Asymmetric Differential Privacy

Shun Takagi
Kyoto University
Kyoto, Japan
takagi.shun.45a@st.kyoto-u.ac.jp

Yang Cao
Kyoto University
Kyoto, Japan
yang@i.kyoto-u.ac.jp

Masatoshi Yoshikawa
Kyoto University
Kyoto, Japan
yoshikawa@i.kyoto-u.ac.jp

ABSTRACT

Recently, differential privacy (DP) is getting attention as a privacy definition when publishing statistics of a dataset. However, when answering a decision problem with a DP mechanism, it causes a two-sided error. This characteristic of DP is not desirable when publishing risk information such as concerning COVID-19. This paper proposes relaxing DP to mitigate the limitation and improve the utility of published information. First, we define a policy that separates information into sensitive and non-sensitive. Then, we define asymmetric differential privacy (ADP) that provides the same privacy guarantee as DP to sensitive information. This partial protection induces asymmetricity in privacy protection to improve utility and allow a one-sided error mechanism. Following ADP, we propose two mechanisms for two tasks based on counting query with utilizing these characteristics: top-k query and publishing risk information of viruses with an accuracy guarantee. Finally, we conducted experiments to evaluate proposed algorithms using real-world datasets and show their practicality and improvement of the utility, comparing state-of-the-art algorithms.

PVLDB Reference Format:
Shun Takagi, Yang Cao, and Masatoshi Yoshikawa. Asymmetric Differential Privacy. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at http://vldb.org/pvldb/format_vol14.html.

1 INTRODUCTION

Differential privacy (DP) [12] is becoming a gold standard privacy notion so that the US census announced that they adopted DP when publishing the '2020 Census results [4] and IT companies such as Google [16], Apple [31], Microsoft [9], and Uber [21] are using DP to protect privacy while collecting data. The flourish is from the mathematical rigorousness under the assumption that an adversary has any knowledge about individuals in the dataset.

This paper considers error on a decision problem that is a query where we can answer by yes or no: *one-sided error* and *two-sided error*, which occur when solving a decision problem by a randomized algorithm such as Monte Carlo algorithm [24]. For example, suppose we answer yes to a decision problem based on Warner's randomized response [33] which satisfies DP. In that case, observers cannot know the true answer is whether yes or no due to the possible error (i.e., two-sided error). A one-sided error means that the randomized mechanism correctly answers for a one-sided answer (e.g., yes), but the opposite answer may include error. If we answer yes using Mangat's randomized response [26], we can know the true answer is yes because the output of the algorithm has the one-sided characteristic.

As an example of a problem of the two-sided characteristic, we consider publishing risky location information for epidemic disease monitoring. Specifically, we want to publish each location is safe or not safe, which means whether it has been visited by many infected people (i.e., counting query). Actually, Korea has succeeded in control of COVID-19 by measures including publishing information of infected people's moving trajectories [30]. Despite its effectiveness, this measure was criticized as an invasion of privacy [30]. One may think to release a DP histogram about "how many infected people visited each location". However, it must include two-sided errors due to the nature of DP. That means, even if published information says safe, it may not be true due to the noise of DP. Without a guarantee of accuracy (one-sided error), published information is not useful to take an appropriate approach.

DP is mathematically defined, so we can induce the theoretical limitations on the error on a decision problem. This paper proves that there is no one-sided error mechanism that satisfies ϵ-DP (i.e., the error of all DP mechanisms must be two-sided). Therefore, we cannot construct a mechanism that satisfies DP with one-sided error. We note that Mangat's randomized response is not following DP.

This paper proposes a new relaxed definition of DP to overcome the limitation, called *Asymmetric Differential Privacy* (ADP). We introduce a *policy* which defines whether a value is sensitive or non-sensitive. Then, we define ADP on the given policy. ADP protects privacy following the given policy so that ADP will protect individuals' sensitive values and not protect non-sensitive values. Concretely, ADP guarantees the indistinguishability of sensitive values from non-sensitive values, but ADP does not guarantee non-sensitive values from sensitive values[1]. The asymmetricity enables constructing a one-sided error mechanism. Then, we propose one-sided error mechanisms to publish the safety information about viruses with a trajectory dataset.

The asymmetricity not only allows a one-sided error mechanism but also improves the existing DP mechanisms. First, we consider a single-dimensional counting query. The Euclidean distance error (i.e., $\mathbb{E}_z[|x-z|]$ where z is the output and x is the true count) has the lower bound $\sqrt{2}/\epsilon$ and the Laplace mechanism [13] is optimal, which was proved by Koufogiannis et al. [23]. Then, we show that ADP allows the Euclidean distance error $1/\epsilon$ due to the asymmetricity. Second, this paper considers two representative DP mechanisms

[1] When DP guarantees the indistinguishability of A from B, DP also guarantees the indistinguishability of B from A, which is the symmetricity of DP.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

using the Laplace mechanism: report noisy arg-max and sparse vector technique [10, 25]. The algorithms aim to make the histogram of top-k counts (i.e., set of counting query). We propose replacing DP of the two algorithms with ADP. The asymmetricity improves the accuracy of output histogram with the same privacy guarantee to the sensitive values.

Finally, we conducted two types of experiments to evaluate our proposed mechanisms using two kinds of real-world datasets, respectively: click-streaming data and trajectory data. First, we show with click-streaming data that the output histogram of top-k items from our ADP mechanisms is more about ten-times accurate w.r.t. the mean squared error than a state-of-the-art mechanism [10] due to the relaxation. Second, we show with trajectory data that our proposed mechanisms are practical to publish safety information of viruses. ADP can guarantee the accuracy of *safety*, but cannot guarantee the one about *dangerousness* (i.e., false-negative occurs). Our experiments show that the false-negative ratio is practically small. In almost the case, that is smaller than 0.01 at $\epsilon = 1$.

Our contribution in this paper is threefold.

- We derive the theorem that shows there are no DP algorithms with one-sided error. Then, we introduce a policy and propose a new privacy definition, called Asymmetric Differential Privacy (ADP), following a policy to allow one-sided error, and we derive the theorems that show improved utility about the one-sided error.
- We introduce a primitive mechanism that satisfies ADP. Then, we propose a policy and mechanisms that follow ADP on the policy for top-k query and monitoring locations' safety for avoiding viruses with an accuracy guarantee.
- We show the improvements from the state-of-the-art mechanism [10] and the practicality for publishing risk information of viruses by the experiments using real-world datasets.

1.1 Related Work

Chen et al. [7] proposed a way of constructing a COVID-19 vulnerability map with geo-indistinguishability. Geo-indistinguishability is a local model so that noise is much larger, so their granularity of the map is needed to be large. Our model is central, so we need a trusted third party, but our map can be any granularity due to the low sensitivity. Contact tracing is now widely used [3, 18] such as BlueTrace in Singapore, Cocoa in Japan, etc., but the model is said to require participants more than about half of the population to decrease infection [29]. Actually, many countries failed due to the lack of participants [8] because of several concerns [28]. Our model requires only infected people to participate, so such a problem does not occur.

As this work proposes, several papers proposed relaxation of DP by defining a policy to improve utility [5, 20, 22]. Their policy is based on a discriminative pair, so the asymmetricity that we utilize does not occur. Therefore, their relaxation cannot achieve our improvement of utility.

To the best of our knowledge, this paper first considers one-sided error in DP. However, it is noted that several relaxation models of DP can achieve one-sided error implicitly [1, 11, 17, 27]. One-sided differential privacy [11] (OSDP) also defines a policy that separates records to non-sensitive and sensitive records to improve the utility of output information by utilizing non-sensitive records. Therefore, OSDP cannot follow our policy that separates values. If a record is a value (i.e., data is one-dimensional), OSDP is identical with ADP, and the asymmetricity in OSDP occurs, but the paper does not mention and utilize the asymmetricity. Actually, if we use OSDP in our tasks, the asymmetricity does not occur because we use multi-dimensional data.

Symbol	Meaning
$D \in \mathcal{D}$	A dataset that is a set of records.
$r = (r_1, r_2, \ldots, r_d)$	A record that is a set of attributes.
$r_i \in \mathcal{X}_i$	An attribute of record r. This paper assumes $\mathcal{X}$ is binary.
$z \in \mathcal{Z}$	An output of a mechanism
$S \subseteq \mathcal{Z}$	A subset of outputs.
λ	A random variable.
$H = (\lambda_1, \lambda_2, \ldots, \lambda_d)$	A set of random variables.
$m : \mathcal{D} \to \mathcal{Z}$	A randomized mechanism.
$m_H : \mathcal{D} \to \mathcal{Z}$	A mechanism using random variables H.
$\mathbb{R}$	The universe of a real number
$\mathbb{N}$	The unuverse of a natural number.
$\epsilon \in \mathbb{R}^+$	A privacy budget.
$f : \mathcal{D} \to \mathcal{Z}$	A query. This paper considers that $\mathcal{Z}$ is $\mathbb{N}$.
$q : \mathcal{D} \to \{0, 1\}$	A decision problem.

Table 1: Notation table.

Mangat's randomized response [26] is following our privacy notion, which we can interpret as asymmetrization of the traditional randomized response (i.e., Warner's randomized response [33]). Utility optimized local differential privacy (ULDP) [27] is also a privacy definition that Mangat's randomized response follows, so ULDP also utilizes asymmetricity. However, ULDP is defined on the local setting on data with one attribute, so we cannot apply ULDP to our task (i.e., counting query). Context-aware local differential privacy [1] and input-discriminative local differential privacy [17] define a precise privacy level for a combination of data, which are generalizations of ULDP. Therefore, they can also utilize asymmetricity in a certain setting (and the papers did not utilize the asymmetricity), but for the same reason as ULDP, we cannot apply them to our task.

2 BACKGROUND

Here, we first introduce DP and related techniques, which are the basis of our proposed notions. Second, we explain counting query and decision problem, which are our target queries to which we introduce our privacy notion.

We show the notations used in this paper in Table 1.

2.1 Differential Privacy

Differential privacy (DP) [12] is a mathematical privacy definition, which quantitatively evaluates the degree of privacy protection when publishing an output by a randomized mechanism. The definition of DP is as follows:

Definition 1 ((ε, δ)-differential privacy). *A randomized mechanism $m : \mathcal{D} \rightarrow \mathcal{Z}$ satisfies (ε, δ)-DP iff for any two neighboring datasets $D, D' \in \mathcal{D}$ and any subset of outputs $S \subseteq \mathcal{Z}$, it holds that*

$$\Pr[m(D) \in S] \leq \exp(\varepsilon) \Pr[m(D') \in S] + \delta. \quad (1)$$

where $\mathcal{D}$ and $\mathcal{Z}$ are the universe of a dataset and output, respectively, and neighboring datasets mean two datasets whose only one record differs.

Practically, we employ a randomized mechanism m that ensures DP for a query $f : \mathcal{D} \rightarrow \mathcal{Z}$. The mechanism m perturbs the output of f to cover f's sensitivity that is the maximum degree of change over any pairs of neighboring datasets D and D'.

Definition 2 (Sensitivity). *The sensitivity of a query f is:*

$$\Delta_f = \sup_{D, D' \in \mathcal{D}} \left| f(D) - f(D') \right|.$$

where $| \cdot |$ is a norm function defined on f's output domain and D and D' are neighboring datasets.

Based on the sensitivity of f, we design the degree of noise to ensure DP.

2.1.1 Laplace Mechanism.

The Laplace mechanism [13] is the most standard mechanism for DP, so many kinds of literature are using the Laplace mechanism. This mechanism adds noise following the Laplace distribution to the answer f as follows.

$$Lap_\epsilon(f, D) = f(D) + \lambda$$

where λ is a random variable following the Laplace distribution $\frac{\epsilon}{2\Delta_f} \exp \frac{|\lambda| \epsilon}{\Delta_f}$. The Laplace mechanism satisfies ϵ-DP.

2.1.2 Composition Theorem.

The composition theorem [15] shows that a mechanism that sequentially applies DP mechanisms satisfies DP. Informally, the composition theorem is as follows.

Theorem 1. *Let m_i be ϵ_i-DP mechanism for $i \in [k]$. Then the sequential mechanism $m_{[k]}$ (i.e., $m_{[k]}(x) = (m_1(x), m_2(x), \ldots, m_k(x))$) satisfies $\sum_{i \in [k]} \epsilon_i$-DP*

2.1.3 Randomness Alignment.

A DP mechanism can be so complicated (e.g., the sparse vector technique described in the following section) that the wrong proofs appear [25]. Then, many verification tools using a proof syntax [2, 32, 34] are proposed to verify the correctness of proof for DP. The idea of the proof syntax is called *randomness alignment*. Here, we explain randomness alignment using pure DP (i.e., $\delta = 0$).

In general, we need to prove that Inequality (1) holds for any output and any pair of neighboring datasets. H denotes a set of random variables used in a mechanism, and let $m_H(D)$ denote the output of m using H. Then, we introduce *randomness alignment* $\phi_{D,D'}$, which is a function that maps noise vectors H into noise vectors H' so that $m_H(D) = m_{H'}(D')$. Here, let D, D' be two neighboring datasets. We follow Ding's formulation about the definitions and notations for randomness alignment [10].

Definition 3 (Randomness Alignment [10]). *A randomness alignment is a function $\phi_{D,D'}$ such that for all H, $m_H(D) = m_{\phi_{D,D'}(H)}(D')$.*

Definition 4 (Local Alignment [10]). *A local alignment for m is a function $\phi_{D,D',z} : S_{D:z} \rightarrow S_{D':z}$ such that for all $H \in S_{D:z}$, we have $m_H(D) = m_{\phi_{D,D',z}(H)}(D')$ where $S_{D:z} = \{H | m_H(D) = z\}$ and z is a possible output of m.*

Definition 5 (Acyclic [10]). *For any $H = (\lambda_1, \lambda_2, \ldots)$, let $H' = (\lambda'_1, \lambda'_2, \ldots)$ denote $\phi_{D,D',z}(H)$. We say that $\phi_{D,D',z}$ is acyclic if there exists a permutation π and piecewise differentiable functions $\psi^{(j)}_{D,D',z}$ such that*

$$\lambda'_{\pi(1)} = \lambda_{\pi(1)} + number\ that\ only\ depends\ on\ D, D', z$$

$$\lambda'_{\pi(j)} = \lambda_{\pi(j)} + \psi^{(j)}_{D,D',z}(\lambda_{\pi(1)}, \ldots, \lambda_{\pi(j-1)})\ for\ j \geq 2$$

Definition 6 (Alignment Cost [10]). *Suppose each λ_i is generated independently from a distribution f_i with the property that $\log(f_i(x)/f_i(y)) \leq |x - y| / \alpha_i$ for all x, y in the domain of f_i. Then the cost of $\phi_{D,D',z}$ is defined as:*

$$cost(\phi_{D,D',z}) = \sum_i \left| \lambda_i - \lambda'_i \right| / \alpha_i$$

Using these formulations, we can derive the following theorem.

Theorem 2. *If the following conditions are satisfied, then m satisfies ϵ-DP [10].*

(1) *m terminates with probability 1.*
(2) *The number of random variables used by m can be determined from its output.*
(3) *Each λ_i is generated independently from a distribution f_i with the property that $\log(f_i(x)/f_i(y)) \leq |x - y| / \alpha_i$ for all x, y in the domain of f_i.*
(4) *For every neighboring dataset D, D' and z there exists a local alignment $\phi_{D,D',z}$ that is acyclic with $cost(\phi_{D,D',z}) \leq \epsilon$.*
(5) *For each neighboring dataset D, D' the number of distinct local alignments is countable. That is, the set $\{\phi_{D,D',z} | z \in \mathcal{Z}\}$ is countable.*

Following this theorem, we can prove that a mechanism satisfies DP. We refer the reader to [10] for detail.

2.2 Counting Query

Counting query is the most basic statistic query. Counting query appears in fractional form, with weights (linear query) or more complex form, but this paper considers the most simple counting query $f : \mathcal{D} \rightarrow \mathbb{N}$, which counts the number of records satisfying a condition. In this case, the theoretical bound of the Euclidean distance error (i.e., $\mathbb{E}_z[|x - z|]$ where z is the output and x is the true count) is as follows

Theorem 3 (Lower bound on the counting query on DP [23]). *Let $\epsilon < 0$, every ϵ-DP mechanism must have the Euclidean distance error more than $\sqrt{2}/\epsilon$.*

The Laplace mechanism 2.1.1 has error $\sqrt{2}/\epsilon$, which means optimal on the counting query.

When the number of counting queries is small, the Laplace mechanism is enough. However, multiple counting queries require much noise due to the composition theorem (Section 2.1.2). Then, the sparse vector technique and the report noisy arg-max algorithm are useful for answering multiple counting queries.

2.2.1 Sparse Vector Technique. Dwork et al. first proposed the sparse vector technique to handle high-sensitive counting query [14]. Then, successors proposed many applications [25]. Recently, Ding et al. found an accuracy-improved version of the sparse vector technique [10].

Here, we explain the most standard sparse vector technique [14]. In nature, if we use sequentially a DP mechanism (e.g., Laplace mechanism) to answer queries $(f_0, f_1, \ldots, f_t)$, the privacy guarantee is $t\epsilon$-DP from the composition theorem (see Section 3.3.3 for detail). This sequential composition is a problem when the number of queries is large, and the sparse vector technique solves this problem.

The sparse vector technique returns a threshold answer (i.e., whether a query answer is above the threshold or not) instead of a noisy count. The algorithm does not consume the privacy budget when outputting a below-the-threshold answer by adding noise to the threshold and query answer using the Laplace mechanism with $\epsilon/2$ and $\epsilon/(2c)$, respectively. It consumes the privacy budget only when outputting an above-the-threshold answer. In other words, the sparse vector technique can find c queries of an above-the-threshold answer. This algorithm satisfies ϵ-DP.

2.2.2 Report Noisy Arg-max. Here, we consider a maximum query (i.e., finding the maximum answer in a set of queries). By counting all attributes, we can find the argument with maximum counts, but this manipulation causes high-sensitivity. Then, a report noisy arg-max algorithm was proposed to handle this high-sensitivity [15]. The algorithm first uses the Laplace mechanism with $\epsilon/2$ for all counts. Then, the algorithms output the argument of the maximum noisy answer. This algorithm satisfies ϵ-DP.

2.3 Decision Problem

A decision problem is a query that returns yes or no. In this paper, we handle the query which answers whether the number of counts is below the threshold or not.

One-sided and Two-sided error. Here, we introduce one-sided error and two-sided error, mainly used in Monte Carlo algorithms to answer a decision problem. In a deterministic algorithm, an error does not occur, but a randomized algorithm such as a Monte Carlo algorithm causes the error.

If a randomized algorithm is always correct when it returns True (False), we call it a true-biased (false-biased) algorithm. We say that if an algorithm is a true-biased (false-biased) algorithm, the algorithm is one-sided. If a randomized algorithm has no-biased, we say that the algorithm is two-sided.

In Section 4.2, we show that the error of any ϵ-DP mechanisms must be two-sided, which means that true-biased (false-biased) publishing is impossible due to the constraint of traditional ϵ-DP.

3 NEW PRIVACY DEFINITION

In this section, we propose a new privacy definition, called Asymmetric Differential Privacy (ADP), which relaxes DP.

3.1 Asymmetric Differential Privacy

Here, we propose the relaxation of DP, called Asymmetric Differential Privacy (ADP). To explain ADP, we use Table 2 as a dataset for an example. Each data represents whether the user has visited locations or not. E.g., this table shows that Bob has visited location 1 and has not visited location 2.

	location 1	location 2
Bob	1	0
Tom	0	0
Alice	1	1
Ema	0	1

Table 2: Example of a dataset.

We first introduce a policy function $P_i : \mathcal{X}_i \rightarrow \{0, 1\}$ where $\mathcal{X}_i$ is the universe of i_{th} attribute. This paper assumes that an attribute's universe is binary (i.e., $\{0, 1\}$).

Definition 7 (Policy function). *A policy function $P_i : \mathcal{X}_i \rightarrow \{0, 1\}$, which takes a value of an attribute i as an input, returns 1 when the input is sensitive and returns 0 otherwise*[2]. *P denotes the set of policy functions $P_1, P_2, \ldots, P_d$ for all attributes, where d is the number of attributes of a dataset. Here, we assume that all individuals in the dataset use the same policy function (i.e., a used policy is public information).*

Example 1 (Policy function). *Let us suppose that users compromise to publish that they have not visited an area but want to hide their visited areas to protect their privacy. Then, the policy function in Table 2 can be defined as follows:*

$$P_i(x) = \begin{cases} 1 & (x = 1) \\ 0 & (x = 0) \end{cases}$$

where $i = 0, 1$.

Example 2 (Policy function of differential privacy). *DP does not use a policy function but we can interpret that DP uses the following policy function for all attributes.*

$$P^{DP}(x) = 1$$

In other words, DP considers that all data is sensitive.

We note that simply publishing non-sensitive information causes a privacy leak because we assume that a policy function is published. For example, if Bob only publishes *location 2*, an adversary knows Bob has visited *location 1* because Bob does not publish the information. Our relaxation is based on this policy function. We relax DP by allowing leakage of non-sensitive value. ADP guarantees that a mechanism hides a record in P-neighboring records defined as follows.

Definition 8 (P-neighboring of a record). *We say that a record $r = (r_1, r_2, \ldots, r_d)$ is P-neighboring to a record $r' = (r'_1, r'_2, \ldots, r'_d)$ iff it holds that:*

$$\forall i, r_i = r'_i \text{ if } P_i(r_i) = 0$$

where r_i and r'_i are i_{th} attributes of r and r', respectively.

[2] Our policy is different from OSDP's policy [11] in that our policy discriminates values but OSDP's policy discriminates records.

EXAMPLE 3 (P-NEIGHBORING OF A RECORD). *Bob's record in Table 2 is P-neighboring to* $[0,0], [1,0]$. *Alice's record is P-neighboring to* $[1,0], [0,1], [0,0]$. *Tom's record is not P-neighboring to any record.*

In DP, there is no notion corresponding to *neighboring record*. Still, we can consider that DP implicitly uses a neighboring record as all records.

Then, we can define the notion of *P-neighboring dataset*.

DEFINITION 9 (P-NEIGHBORING OF A DATASET). *We say that a dataset D is P-neighboring to a dataset D' iff only one record of D differs from D' and the differing record is P-neighboring. Let $N_P(D)$ denote the set of datasets, each of which is P-neighboring dataset of D.*

Note that the neighboring relationship in DP is symmetric, but the P-neighboring relationship can be asymmetric. In other words, if dataset D is neighboring to D', D' is neighboring to D. However, even if D is P-neighboring to D', D' is not always P-neighboring to D. The name *asymmetric* is given by this fact.

Using the P-neighboring relationship, we can define our new privacy notion, called Asymmetric Differential Privacy (ADP).

DEFINITION 10 ((P,ϵ)-ASYMMETRIC DIFFERENTIAL PRIVACY). *A randomized mechanism m satisfies (P,ϵ)-Asymmetric Differential Privacy (ADP) iff $\forall S \subseteq \mathcal{Z}$ and $\forall D, D' \in N_P(D)$*

$$\Pr[m(D) \in S] \le \mathrm{e}^{\epsilon} \Pr[m(D') \in S] \tag{2}$$

The only difference from DP is in the neighboring relationship. Our privacy definition guarantees that it is difficult for an adversary to distinguish D from $D' \in N_P(D)$ to the degree of ϵ. From an individual's point of view, the own record has indistinguishability from neighboring records. In other words, an ADP algorithm protects a sensitive value (i.e., $P(r_i) = 1$) by hiding in other values. However, a non-sensitive value defined by a policy function (i.e., $P(r_i) = 0$) can leak due to the policy.

EXAMPLE 4 (A PRIVACY GUARANTEE). *Bob in Table 2 has visited location 1 and the information is protected because $P_{location\ 1}(1) = 1$. However, the information that Bob has not visited location 2 can leak because $P_{location\ 2}(0) = 0$. Tom's record has no P-neighboring record, so Tom's record can leak.*

3.2 P-sensitivity

The asymmetric characteristic on the P-neighboring relationship (see Section 3.1 for detail) can induce monotonicity of sensitivity. Monotonicity enables us to construct a more accurate and one-sided error algorithm. We note that the sensitivity of DP has a characteristic of not monotonicity since DP uses the symmetric neighboring relationship, which is the main motivation to introduce ADP instead of DP.

3.2.1 Definition.

We define P-sensitivity instead of sensitivity of DP by changing the neighboring relationship.

DEFINITION 11 (P-SENSITIVITY).

$$\Delta_{f,P} = \sup_{D\in\mathcal{D}, D'\in N_P(D)} \left|f(D') - f(D)\right|$$

where f and P are a query and a policy function, respectively.

3.2.2 Monotonicity.

We define monotonicity in sensitivity as follows:

DEFINITION 12 (MONOTONICITY IN P-SENSITIVITY). *P-sensitivity of f is called monotonically increasing (decreasing) iff $\forall D \in \mathcal{D}, D' \in N_P(D)$*

$$f(D) \le f(D')\ (f(D) \ge f(D')).$$

As described above, DP requires that sensitivity is not monotonic due to the symmetric neighboring relationship.

THEOREM 4. *Sensitivity in DP (i.e., P^{DP}-sensitivity) is not monotonic except the query where $\forall D, D', f(D) = f(D')$*[3].

PROOF. The neighboring relationship of DP is symmetric, so if $f(D) - f(D') > 0$, it holds that $f(D') - f(D) < 0$. Therefore, sensitivity is not monotonic except the case where $\forall D, D', f(D) = f(D')$.

Q.E.D. □

EXAMPLE 5 (MONOTONICALLY DECREASING). *Assume a counting query f that counts the number of people who have visited a location. Then, P-sensitivity of f where P is the policy in Example 1 is monotonically decreasing since a change of a sensitive value only decreases the count.*

3.3 Characteristics of ADP

First, we describe the privacy guarantees of ADP against a strong adversary. Second, we analyze a relationship between ADP and DP. Third, we derive the composition theorem for ADP. Here, we consider that a dataset D consists of only one record for simplicity (i.e., the setting of local differential privacy) without losing generality.

3.3.1 Privacy Guarantee Against a Strong Adversary.

A strong adversary is knowledgeable about a target individual so that the adversary knows non-sensitive values but does not know sensitive values. In other words, the adversary knows that a true record is in $N_P(D)$ (note that here $N_P(D)$ is the set of records due to the assumption of $|D| = 1$). Then, from Definition 10,

$$\frac{\Pr[r|z]}{\Pr[r'|z]} \le \mathrm{e}^{\epsilon} \frac{\Pr[r]}{\Pr[r']}$$

where r is a true record and $\{r'\} \in N_P(\{r\})$. Here, the strong adversary knows that the true record is in $N_P(\{r\})$, so the privacy guarantee against the strong adversary is the same as DP. In other words, the strong adversary cannot improve her/his knowledge up to e^{ϵ} even if s/he sees z output by a mechanism that satisfies (P,ϵ)-ADP. If an adversary is not knowledgeable about a target individual, s/he can know that a true record is in $N_P(\{r\})$.

3.3.2 Relationship to Differential Privacy.

As a relationship between ADP and DP, we can prove that ADP is a relaxation of DP. Informally, the following theorems hold.

THEOREM 5. (1) *If a mechanism m satisfies (P^{DP},ϵ)-ADP, m satisfies ϵ-DP.*
(2) *If a mechanism m satisfies ϵ-DP, m satisfies (P,ϵ)-ADP for any policy function P.*

[3] This query is not sensitive because this always returns the same value regardless of an input database.

PROOF. (1) P^{DP}-neighboring records are all records. This means that P^{DP}-neighboring datasets are the same as the ones of the neighboring datasets in DP. Therefore, the theorem holds.

(2) Neighboring datasets of D in DP (i.e. P^{DP}-neighboring datasets) include P-neighboring datasets of D for any P. Therefore, if m satisfies ϵ-DP, Inequality (4) holds for any neighboring datasets, so Inequality (2) also holds for P-neighboring datasets.
Q.E.D.

□

3.3.3 Composition Theorem. ADP satisfies the composition theorem like DP.

THEOREM 6. *We consider the sequential mechanism M, which sequentially applies $m_1, m_2, \ldots, m_j$ to dataset D to output $z_1, z_2, \ldots, z_j$. Assume that $m_1, m_2, \ldots, m_j$ satisfy (P, ϵ_1), (P, ϵ_2), …, (P, ϵ_j)-ADP, respectively. Then, the sequential mechanism M satisfies $(P, \sum_{i=1}^{j} \epsilon_i)$-ADP.*

PROOF. From the definition of ADP, it holds that $\forall S \subseteq \mathcal{Z}, D, D' \in N_P(D), i \in [j]$

$$\Pr[m_i(D) \in S] \leq \mathrm{e}^{\epsilon_i} \Pr[m_i(D') \in S]$$

Therefore, the following inequality holds.

$$\frac{\Pi_{i=1}^{j} \Pr[m_i(D) \in S]}{\Pi_{i=1}^{j} \Pr[m_i(D') \in S]} = \frac{\Pi_{i=1}^{j} \Pr[M(D) \in S]}{\Pi_{i=1}^{j} \Pr[M(D') \in S]} \leq \mathrm{e}^{\sum_{i=1}^{j} \epsilon_i}$$

Q.E.D. □

3.3.4 Randomness Alignment in Asymmetric Differential Privacy. As described above, the difference between ADP and DP is in neighboring relationships. Therefore, we can apply a randomness alignment technique to prove that a mechanism satisfies (P, ϵ)-ADP as follows.

THEOREM 7. *If the following conditions are satisfied, then m satisfies (P, ϵ)-ADP.*

(1) *m terminates with probability* 1.
(2) *The number of random variables used by m can be determined from its output.*
(3) *Each λ_i is generated independently from a distribution f_i with the property that* $\log(f_i(x)/f_i(y)) \leq |x - y| / \alpha_i$ *for all x, y in the domain of f_i.*
(4) *For every $D \in \mathcal{D}, D' \in N_P(D)$ and z there exists a local alignment $\phi_{D,D',z}$ that is acyclic with $cost(\phi_{D,D',z}) \leq \epsilon$.*
(5) *For each $D \in \mathcal{D}, D' \in N_P(D)$ the number of distinct local alignments is countable. That is, the set $\{\phi_{D,D',z} | z \in \mathcal{Z}\}$ is countable.*

PROOF. Theorem 2 means that if a mechanism satisfies the conditions in Theorem 2 it holds that

$$\Pr[m(D) \in S] \leq \mathrm{e}^{\epsilon} \Pr[m(D') \in S] \tag{3}$$

for any D, D', and $S \in \mathcal{Z}$. Here, we instead use P-neighboring datasets $D \in \mathcal{D}, D' \in N_P(D)$. Therefore, for any $D, D' \in N_P(D)$, Inequality (3) holds, which is the definition of ADP.

Q.E.D. □

4 THEORETICAL BOUND OF ONE-SIDED ERROR

Here, we first analyze a true-biased (false-biased) algorithm in DP (i.e., an algorithm with one-sided error). Then, we prove that there is no true-biased (false-biased) algorithm that satisfies ϵ-DP. Finally, we show that ADP allows a true-biased (false-biased) algorithm and the theoretical bound of one-sided error.

4.1 One-sided True Positive

We call the output of True by a true-biased algorithm (i.e., true positive) *one-sided true positive* (OTP). From here, the boolean is denoted by $\{0, 1\}$. We consider a decision problem $q : \mathcal{D} \rightarrow \{0, 1\}$ that takes a dataset as input and returns a boolean. To answer a query with privacy protection, we use a randomized mechanism $m : \mathcal{D} \rightarrow \{0, 1\}$ instead of $q(D)$, which causes an error. We say that the output is true positive (negative) iff $m(D) = 1(0)$ and $q(D) = 1(0)$, respectively. Then, the definition of OTP is as follows.

DEFINITION 13 (ONE-SIDED TRUE POSITIVE (OTP)). *OTP is output* 1 *of a true-biased algorithm. That is, OTP is output* 1 *of m where:*

$$\Pr[m(D) = 1; q(D) = 1] > 0 \ \wedge \ \Pr[m(D) = 1; q(D) = 0] = 0$$

If we see output 1 *of $m(D)$, we can validate $q(D) = 1$.*

We omit the notation of *one-sided true negative* to make it easy to read, but it will be identical to OTP by replacing 1 with 0.

4.2 OTP in DP

From here, we consider a mechanism m, which satisfies (ϵ, δ)-DP. From the conclusion as the following theorem, the mechanism achieves OTP with the probability of the proportional to δ, which means pure DP (i.e., $\epsilon = 0$) does not achieve OTP.

THEOREM 8 (OTP IN (ϵ, δ)-DIFFERENTIAL PRIVACY). *Let us suppose that m satisfies (ϵ, δ)-DP and the minimum hamming distance to change the query answer of D is k (i.e., $\forall D', d_h(D, D') \geq k$ where $q(D) = 1$ and $q(D') = 0$)*[4]. *Then, it holds that:*

$$\Pr[OTP; q(D) = 1] \leq \delta \sum_{i=1}^{k} \mathrm{e}^{(i-1)\epsilon}$$

where $\Pr[OTP; q(D) = 1]$ *is the probability that z is OTP when* $q(D) = 1$.

PROOF. Since m satisfies (ϵ, δ)-DP,

$$\Pr[m(D) \in S] \leq \mathrm{e}^{\epsilon} \Pr[m(D') \in S] + \delta \tag{4}$$

where D and D' are neighboring datasets, and S is any subset in output domain. Here, we assume $q(D) = 1$ and S to consist of all outputs that are OTP. Then, $\Pr[m(D) \in S]$ is the same as $\Pr[OTP; q(D) = 1]$. Since the minimum hamming distance to change the query answer

[4] Here, D and D' are not limited to neighboring datasets.

is k, we can derive the following inequality by iteratively applying Inequality (4).

$$\Pr[m(D) \in S] \leq \mathrm{e}^{k\epsilon} \Pr[m(D_k) \in S] + \delta \sum_{i=1}^{k} \mathrm{e}^{(i-1)\epsilon}$$

where D_k is a dataset such that $d_h(D, D_k) = k$ and $q(D_k) = 0$. From the definition of OTP (Definition 13), we have $\Pr[m(D') \in S] = 0$. Therefore:

$$\Pr[m(D) \in S] = \Pr[OTP; q(D) = 1] \leq \delta \sum_{i=1}^{k} \mathrm{e}^{(i-1)\epsilon}$$

Q.E.D. □

In general, δ should be small (conventionally, δ is set as $1/n$ where n is the number of data) because δ means the probability that DP breaks. We note that δ of ϵ-DP is 0, which means that ϵ-DP cannot achieve OTP. If $k = 1$ and $\delta = 1/n$, we achieve OTP with only the probability $1/n$. Therefore, DP is not suited for our setting where OTP is required.

4.3 OTP in ADP

Here, we derive the upper bound of a probability that ADP achieves OTP, which means that there are one-sided error mechanisms. Our neighboring notion is asymmetric as described above, so the hamming distance in DP is not defined. Then, instead of the hamming distance, we use the minimum step.

Definition 14 (the minimum step). *The minimum step from D to D' is the minimum number of steps to take to change D to D' via P-neighboring datasets.*

If the minimum step from D_0 to D_k is k, there are datasets $D_0, D_1, \ldots, D_k$ such that $D_i \in N_P(D_{i-1})$ for all $1 \leq i \leq k$.

Theorem 9 (OTP in (P, ϵ)-ADP). *Let us assume that m satisfies (P, ϵ)-ADP, and the minimum step to change the query answer from D is k. Then, the following inequalities hold:*

$$\Pr[OTP; q(D) = 1] \leq 1 - \frac{1}{\mathrm{e}^{k\epsilon}}$$

Proof. Let S denote the set consisting of all outputs that are OTP. Then, we let S' denote the complement set of S. Since m satisfies (P, ϵ)-ADP, iteratively using inequality (2),

$$\Pr[m(D) \in S'] \leq \mathrm{e}^{k\epsilon} \Pr[m(D_k) \in S']$$

where D_k is a dataset such that the minimum step from D is k. Here, assuming $q(D) = 0$ and $q(D_k) = 1$, we can reformulate the above inequality as follows:

$$1 \leq \mathrm{e}^{k\epsilon}(1 - \Pr[m(D_k) \in S]) \tag{5}$$

because $\Pr[m(D_k) \in S] + \Pr[m(D_k) \in S'] = 1$ and S is the set of OTP. Since $q(D_k) = 1$, $\Pr[m(D_k) \in S]$ is the probability that we achieve OTP. Therefore, by reformulating (5),

$$\Pr[OTP; q(D) = 1] \leq 1 - \frac{1}{\mathrm{e}^{k\epsilon}}$$

Q.E.D. □

5 MECHANISMS

Here, we propose mechanisms that satisfy ADP. First, we introduce a basic mechanism that can be a component of ADP mechanisms, called the asymmetric Laplace mechanism. Then, we propose mechanisms for two use-cases, respectively: top-k query and location monitoring.

5.1 Asymmetric Laplace Mechanism

First, we introduce the asymmetric Laplace mechanism (aLap) that is an ADP version of the Laplace mechanism[5].

Definition 15 (asymmetric Laplace mechanism). *Let us assume a counting query $f : \mathcal{D} \to \mathbb{N}$. Then, the asymmetric Laplace mechanism aLap is*

$$aLap_\epsilon(f, D) = f(D) + \lambda$$

where λ is following the asymmetric Laplace distribution with ϵ as follows:
if P-sensitivity of f is monotonically decreasing,

$$\begin{cases} \frac{\epsilon}{\Delta_{f,P}} \exp \frac{-\lambda\epsilon}{\Delta_{f,P}} & (\lambda \geq 0) \\ 0 & (otherwise) \end{cases}$$

else if P-sensitivity of f is monotonically increasing,

$$\begin{cases} \frac{\epsilon}{\Delta_{f,P}} \exp \frac{\lambda\epsilon}{\Delta_{f,P}} & (\lambda \leq 0) \\ 0 & (otherwise) \end{cases}$$

else,

$$\frac{\epsilon}{2\Delta_{f,P}} \exp \frac{|\lambda|\,\epsilon}{\Delta_{f,P}}$$

The distribution in the case where f is not monotonic is the same as the Laplace distribution. When P-sensitivity is monotonic, the distribution is the exponential distribution, which has a smaller variance of $1/\epsilon^2$. Therefore, when the query is monotonic, we can improve accuracy. We note that the estimator of the answer with the Laplace noise is the answer itself, but the estimator for the asymmetric Laplace mechanism is $aLap_\epsilon(f, D) - 1/\epsilon$.

Theorem 10. *aLap satisfies (P, ϵ)-ADP.*

Proof. When P-sensitivity of f is not monotonic, the asymmetric Laplace mechanism satisfies (P, ϵ)-ADP from Theorem 5 because the Laplace mechanism satisfies ϵ-DP.

When P-sensitivity of f is monotonically decreasing, we need to show that $\forall z \in \mathcal{Z}$

$$\Pr[aLap_\epsilon(f, D) = z] \leq \mathrm{e}^{\epsilon} \Pr[aLap_\epsilon(f, D') = z]$$

where D' is a P-neighboring dataset of D. We consider the case where $aLap_\epsilon(f, D) = aLap_\epsilon(f, D')$ (i.e., $f(D) + \lambda = f(D') + \lambda'$). The difference is the largest when $\lambda' = \lambda + \Delta_{f,P}$ since P-sensitivity is monotonically decreasing. In other words, $\Pr[\lambda']$ is the smallest when the difference is P-sensitivity (i.e., $\Delta_{f,P}$). Therefore, $\Pr[\lambda]/\Pr[\lambda'] \leq \mathrm{e}^{\epsilon}$ for all λ, which proves the theorem. When the sensitivity of f is monotonically increasing, we consider $\lambda' =$

[5] aLap is similar to OsdpLaplace [11], but we can only use OsdpLaplace for the count of non-sensitive records. However, we can use aLap for any records since aLap utilizes the asymmetricity.

$\lambda - \Delta_{f,P}$ and the following proof is the same as when sensitivity is monotonically decreasing.

Q.E.D. □

When P-sensitivity is monotonic, the variance is $1/\epsilon^2$, which is smaller than the Laplace mechanism, but when P-sensitivity is not monotonic, the error is the same as the Laplace mechanism. Therefore, if we can construct a policy function P such that P-sensitivity is monotonic, we should use ADP to make the output more accurate. Otherwise, we need not use ADP and should use DP since ADP's privacy protection is weaker than DP.

Remark. From the theorem of Hardt and Talwar [19], the lower bound of the Euclidean distance error is proportional to the volume of the sensitivity hull (see [19] for detail). Asymmetric neighboring relationship halves the volume, so the lower bound is also halved, which means $\sqrt{2}/(2\epsilon)$. Therefore, the asymmetric Laplace mechanism does not reach optimal because the sensitivity hull is not isotropic. Hardt and Talwar's isotropic transformation method may be useful to achieve optimally, but this is beyond this paper's scope.

5.1.1 Decision Problem with the Asymmetric Laplace Mechanism. We can construct a mechanism for a decision problem with *aLap*. Here, we consider as a decision problem the query $q : \mathcal{D} \to \{0, 1\}$ which asks whether a counting query $f(D)$ is under the threshold or not. By using $aLap_\epsilon(f, D)$ for $f(D)$, q satisfies (P, ϵ)-ADP.

THEOREM 11. *When P-sensitivity of f is monotonic, the above mechanism q achieves OTP with the probability of $1 - \frac{1}{e^{k\epsilon}}$ where k is the minimum step.*

PROOF. Here, we consider the case where P-sensitivity of f is monotonically decreasing. In this case, OTP is the output 1 (i.e., below-the-threshold) because the noise of *aLap* is always positive so that the noise does not change the above-the-threshold answer, which means $\Pr[m(D) = 1; q(D) = 0] = 0$ (i.e., the definition of OTP). The probability of OTP is the probability that the variable of the exponential distribution is under k. Therefore,

$$1 - \int_k^\infty \frac{\epsilon}{\Delta_{f,P}} \exp \frac{-\lambda\epsilon}{\Delta_{f,P}} d\lambda$$

Since the query is counting query (i.e., $\Delta_{f,P} = 1$), the probability is $1 - \frac{1}{e^{k\epsilon}}$.

Q.E.D. □

Therefore, we can output OTP with the highest probability by using *aLap* for this decision problem.

5.2 Top-k Query

Here, we propose two ADP mechanisms for a top-k query: asymmetric report noisy max algorithm and asymmetric sparse vector technique, which are based on the popular algorithms for DP [15]. Due to the asymmetricity of ADP, our algorithms can output a more accurate top-k histogram than DP.

5.2.1 Asymmetric Report Noisy Max Algorithm. The report noisy arg-max is an algorithm satisfying DP to answer the argument of a noisy max value (see Section 2.2.2 for detail). Dwork et al. first proposed the algorithm [15] outputting only the argument with a maximum value. Subsequently, Ding et al. proposed an algorithm that can output more information than Dwork's one [10]. Ding's algorithm can output top-k arguments with gap-information without additional privacy leakage. Here, we propose a new algorithm that satisfies ADP instead of DP. By changing DP to ADP, the algorithm can output more information than Ding's algorithm. Precisely, we can output not the arguments but the top-k noisy values. Algorithm 1 is the pseudocode.

Algorithm 1 Asymmetric Report Noisy Max Algorithm

```
Input: $D$ : dataset, $\epsilon$ : privacy budget, $F$ : a set of queries whose
    $P$-sensitivity are monotonically decreasing and 1,
Output: Top-$k$ noisy values.
for each query $f_i \in F$ do
    $\lambda \sim \mathrm{aLap}_{\epsilon/k}(f_i, D)$
    $\tilde{f_i}(D) \Leftarrow f_i(D) + \lambda$
end for
$(j_1, j_2, \ldots, j_k) \Leftarrow argmax_k\{\tilde{f_0}, \tilde{f_1}, \ldots, \tilde{f_t}\}$
return $((j_1, \tilde{f_{j_1}}(D)), (j_2, \tilde{f_{j_2}}(D)), \ldots, (j_k, \tilde{f_{j_k}}(D))$
```

There is a difference in the assumption from Ding's algorithm. We assume that P-sensitivity of counting queries is monotonically decreasing due to the policy function. This assumption enables the mechanism to improve output information for two aspects. First, the proposed algorithm can use the asymmetric Laplace distribution with k/ϵ instead of the Laplace mechanism with $2k/\epsilon$, which means the Euclidean distance error is $1/(2\sqrt{2})$ times smaller than the one of the Laplace distribution. Second, the proposed algorithm can output top-k noisy values instead of arguments with gap-information. Gap information is a gap between two neighboring answers (e.g., $f_{j_1}(D) - f_{j_2}(D)$), so Ding's method needs to use privacy budget to measure the top-k values. Our method does not need to measure them because it outputs noisy values.

THEOREM 12. *Algorithm 1 satisfies (P, ϵ)-ADP.*

PROOF. Let $H = (\lambda_1, \lambda_2, \ldots, \lambda_t)$ denote the noise vector whose each element λ_i is noise for query f_i used in Algorithm 1. Then, we consider the following noise vector $H' = (\lambda'_1, \lambda'_2, \ldots, \lambda'_t)$.

$$\lambda'_i = \begin{cases} \lambda_i & (i \in \mathcal{I}_z^c) \\ \lambda_i + f_i(D) - f_i(D') & (i \in \mathcal{I}_z) \end{cases}$$

where $\mathcal{I}_z$ represents top-k arguments and $\mathcal{I}_z^c$ represents other arguments. Since the sensitivity of queries is monotonically decreasing, the answer of $m_H(D)$ (i.e., the top-k arguments and values corresponding to the arguments) is the same as $m_{H'}(D)$. It holds that $m_H(D) = m_{H'}(D')$, so we can define $\phi_{D,D',z}(H) = H'$ as acyclic local alignments. Additionally, due to the monotonicity, each variable is following the exponential distribution, and the cost is ϵ. Therefore, the conditions of Theorem 7 hold.

Q.E.D. □

5.2.2 Asymmetric Sparse Vector Technique. Next, we propose the asymmetric sparse vector technique, which is the ADP version of the sparse vector technique [14]. We consider a set of counting queries $F = \{f_1, f_2, \ldots, f_t\}$ and assume that P-sensitivity of the queries satisfies decreasing monotonicity and $\Delta_{f_i,P} = 1$. Naively answering each query with the asymmetric Laplace mechanism using ϵ finally consumes privacy budget $t\epsilon$ due to the sequential composition (Theorem 6). The asymmetric sparse vector technique uses a decision problem that asks whether $f_i(D)$ is under the threshold T_i or not instead of the counting query (see Section 5.1.1 for detail) to save the privacy budget. If the query answer is OTP (i.e., the noisy answer is below-the-threshold), the algorithm answers the decision problem without consuming the privacy budget. If the query answer is above-the-threshold, the algorithm consumes the privacy budget, so the algorithm answers the counting query with $aLap_{\epsilon/c}$. Algorithm 2 is the pseudocode.

Algorithm 2 The asymmetric sparse vector technique

```
Input: $\epsilon, c$: the number of above-the-threshold answers, $F = \{f_0, f_1, \dots, f_t\}$: a set of queries whose $P$-sensitivity are monotonically decreasing and 1, $T = \{T_0, T_1, \dots, T_t\}$: a set of thresholds
Output: out
count= 0
for each query $f_i \in F$ do
    $z \Leftarrow aLap_{\epsilon/c}(f_i, D)$
    if $z \geq T_i$ then
        Output $a_i = z$
        count=count+1, abort if count$\geq$ $c$.
    else
        Output $a_i = 0$
    end if
end for
```

Theorem 13. *Algorithm 2 satisfies (P,ϵ)-ADP.*

Proof. We use the technique of randomness alignment to prove this theorem. We assume that the algorithm outputs $A = (a_0, a_1, \dots, a_t)$ using $H = (\lambda_0, \lambda_1, \dots, \lambda_t)$ as the random variables for $aLap$ (i.e., $m_H(D) = A$). Then, we consider the following randomness alignment $H' = (\lambda'_0, \lambda'_1, \dots, \lambda'_m)$.

$$\lambda'_i = \begin{cases} \lambda_i & (a_i = 0) \\ \lambda_i + f_i(D) - f_i(D') & (a_i = z) \end{cases}$$

It holds that $m'_H(D') = A$, which means that $\phi_{D,D',z}(H) = H'$ is an acyclic local alignment. This is because f_i is monotonically decreasing so that the below-the-threshold answer does not change and $\lambda' + f_i(D') = \lambda + f_i(D)$. Therefore, the cost is $c * \epsilon/c = \epsilon$, so the conditions of Theorem 7 hold.

Q.E.D. □

We note that when the query is monotonically increasing, we can get below-the-threshold answers in the same way.

The asymmetricity improves three aspects from the sparse vector technique (Section 2.2.1). First, the asymmetric sparse vector technique does not require noise for the threshold. Second, we can use the asymmetric Laplace mechanism with ϵ/c, which has the $1/(2\sqrt{2})$ times smaller error than the Laplace mechanism. Third, we can answer a noisy counting query when the algorithm consumes the privacy budget. These three differences improve accuracy as shown in Section 6.

The traditional sparse vector technique needs to measure the top-k values in addition to the sparse vector technique to make a top-k histogram. Therefore, Ding et al. split the privacy budget in half for the sparse vector technique and measure. However, our asymmetric sparse vector technique does not require the measurement because it can output noisy values.

5.3 Location Monitoring

Since trajectory information is high-sensitive [6], DP requires much noise to publish a trajectory itself, which also applies to ADP. Therefore, instead of publishing a trajectory itself, we propose mechanisms to publish information about a location's safety using counting query as a use-case of a decision problem. We define *safe* as the number of target people (e.g., infected people) who have visited the location is under a threshold.

We propose two types of queries to monitor safety.

(1) query whether a location is safe in a range of time or not
(2) query whether locations were safe at a certain time or not

In the first query, we designate a location we want to monitor. In the second query, we designate a time where we want to monitor locations. By defining a policy function as Example 1 (i.e., not visiting is non-sensitive and visiting is sensitive), we can construct ADP mechanisms with one-sided error. Here, we assume that a dataset is frequently updated when data come or data is expired because the risk information change according to updated data.

5.3.1 Monitoring a Location.

Here, we consider the query $q : \mathcal{D} \rightarrow \{0, 1\}$, which asks whether the target location is safe or not. In other words, the query answers whether the number of target people who have recently visited the location is under the threshold or not. Algorithm 3 is the pseudocode of the mechanism for this query.

Algorithm 3 Monitoring a location

```
Input: $\epsilon, f : counting\ query, T : threshold$
Output: out
 1: initialize Queue
 2: while batch at input do
 3:     initialize data in batch to be False
 4:     enqueue all data in batch to Queue
 5:     while Queue.tail is expired do
 6:         Queue.dequeue()
 7:     end while
 8:     if Any data.mark in Queue is True then
 9:         back to line 2
10:     end if
11:     if $z = aLap_\epsilon(f, Queue) \geq T$ then
12:         Output $a_i = z$
13:         for data in Queue do
14:             data.mark = True
15:         end for
16:     else
17:         Output $a_i = 0$
18:     end if
19: end while
```

The algorithm outputs whether the location is safe or not every time a new batch of data comes (Line 2). The dataset is then updated by adding a new batch and removing expired data (Line 3-7). We note that we predefine the expired period of data. E.g., if we predefine the period as two weeks, the data is expired when two weeks have passed since the user of the data has visited the location. Using aLap, we perturb the count and compare the perturbed count with the given threshold. If the perturbed count is above the threshold, we output the noisy value and mark all data in Queue (Line 11-15). This marking represents the consumption of ϵ, so we cannot continue to output if the data is marked (Line 8-9). If the perturbed count is below the threshold, we output 0, which does not consume ϵ due to the asymmetric sparse vector technique (Line 17).

This algorithm satisfies (P, ϵ)-ADP, and output 0 is OTP (i.e., safe information is accurate). We can continue to monitor the safety of the location with the highest probability of OTP due to the property of aLap described in Section 5.1.1.

We note that to monitor multiple locations, we need to separate the privacy budget (e.g., if we want to monitor three locations, we need to use $\epsilon/3$ for each query).

THEOREM 14. *Algorithm 3 satisfies (P, ϵ)-ADP.*

PROOF. We consider all batches used in the algorithm as dataset D. We assume that output is $m_H(D) = A = (a_0, a_1, \ldots, a_t)$ where the used noise vector is $H = (\lambda_0, \lambda_1, \ldots, \lambda_t)$. Consider the following randomness alignment.

$$\lambda'_i = \begin{cases} \lambda_i & (a_i = 0) \\ \lambda_i + f_i(D) - f_i(D') & (a_i = z) \end{cases}$$

where f_i is the query used at ith update (i.e., $f_i(D) = f(D_{input_i})$ where D_{input_i} is the input at ith update). Then, it holds that $m_{H'}(D') = A' = A$ where $H' = (\lambda'_0, \lambda'_1, \ldots, \lambda'_t)$ due to the monotonicity. Since we stop the output when the above-the-threshold answer appears and until the corresponding record disappears, a change of one record only affects one query answer. Therefore, it holds that $\lambda'_i = \lambda_i$ expect one of the indices. Therefore, the alignment cost is ϵ, so the conditions of Theorem 7 hold, which means that Algorithm 3 satisfies (P, ϵ)-ADP.

Q.E.D.

□

5.3.2 Monitoring Locations at a Designated Time. The first mechanism adds up the needed privacy budget when locations to monitor increase. By designating a time, we can monitor the unlimited number of locations with a fixed privacy budget since each individual is at one location at a certain time. In other words, the query answers whether the number of target people who have been at the location at the time is below the threshold. Algorithm 4 is the pseudocode of the mechanism for this query.

Like Algorithm 3, the mechanism perturbs the answer of f_l, which is the counting query for location l using aLap. If the perturbed count is above the threshold, the mechanism outputs noisy answer z_{li} and marks the location because the mechanism consumes ϵ (Line 9-11). If the mechanism consumed ϵ for location l, we would skip the output for l (Line 6-8). If the perturbed count is below the threshold, we output 0 without consuming ϵ due to the asymmetric property (Line 13).

Algorithm 4 Monitoring locations at a certain time

```
Input: $\epsilon, L, T, \{f_l\}_{l \in L}$
Output: out
 1: initialize Array
 2: initialize marks of all locations to be False
 3: while batch at input do
 4:   append all data in batch to Array
 5:   for each location $l \in L$ do
 6:     if $l$.mark = True then
 7:       back to Line 4
 8:     end if
 9:     if $z_{li} = aLap_\epsilon(f_l, Array) \geq T$ then
10:       $l$.mark = True
11:       Output $a_{li} = z_{li}$
12:     else
13:       Output $a_{li} = 0$
14:     end if
15:   end for
16: end while
```

We note that to monitor multiple times, we need to separate the privacy budget (e.g., if we monitor three designated times, we need to use $\epsilon/3$ for each query).

THEOREM 15. *Algorithm 4 satisfies (P, ϵ)-ADP*

PROOF. We consider all batches used in the algorithm as dataset D. We assume that output is $m_H(D) = A = (a_{l_0 0}, a_{l_0 1}, \ldots, a_{l_{|L|} t})$ where the used noise vector is $H = (\lambda_{l_0 0}, \lambda_{l_0 1}, \ldots, \lambda_{l_{|L|} t})$. Consider the following randomness alignment.

$$\lambda'_{li} = \begin{cases} \lambda_{li} & (a_{li} = 0) \\ \lambda_{li} + f_{li}(D) - f_{li}(D') & (a_{li} = z) \end{cases}$$

where f_{li} is the query used at ith update (i.e., $f_{li}(D) = f_l(D_{input_i})$ where D_{input_i} is the input at ith update). Then, it holds that $m_{H'}(D') = A' = A$ where $H' = (\lambda'_{l_0 0}, \lambda'_{l_0 1}, \ldots, \lambda'_{l_{|L|} t})$ due to the monotonicity. Since an individual can be one place at a certain time and we stop the output when the above-the-threshold answer appears and until the corresponding record disappears, the change of one record only affects one query answer. Therefore, it holds that $\lambda'_{li} = \lambda_{li}$ expect one of the indices. Therefore, the alignment cost is ϵ, so the conditions of Theorem 7 hold, which means that Algorithm 4 satisfies (P, ϵ)-ADP.

Q.E.D. □

6 EXPERIMENTS

We conducted simulations of the two use-cases using real-world datasets and evaluate their performance. We open the source code used in these experiments on https://github.com/tkgsn/adp-algorithms.

6.1 Datasets and a Policy

First, we explain the datasets used for our experiments. We use different datasets for each use-case.

Top-k Query. For evaluating the asymmetric sparse vector technique and the asymmetric report noisy max algorithm, we use two real-world datasets from [25] and a synthetic dataset created by the

Dataset	# Records	# Unique Items
BMS-POS	515,597	1,657
Kosarak	990,002	41,270
T40I10D100K	100,000	942

Table 3: Statistics of datasets

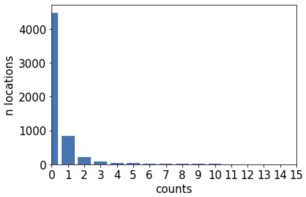

Figure 1: The number of locations with respect to the counts of visited people.

generator from the IBM Almaden Quest research group: BMS-POS, Kosarak, and T40I10D100K. These datasets are collections of click-stream, and each record is a stream of items. We show the statistics of each dataset in Table 3.

Location Monitoring. For evaluating our proposed location monitoring mechanisms, we use the Peopleflow dataset[6] as trajectory data of infected people. Trajectory data in the Peopleflow dataset includes *location_id*, which represents the category of the location (e.g., restaurant or amusement facility). There are 5835 locations in the Peopleflow dataset, and we assume that if a point includes a *location_id*, the individual has visited the location. We randomly separate individuals into a batch with the size of 500. We plot the number of locations according to the counts of visited people using a randomly chosen batch in Figure 1. We can see that most locations are not visited. Counts concentrate on locations such as a station.

Policy. Here, we use the policy in Example 1. That is, we assume that the fact that a user has clicked an item is sensitive, and the fact that a user has not clicked an item is non-sensitive for the click datasets. For the trajectory dataset, we assume that the fact that a user has visited a location is sensitive, and the fact that a user has not visited a location is non-sensitive. This policy induces the monotonically decreasing at P-sensitivity when counting the fact that a user has clicked an item (visited a location) for the same reason as Example 5.

6.2 Top-k Query

We here compare our algorithms with the state-of-the-art algorithms satisfying DP [10], which we call the free-gap algorithm. To see how the parameters impact the results, we show the results varying ϵ and k. Here, we use two measurements: the mean squared error and accuracy of top-k items. I.e., we evaluate how accurate the published histogram is and top-k items are. We iterated this evaluation 10000 times, and we show the average of them in a solid line and its 95% bootstrapped confidential interval in the shade.

[6] http://pflow.csis.u-tokyo.ac.jp/

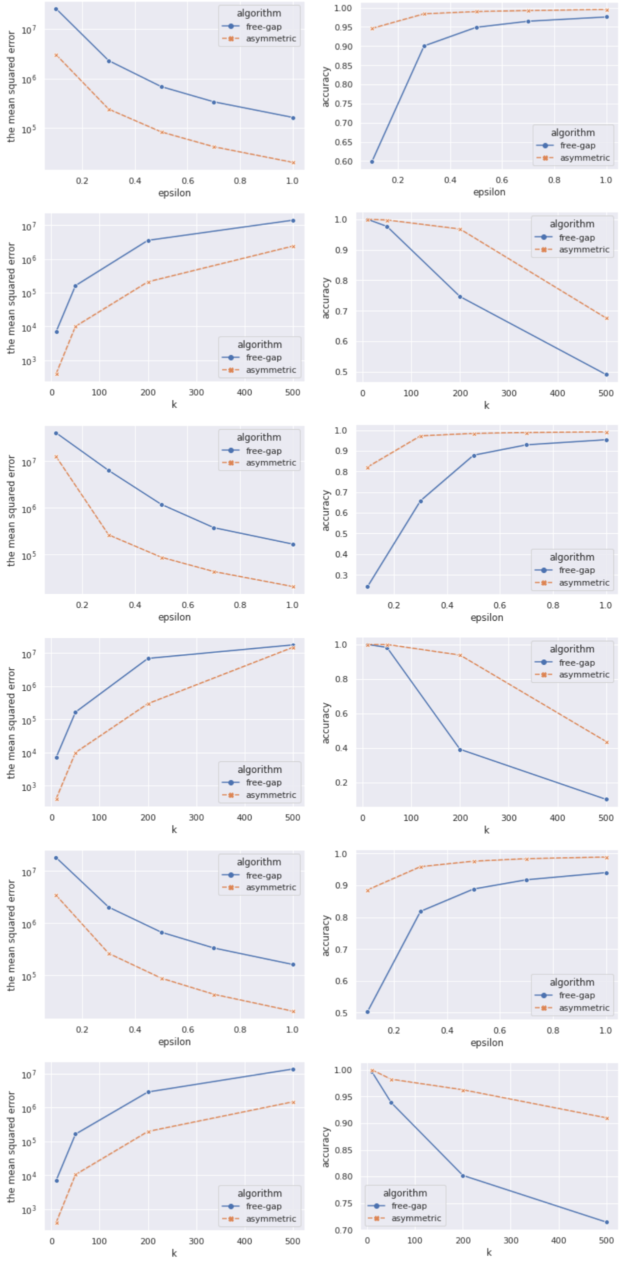

Figure 2: The results of the asymmetric report noisy k-max algorithm for BMS-POS, Kosarak, and T40I10D100K varying ϵ and k.

6.2.1 Asymmetric Report Noisy Max Algorithm.

Varying ϵ value. Here, we fix k as 100 and plot the results in above Figure 2 with varying the value of ϵ. We can see that our algorithm can achieve nearly 1 at accuracy for any dataset in $\epsilon = 0.5$ although the free-gap algorithm cannot achieve 1 at even $\epsilon = 1$. The histogram from our algorithm is about ten-times more accurate than the one from the free-gap algorithm.

Varying k value. Here, we fix ϵ as 0.5 and plot the results in below Figure 2 with varying the value of k. We can see that the free-gap algorithm more shapely decreases accuracy as k increases than our algorithm.

6.2.2 Asymmetric Sparse Vector Technique. The sparse vector technique requires a threshold. However, we cannot know the threshold for the top-k query. We took the same measure for this problem as the one of Ding et al. [10]. We randomly choose an integer i from $[k, 2k]$ and we use the value indexed by the chosen integer (i.e., ith from the top) as the threshold.

Varying ϵ value. We fix k as 100 and plot the results in above Figure 3. We can see that our algorithm is about ten-times more accurate than the free-gap algorithm. When $\epsilon = 1$, accuracy is almost the same. This is because the sparse vector technique requires the threshold, which is sensitive information so that the maximum accuracy is limited. Our algorithm reaches the maximum accuracy at lower ϵ than the free-gap algorithm.

Varying k value. Here, we fix ϵ as 0.5 and plot the results in below Figure 3 with varying the value of k. The same as the above results, when k is small, accuracy is almost the same, and our algorithm achieves similar results on large k although the free-gap algorithm sharply decreases accuracy.

6.2.3 Remark. Our algorithms output a more accurate answer to a top-k query for any dataset, k, and ϵ than Ding's algorithm by providing the privacy guarantee of DP to only sensitive values defined by the policy. Therefore, if a user wants to hide non-click information, these algorithms are not appropriate. Still, if a user can compromise the policy, we can improve the accuracy.

The asymmetric report noisy k-max algorithm is more accurate than the asymmetric sparse vector technique for the top-k query. However, we note that the asymmetric sparse vector technique outputs more information: noisy values of above-the-threshold answers and arguments of below-the-threshold answers. Additionally, since the decision problem in the asymmetric sparse vector technique is one-sided, arguments of below-the-threshold answers are accurate.

6.3 Location Monitoring

Here, the goal is to publish one-sided *safety* information. In other words, the algorithm outputs whether the number of target people who have visited the location is under the threshold or not. Therefore, we guarantee that the false-positive ratio = 0. However, we cannot guarantee that the false-negative ratio = 0 due to the constraint of ADP. Then, we use the false-negative ratio as the utility measure.

6.3.1 Monitoring a Location. Here, we evaluate Algorithm 3. Algorithm 3 continues to output until infected people do not appear. Then, we evaluate the output for each update and plot the result in Figure 4. The false-negative ratio increases when the true counts (i.e., $f(D)$) or the threshold decreases. If the true count is 0, the false negative ratio is less than 0.01 for $\epsilon = 1$ and threshold= 5. This means that we can publish accurate information with more than 99% for each update.

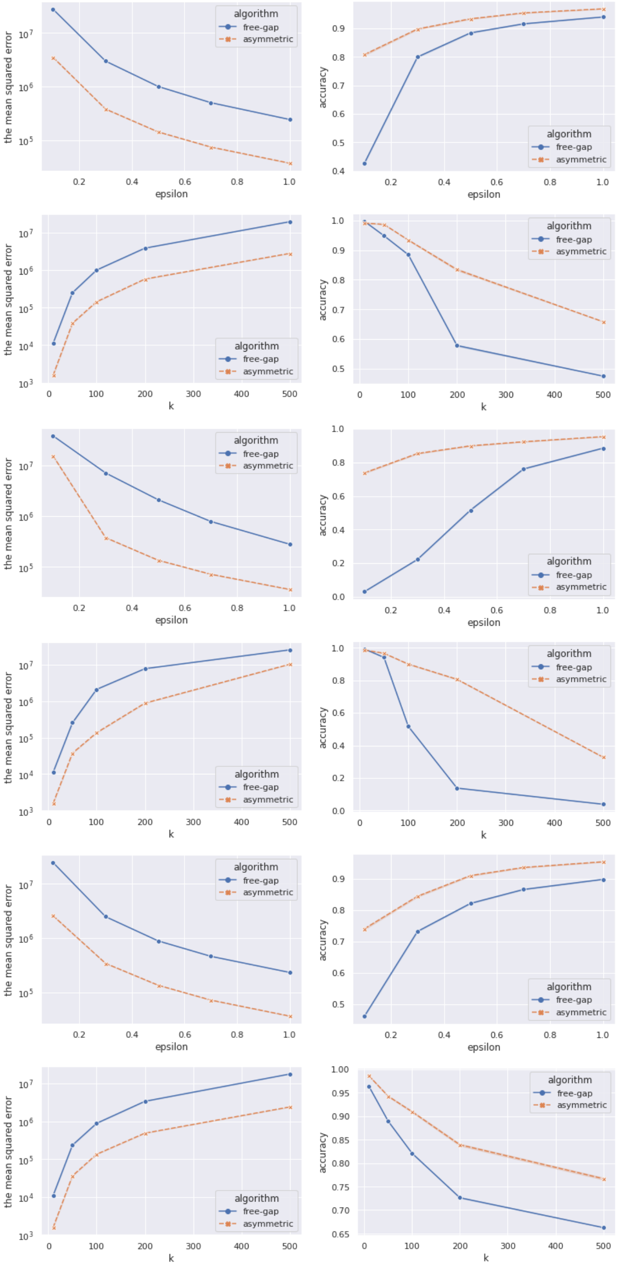

Figure 3: The results of the asymmetric sparse vector technique for BMS-POS, Kosarak, and T40I10D100K varying ϵ and k.

6.3.2 Monitoring Locations at a Designated Time. Here, we evaluate Algorithm 4. First, we visualize the outputs for each case of the ground truth ($\epsilon = \infty$), $\epsilon = 0.5$, $\epsilon = 1$ and $\epsilon = 2$ in Figure 5. The threshold T is set as 10. In this simulation, we fix the batch size as 500 and update 5 times to add 5 batches, which means that the dataset's final size is 2500. The blue dots represent safe locations, and orange dots are the hot spots. Comparing with the ground truth, we can see that the orange dots increase. These additional orange dots are the error (i.e., false negative), and the blue dots represent

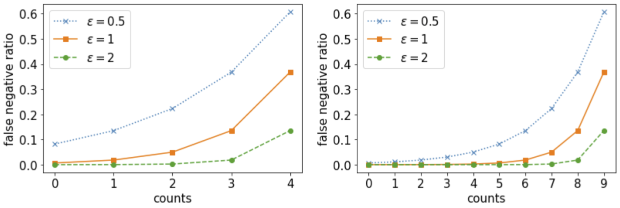

Figure 4: False negative ratio for each update when varying the counts. The left figure is for threshold= 5 and the right figure is for threshold= 10.

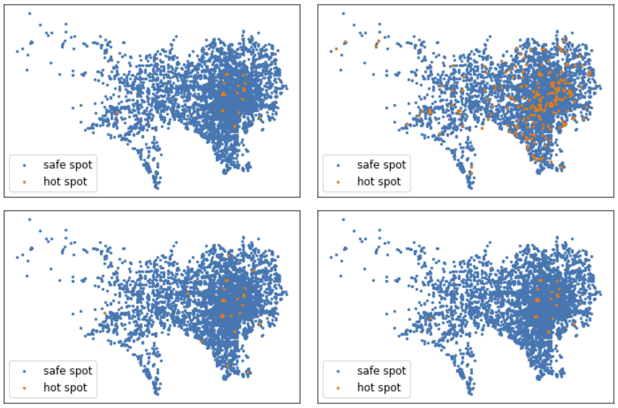

Figure 5: The monitored locations and hot spots (counts ≥ 10). From left, ground truth, $\epsilon = 0.5$, $\epsilon = 1$ and $\epsilon = 2$.

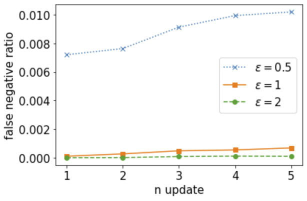

Figure 6: The false negative ratio at each update.

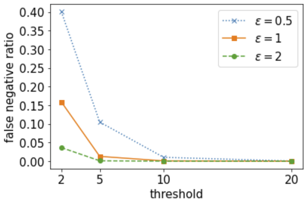

Figure 7: The false negative ratio varying the threshold.

OTP (i.e., accurate safe information). The smaller ϵ is, the more orange dots exits due to the stronger requirement ADP.

Next, we evaluate the false negative ratio. Here, the false-negative ratio is the ratio of the number of wrong orange dots to the number of all right blue dots in Figure 5. We plot the false negative ratio at each update in Figure 6. From this figure, the false-negative ratio is at most 0.01. This means that we can publish 99% safe areas with the guarantee of accuracy.

Finally, we evaluate the false negative ratio varying the threshold. When the threshold is small (e.g., 2), the false-negative ratio is very high. By setting the smaller threshold, we can know the precise information, but the output includes many errors. Setting the bigger threshold obscures the published information but the number of errors decreases. The threshold is an important key to adjust this trade-off.

7 CONCLUSION

We proposed asymmetric differential privacy, which is the relaxation of differential privacy to mitigate the constraints of differential privacy. We proved that ADP improves the Euclidean error and allows one-sided error. Then, we proposed practical algorithms satisfying ADP. We show by experiments with real-world datasets that we can get a more accurate top-k histogram and one-sided information about the safety of locations using our proposed algorithms.

In this paper, we consider the single-dimensional counting query as the use case of ADP. However, ADP is the general definition so that we may introduce ADP to other queries to improve utility, which is one of the future directions.

REFERENCES

[1] Jayadev Acharya, Kallista Bonawitz, Peter Kairouz, Daniel Ramage, and Ziteng Sun. 2020. Context Aware Local Differential Privacy. In *International Conference on Machine Learning*. PMLR, 52–62.

[2] Aws Albarghouthi and Justin Hsu. 2017. Synthesizing Coupling Proofs of Differential Privacy. *Proc. ACM Program. Lang.* 2, POPL, Article 58 (Dec. 2017), 30 pages. https://doi.org/10.1145/3158146

[3] Johannes K Becker, David Li, and David Starobinski. 2019. Tracking anonymized bluetooth devices. *Proceedings on Privacy Enhancing Technologies* 2019, 3 (2019), 50–65.

[4] USC Bureau. [n.d.]. On the map: Longitudinal employer-household dynamics, https://lehd.ces.census.gov/applications/help/onthemap.html#confidentiality_protection.

[5] Yang Cao, Yonghui Xiao, Shun Takagi, Li Xiong, Masatoshi Yoshikawa, Yilin Shen, Jinfei Liu, Hongxia Jin, and Xiaofeng Xu. 2020. PGLP: Customizable and Rigorous Location Privacy Through Policy Graph. In *European Symposium on Research in Computer Security*. Springer, 655–676.

[6] Rui Chen, Benjamin CM Fung, Noman Mohammed, Bipin C Desai, and Ke Wang. 2013. Privacy-preserving trajectory data publishing by local suppression. *Information Sciences* 231 (2013), 83–97.

[7] Rui Chen, Liang Li, Jeffrey Jiarui Chen, Ronghui Hou, Yanmin Gong, Yuanxiong Guo, and Miao Pan. 2020. COVID-19 Vulnerability Map Construction via Location Privacy Preserving Mobile Crowdsourcing. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 1–6.

[8] Eva Clark, Elizabeth Y Chiao, and E Susan Amirian. 2020. Why contact tracing efforts have failed to curb coronavirus disease 2019 (covid-19) transmission in much of the united states. *Clinical Infectious Diseases* (2020).

[9] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting telemetry data privately. In *Advances in Neural Information Processing Systems*. 3571–3580.

[10] Zeyu Ding, Yuxin Wang, Danfeng Zhang, and Daniel Kifer. 2020. Free gap information from the differentially private sparse vector and noisy max mechanisms. *Proceedings of the VLDB Endowment* (2020).

[11] Stelios Doudalis, Ios Kotsogiannis, Samuel Haney, Ashwin Machanavajjhala, and Sharad Mehrotra. 2020. One-sided differential privacy. *ICDE* (2020).

[12] Cynthia Dwork. 2006. Differential privacy. In *Proceedings of the 33rd international conference on Automata, Languages and Programming-Volume Part II*. Springer-Verlag, 1–12.

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

[14] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. 2009. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 381–390.

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

[16] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 1054–1067.

[17] Xiaolan Gu, Ming Li, Li Xiong, and Yang Cao. 2020. Providing input-discriminative protection for local differential privacy. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 505–516.

[18] Yaron Gvili. 2020. Security analysis of the COVID-19 contact tracing specifications by Apple Inc. and Google Inc. *IACR Cryptol. ePrint Arch.* 2020 (2020),

428.

[19] Moritz Hardt and Kunal Talwar. 2010. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 705–714.

[20] Xi He, Ashwin Machanavajjhala, and Bolin Ding. 2014. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 1447–1458.

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

[22] Daniel Kifer and Ashwin Machanavajjhala. 2014. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)* 39, 1 (2014), 1–36.

[23] Fragkiskos Koufogiannis, Shuo Han, and George J Pappas. 2015. Optimality of the laplace mechanism in differential privacy. *arXiv preprint arXiv:1504.00065* (2015).

[24] Robert Kudelić. 2016. Monte-Carlo randomized algorithm for minimal feedback arc set problem. *Applied Soft Computing* 41 (2016), 235–246.

[25] Min Lyu, Dong Su, and Ninghui Li. 2016. Understanding the sparse vector technique for differential privacy. *arXiv preprint arXiv:1603.01699* (2016).

[26] Naurang S Mangat. 1994. An improved randomized response strategy. *Journal of the Royal Statistical Society: Series B (Methodological)* 56, 1 (1994), 93–95.

[27] Takao Murakami and Yusuke Kawamoto. 2019. Utility-optimized local differential privacy mechanisms for distribution estimation. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 1877–1894.

[28] Simon Nicholas, Chris Armitage, Tova Tampe, and Kimberly Dienes. 2020. Public attitudes towards COVID-19 contact tracing apps: a UK-based focus group study. (2020).

[29] Yuto Omae, Jun Toyotani, Kazuyuki Hara, Yasuhiro Gon, and Hirotaka Takahashi. 2020. A Calculation Model for Estimating Effect of COVID-19 Contact-Confirming Application (COCOA) on Decreasing Infectors. *arXiv preprint arXiv:2010.12067* (2020).

[30] Sangchul Park, Gina Jeehyun Choi, and Haksoo Ko. 2020. Information technology–based tracing strategy in response to COVID-19 in South Korea—privacy controversies. *Jama* (2020).

[31] A. D. P. Team. [n.d.]. Learning with privacy at scale.

[32] Yuxin Wang, Zeyu Ding, Guanhong Wang, Daniel Kifer, and Danfeng Zhang. 2019. Proving differential privacy with shadow execution. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*. 655–669.

[33] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.

[34] Danfeng Zhang and Daniel Kifer. 2017. LightDP: Towards Automating Differential Privacy Proofs. *SIGPLAN Not.* 52, 1 (Jan. 2017), 888–901. https://doi.org/10.1145/3093333.3009884