noise to each node's points count. However the Laplace distribution is symmetric about the origin, the mean of a large number of queries may cancel the Laplace noise and reveal privacy. Existing methods take the solution by limiting the number of queries. But in private tree the points count of intermediate node may be real since the summation of all its descendants may cancel the Laplace noise. To address these problems of differentially private spatial decompositions, we propose a more secure algorithm to make the Laplace noise not be canceled. That splits the domains depending on its real points count not noisy, and only adds indefeasible Laplace noise to leaves. That the ith randomly selected leaf of one intermediate node is added noise by $\frac{(\beta-i+1)+1+\beta}{(\beta-i+1)+\beta} Lap(\lambda)$. We also propose the definition of $Lap_{min}(\lambda)$ whose absolute value is not greater than $Sensitivity(f)$. It is proved that adding $Lap_{min}(\lambda)$ noise to query answer guarantees both differential privacy and minimal relative error comparing with unlimited Laplace noise. The experiment results show that our algorithm performs better both on synthetic and real datasets with higher security and data utility, and the noises costs is highly decreased.

Keywords Differential privacy · Indefeasible Laplace noise · $Lap_{min}(\lambda)$ · Spatial decompositions

1 Introduction

In the information age, the explosive growth of various data makes data miners mine knowledge driven by diverse purposes, such as business decision, military intelligence analysis,

This article belongs to the Topical Collection: *Special Issue on Trust, Privacy, and Security in Crowdsourcing Computing*
Guest Editors: An Liu, Guanfeng Liu, Mehmet A. Orgun, and Qing Li

✉ Chunhua Li
li.chunhua@hust.edu.cn

Extended author information available on the last page of the article.

disease control and prevention, and so on. However, with the rapid development of data mining technology [1], people's privacy is facing a huge threat as their sensitive information is revealed. For this reason, data privacy was highly valued in the past two decades and many approaches were token into the process of data publishing [2, 3]. Geographic information has become increasingly abundant, such as the distribution of endangered species, the distribution of hotel occupancy in a certain city, the trip distribution of occupied taxis, and so on. The statistic of location information can help biological scientists, government, business decision-makers etc. make the correct and effective decision.

Most algorithms obtain the statistic of location information through spatial decomposition. Spatial decompositions take the solution of splitting the dataset D of tuples in domain Ω, recursively decompose Ω into a set of sub-domains if the point count of current domain is larger than the given threshold θ. When met the recursive termination condition, a hierarchical spatial decompositions tree was generated. Through retrieving this tree, the location information in a certain region can be obtained. We assume that only the statistic information of database can be queried. Each one individual's location should not be queried for privacy protection [4–6]. However the spatial decompositions tree may reveal individuals' privacies. For example, when this tree is deep enough, many leaves of this tree may only contain one tuple. Some malicious attackers can get individuals' precise information by various technological means such as data mining algorithms via retrieving the spatial decompositions tree, as a result that will reveal the users' privacy. To address this problem, we can perturb the information of this tree to achieve preserving privacy [7] before publishing, which is called preserving privacy data publishing [8, 9].So that users can get the statistical information of the database as a whole but not an individual's information when users access the database.

The research of privacy preserving is mainly classified into two main categories, data clustering and theoretical framework [10]. The algorithms of data clustering are developed from k-anonymity [11] and l-diversity [12] to t-closeness [13]. The theoretical framework direction mainly includes differential privacy and its further developments.

Dwork et al. first gave the notion of differential privacy [14]: deleting an element from a statistical database should not substantially increase the risk of the record owner's privacy. ε-differential privacy is the first algorithm with strict theoretical framework which guarantees individual's the privacy even the adversary knows all other records and has sufficient background knowledge. Consequently, Dwork proposed a theoretical framework [15–17] called ε-differential privacy, and proved that the *Laplace mechanism* [18] can achieve differential privacy for numerical queries. In 2007, Frank M. et al. proposed the *Exponential mechanism* [19, 20] to preserve the non-numerical queries. In the last decade, differential privacy was applied in many algorithms, such as histogram-based data publishing [21], batch query [22], decision tree [23] and so on. In the last decade, ε-differential privacy was applied to various fields such as data publishing, data mining and machine learning [24, 25]. Differential privacy was applied in data publishing including histogram data publishing, tree structure based and grid based data publishing. All the existing algorithms focus on the data privacy and the application of differential privacy, but not take an important consideration of the data utility. Data privacy and data utility are both very important in data analysis. Though the trade-off between them is difficult to both, we concentrate on improving the data utility under the premise of data privacy.

Graham Cormode proposed the DPSD in 2012, that differential privacy was firstly applied to spatial decompositions [26]. this method took the approach of adding Laplace noise to each node of the spatial decompositions tree and publishing the noisy private tree to achieve the purpose of privacy preserving. Then more excellent algorithms [27–29] emerged

for spatial decompositions based on differential privacy, of which the PrivTree proposed by Jun Zhang 2016 was especially noteworthy. The existing spatial decompositions in differential privacy generated a private tree by spatial decompositions. The region of of the whole dataset Ω will be recursively split to subregions. Each split of one node will generate 2^d children, where d is the dimension of the dataset, which finally outputs a 2^d-tree. For preserving privacy, this 2^d-tree will be perturbed by adding Laplace noise to each node. Users can get the answers of their queries of the point count of a given area by retrieving the private tree.

1.1 Motivation

Differential privacy indeed can achieve the goal of privacy protection. However, The Laplace distribution curve is symmetric with respect to $x = 0$. In [15] Dwork indicated that through adding Laplace noise to the answers to achieve differential privacy is delicate, since Laplace noise is symmetric about of the origin and the same question is asked many times, the responses may be averaged, which can cancel out the Laplace noise. So Dwork suggested that the database manger can limit the number of queries to address this problem. However in the private spatial decompositions tree, each intermediate node's points count equals the summation of all its descendants' points counts, it may reveal the intermediate node's privacy for the summation of all its descendants Laplace noise may be canceled.

Besides, it is no need to add Laplace noise to every node of the tree. Since one intermediate node in the tree corresponds to its specific region, the regions corresponding to its children are the partitions of it. Adding a Laplace noise to one leaf can perturb each node's points count who are in the path between this leaf to root. Adding Laplace noise to each node of the tree makes domain split imprecise as the noise snowballs from root to leaf; besides, the noises costs of the private tree is highly increased.

Main problems of existing algorithms for spatial decompositions are that (i) when querying statistic information by retrieving private tree, the response may be true answer as the distribution of Laplace is symmetric about 0, and the summation of these nodes may cancel the Laplace noise. (ii) A great number of Laplace noises are needed for preserving privacy since each node of the private tree is added by Laplace noise, which will lead to excessive noises costs. (iii) The depth of the private tree h, also the maximum depth of recursion should be predefined when splitting Ω. However, the choice of h can be challenging, since a smaller h will cause coarse splitting of Ω, while a larger h will lead to high depth of the private tree too high, as a result the noise added to the private tree will increase.

1.2 Contribution

To address the limitations above, we propose a more secure spatial decompositions algorithm via indefeasible Laplace noise in differential privacy. We (i) give a minimal split unit u instead of h, which is more intuitive for users, (ii) only add Laplace noises to leaves of the private tree, which decreases the noises costs of the private tree. We (iii) first propose the notion of indefeasible Laplace noise, which solves the problem of symmetric distribution of Laplace, and guarantees the ε-differential privacy.

In summary, we make the following contributions in this paper:

1. we propose a more secure spatial decompositions algorithm via indefeasible Laplace noise in differential privacy, which only adds noise to leaves but not to intermediate nodes. We use a more intuitive threshold u, the minimal split unit to limit the maximum depth of the private tree.

2. We add indefeasible noise to the ith randomly selected leaf child of each intermediate node of the private tree through multiplying Laplace noise by coefficient of $\frac{(\beta-i+1)+\beta+1}{(\beta-i+1)+\beta}$. It satisfies ε-differential privacy and the answer to query will be not real because the added noise cannot be cancelled. It is also proved that the noises costs is lower than existing algorithms.
3. We take the solution of adding $Lap_{min}(\lambda)$ noise to query answer which guarantees the maximal data utility.
4. We conduct extensive experiments to demonstrate the performance of our algorithm

2 Preliminaries

In this section, we give the background of differential privacy and introduce the problem of spatial decomposition, which will be addressed by our algorithm in Section 3.

2.1 Differential privacy

Dwork et al. first proposed the concept of differential privacy, in which a randomized function F is applied by the data publisher when releasing information. The input is the dataset D and the output is the released information, named transcript. Differential privacy requires that the outputs of D and D' which differ in at most one tuple can't be distinguished.

Definition 1 (NEIGHBORING DATASETS) *The dataset D and D' are neighbors if D and D' differ in at most one element.*

Definition 2 (ε-DIFFERENTIAL PRIVACY) *The randomized mechanism M satisfies ε-differential privacy if, for any two neighboring datasets D and D' and for all output $O \in Range(M)$,*

$$\frac{Pr\left[M\left(D\right)=O\right]}{Pr\left[M\left(D'\right)=O\right]} \leq e^{\varepsilon} \tag{1}$$

Where $Pr[\cdot]$ denotes the probability of an event.

Definition 3 (SENSITIVITY) *Let f be a function that maps a dataset D to a vector of real numbers. The global sensitivity of f is defined as:*

$$S\left(f\right)=\max_{D,D'}\left\|f\left(D\right)-f\left(D'\right)\right\|_1 \tag{2}$$

Where $\|\cdot\|_1$ denotes the L1 norm.

The $S(f)$ represents the maximum change of the values of the output of function f. when input two neighboring datasets D and D'. For many types of numerical queries $S(f)$ are very small. In particular, the simple query such as the sum has the $S(f)$=1.

There exist typically two mechanisms to achieve ε-differential privacy which are *Laplace Mechanism* and *Exponential mechanism*. The *Laplace Mechanism* is the fundamental algorithm used in numerical function through adding *i.i.d.* noise into each output value, whereas the *Exponential mechanism* is used in non-numerical function. The noise follows *Laplace distribution* with the following probability density function:

$$Pr\left(x\right)=\frac{1}{2\lambda}e^{-\frac{|x|}{\lambda}} \tag{3}$$

We notate the above distribution as $Lap(\lambda)$, where λ denotes the scale. The perturbed value of $f(x)$ satisfies differential privacy when adding Laplace noise with $\lambda = \frac{S(f)}{\varepsilon}$, as $Lap\left(\frac{S(f)}{\varepsilon}\right)$.

Theorem 1 *Let $F_1, F_2, \cdots F_k$ be k algorithms, each of them satisfies ε_i-differential privacy ($i \in [1, k]$). Then the sequential composition $(F_1, F_2, \cdots F_k)$ satisfies $(\sum_{i=1}^{k} \varepsilon_i)$-differential privacy.*

The theorem 1 is a serial combination of ε-differential privacy and its proof can be seen in Appendix. The illustration of ε-differential privacy's serial combination is show in Figure 1:

2.2 Spatial decompositions

Spatial data query is a fundamental function in geographic information system. Suppose that we make a rectangular selection on a map layer to determine which tuples on this layer are completely contained in the rectangle [30]. Without spatial indexes, all the tuples should be retrieved to judge whether the tuple is located in the rectangle. The time complexity will be $O(n)$, where n is the number of tuples, which is not reasonable even unachievable in practice. If we make a grid index of the map layer, it is unnecessary to do the intersect of all the tuples and the rectangle, but only the tuples located in the grids exactly contained within the given rectangle need to be examined. The time complexity will be greatly reduced and the query efficiency will be greatly improved.

Spatial decomposition was usually classified into two categories, which are data-dependent decomposition and data-independent decomposition. The data dependent decomposition indicates that the partition of space is dependent on the input data. The most common data-dependent decomposition are KD-tree [31] and R-tree [32]. The KD-tree is generated by recursively splitting the spatial nodes via the lines passing through the median data value along the coordinate axis and the splitting axis changes each time cyclically. The data-independent decomposition indicates that the partition of the spatial nodes is independent on the input data. It is computed by splitting the space to two average parts on each coordinate. The best known is quadtree in two dimensions and 2^d-tree [33, 34] in higher dimensions. In our algorithm we use data-independent decomposition as the spatial decompositions.

Let D be a dataset consisting of data points in a d-dimensional space Ω. A spatial decomposition of D generates a 2^d-tree, which decomposes Ω into its sub-domains, along with

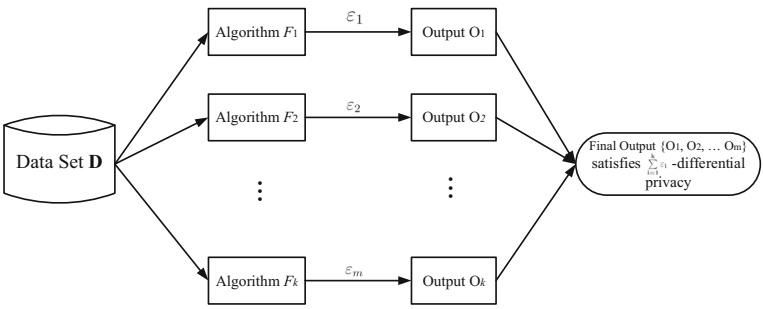


Figure 1 The serial combination of ε-differential privacy

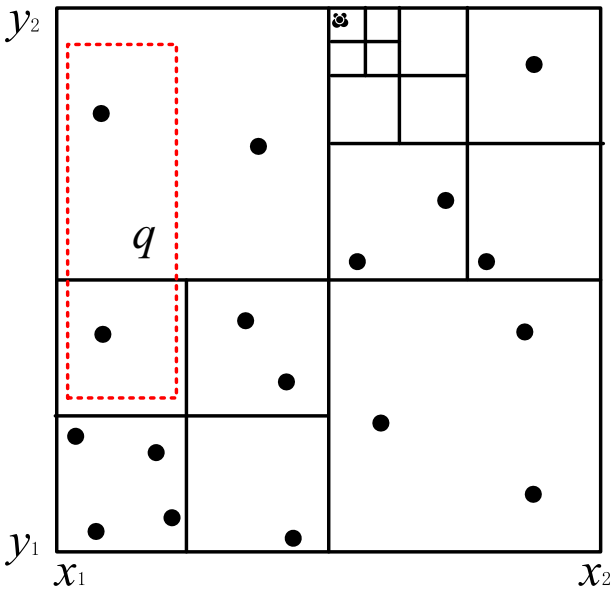


Figure 2 Spatial decompositions of a dataset D in 2-dimensional space Ω

partitioning of its data points into the leaves of the decomposition tree. In Figure 2, gives a 2- dimensional dataset D containing 22 points, it recursively divides into four regions until the number of its data points is less than a given threshold. Figure 3 is the generation of a 2^d-tree (quadtree with d=2) of D. The root of the quadtree v_1 corresponds to the region that covers the entire domain Ω, it has four child nodes v_2, v_3, v_4 and v_5 respectively corresponding to the four sub-domains of Ω. Each node of the quadtree consists of its corresponding domain and the number of data points contained in its region.

The 2^d-tree is widely used for querying the spacial region. For example, the quadtree in Figure 2 can be used to answer the query [35] of data points count of range as the rectangle q inclosed by dotted line in Figure 2, through up-down traversing the quadtree from the root. The query answer is initialized as $ans = 0$ at beginning of the traversal. As it traverses each node v of the quadtree, by examining v's region $v.dom$, four different cases may be encountered:

1. If $v.dom$ and q have no intersection, we skip the node v;

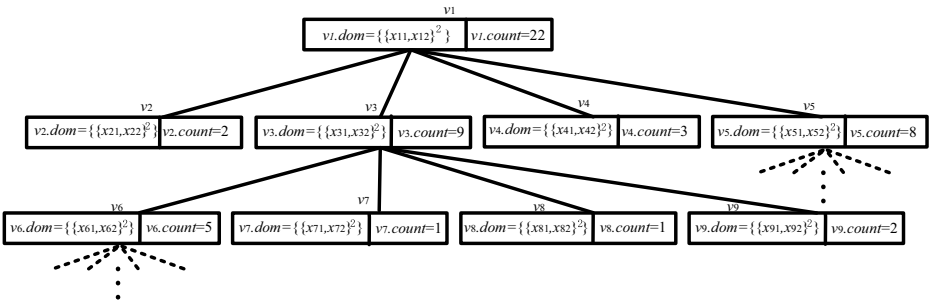


Figure 3 The generation of a quadtree of the dataset in Figure 1

2. If $v.dom$ is a subset of q, we add $v.count$ to ans;
3. If $v.dom$ and q have partial intersection and v is not a leaf node, the recursive queries on v'children are performed;
4. If $v.dom$ and q have partial intersection and v is a leaf node, we add the points count located in intersection region of $v.dom$ and q to ans.

When the traversal terminates, we return ans as the result.

3 Related work

In this section we first review private spatial decompositions algorithms, then discuss their disadvantages and the dilemmas.

Graham Cormode et al. first proposed the differentially private spatial decompositions through quadtree in 2012, as called DPSD. It takes the solution of publishing the perturbed version of the spatial decomposition quadtree, which adds Laplace noise to each node of quadtree to achieve ε-differential privacy. Algorithm 1 presents the main approach of DPSD. The input contains two thresholds h and θ, which respectively represent the maximum recursion depth and the decision value of split as discussed in 2.2, and $\overline{v_i.count}$ is the noisy version of the $v_i.count$.

Algorithm 1 DPSD(D,λ, θ,h).

initialize a quadtree T with a root node v_1;
set $v_1.dom = \Omega$, and mark v_1 as visited ;
while there exists an unvisited node v_i **do**
    mark v_i as visited ;
    computer the number $v_i.count$ of points in D that are contained in $v_i.dom$;
    computer the noisy version of $v_i.count$, $\overline{v_i.count} = v_i.count + Lap(\lambda)$;
    if $\overline{v_i.count} > \theta$ and $depth(v_i) < h - 1$ **then**
        split v_i, and add its children to T;
        mark the children of v_i as unvisited;
    end if
end whilereturn T;

Assume that D and D' are neighboring datasets. The quadtrees of D and D' have at most h nodes with different point count, and these h nodes form a path from the root to one leaf. That indicates that the sensitivity of quadtree is h, so the algorithm 1 satisfies ε-differential privacy if $\lambda \geq \frac{h}{\varepsilon}$. For privacy preserving data publishing, the quadtree T satisfies $n\varepsilon$-differential privacy by theorem 1, where n donates the nodes count of the quadtree, since each node of T adds noise Lap(λ). In algorithm 1 there is n Laplace noise added to the private tree, the noise cost of the private tree is $\sum_{i=1}^{n} |Lap(\lambda)|$. The main limitation of algorithm 1 is that the privacy cost depends on the recursive depth h.

Jun Zhang et al. in 2016 suggested that after constructing a quadtree T by algorithm 1, remove the noises of all intermediate nodes, and remain the noises of the leaves, which generates a modified private quadtree T'. When publishing the quadtree T', the points count of intermediate nodes can be obtained by the summing up of the count of the leaves under it. It was proved that the privacy cost of T' is as same as T's. To address this problem, Jun Zhang proposed PrivTree, as shown in algorithm 2. The biased count of a node

makes the privacy cost independent with h and also introduces a new parameter θ.The line 5 in algorithm 2 shows the method of computing the biased count of v_i, $b_i.count = \max\{\theta - \delta, v_i.count - depth(v_i) \cdot \delta\}$.

Algorithm 2 PrivTree(D,λ, θ, δ).

```
1: initialize a quadtree T with a root node v_1;
2: set v_1.dom = Ω, and mark v_1 as visited ;
3: while there exists an unvisited node v_i do
4:     mark v_i as visited ;
5:     computer a biased point count for v_1 with decaying factor θ, b_i.count =
       max {θ − δ, v_i.count − depth (v_i) · δ}
6:     computer the noisy version of b_i.count, b_i.count = b_i.count + Lap(λ);
7:     if b_i.count > θ then
8:         split v_i, and add its children to T;
9:         mark the children of v_i as unvisited;
10:    end if
11: end whilereturn T with all point counts removed;
```

Both of these two algorithms consider the split problem by the noisy version of each node of the private tree, which will lead to higher privacy cost of the private tree. In PrivTree, the choice of δ in PrivTree is a difficulty as $b_i.count$ at least equals $\theta - \delta$. If δ is very small, $b_i.count$ plus Lap(λ) may easily be larger than δ, which results in unnecessary splitting. It was proved that the setting of $\theta = \lambda \cdot ln(\beta)$ guarantees any node v_i with $b_i.count = \theta - \delta$ with the probability of $\frac{1}{2\beta}$ to be split, where β is the fanout of the private tree.

4 A low cost and un-cancelled laplace noise based differential privacy algorithm for spatial decompositions

In this section, we propose a low cost spatial decompositions algorithm based on un-cancelled Laplace noise InLN_DPSD, which is more secure by adding indefeasible Laplace noise to the private tree to avoid the Laplace noise to be canceled and has lower noise cost.

4.1 Low noise cost private tree for spatial decompositions

According to the above-mentioned analysis, we propose a low noise cost private tree based differential privacy algorithm.

We improved the algorithm 1 with a new parameter minimal split unit u instead of h, which is more intuitive for split termination, and whether a node should be split depends on its real point count $v_i.count$ instead of $\overline{v_i.count}$. We replace the $\overline{v_i.count} > \theta$ and $depth(v_i) < h - 1$ by $v_i.count > \theta$ and $\frac{\Omega}{2^{depth(v_i)*d}} > u$ of algorithm 1 in line 7, which indicates that if $v_i.dom$, $\frac{\Omega}{2^{depth(v_i)*d}}$ is smaller than u, even if $v_i.count > \theta$, stop spiltting v_i, where d is the dimensionality of Ω and $depth(v_i)$ denotes the depth of v_i in the private tree. The private tree T generated by the modified algorithm has all leaves with Laplace noise, while all intermediate nodes do not. For privacy preserving data publishing, we also get the point count of intermediate nodes by summing up the count of the leaves under it. To evaluate the performance of our improved algorithm, we make the same assumption that D and D' are neighboring datasets differing only by one element. There are at most $\max(depth(v_i))$ nodes

with different point count in quadtrees of D and D', and these nodes must form a path from the root to the one leaf. In particular, we let $\max(depth(v_i))$ equal h. Then,

$$ln\frac{Pr[D \to T]}{Pr[D' \to T]} = \sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + Lap(\lambda) > \theta]}{Pr[v_i'.count + Lap(\lambda) > \theta]}$$

$$+ln\frac{Pr[v_h.count + Lap(\lambda) = \overline{v_h.count}]}{Pr[v_h'.count + Lap(\lambda) = \overline{v_h.count}]}$$

Although the first h-1 intermediate nodes do not add Laplace noise, the D and D' can also approximately output a same T satisfying ε-differential privacy if $\lambda > \frac{h}{\varepsilon}$. Every node v_i for any $i \in [1, h-1]$ is the ancestor of v_h and $v_h.dom \subset v_i.dom$, that means $v_i.count$ has the same Laplace noise with the leaf's.

For any $i \in (1, h-1)$ with $v_i.count \leq \theta$, $ln\frac{Pr[v_i.count+Lap(\lambda)>\theta]}{Pr[v_i'.count+Lap(\lambda)>\theta]}$ equals $\frac{1}{\lambda}$. Otherwise it is less than $\frac{1}{\lambda}$, which has been proved by Jun Zhang et al. in [27]. So,

$$\sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + Lap(\lambda) > \theta]}{Pr[v_i'.count + Lap(\lambda) > \theta]} \leq \frac{h-1}{\lambda} \tag{4}$$

$$ln\frac{Pr[v_h.count + Lap(\lambda) = \overline{v_h.count}]}{Pr[v_h'.count + Lap(\lambda) = \overline{v_h.count}]} = \frac{1}{\lambda} \tag{5}$$

$$ln\frac{Pr[D \to T]}{Pr[D' \to T]} \leq \frac{h}{\lambda} = \varepsilon \tag{6}$$

Assume that there are n points in T, m is the number of intermediate nodes, t is the number of leaf nodes, and β is the fanout of T and these parameters satisfy (7). The total noise added to this private tree T is $\sum_{i=1}^{t} |Lap(\lambda)|$. It then can be seen that the noise cost of modified algorithm is smaller than the private tree generated by algorithm 1, which is $\sum_{i=1}^{n} |Lap(\lambda)|$.

$$\begin{cases} m + t = n; \\ \beta * m + 1 = n; \end{cases} \tag{7}$$

4.2 Spatial decompositions based on indefeasible Laplace noise

As discussed in 4.1, we proposed only adding Laplace to leaves but not to the intermediate nodes to reduce the total privacy cost of the private tree. However due to the fragility of differential privacy. The noise of intermediate node was generated by the sum of its descendant leaves' Laplace noises. The noise of an intermediate node which has β leaves equals $\sum_{i=1}^{k} Lap(\lambda)$ which may be 0 with a very high probability as Laplace is symmetric about the origin. As a result the intermediate node's point count approximately to be real.

Like wise we make the same assumption as the modified algorithms. Extremely, we assume a private tree T is a full β-tree and each leaf has $\beta - 1$ brother leaves. Let $L = \{l_1, l_2, \cdots, l_\beta\}$ be the set of leaves of one intermediate node, $|L| = \beta$. Each time we randomly select one leaf l_i from the set L and add Laplace noise $\frac{i+1+\beta}{i+\beta} Lap(\lambda)$ into its point count, then update $L = L \setminus \{l_i\}$. Repeat the above operation until $L = \Phi$. The noise of intermediate

node which has β leaves equals $\sum_{i=1}^{k} \frac{i+1+\beta}{i+\beta} Lap(\lambda) \neq 0$, therefore the noise of intermediate node will not be cancelled out.

$$\overline{v_i.count} = v_i.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) \tag{8}$$

Lemma 1

$$ln\frac{Pr[D \to T]}{Pr[D' \to T]} = \sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) > \theta]}{Pr[v_i'.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) > \theta]}$$
$$+ln\frac{Pr[v_h.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) = \overline{v_h.count}]}{Pr[v_h'.count + \frac{i+1+\beta}{i+\beta} Lap(\lambda) = \overline{v_h.count}]} < \frac{h}{\lambda}$$

The proof of Lemma 1 can be seen in Appendix.

The Laplace noise is proportional to $S(f)$ and inversely proportional to ε. While $S(f)$ and ε are fixed values, increasing the noise with proportion of $\frac{i+1+\beta}{i+\beta}$ can also satisfy the ε-differential privacy.

Lemma 2 *The total noise cost of the private tree generated by improved algorithm is* $\frac{t}{\beta} \sum_{i=1}^{\beta} |\frac{i+1+\beta}{i+\beta} Lap(\lambda)|$, *smaller than the noise cost of private tree generated by algorithm 1,which is* $\sum_{i=1}^{n} |Lap(\lambda)|$.

The proof of Lemma 2 can be seen in Appendix.

4.3 The general private β-tree with modified indefeasible Laplace noise

In practice, the spatial decomposition tree is not a full β-tree, but a general β-tree which each intermediate node has the same fanout β. We modify our strategy by adding noise into the ith selected leaf child with $\frac{(\beta-i+1)+1+\beta}{(\beta-i+1)+\beta} Lap(\lambda)$, causing its first selected leaf child with noise of $\frac{\beta+1+\beta}{\beta+\beta} Lap(\lambda)$.

It can be computed that $m = (t-1)/(\beta-1)$. We define the function $f(k) = \frac{k+\beta+1}{k+\beta}$, for any $k \in [1, \beta]$. $f(k)$ is a monotonically decreasing function, that is $f(i) > f(i+1)$ for any $i \in [1, \beta-1]$.

Lemma 3 *Assume that there are* x_k *leaves which are kth selected and add noises with* $f(k)Lap(\lambda)$ *in the* β*-tree, for any* $k \in [1, \beta]$. *The total noise cost of the private tree* $\sum_{k=1}^{\beta} |x_k f(k) Lap(\lambda)| = (x_1 f(1) + \cdots x_\beta f(\beta))|Lap(\lambda)|$ *(and* $\sum_{k=1}^{\beta} x_k = t$*) is smaller than the noise cost of the full* β*-tree, which has the same leaves count and intermediate nodes count with the general* β*-tree. Besides, the noise cost of full* β*-tree can be symbolically simplified as* $\frac{t}{\beta} \sum_{k=1}^{\beta} |f(k)Lap(\lambda)|$.

The proof of Lemma 3 can be seen in Appendix.

4.4 Pruning redundant branches of the private tree

There is a situation in which a local area gathers much more points, whose count is larger than θ, while the other region has minimal points. It also exists such a situation in split, where one region's points count of one split equals its parent's points count while the other $2^d - 1$ regions' points count equals 0. It is obvious that the nodes with 0 points count are leaves, which will be added with Laplace noise. The total noises costs of the private tree increases for the 0 points count leaves, while those nodes are not necessary to be preserved. We take the solution of pruning the redundant branches, i.e.,if a non-leaf node v_{child}'s points count equals its parent v_{parent}'s points count, we replace v_{parent} with v_{child}.

The pruning process only occurs in the intermediate nodes, otherwise it will make the fanout less than β, violating the definition of private tree.

4.5 The spatial decomposition algorithm InLN_DPSD

Our technique for private spatial decompositions is presented in Algorithm 2. We first generate a β-tree applying low noise cost private tree for spatial decompositions(line 1∼10). We split the node according to the real point count $v_i.count$ and the algorithm terminates depending on the minimum unit u, which is more intuitive than the maximal depth of recursion h(line 6).

Then we pruned the redundant branches of our generated private tree(line 11∼20) that the 0 count node would not be added Laplace noise. Next, add noise to the leaves of the β-tree (line 20∼34), which is the biggest innovation of InLN_DPSD. We first give the notion of indefeasible Laplace noise, which multiplies Laplace noise with coefficient $\frac{(\beta-i+1)+\beta+1}{(\beta-i+1)+\beta}$, where i indicates the ith leaf randomly selected from the intermediate node's children, and β is the fanout of the private tree.

In the process of adding indefeasible noise, we check all nodes of the β-tree. If the node is not a leaf, add noise to its children who are leaves. An intermediate node's child leaves are added to Laplace noise multiplied with coefficient $\frac{k+\beta+1}{k+\beta}$, where k decreases from β to 1. Usually k may be greater than 1 for not all the β children are leaves, and k only is decreased by 1 when a child leaf is processed(line 27∼29). It has been proved in Section 4.2 that when the private tree was added to Laplace noise multiplied $\frac{k+\beta+1}{k+\beta}$ where k decreases from β, its total noise cost is less than the DPSD algorithm. Otherwise if the k increases from 1 to β, the noise cost is larger than existing algorithms'.

We analyzed the computation complexity our proposed algorithm from space complexity and time complexity. In terms of spatial complexity $S(n)$, because our algorithm only adds Laplace to the leaves of the private tree, in extreme cases, the number of leaves is the size of data set n, so the space complexity of Laplace noise is $S(n)$. While other existing algorithms' added Laplace noises to the whole private tree, whose space complexity is much more than our algorithm. In the terms of time complexity, first we split the whole regain Ω to generate a private tree and as its time complexity is $n * log_{\beta}^{n}$. The time of adding Laplace noise to leaves is n. So the total time complexity is $O(n * log_{\beta}^{n})$, which is as same as the existing algorithms' time complexity. Through the above analysis, our algorithm's computation complexity is less than the existing algorithms, and the performance of algorithm is more better.

Algorithm 3 InLN_DPSD(D, λ, θ, u).

initialize a quadtree T with a root node v_1;
set $v_1.dom = \Omega$, and mark v_1 as visited ;
while there exists an unvisited node v_i **do**
    mark v_i as visited ;
    computer the number $v_i.count$ of points in D that are contained in $v_i.dom$;
    if $v_i.count > \theta$ and $\frac{\Omega}{2^{depth(v_i)*d}} > u$ **then**
        split v_i, and add its children to T;
        mark the children of v_i as unvisited;
    end if
end while
for each $i \in [1, n]$ **do**
    for each $j \in [1, \beta]$ **do**
        $v_j = v_i \rightarrow child[j]$;
        if $v_j.count == v_i.count$ **then**
            $v_j.parent = v_i.parent$;
            free v_j's brothers;
            break;
        end if
    end for
end for
for each $i \in [1, n]$ **do**
    if $isleaf(v_i) == 0$ **then**
        initialize a children set $L = \phi$;
        add all v_i's children to L;
        for each $j \in [1, \beta]$, $k = \beta$ **do**
            randomly select v_j from L;
            if $isleaf(v_j) == 1$ **then**
                $\overline{v_j.count} = v_j.count + \frac{k+\beta+1}{k+\beta} * Lap(\lambda)$;
                $k--$;
            end if
            $L = L \setminus v_j$;
        end for
    end if
end for
return T;

5 The optimal laplace noise for differential privacy $Lap_{min}(\lambda)$

In this section we propose the conception of $Lap_{min}(\lambda)$ whose absolute value is no bigger than the query function's global sensitivity $S(f)$. Adding noise $Lap_{min}(\lambda)$ instead of unlimited Laplace noise to query answers both satisfies ε-differential privacy and optimal data utility.

5.1 The $Lap_{min}(\lambda)$ satisfy ε-differential privacy

we define that $Lap_{min}(\lambda)$ satisfying Laplace distribution is the minimal Laplace added to the query answer and $|Lap_{min}(\lambda)| \leq S(f)$.

$$Lap_{min}(\lambda) = \frac{1}{2\lambda} e^{\frac{|x|}{\lambda}}, \quad |x| \leq S(f) \tag{9}$$

Assume that D and D' are neighbouring databases, $f(D)$ and $f(D')$ are respectively the real query answer of D and D'. Without loss of generality we assume that $f(D) > f(D')$, then,

$$\begin{aligned}
&\frac{Pr\left[M\left(D\right) = O\right]}{Pr\left[M\left(D'\right) = O\right]} \\
&= \frac{Pr\left[f\left(D\right) + Lap_{min}(\lambda) = O\right]}{Pr\left[f\left(D'\right) + Lap'_{min}(\lambda) = O\right]} \\
&= \frac{Pr\left[Lap_{min}(\lambda) = O - f\left(D\right)\right]}{Pr\left[Lap'_{min}(\lambda) = O - f\left(D'\right)\right]} \\
&= \frac{\frac{1}{2\lambda} e^{\frac{-|O-f(D)|}{\lambda}}}{\frac{1}{2\lambda} e^{\frac{-|O-f(D')|}{\lambda}}} \\
&= e^{-\frac{|O-f(D)|-|O-f\left(D'\right)|}{\lambda}} \\
&\leq e^{\frac{|f(D)-f\left(D'\right)|}{\lambda}}
\end{aligned}$$

$$\begin{aligned}
&According\ DEFINITION\ 3, S(f) = |f(D) - f(D')|_{max}, \\
&so\ |f(D) - f(D')| \leq S(f).\ Substitute\ \lambda = S(f)/\varepsilon\ into \\
&the\ above\ formula. \\
&\leq e^{\frac{S(f)}{\frac{S(f)}{\varepsilon}}} \\
&\leq e^{\varepsilon}
\end{aligned}$$

In the special case, assume that $\varepsilon = 0$ and $f(D) - f(D') = S(f)$, that means $f(D) + Lap_{min}(\lambda) = f(D') + Lap'_{min}(\lambda)$ and $Lap'_{min}(\lambda) - Lap_{min}(\lambda) = f(D) - f(D') = S(f)$. As we proposed that the absolute value of Laplace noise added to query answers is no more than S(f), so the $Lap_{min}(D)$ must be less than 0 and the $Lap'_{min}(D)$ must be great than 0 that satisfies the ε-differential privacy.

We take $Lap(\lambda)$ as unlimited Laplace noise, so $Lap(\lambda) = Lap_{min}(\lambda) + Z * S(f)$ where Z is integer. Then,

$$\begin{aligned}
&\frac{Pr\left[f\left(D\right) + Lap(\lambda) = O\right]}{Pr\left[f\left(D'\right) + Lap'(\lambda) = O\right]} \leq e^{\varepsilon} \\
&\Leftrightarrow \frac{Pr\left[f\left(D\right) + Lap_{min}(\lambda) + Z * S(f) = O\right]}{Pr\left[f\left(D'\right) + Lap'_{min}(\lambda) + Z * S(f) = O\right]} \leq e^{\varepsilon} \\
&\Leftrightarrow \frac{Pr\left[f\left(D\right) + Lap_{min}(\lambda) = O - Z * S(f)\right]}{Pr\left[f\left(D'\right) + Lap'_{min}(\lambda) = O - Z * S(f)\right]} \leq e^{\varepsilon} \\
&\Leftrightarrow \frac{Pr\left[f\left(D\right) + Lap_{min}(\lambda) = O'\right]}{Pr\left[f\left(D'\right) + Lap'_{min}(\lambda) = O'\right]} \leq e^{\varepsilon}
\end{aligned}$$

In the above derivation, we let the $O' = O - Z * S(f)$ take place O. As we discussed above, if $f(D) \geq f(D')$ and $0 \leq Lap(\lambda) \leq Lap'(\lambda)$,

$Lap_{min}(\lambda) \leq 0$ and $Lap'_{min}(\lambda) \geq 0$, $(Z-1) * S(f) \leq Lap(\lambda) \leq Z * S(f)$ and $Z * S(f) \leq Lap'(\lambda) \leq (Z+1) * S(f)$, so $Z = Lap'(\lambda)/S(f)$. Else if $f(D) \geq f(D')$ and $Lap(\lambda) \leq Lap'(\lambda) \leq 0$, $Z = Lap(\lambda)/S(f)$.

Otherwise, if $f(D) \leq f(D')$ and $0 \leq Lap(\lambda) \leq Lap'(\lambda)$, $Z = Lap(\lambda)/S(f)$. Else if $f(D) \leq f(D')$ and $Lap(\lambda) \leq Lap'(\lambda) \leq 0$, $Z = Lap'(\lambda)/S(f)$. It was proved that reducing the unlimited Laplace noise $Lap(\lambda)$ to $Lap_{min}(\lambda)$ by $Lap_{min}(\lambda) = Lap(\lambda) - Z * S(f)$ can also satisfy the ε-differential privacy.

5.2 The analysis of data utility

The trade-off between privacy and data utility has been attracting high attention of researchers. Under the premise of ε-differential privacy, we hope that the data utility is as high as possible. We take the relative error RE [36] as the evaluation standard of data utility.

$$RE = \frac{|M(D) - f(D)|}{f(D)} = \frac{|Lap(\lambda)|}{f(D)} \tag{10}$$

Where $M(D)$ denotes the perturbed query answer of D and $f(D)$ denotes the real query answer of D.

As discussed above, if $Lap(\lambda) > 0$, so $Z \geq 0$, $0 \leq Lap_{min}(\lambda) = Lap(\lambda) - Z * S(f) \leq Lap(\lambda)$, $\frac{|Lap_{min}(\lambda)|}{f(D)} \leq \frac{|Lap(\lambda)|}{f(D)}$. Otherwise if $Lap(\lambda) < 0$, so $Z \leq 0$, $Lap(\lambda) \leq Lap(\lambda) - Z * S(f) = Lap_{min}(\lambda) \leq 0$, $\frac{|Lap_{min}(\lambda)|}{f(D)} \leq \frac{|Lap(\lambda)|}{f(D)}$.

From the analysis we can see that adding Laplace noise $Lap_{min}(\lambda)$ satisfying the ε-differential to query answers make the RE lower than the unlimited Laplace noise. Maintaining the previous privacy levels, the data utility is highly raised obviously.

5.3 The generation of Laplace noise $Lap_{min}(\lambda)$

The Laplace distribution function is as follow:

$$F(x) = \begin{cases} \frac{1}{2}e^{\frac{x}{\lambda}}, & x < 0 \\ 1 - \frac{1}{2}e^{\frac{-x}{\lambda}}, & x \geq 0 \end{cases} \tag{11}$$

As $F(x) \in [0, 1]$, we take random values of $F(x)$ from 0 to 1 to get unlimited Laplace noise. While it was declared that $|Lap_{min}(\lambda)| \leq S(f)$,

$F(-S(f)) = \frac{1}{2}e^{\frac{-S(f)}{\lambda}} = \frac{1}{2}e^{\frac{-S(f)}{\frac{S(f)}{\varepsilon}}} = \frac{1}{2}e^{-\varepsilon}$ and $F(S(f)) = 1 - \frac{1}{2}e^{-\varepsilon}$. $\forall \xi \sim Uni(\frac{1}{2}e^{-\varepsilon}, 1 - \frac{1}{2}e^{-\varepsilon})$, ξ is random valure in $[\frac{1}{2}e^{-\varepsilon}, 1 - \frac{1}{2}e^{-\varepsilon}]$. Let ξ take place $F(x)$, then,

$$x = \begin{cases} \lambda ln(2\xi) & \frac{1}{2}e^{-\varepsilon} \leq \xi < 0.5 \\ -\lambda ln(2(1-\xi)), & 0.5 \leq \xi \leq 1 - \frac{1}{2}e^{-\varepsilon} \end{cases} \tag{12}$$

By observing the piecewise function of x, we take $\xi' = \xi - 0.5$, $\xi' \sim Uni(-0.5 + \frac{1}{2}e^{-\varepsilon}, 0.5 - \frac{1}{2}e^{-\varepsilon})$. Put the $\xi = \xi' + 0.5$ into function (6),then,

$$x = \begin{cases} \lambda ln(1 + 2\xi') & -0.5 + \frac{1}{2}e^{-\varepsilon} \leq \xi' < 0 \\ -\lambda ln(1 - 2\xi'), & 0 \leq \xi' \leq 0.5 - \frac{1}{2}e^{-\varepsilon} \end{cases} \tag{13}$$

$x = -\lambda sign(\xi') ln(1 - abs(2\xi'))$

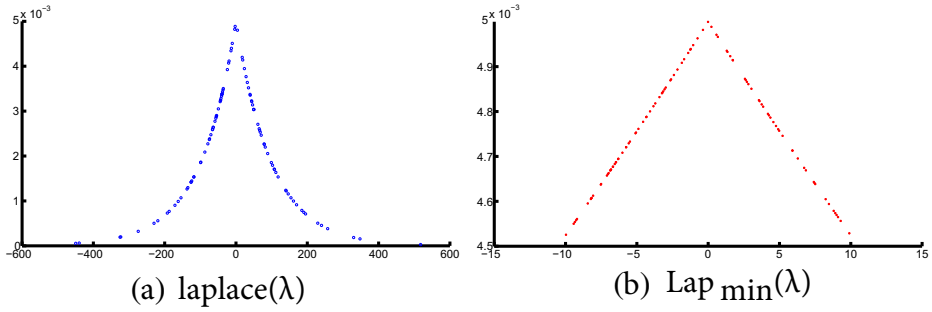


Figure 4 The distribution of 100 random Laplace noises and $Lap_{min}(\lambda)$ noises with $S(f) = 10$ and $\varepsilon = 0.1$

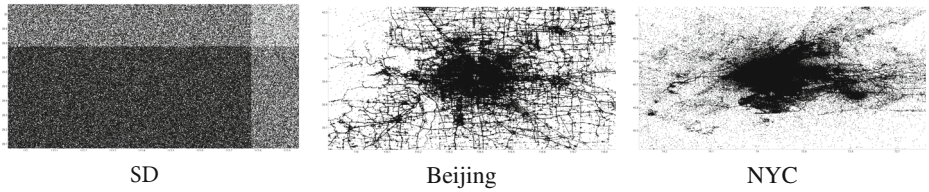


Figure 5 The distribution of each dataset

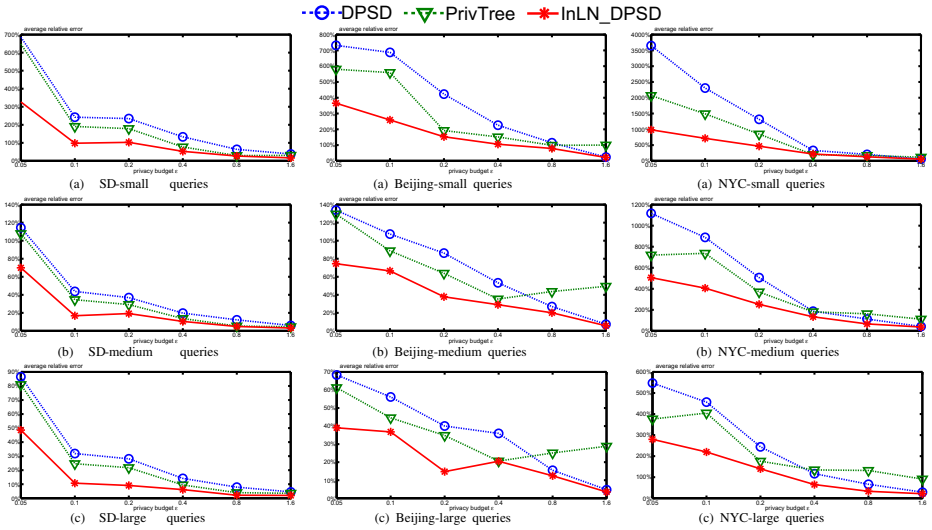


Figure 6 The results of average relative error of range count queries on each dataset

The $Lap_{min}(\lambda)$ can be gotten through $Matlab$ langeuage as follows:
$x = unifrnd(-0.5 + 1/2 * exp(-\varepsilon), 0.5 - 1/2 * exp(\varepsilon), n);$
$y = -(S(f)/\varepsilon)sign(x). * log(1 - 2 * abs(x))$

The first line generate $n * n$ random values x ranged of $[-0.5 + \frac{1}{2}e^{-\varepsilon}, 0.5 - \frac{1}{2}e^{-\varepsilon}]$, the second line generate $n * n$ Laplace noise y ranged of $[-S(f), S(f)]$, which is one-to-one corresponded to x.

Figure 4 shows the distribution of 100 random unlimited Laplace noises and $Lap_{min}(\lambda)$ noises with $S(f) = 10$ and $\varepsilon = 0.1$, i.e.,$\lambda = 100$. From Figure 4a we can see that the value of $|Laplace(\lambda)|$ may be very large, e.g. -441, adding it to query answer may lead to very large RE, the outcome is meaningless. However, $Lap_{min}(\lambda)$ will not be the case.

6 Experiment

In this section, we conducted differential privay spatial decompositions experiments on three publicly available datasets to evaluate our algorithm against the state-of-the-art algorithms of spatial decompositions based on differential privacy.

6.1 Competing methods and testing datasets

Competing Methods To evaluate the efficacy of the proposed approaches, we compare InLN_DPSD with the DPSD and PrivTree. Besides, we execute the DPSD with unlimited Laplace noise and $Lap_{min}(\lambda)$ respectively to evaluate our approach.

Testing Datasets In the experiments we employ one synthetic two-dimensional dataset and two real spatial datasets. SD contains 1 million location information; Beijing[1] is two-dimensional real dataset which contains 15 million records of pickup locations of Beijing taxis; NYC[2], a four-dimensional real dataset containing one hundred million records of pickup and drop-off locations of NYC taxis in 2013. The distribution of these datasets is shown in Figure 5.

6.2 Evaluation measures

We run each algorithm on every dataset to evaluate their performances. We respectively generate ten thousand queries on the region covering [0.1%, 1%), [1%, 10%), [10%, 100%) of the dataset domain and get their answers with each algorithm. For evaluating the query accuracy, we define the relative error RE [26, 36] as the measure accuracy of a perturbed answer $\overline{q(D)}$ to a query q by its real answer $q(D)$.

$$RE(\overline{q(D)}) = \frac{|\overline{q(D)} - q(D)|}{\max\{q(D), \Delta\}}$$

[1] http://research.microsoft.com/apps/pubs/?id=152883
[2] http://publish.illinois.edu/dbwork/open-data/

Table 1 The number of Laplace noises added to the private tree

	DPSD	PrivTree	InLN_DPSD
SD	736449	755261	227677
Beijing	405269	371705	145216
NYC	130209	109041	63527

where Δ is a smoothing factor [37] set to 1% of the dataset cardinality n. We repeat each experiment 1000 times and report the average relative error of each method for each query set.

6.3 Results and analyses

Figure 6 shows the average relative error of the three different range queries on each dataset applying every algorithm with different privacy budget ε. From the results we can see:

(1) The relative error of InLN_DPSD algorithm is always smaller than the other two algorithms. That shows our algorithm outperforms.
(2) The relative error of NYC dataset is much larger than SD and Beijing datasets. The more skewed the dataset is, the larger relative error it has.
(3) The relative error increases as privacy budget ε increases. Since the noise added to dataset is inversely proportional to ε.
(4) The relative error is inversely proportional to the query area. That indicates the larger the query area is, the more accurate the answer is. If the query area is much smaller, even to one tuple's location, the relative error will be very large, which preserves the individual's privacy.

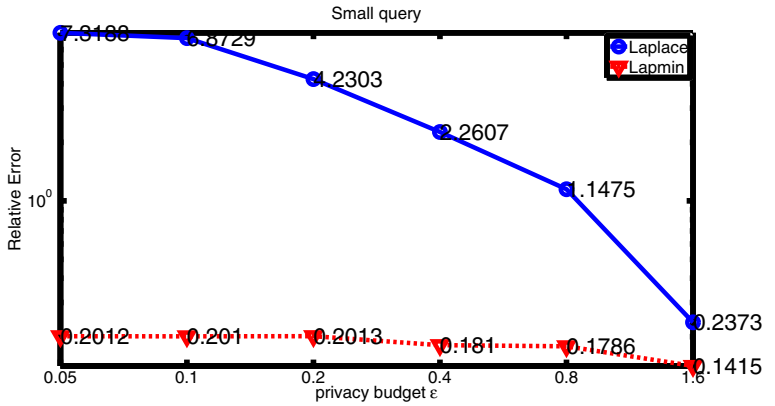
Table 1 shows the number of noises added to the private tree in different datasets when applying these three algorithms. From the table we can see that the InLN_DPSD algorithm adds fewer noises to the private tree.

Figure 7 shows the average relative error of Beijing queries with limited and unlimited Laplace noise. From the experimental results we can see that:

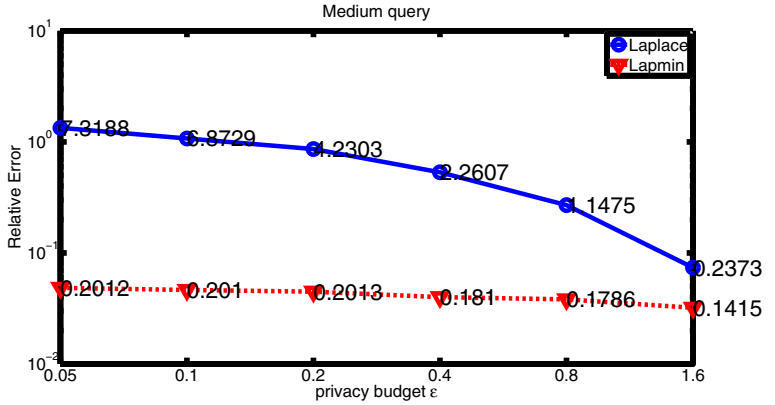
(i) The average RE of small queries, medium queries and large queries added by $Lap_{min}(\lambda)$ are all far smaller than the unlimited Laplace noise.
(ii) The relative errors added by $Lap_{min}(\lambda)$ in different privacy budget ε vary slight relative to unlimited Laplace noise as the limited Laplace noise $Lap_{min}(\lambda)$ is bounded by sensitivity $S(f)$.
(iii) Differential private spatial decompositions with $Lap_{min}(\lambda)$ can greatly reduce the cost of privacy while not increase the computational complexity.

7 Conclusion

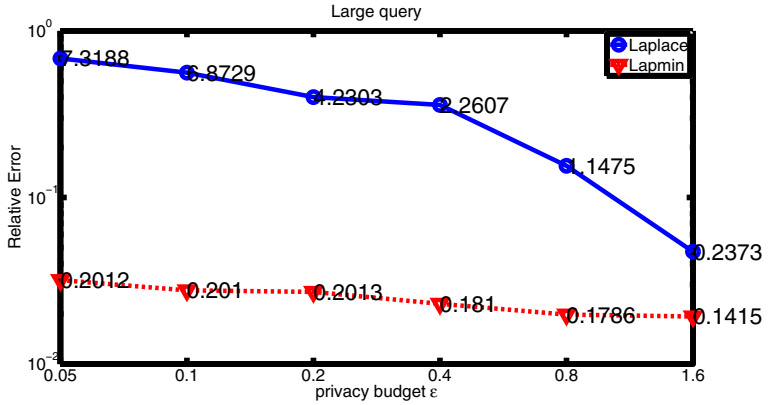
In this paper, we study the problem of spatial decompositions based differential privacy. The traditional differential privacy is delicate as the Laplace distribution is symmetric about



(a) small query



(b) medium query



(c) large query

Figure 7 Different Beijing queries with limited and unlimited Laplace noise

the origin, the sum of several Laplace noises may be 0 so that the added noises may be canceled and the privacy may be compromised. We take the strategy of only adding indefeasible Laplace noises to leaves, the noises will not be canceled and the domain splitting will be more accurate which leads to higher data utility and the total noise cost of the private tree is highly decreased. Based on this strategy we propose InLN_DPSD, a more secure spatial decompositions algorithm via indefeasible Laplace noise in differential privacy. we also propose that the minimal Laplace noise $Lap_{min}(\lambda) \leq S(f)$ is optimal for ε-differential privacy, which ensures optimal trade-off between data privacy and data utility. The experiment results of the boolean dataset Exam Result and location dataset Beijing, i.e., the data utility is highly raised by reducing added noise under the premise of ε-differential privacy. The experiment results show that InLN_DPSD outperforms the DPSD and PrivTree both in a synthetic dataset and real dataset.

Acknowledgments The authors would like to thank Ping Huang for helping revise the paper over and over again. This work was supported by the National Natural Science Foundation of China under grants No. 61821003 and No.61902135, the National Key Research and Development Program of China under grant No. 2016YFB0800402, ARC Discovery Early Career Researcher Award (DE160100308) and ARC Discovery Project (DP170103954;DP190101985).

Appendix

Proof of Theorem 1

Let $F_1, F_2, \cdots F_k$ be k algorithms, each of them satisfies ε_i-differential privacy ($i \in [1, k]$). Then the sequential composition $(F_1, F_2, \cdots F_k)$ satisfies $(\sum_{i=1}^{k} \varepsilon_i)$-differential privacy.

Proof The theorem is a serial combination of ε-differential privacy. The algorithm $F_i(D)$ satisfies ε_i-differential privacy, so:

$$\forall O_i \in Range(F_i),\ Pr[F_i(D) = O_i] \leq e^{\varepsilon_i} * Pr[F_i(D') = O_i]$$

Assume that $F(D)$ is the serial combination of all the $F_i(D)$, and its output is $\{O_1, O_2, ..., O_k\}$. As any two $F_i(D)$ are independent, so:

$$\forall O = \{O_1, O_2, ..., O_k\} \in Range(F),\ Pr[F(D) = O] = \prod_{i=1}^{k} Pr[F_i(D) = O_i]$$

$$Pr[F(D) = O] = \prod_{i=1}^{k} Pr[F_i(D) = O_i] \leq \prod_{i=1}^{k} (e^{\varepsilon_i} \times Pr[F_i(D') = O_i])$$

$$= e^{\sum_{i=1}^{k} \varepsilon_i} \times \prod_{i=1}^{k} Pr[F_i(D') = O_i] = e^{\sum_{i=1}^{k} \varepsilon_i} \times Pr[F(D') = O]$$

We can see that $Pr[F(D) = O] = e^{\sum_{i=1}^{k} \varepsilon_i} \times Pr[F(D') = O]$, so the $F_i(D)$'s serial combination $F(D)$ satisfies $\sum_{i=1}^{k} \varepsilon_i$-differential privacy. □

Proof of Lemma 1

$$ln\frac{Pr[D \to T]}{Pr[D' \to T]} = \sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) > \theta]}{Pr[v_i'.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) > \theta]}$$
$$+ln\frac{Pr[v_h.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) = \overline{v_h.count}]}{Pr[v_h'.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) = \overline{v_h.count}]} < \frac{h}{\lambda}$$

Proof

$$\sum_{i=1}^{h-1} ln\frac{Pr[v_i.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) > \theta]}{Pr[v_i'.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) > \theta]}$$
$$= \sum_{i=1}^{h-1} ln\frac{Pr[Lap(\lambda) < (v_i.count - \theta)\frac{i+1+\beta}{i+\beta}]}{Pr[Lap(\lambda) < (v_i'.count - \theta)\frac{i+1+\beta}{i+\beta}]}$$

For Laplace distribution function is:

$$F(x) = \begin{cases} \frac{1}{2}e^{\frac{x}{\lambda}} & x < 0 \\ 1 - \frac{1}{2}e^{-\frac{x}{\lambda}} & x \geq 0 \end{cases}$$

$if\ v_i.count < \theta, F(x) = \frac{1}{2}e^{\frac{x}{\lambda}}, then$:

$$\sum_{i=1}^{h-1} ln\frac{Pr[Lap(\lambda) < (v_i.count - \theta)\frac{i+1+\beta}{i+\beta}]}{Pr[Lap(\lambda) < (v_i'.count - \theta)\frac{i+1+\beta}{i+\beta}]}$$
$$= \sum_{i=1}^{h-1} ln(e^{\frac{1}{\lambda}\cdot\frac{i+1+\beta}{i+\beta}[(v_i.count-\theta)-(v_i'.count-\theta)])})$$
$$= \sum_{i=1}^{h-1} \frac{1}{\lambda}\cdot\frac{i+1+\beta}{i+\beta} < \sum_{i=1}^{h-1}\frac{1}{\lambda} = \frac{h-1}{\lambda}$$

$else\ if\ v_i.count \geq \theta, F(x) = 1 - \frac{1}{2}e^{-\frac{x}{\lambda}}$,and in [27] it has proved that:

$$ln\frac{Pr[Lap(\lambda) < (v_i.count - \theta)]}{Pr[Lap(\lambda) < (v_i'.count - \theta)]} \leq \frac{1}{\lambda}$$

$$\sum_{i=1}^{h-1} ln\frac{Pr[Lap(\lambda) < (v_i.count - \theta)\frac{i+1+\beta}{i+\beta}]}{Pr[Lap(\lambda) < (v_i'.count - \theta)\frac{i+1+\beta}{i+\beta}]}$$
$$\leq \sum_{i=1}^{h-1}\frac{1}{\lambda} = \frac{h-1}{\lambda}$$

$$
\begin{aligned}
&ln\frac{Pr[v_h.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) = \overline{v_h.count}]}{Pr[v'_h.count + \frac{i+1+\beta}{i+\beta}Lap(\lambda) = \overline{v_h.count}]} \\
&= ln\frac{Pr[\frac{i+1+\beta}{i+\beta}Lap(\lambda) = \overline{v_h.count} - v_h.count]}{Pr[\frac{i+1+\beta}{i+\beta}Lap(\lambda) = \overline{v_h.count} - v'_h.count]} \\
&= ln\frac{Pr[Lap(\lambda) = (\overline{v_h.count} - v_h.count)\cdot\frac{i+\beta}{i+1+\beta}]}{Pr[Lap(\lambda) = (\overline{v_h.count} - v'_h.count)\cdot\frac{i+\beta}{i+1+\beta}]} \\
&= ln\, e^{\frac{i+\beta}{i+1+\beta}\cdot\frac{|(\overline{v_h.count}-v_h.count)|-|(\overline{v_h.count}-v'_h.count)|}{\lambda}} \\
&\leq \frac{i+\beta}{i+1+\beta}\cdot\frac{1}{\lambda} \\
&< \frac{1}{\lambda}
\end{aligned}
$$

□

Proof of Lemma 2

The total noises cost of the private tree generated by improved algorithm is $\frac{t}{\beta}\cdot\sum_{i=1}^{\beta}|\frac{i+1+\beta}{i+\beta}Lap(\lambda)|$, it is smaller than the noises cost of private tree generated by algorithm 1,which is $\sum_{i=1}^{n}|Lap(\lambda)|$.

Proof The proof of $\frac{t}{\beta}\cdot\sum_{i=1}^{\beta}|\frac{i+1+\beta}{i+\beta}Lap(\lambda)| < \sum_{i=1}^{n}|Lap(\lambda)|$ is equivalent to the proof of $\frac{t}{\beta}\cdot\sum_{i=1}^{\beta}\frac{i+1+\beta}{i+\beta} < n$.

$$
\begin{aligned}
&\frac{t}{\beta}\cdot\sum_{i=1}^{\beta}\frac{i+1+\beta}{i+\beta} = \frac{t}{\beta}\cdot\sum_{i=1}^{\beta}(1+\frac{1}{i+\beta}) \\
&= \frac{t}{\beta}(\beta + \sum_{i=1}^{\beta}(\frac{1}{i+\beta})) = t + \frac{t}{\beta}\sum_{i=1}^{\beta}(\frac{1}{i+\beta}) \\
&< t + \frac{t}{\beta}\sum_{i=1}^{\beta}(\frac{1}{\beta}) = t + \frac{t}{\beta} \\
&< t + \frac{t-1}{\beta-1} = t + m = n
\end{aligned}
$$

□

Proof of Lemma 3

Assume that there are x_k leaves which are kth selected and add noises with $f(k)Lap(\lambda)$ in the β-*tree, for any* $k \in [1, \beta]$. The total noises cost of the private tree $\sum_{k=1}^{\beta}|x_k f(k)Lap(\lambda)|$

$=(x_1 f(1)+\cdots+x_\beta f(\beta))|Lap(\lambda)|$ (and $\sum_{k=1}^{\beta} x_k = t$) is smaller than the noises cost of the full β-tree, which have the same leaves count and intermediate nodes count with the general β-tree. Besides, the noises cost of full β-tree can be symbolic simplified $\frac{t}{\beta}\sum_{k=1}^{\beta}|f(k)Lap(\lambda)|$.

Proof The proof of $\sum_{k=1}^{\beta}|x_k f(k)Lap(\lambda)| < \frac{t}{\beta}\sum_{k=1}^{\beta}|f(k)Lap(\lambda)|$ is equivalent to the proof of $\sum_{k=1}^{\beta} x_k f(k) < \frac{t}{\beta}\sum_{k=1}^{\beta} f(k)$, and $\sum_{k=1}^{\beta} x_k = t$;

Because the β-tree adds noise into an intermediate node's ith selected child leaf with $\frac{(\beta-i+1)+1+\beta}{(\beta-i+1)+\beta}Lap(\lambda)$, where i increases from 1 to β. For any $k \in [1, \frac{\beta}{2}]$, $\frac{t}{\beta} - x_k = x_{\beta+1-k} - \frac{t}{\beta} > 0$.

$$
\begin{aligned}
D_{PrivCost} &= \sum_{k=1}^{\beta} x_k f(k) - \frac{t}{\beta}\sum_{k=1}^{\beta} f(k) \\
&= x_1 f(1) + \cdots + x_\beta f(\beta) - c(f(1) + \cdots + f(\beta)) \\
&= (x_1 - \frac{t}{\beta})(f(1) - f(\beta)) + \cdots + (x_{\frac{\beta}{2}} - \frac{t}{\beta})(f(\frac{\beta}{2}) \\
&\quad - f(\frac{\beta}{2}+1)) \\
&= \sum_{k=1}^{\frac{\beta}{2}}(x_k - \frac{t}{\beta})(f(k) - f(\beta+1-k))
\end{aligned}
$$

For any $k \in [1, \frac{\beta}{2}]$, $x_k - \frac{t}{\beta} < 0$, $f(k) - f(\beta+1-k) > 0$, so $D_{PrivCost} < 0$. □

References

1. Shen, F., Mu, Y., Yang, Y., Liu, W., Li, L., Song, J., Shen, H.T.: Classification by retrieval: Binarizing data and classifiers. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, pp. 595–604 (2017)
2. An, L., Wang, W., Shang, S., Li, Q., Zhang, X.: Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. GeoInformatica **22**(2), 335–362 (2018)
3. An, L., Li, Z., Liu, G., Zheng, K., Zhang, M., Li, Q., Zhang, X.: Privacy-preserving task assignment in spatial crowdsourcing. J. Comput. Sci Technol. **32**(5), 905–918 (2017)
4. Xiao, M., Ma, K., Liu, A., Zhao, H., Li, Z., Zheng, K., Zhou, X.: Sra: Secure reverse auction for task assignment in spatial crowdsourcing. IEEE Trans. Knowl. Data Eng. **PP**, 1–1 (2019). https://doi.org/10.1109/TKDE.2019.2893240
5. Li, X., Song, J., Gao, L., Liu, X., Huang, W., He, X., Gan, C.: Beyond rnns: Positional Self-Attention with Co-Attention for Video Question Answering. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, pp. 8658–8665 (2019)
6. Zhai, D., Sun, Y., An, L., Li, Z., Liu, G., Zhao, L., Zheng, K.: Towards secure and truthful task assignment in spatial crowdsourcing. World Wide Web **22**(5), 2017–2040 (2019)

7. Friedman, A., Schuster, A.: Data mining with differential privacy. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, pp. 493–502 (2010)
8. Fung, B.C.M., Ke, W., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Comput Surv. **42**(4), 14:1–14:53 (2010)
9. Hardt, M., Ligett, K., McSherry, F.: A simple and practical algorithm for differentially private data release. In: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held, Lake Tahoe, Nevada, pp. 2348–2356 (2012)
10. Yu, S.: Big privacy: Challenges and opportunities of privacy study in the age of big data. IEEE Access **4**, 2751–2763 (2016)
11. Sweeney, L.: k-anonymity: A model for protecting privacy. Int. J. Uncertain. Fuzz. Knowl.-Based Syst. **10**(5), 557–570 (2002)
12. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, Atlanta, GA, pp. 24 (2006)
13. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, pp. 106–115 (2007)
14. Dwork, C.: Differential Privacy. In: Automata, Languages and Programming, 33Rd International Colloquium, ICALP 2006, Venice, Italy, Proceedings, Part II, pp. 1–12 (2006)
15. Dwork, C.: Differential Privacy: A Survey of Results. In: Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, Proceedings, pp. 1–19 (2008)
16. Dwork, C.: A firm foundation for private data analysis. Commun. ACM **54**(1), 86–95 (2011)
17. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3-4), 211–407 (2014)
18. Dwork, C.: A firm foundation for private data analysis. Commun. ACM **54**(1), 86–95 (2011)
19. McSherry, F., Talwar, K.: Mechanism Design via Differential Privacy. In: 48Th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, Proceedings, pp. 94–103 (2007)
20. McSherry, F., Mironov, I.: Differentially private recommender systems: Building privacy into the netflix prize contenders. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 627–636 (2009)
21. Xu, J., Zhang, Z., Xiao, X., Yang, Y., Yu, G., Winslett, M.: Differentially private histogram publication. VLDB J. **22**(6), 797–822 (2013)
22. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. In: Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, Long Beach, California, pp 225–236 (2010)
23. Mohammed, N., Chen, R., Fung, B.C.M., Yu, P.S.: Differentially private data release for data mining. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, pp. 493–501 (2011)
24. Verykios, V.S., Bertino, E., Fovino, I.N., Provenza, L.P., Saygin, Y., Theodoridis, Y.: State-of-the-art in privacy preserving data mining. SIGMOD Rec. **33**(1), 50–57 (2004)
25. Xu, Y., Ma, T., Tang, M., Tian, W.: A survey of privacy preserving data publishing using generalization and suppression (2014)
26. Cormode, G., Procopiuc, C.M., Srivastava, D., Shen, E., Yu, T.: Differentially Private Spatial Decompositions. In: IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, pp. 20–31 (2012)
27. Zhang, J., Xiao, X., Xie, X.: Privtree: A differentially private algorithm for hierarchical decompositions. In: Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, pp. 155–170 (2016)
28. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Privbayes: Private Data Release via Bayesian Networks. In: International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, pp. 1423–1434 (2014)
29. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Private release of graph statistics using ladder functions. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, pp. 731–745 (2015)
30. Xiao, M., Wu, J., Huang, L., Cheng, R., Wang, Y.: Online task assignment for crowdsensing in predictable mobile social networks. IEEE Trans. Mob Comput. **16**(8), 2306–2320 (2017)
31. Xiao, Z., Huang, W.: Kd-Tree Based Nonuniform Simplification of 3D Point Cloud. In: 2009 Third International Conference on Genetic and Evolutionary Computing, WGEC 2009, Guilin, China, pp. 339–342 (2009)

32. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: SIGMOD'84, Proceedings of Annual Meeting, Boston, Massachusetts, pp. 47–57 (1984)
33. Hans, L.: Bodlaender a linear time algorithm for finding tree-decompositions of small treewidth (1996)
34. Demaine, E.D., Mozes, S., Rossman, B., Weimann, O.: An Optimal Decomposition Algorithm for Tree Edit Distance. In: Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, Proceedings, pp. 146–157 (2007)
35. Li, C., Hay, M., Rastogi, V., Miklau, G., McGregor, A.: 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, Indianapolis, Indiana, pp. 123–134 (2010)
36. Qardaji, W.H., Yang, W., Li, N.: Differentially Private Grids for Geospatial Data. In: 29Th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013, Pp 757–768 (2013)
37. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. IEEE Trans. Knowl. Data Eng. **23**(8), 1200–1214 (2011)

Affiliations

Xiaocui Li[1] · Yangtao Wang[1] · Jingkuan Song[2] · Yu Liu[1] · Xinyu Zhang[3] · Ke Zhou[1] · Chunhua Li[1]

Xiaocui Li
LXC@hust.edu.cn

Yangtao Wang
ytwbruce@hust.edu.cn

Jingkuan Song
jingkuan.song@gmail.com

Yu Liu
liu_yu@hust.edu.cn

Xinyu Zhang
zhangxinyu@whu.edu.cn

Ke Zhou
k.zhou@hust.edu.cn

1 Wuhan National Laboratory for Optoelectronics and School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

2 University of Electronic Science and Technology of China, Chengdu, China

3 Wuhan University, Wuhan, China