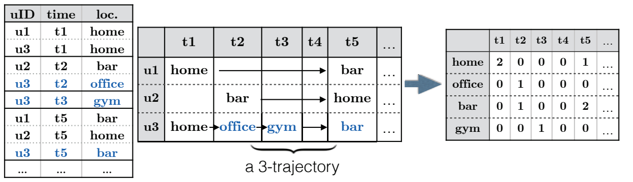
Differentially Private Real-time Data Release over Infinite Trajectory Streams

Yang Cao, Mashatoshi Yoshikawa
{soyo@db.soc., yoshikawa@}i.kyoto–u.ac.jp
Department of Social Informatics, Kyoto University

***Abstract*—Recent emerging mobile and wearable technologies make it easy to collect personal spatiotemporal data such as activity trajectories in daily life. Releasing real-time statistics over trajectory streams produced by crowds of people is expected to be valuable for both academia and business, answering questions such as "How many people are in Central Station now?" However, analyzing these raw data will entail risks of compromising individual privacy. *ϵ-Differential Privacy* has emerged as a de facto standard for private statistics publishing because of its guarantee of being rigorous and mathematically provable. Since user trajectories will be generated infinitely, it is difficult to protect every trajectory under ϵ-differential privacy. To this end, we propose a flexible privacy model of *ℓ-trajectory privacy* to ensure every length of ℓ trajectories under protection of ϵ-differential privacy. Then we hierarchically design algorithms to satisfy *ℓ-trajectory privacy*. Experiments using four real-life datasets show that our proposed algorithms are effective and efficient.**



uID	time	loc.
u1	t1	home
u3	t1	home
u2	t2	bar
u3	t2	office
u3	t3	gym
u1	t5	bar
u2	t5	home
u3	t5	bar
...	...	...

	t1	t2	t3	t4	t5	…
u1	home	→	→	→	bar	…
u2		bar	→	→	home	…
u3	home →	office →	gym	→	bar	…

a 3-trajectory

	t1	t2	t3	t4	t5	…
home	2	0	0	0	1	…
office	0	1	0	0	0	…
bar	0	1	0	0	2	…
gym	0	0	1	0	0	…

(a) Raw data. (b) Raw data in trajectory representation.(c) Real-time statistics.

Fig. 1: Raw data and aggregate information. Each row in Table (a) is a spatiotemporal data point. Each user's trajectory in Table (b) is composed of multiple data points successively (maybe not adjacently) on the timeline. ℓ-trajectory is a user's any ℓ successive data points. For example, *home* $\rightarrow$ *office* $\rightarrow$ *gym* and *office* $\rightarrow$ *gym* $\rightarrow$*bar* are $u3$'s two overlapped 3-trajectories.

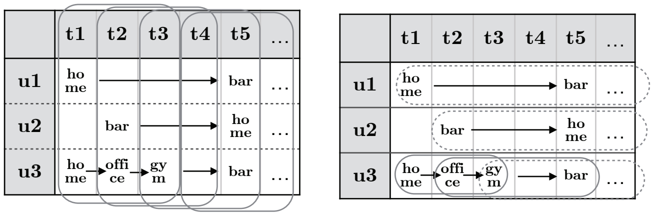
I. Introduction

In recent years, personal data have been increasingly collected, stored, and analyzed. The information is sensitive, especially location history data that one has visited. These successive points of interest (POIs), along with timestamps, constitute our moving trajectories in daily life. Releasing real-time statistics of trajectory streams produced by crowds of people is expected to be extremely valuable for data-based analysis and decision making. Consider the scenario illustrated in Fig.1. A trusted server is collecting many people's moving trajectories continuously. The raw data are spatiotemporal points, as illustrated in Fig.1(a). A user's *ℓ-trajectory* is defined as ℓ successive spatiotemporal data points on the timeline, as illustrated in Fig.1(b). Then statistics at each timestamp such as "How many people are in location l_1 at the current time?" can be released such as Fig.1(c). These statistical answers are expected to be useful for marketing analysis[1], real-time traffic analysis[2], intelligent navigation systems [3], web browsing behavior mining[4] (users' trajectories in the cyberspace) and so on.

To protect privacy, such sensitive data should be anonymized. However, previous research has pointed out that even anonymized information can be used to identify a particular person. Several well-known privacy disclosure cases have arisen, such as an attack on the Netflix database [5] and medical records of the governor of Massachusetts [6]. Such attacks are called "linkage attack" [7] by which an adversary uses auxiliary information to obtain sensitive data about a target. Especially, people's moving trajectories are extremely vulnerable because they have patterns that are readily apparent. Research[8] shows that only *four* spatiotemporal points from anonymized mobile datasets are sufficient to identify 95% of individuals uniquely. In addition, a trajectory provides more contextual information as successive spatiotemporal points. On the other hand, to provide a good trade-off between privacy and utility, a flexible model is needed.

ϵ-Differential privacy (DP) has emerged as a de facto standard for privacy preserving data publishing (PPDP) because of rigorous theoretical guarantees [9], [10]. It ensures that the modification of any single record does not have a significant effect on the outcome of analysis. ϵ is a positive parameter called *privacy budget* which is given in advance to control the privacy level. The value of ϵ is inversely propositional to the privacy level. The literature related to DP provides rich results including application of DP to streaming data [9], [10], [11], [12]. Various privacy goals and diverse application scenarios exist. In the setting of streaming data, differential privacy comes with two privacy definitions: *user-level* and *event-level* privacy [9], [10]. Roughly speaking, for trajectory data streams, *user-level* privacy means to protect the whole trajectory history of any user, and *event-level* privacy only promises to protect any single spatiotemporal data point. However, in our scenario, the former is nearly impossible because the trajectory streams are infinite, whereas the latter is not sufficiently safe because of the vulnerable nature of trajectory data with respect to attacks with background knowledge.

A new streaming data privacy model of *w-event privacy* [11] was proposed recently to strike a nice balance between two former privacy definitions(Fig.2(a)). The model emphasizes protection of data points belonging to every w *contiguous* timestamps in a sliding window. Unfortunately, *w-event privacy* is not sufficient to protect trajectory streams. First,



(a) w-event privacy (w=3).

(b) l-trajectory privacy (l=3).

Fig. 2: (a) illustrates a privacy model of previous research based on event-level privacy. (b) illustrates our proposed privacy model based on user-level privacy. Trajectory in dashed ellipse in (b) means ongoing 3-trajectories.

in general, users' trajectories are sparse on the timeline. Data points of trajectories are not always appearing successively at contiguous timestamps[1], but w-event privacy and its sliding window method cannot protect every ℓ-trajectory, because ℓ-trajectory does not necessarily fits in a sliding window of w length. For example, regarding Fig. 2(a), a 3-trajectory *office→gym→bar* of $u3$ does not fit any three contiguous timestamps. The 2-trajectory *home→bar* of $u1$ spans five contiguous timestamps. In general, there is no upper bound in the number of contiguous timestamps a ℓ-trajectory spans. Second, the fixed parameter w of w-event privacy in our setting is insufficient to provide uniform and personalized protection. The reason is that the density (rate of data points being generated) of trajectory streams will be varied on the timeline. For example, relatively dense trajectories usually appear in the daytime (people are moving fast between POIs), whereas relatively sparse trajectories mostly appear in the nighttime (people generate fewer data points in a certain period). Moreover, the density of trajectories will vary with different moving speeds of individuals. In this sense, using trajectory length ℓ as a parameter (Fig.2(b)) to control the privacy level allows us to define a uniform and semantically clear privacy framework.

In this paper, we first formulate ℓ-trajectory privacy under ϵ-differential privacy, and provide two algorithms to achieve ℓ-trajectory privacy. At each timestamp, there is a privacy preserving data publishing algorithm whose privacy budget is properly allocated. Then, ℓ-trajectory privacy is preserved when the sum of privacy budgets inside timestamps of any ℓ-trajectory is at most the total privacy budget ϵ. A salient challenge to achieve ℓ-trajectory privacy is that the upcoming spatiotemporal data points are unknown, hence, a naive method that pre-allocates the local budgets will lead to inferior data utility. To this end, we proposed a framework that adopts a dynamic budget allocation based on approximation strategies, and we implement two algorithms Adj and MMD as two different approximation strategies.

II. Related Work

For stream data, *user-level* privacy and *event-level* privacy have been proposed [9], [10] as two different privacy goals. Previous work on differential privacy of streaming data mainly focuses on event-level privacy on finite or infinite streams [13], [14], [11], and user-level privacy on finite streams [12], [2], [4]. Fan and Xiong [2], [4] proposed FAST framework for publishing time-series data in a user-level private way. FAST use sampling and filtering components to reduce the noise; given a specified number of samples, the filtering component predict the future data and correct its prior by noisy samples. However, this scheme takes as input the total amount of timestamps T, which leads to inapplicability in our infinite scenario.

Chen et al. [15] proposed a data-dependent sanitation mechanism by grouping trajectories to produce a noisy prefix tree. Jiang et al. [16] also examined differentially publish private trajectories by sampling suitable direction and distance of trajectory data. Ho et al. [17], [18] proposed a differentially private approach to mine interesting geographic location patterns from trajectory data. These work address the scenario of directly publishing user-level trajectories or the results of specific data mining tasks, which are different from publishing sequential statistics for sets of users' trajectories.

III. Problem Definition and Privacy Model

A. Data Model

We consider scenario of a trusted server collects datasets D_i of users' spatiotemporal data points continuously at each timestamp $i \in [1, t]$. Let t denotes the current timestamp. Each data point $\langle uid, time, loc \rangle$ is a row in D_i (Fig.1(a)). Assuming that $\boldsymbol{locs}$ is a set of all locations in which we were interested (POIs), then the server wishes to publish a vector $\boldsymbol{r}_i$ as the counts of $loc \in \boldsymbol{locs}$ appeared in D_i, i.e., the answer of count query $Q^c : D_i \rightarrow \mathbb{R}^{|\boldsymbol{locs}|}$ at each timestamp t. We assume that a user only appear at most *one* location at each timestamp, then $\langle uid, time \rangle$ is the primary key in the whole database.

Definition 1 (ℓ-trajectory): A sequence of successive spatiotemporal data points produced by the same user is a ℓ-trajectory if the number of data points is equal to ℓ. A ℓ-trajectory ends at timestamp k, as denoted by $\ell_{u,k} = \{\langle u, i, loc \rangle, \cdots, \langle u, k, loc' \rangle\}$. We say that $\ell_{u,k}$ *dominate* the set of timestamps appeared in these data points (i.e., $\{i, \cdots, k\}$), denote as $\tau_{u,k}$.

1) Privacy risk: In the scenario above, the users' trajectory will be at risk. As described above, even releasing anonymized and aggregate information is not safe because of the adversary's unforeseen background knowledge. For instance, in the example of Fig. 1, assuming that an adversary knows that $u3$ has stayed at some places at timestamps $t1, t3, t5$, then by the published data Fig. 1(c), the trajectory of $u3$ can be inferred as $home \rightarrow gym \rightarrow bar$ with probability $2/3$.

To avoid such risk, we will propose a privacy model based on differential privacy that does not make any assumption on the background knowledge of the attacker.

We will give a formal definition of our privacy goal under ϵ-differential privacy in section III-D .

B. Differential Privacy

Differential privacy was proposed by Dwork et al. [19]. A widely used method to achieve differential privacy is the

[1] In some cases, users' spatiotemporal data points are collected at every timestamp as initial raw data. However, in most cases, only interesting attributes (e.g., POIs) will be reserved during data cleaning processes.

Laplace mechanism [20], which adds random noise to actual data to prevent the disclosure of sensitive information.

Roughly speaking, differential privacy attempts to limit the impact of an individual's record relative to the answer of a query on the database. In other words, whether the sensitive information is present or not in the database, the answers of a query should have little difference. In this way, an adversary cannot associate any piece of information with a specific individual from mining the answers of queries.

Definition 2 (ϵ-Differential Privacy): D, D' are two possible neighbor databases that differ in one row that is modified. Λ is a randomized algorithm. $\boldsymbol{R}$ represents all possible outputs of Λ. Algorithm Λ satisfies ϵ-differential privacy, if for any $r \in \boldsymbol{R}$ and any two neighbor databases D, D', we have the following.

$$Pr[\Lambda(D) = r] \leq exp(\epsilon) \cdot Pr[\Lambda(D') = r] \qquad (1)$$

In Inequality (1), ϵ is a privacy budget given in advance. It is used to control the privacy level. We note that lower ϵ signifies higher privacy, and vice versa. *Global sensitivity* (or *sensitivity* for short) $GS(Q)$ of query Q is the maximum L_1 distance between the query results for any two neighboring databases.

To avoid misunderstandings, we note that there are different ways to define differential privacy, based on the definition of neighboring databases that differ by adding/removing an individual's record (*unbounded*) [19] or modifying an individual's record but and use a database of the same size (i.e., number of records) (*bounded*) [20]. Unless we explicitly state otherwise, we use *bounded* differential privacy in this paper. Further discussion of these differences can be found in [21].

Theorem 1 (Laplace Mechanism [20]): ϵ-Differential privacy can be achieved by adding independent Laplace random noise x to the answer of query Q. In the following, n and D denote the noisy data and a given database, respectively.

$$n = Q(D) + x \qquad (2)$$

$$x \sim Pr(x|\lambda) = \tfrac{1}{2\lambda} \cdot exp(-|x|/\lambda) \qquad (3)$$

where λ is a scale parameter of Laplace distribution, which equals to $GS(Q)/\epsilon$.

Another way to obtain differential privacy is through the exponential mechanism[22]. Given a quality function q that scores results of a specified query, where higher scores are better. Then returning results in the probability of exponentially proportional to the corresponding score. In other words, the result with a higher score is exponentially more likely to be chosen. This will ensure differential privacy.

Theorem 2 (Exponential Mechanism [22]): Let D^n denotes all possible databases in the universe, and $D, D' \in D^n$ are two instances of neighboring databases. Let $q : (D^n \times R) \to \mathbb{R}$ be a quality function that, R as a certain query's all possible sensitive outcome, given a database$D \in D^n$, assign a score to each outcome $r \in R$. Then we define global sensitivity $GS(q)$ as $max_r|q(D, r) - q(D', r)|, D, D' \in D^n$. Exponential mechanism M is obtained by returning the sensitive answer r with probability proportional to $exp(\frac{\epsilon q(D,r)}{2GS(q)})$.

C. Utility metrics

In order to protect the individual's sensitive data , we need to add randomness to the real statistics. Therefore, the utility will be measured as the magnitude of random noise. We adopt the classical statistical metric of Mean Absolute Error (MAE) as shown below.

Here, $\boldsymbol{locs}$ is a set of POIs, $\boldsymbol{r_i}$ and $\boldsymbol{n_i}$ as vectors of length $|\boldsymbol{locs}|$ are a real and noisy data at timestamp i respectively. $\boldsymbol{R}$ and $\boldsymbol{N}$ are all timestamps' real and noisy data respectively. Then the error magnitude at each timestamp i is measured by Mean of Absolute Error as Equation (4).

$$MAE(\boldsymbol{R}, \boldsymbol{N}) = \frac{1}{T * |\boldsymbol{locs}|} \cdot \sum_{i=1}^{T} \sum_{j=1}^{|\boldsymbol{locs}|} |\boldsymbol{r_i}[j] - \boldsymbol{n_i}[j]| \qquad (4)$$

D. Proposed Privacy Model

Motivated by the vulnerability of the trajectory data, our privacy goal is to protect any ℓ-length trajectory of any user under ϵ-differential privacy. To define this under ϵ-differential privacy, we must first clarify the definition of *neighboring datasets* at each timestamp and *neighboring trajectory stream prefixes*.

Definition 3 (Neighboring dataset at each timestamp): If two datasets D_i, D'_i are collected at timestamp $i \in [1, t]$ and differ in a single $\underline{loc}$ of user u, then we say that D_i, D'_i is a pair of neighboring datasets with respect to *u*.

Since a user appears at most one loc at each timestamp, modifying a loc either by changing a single loc to another $loc' \in locs$ or to a $loc' \notin locs$, the sensitivity of Q^c is 2.

A trajectory stream prefix corresponds to all data of the infinite trajectory streams up to the current timestamp t .

Definition 4 (ℓ-trajectory neighboring stream prefixes): Let $S_t = \{D_1, \cdots, D_t\}$ and $S'_t = \{D'_1, \cdots, D'_t\}$ be two trajectory stream prefixes ending with the current timestamp t. S_t and S'_t are ℓ-trajectory stream prefixes neighboring each other if one is obtained from another by modifying all $\underline{locs}$ in any *one* ℓ-trajectory $\ell_{u,k}$ (recall that a ℓ-trajectory is a set of ℓ spatiotemporal data points). We say that S_t and S'_t are neighboring with respect to $\ell_{u,k}$.

Then we can define *ℓ-trajectory privacy* under ϵ-differential privacy as follows. Our guarantee captures the impact of any ℓ-trajectory relative to the sensitive answer of a query.

Definition 5 (ℓ-trajectory ϵ-differential privacy): Let Λ be an algorithm that takes prefixes of trajectory streams $S_t = \{D_1, \cdots, D_t\}$ as inputs. Let $\boldsymbol{N}_t = \{\boldsymbol{n}_1, \cdots, \boldsymbol{n}_t\}$ be a possible perturbed output stream of Λ. If for any ℓ-trajectory neighboring S_t and S'_t, the following holds,

$$Pr\left[\Lambda(S_t) = \boldsymbol{N}_t\right] \leq e^{\epsilon} \cdot Pr\left[\Lambda(S'_t) = \boldsymbol{N}_t\right] \qquad (5)$$

then we say that Λ satisfies ℓ-trajectory ϵ-differential privacy (simply, *ℓ-trajectory privacy*).

E. Methodology to Achieve ℓ-Trajectory Privacy

The next question is how to achieve ℓ-trajectory privacy. Inspired by *w-event*'s [11] scheme of ensuring the sum of privacy budget in a sliding window less than total budget ϵ, we prove Theorem 3. Roughly speaking, to satisfy ℓ-trajectory privacy, the sum of privacy budgets allocated to timestamps of any single ℓ-trajectory must be less than the total privacy budget ϵ.

Theorem 3: Let Λ be an integrated algorithm which takes prefixes of streams $S_t = \{D_i, i \in [1,t]\}$ as inputs, and $\boldsymbol{N_t} = \{\boldsymbol{n}_i, i \in [1,t]\}$ as outputs. Λ consists of a series of algorithms$\{A_i, i \in [1,t]\}$, each one of which takes D_i as inputs, and outputs noisy data $\boldsymbol{n}_i$ with independent randomness. Let $\tau_{u,k}$ be the set of timestamps dominated by trajectory $\ell_{u,k}$ for a specific user u and timestamp k.

Presume ε_i is a privacy budget of A_i (i.e., A_i satisfies ε_i-differential privacy), then Λ satisfies ℓ-trajectory privacy, if

$$\forall u, \forall k, \sum_{i \in \tau_{u,k}} \varepsilon_i \leq \epsilon. \tag{6}$$

Due to space limitations, we only sketch this methodology and omit the proof (as well as proofs in the following sections).

IV. Algorithms

A. First-cut Solution

The straightforward idea is to *uniformly* allocate the privacy budget of ϵ/ℓ on each timestamp. We implemented this technique in Algorithm 1 as our baseline. The integrated algorithm UNIFORM contains UNIFORM$_i$ of each timestamp $i \in [1,t]$.

```
Algorithm 1: UNIFORM_t
  Input: Real statistics $\boldsymbol{r}_t$; Trajectory length $\ell$; Privacy budget $\epsilon$
  Output:  noisy data $\boldsymbol{n}_t$
  Calculate real statistics $\boldsymbol{r}_t \leftarrow Q^c(D_t)$
  Add scaled Laplace noises $\boldsymbol{n}_t \leftarrow \boldsymbol{r}_t + \langle Lap(2\ell/\epsilon)\rangle^{|\boldsymbol{locs}|}$
  return $\boldsymbol{n}_t$
```

B. Proposed Framework

UNIFORM leaves us no space for optimization. To optimize privacy budget allocation, an idea is dynamically allocate privacy budget. As we described in Section I, the varying density is an intrinsic characteristic of trajectory streams, so that it is expected to spend less privacy budget when relatively less users are producing data points, whereas to spend more privacy budget when it is "worth". How to measure whether spending privacy budget is a worthwhile investment or not ? This question leads us to develop an approximation strategy cooperating with dynamic budget allocation component to raise data utility.

Approximation strategies were investigated in earlier research, such as histogram publishing [23], [24], and statistics on data stream publishing [11], [2]. Instead of directly adding noise to a real statistics, they function by transformation of original data or a query structure to achieve better overall utility. In our case, we will republish an appropriate previously released noisy data if it is "close to" the real statistics of the current timestamp. The distance from the real statistics to noisy data will be utilizing MAE which defined in Equation (4).

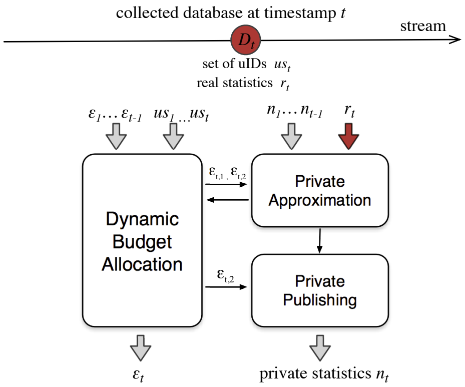


Fig. 3: Illustration of proposed framework. Red object contains sensitive data.

In Fig.3, we present the proposed framework as well as its implementation of Algorithm 2. It is composed of three key components. *Dynamic budget allocation* component allocates budgets to algorithms on each timestamp and ensure Theorem 3. *Private approximation* component spends a part of budget to make a decision of approximately publishing data or not. *Private publishing* component receives information from the *private approximation*, and releases private data.

```
Algorithm 2: PPDP algorithm at timestamp $t$
  Input: Real statistics $\boldsymbol{r}_t$; Users sets $us_i, i \in [1,t]$; Protected
         trajectory length $\ell$; Total privacy buget $\epsilon$; Previously allocated
         budget $\varepsilon_i, i \in [1, t-1]$; noisy data $\boldsymbol{n}_i, i \in [1, t-1]$
  Output: Noisy statistics $\boldsymbol{n}_t$, Allocated budget $\varepsilon_t$
  $\varepsilon_{t,1}, \varepsilon_{t,2} \leftarrow$ Dynamic Budget Allocation$_t$
  $appx \leftarrow$ Approximation Strategy$_t$
  // Private Publishing
  if $appx = t$ then
      Add scaled Laplace noises $\boldsymbol{n}_t \leftarrow \boldsymbol{r}_t + \langle Lap(2/\varepsilon_{t,2})\rangle^{|\boldsymbol{locs}|}$
  else
      $\boldsymbol{n}_t \leftarrow \boldsymbol{n}_{appx}$ , $\varepsilon_{t,2} \leftarrow 0$
  $\varepsilon_t \leftarrow \varepsilon_{t,1} + \varepsilon_{t,2}$
  Return $\boldsymbol{n}_t, \varepsilon_t$
```

We note that the users set us_i at each timestamp is considered as non-sensitive data, that is because under the definition 3 of neighboring datasets, us_i will not change between any two neighboring dataset at timestamp i.

C. Dynamic budget Allocation and Approximation

Algorithm 3 described the dynamic budget allocation in an exponentially decay fashion. Now we briefly examine whether this algorithm is ℓ-trajectory private or not In Line 2 and 3, it calculates the max constraint of spent privacy budget by checking budget on $\tau_{u,t}$ of users who have a data point at the current timestamp t (i.e., us_t). Then, in line 4, in order to spare some budget for the future upcoming data points, it heuristically save half of total available budget for the future data and assign another half to $\varepsilon_{t,2}$ which is dedicated to possible Laplace noise. Therefore, for algorithms A_i, i belongs to any $\in \tau_{u,k}$ (i.e., set of timestamps inside any ℓ-trajectory), the sum of allocated budgets will less than the total budget ϵ. According to Theorem 3, it ensures ℓ-trajectory privacy.

```
Algorithm 3: Dynamic Budget Allocation: GA$_t$
Input: Users set $us_i, i \in [1, t]$ ; Previous privacy budget
        $\varepsilon_i, i \in [1, t-1]$ ; Total privacy budget $\epsilon$
Output: privacy budget $\varepsilon_{t,1}, \varepsilon_{t,2}$
Allocate fixed budget $\varepsilon_{t,1} \leftarrow \epsilon/(2 * \ell)$ , temporary budget $\varepsilon_{t,2} \leftarrow 0$
Calculate spent budget $\varepsilon_t^s \leftarrow max_{u \in us_t}\{\sum \varepsilon_i, i \in \tau_{u,t}\}$
Calculate remainder budget $\varepsilon_t^r \leftarrow \epsilon/2 - \varepsilon_t^s$
// exponentially decreasing
Allocate dynamic budget $\varepsilon_{t,2} \leftarrow \varepsilon_t^r/2$
return $\varepsilon_{t,1}$, $\varepsilon_{t,2}$
```

Now we will describe two kinds of approximation strategies which are implemented as Algorithms 4 and 5. Because statistics on trajectory streams are collected in each short period then publish at each timestamp, it can be considered as a kind of slide windows publishing, which means each count on the timeline will be numerically close to its adjacent value. According to this observation, we adopt a scheme of approximately re-publishing *adjacent* noisy data. In Line 1 of Algorithm 4, it trades off lower error between Laplace error and republishing error. Because this procedure need to access real statistics by *MAE* query, a scaled noise need to perturb MAE value. For result of *MAE* query, the global sensitivity is $2/|\boldsymbol{locs}|$, and we have privacy budget $\varepsilon_{t,1}$, according to the Laplace Mechanism 1, the scaled noise should be $Laplace(2/(|\boldsymbol{locs}| * \varepsilon_{t,1}))$.

```
Algorithm 4: Approximation Strategy: re-publish $\boldsymbol{n}_{Adj}$
  Input: Real statistics $\boldsymbol{r}_t$; Privacy budget $\varepsilon_{t,1}, \varepsilon_{t,2}$
  Output: index of noisy data $appx$
1 if $MAE(\boldsymbol{r}_t, \boldsymbol{n}_{t-1}) + Lap(2/(|\boldsymbol{locs}| * \varepsilon_{t,1})) \leq 2/(|\boldsymbol{locs}| * \varepsilon_{t,2})$
  then
2     $appx \leftarrow t-1$
3 else
4     $appx \leftarrow t$
5 return $appx$
```

On the other hand, we observe that the real statistics on trajectory data usually have a certain repeating pattern (e.g., for people flow trajectory data, in general, the peak-hour count appear around 8 a.m. or 6 p.m. everyday). So we develop a scheme of re-publishing the most appropriate noisy data by searching $\boldsymbol{n}_{appx}$ of Minimum Manhattan Distance (MMD) to the current real statistics $\boldsymbol{r}_t$ among all noisy data $\boldsymbol{n}_i, i \in [1, t-1]$. A challenge arises when we try to retrieval $\boldsymbol{n}_{MMD}$ because MD query access $t-1$ times real statistics, which means if we adopt Laplace mechanism it will lead to an unbounded scaled noise. By adopting exponential mechanism (Theorem 2) and design an appropriate quality function, we can reduce this noise within acceptable limits. Algorithm 5 implements this idea and uses the negative value of *MD* query as quality function. In Line 1, it spent half of privacy budget of $\varepsilon_{t,1}$ to approximately retrieving $\boldsymbol{n}_{\text{MMD}}$. In Line 2,3, it calculates the quality score and randomly return the $\boldsymbol{n}_i$ according the probability of proportional to $exp(\frac{\varepsilon_{t,1}^m * score_i}{4})$. The divisor 4 comes from multiplication of sensitive 2 and Exponential mechanism's defined factor 2. The remainder of Algorithm 5 (from Line 4 to 8) is similar as Algorithm 4.

```
Algorithm 5: Approximation Strategy: re-publish $\boldsymbol{n}_{MMD}$
  Input: Real statistics $\boldsymbol{r}_t$; Privacy budget $\varepsilon_{t,1}, \varepsilon_{t,2}$
  Output: index of noisy data $appx$
1 $\varepsilon_{t,1}^m \leftarrow \frac{\varepsilon_{t,1}}{2}$ , $\varepsilon_{t,1}^x \leftarrow \frac{\varepsilon_{t,1}}{2}$
2 Calculate $score_i$ of quality function $q : -\text{MD}(\boldsymbol{r}_t, \boldsymbol{n}_i), i \in [1, t-1]$
3 $\boldsymbol{n}_{\text{MMD}} \leftarrow \boldsymbol{n}_i \propto exp(\frac{\varepsilon_{t,1}^m * score_i}{4}), i \in [1, t-1]$
4 if
  $MAE(\boldsymbol{r}_t, \boldsymbol{n}_{MMD}) + Laplace(2/(|\boldsymbol{locs}| * \varepsilon_{t,1}^x)) \leq 2/(|\boldsymbol{locs}| * \varepsilon_{t,2})$
  then
5     $appx \leftarrow MMD$
6 else
7     $appx \leftarrow t$
8 return $appx$
```

V. Experiments

In this section, we evaluate the data utility of the proposed algorithm and a competitor FAST[2]. We evaluate our algorithms on four real trajectory dataset, that is PeopleFlow[2], Geolife[3], T-Drive[4], WorldCup98[5].

	PeopleFLow	Geolife	T-Drive	WorldCup98
Timestamps Amt.	1,694	1,440	886	722
Users Amt.	11,406	170	2,698	550,762
POIs Amt.	18	56	21	1,000
Data Points Amt.	102,468	240,990	37,255	1,258,542

As we introduced in Section II, FAST[6] is a sampling and filtering framework for privacy preserving time series data publishing. They designed two kinds of algorithms based on FAST, that is FAST with adaptively sampling and with fixed sample rate. Since the former one need to know amount of total timestamps T in advance that leads to inapplicability of infinite trajectory streams. Therefore, we use FAST with fixed sampling rate as our utility competitor, denoted as FAST$_{\text{fixed}}$. We configure it according to the original paper [2]. Since the output of these algorithms are including randomness, in order to compare the average case, each algorithm runs 50 times then outputs the average results.

A. Utility Evaluation

GA+MMD is best on all dataset by varying ℓ or ϵ. UNIFORM's MAE become larger almost linearly with the increasing of ℓ, and exponentially with decreasing of ϵ. FAST$_{\text{fixed}}$ is not stable. Its utility is higher than UNIFORM in most of the cases except the extremely larger or smaller ϵ, and lower than GA+Adj in most of the cases except on dataset Geolife.

Varying ℓ : The upper part of Fig. 4 are results of MAE under different ℓ. Longer ℓ higher the privacy level. We set $\epsilon = 1$. GA+MMD always perform better than other algorithms. We notice that when ℓ increases, the error of other algorithms

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

[3] http://research.microsoft.com/en-us/projects/geolife/

[4] http://research.microsoft.com/en-us/projects/tdrive/

[5] http://ita.ee.lbl.gov/html/contrib/WorldCup.html

[6] http://www.mathcs.emory.edu/aims/FAST/index.html

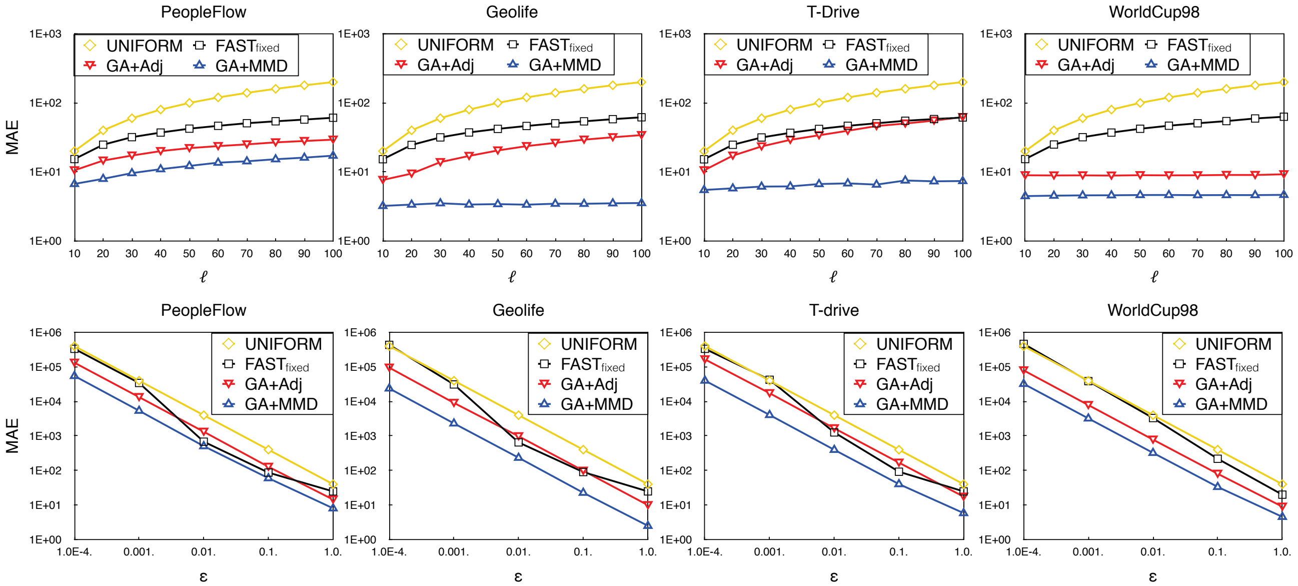


Fig. 4: MAE on four Datasets by varying ℓ and varying ϵ.

will be higher, while thanks to the dynamic budget allocation and MMD approximation strategy, GA+MMD keeps a good utility among different datasets. When $|\boldsymbol{locs}|$ is higher (WorldCup98), GA+Adj will approach the utility of GA+MMD.

Varying ϵ : The lower part of In Fig. 4 are results of MAE under varying ϵ. Larger ϵ lower the privacy level. We set $\ell = 20$. GA+MMD always perform better and more stable than other algorithms. Especially, FAST are not stable among different ϵ.

VI. Conclusion

We conclude that the proposed GA+MMD can work efficiently to release privacy preserved data in real-time, while satisfying ℓ-trajectory privacy. It is sufficient to adaptively adjust privacy budget allocation dependent on underlying data distribution to achieve good performance. As future work, we will investigate application of this model to several other kinds of data and other kinds of data mining tasks.

Acknowledgment

We are grateful to the anonymous referees for useful comments and suggestions.

References

[1] https://www.nttdocomo.co.jp/corporate/disclosure/mobile_spatial_statistics/.

[2] L. Fan, L. Xiong, and V. Sunderam, "FAST: differentially private real-time aggregate monitor with filtering and adaptive sampling," in SIGMOD '13.

[3] X. Zhang, H. Kitabayashi, Y. Asano, and M. Yoshikawa, "A health-aware pedestrian navigation system by analysis of spatiotemporal vital sign data", Workshop on Personal Data Analytics in VLDB'14.

[4] L. Fan, L. Bonomi, L. Xiong, and V. Sunderam, "Monitoring web browsing behavior with differential privacy," in WWW '14.

[5] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE* Symposium on Security and Privacy, 2008.

[6] L. Sweeney, "K-anonymity: A model for protecting privacy," Int. J. Uncertain. Fuzziness Knowl.-Based Syst. , Oct. 2002.

[7] C. Dwork, "A firm foundation for private data analysis," Commun. ACM, vol. 54, no. 1, p. 86–95, Jan. 2011.

[8] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility,"Sci. Rep., vol. 3, Mar. 2013.

[9] C. Dwork, "Differential privacy: A survey of results," LNCS,2008

[10] C. Dwork, "Differential privacy in new settings." in *SODA*'10.

[11] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," in VLDB'14.

[12] D. Mir, S. Muthukrishnan, A. Nikolov, and R. N. Wright, "Pan-private algorithms via statistics on sketches," in PODS '11.

[13] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Trans. Inf. Syst. Secur.*, 2011.

[14] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft, "Private decayed predicate sums on streams," in ICDT '13.

[15] R. Chen, B. C. M. Fung, N. Mohammed, B. C. Desai, and K. Wang, "Privacy-preserving trajectory data publishing by local suppression," *Information Sciences*, vol. 231, pp. 83–97, May 2013.

[16] K. Jiang, D. Shao, S. Bressan, T. Kister, and K.-L. Tan, "Publishing trajectories with differential privacy guarantees," in SSDBM,2013.

[17] S.-S. Ho and S. Ruan, "Differential privacy for location pattern mining," in SIGSPATIAL '11.

[18] S.-S. Ho and S. Ruan, "Preserving privacy for interesting location pattern mining from trajectory data," *Transactions on Data Privacy*, 2013.

[19] C. Dwork, "Differential privacy," in *Automata, languages and programming*. Springer, 2006, p. 1–12.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in TCC'06, pp. 265–284.

[21] D. Kifer and A. Machanavajjhala, "No free lunch in data privacy," in SIGMOD '11. New York, NY, USA: ACM, 2011, p. 193–204.

[22] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in FOCS '07.

[23] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu, "Differentially private histogram publication," in ICDE '12.

[24] X. Zhang, R. Chen, J. Xu, X. Meng, and Y. Xie, "Towards accurate histogram publication under differential privacy," in SIAM'14,