Efficient Frequency-based Randomization for Spatial Trajectories under Differential Privacy

Fengmei Jin, Wen Hua✉, Lei Li, Boyu Ruan, Xiaofang Zhou, *Fellow, IEEE*

Abstract—The uniqueness of trajectory data for user re-identification has received unprecedented attention as the increasing popularity of location-based services boosts the excessive collection of daily trajectories with sufficient spatiotemporal coverage. Consequently, leveraging or releasing personally-sensitive trajectories without proper protection severely threatens individual privacy despite simply removing IDs. Trajectory privacy protection is never a trivial task due to the trade-off between privacy protection, utility preservation, and computational efficiency. Furthermore, *recovery attack*, one of the most threatening attacks specific to trajectory data, has not been well studied in the current literature. To tackle these challenges, we propose a frequency-based randomization model with a rigorous differential privacy guarantee for privacy-preserving trajectory data publishing. In particular, two randomized mechanisms are introduced for perturbing the local/global frequency distributions of a limited number of significantly essential locations in trajectories by injecting special Laplace noises. To reflect the perturbed distributions on the trajectory level without losing privacy guarantee or data utility, we formulate the trajectory modification tasks as kNN search problems and design two hierarchical indices with powerful pruning strategies and a novel search algorithm to support efficient modification. Extensive experiments on a real-world dataset verify the effectiveness of our approaches in resisting individual re-identification and recovery attacks simultaneously while still preserving desirable data utility. The efficient performance on large-scale data demonstrates the feasibility and scalability in practice.

Index Terms—differential privacy, re-identification attack, recovery attack, frequency randomization, hierarchical grid index

1 Introduction

Nowadays, GPS-enabled devices and location-based applications have become ubiquitous, and an increasing amount of spatiotemporal data has been collected, such as vehicle trajectories, phone call records, and user check-ins. Knowledge discovery from such data promotes the development of many advanced technologies like route planning, which bring massive daily benefits. However, as a side effect, trajectories become vulnerable and could potentially expose sensitive information. A widely-encountered risk in trajectory data is the *re-identification* attack that recognizes an individual from their moving history. Recent studies [1]–[4] have demonstrated that individuals can be identified with a sufficiently high success rate ($> 80\%$) by exploring their personalized movement patterns. Hence, to protect people from re-identification, many privacy models have been proposed for trajectory data release [5]–[9]. Among these, differential privacy (DP) [10] mathematically provides a superior data privacy guarantee to ensure the attacker cannot infer too much about any specific person. [11] empirically compared some representative trajectory privacy models and concluded that DP achieves the best protection against re-identification attacks. Nevertheless, there are still several notable limitations in existing trajectory DP models:

- *Fengmei Jin and Wen Hua are with The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.*
  E-Mail: {fengmei.jin, wency.hua}@polyu.edu.hk
- *Lei Li is with The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong, China.*
  Email: thorli@ust.hk
- *Boyu Ruan and Xiaofang Zhou are with The Hong Kong University of Science and Technology, Clear Water Bay, New Territories, Hong Kong.*
  E-Mail: {boyuruan, zxf}@ust.hk

1) Privacy vs. Utility. In the context of trajectory data, most approaches achieve protection by adding various types of noise to the entire trajectory, to each point in the trajectory, or to each coordinate of the trajectory point. Unfortunately, excessive modification caused by noise injection, no matter at which spatial resolution, dramatically affects the utility of the anonymized data. As a result, differentially private trajectories are often useless in practice due to twisting shapes or unrealistic paths on road networks [12]. For example, as an innovative DP-based model, DPT [9] captures trajectory patterns with prefix trees, adds randomized noise, and synthesizes trajectories based on the noisy prefix trees. In this way, it offers a strong privacy guarantee yet cannot retain the truthfulness of data since every synthetic trajectory is constructed based on the differentially private movement patterns rather than any real trajectory. As discussed in [11], although DPT provides the best privacy protection against linkage attacks, it fails to preserve adequate data utility after protection steps. Thus, the trade-off between privacy protection and utility preservation has been a major bottleneck for DP-based methods.

2) Recovery Attack. In fact, it is unnecessary to generate synthetic trajectories from scratch or brutally introduce noise to every single element of the trajectories. Knowing some non-sensitive public locations a user visited cannot enhance the inference about any personal information. Recently, [1], [3] illustrated that only a limited number of spatial points (called *signature* hereafter) contribute the majority to re-identifying an individual from the trajectory dataset. These signature points are the most representative and distinctive in a user's moving history and hence are sensitive locations (e.g., home and workplace) to be carefully protected. Inspired by this, [4] attempted to remove

all signature points from the original trajectories and preserve the remaining points. This successfully lowered the re-identification risk and achieved satisfactory data utility, showing a good balance between privacy protection and utility preservation. However, it is insufficient because original traces can be recovered using well-developed techniques such as map-matching and path inference. According to our empirical study, 60% of original trajectories can be reconstructed, making user identity still vulnerable. Such kind of *recovery attack* is a critical threat to trajectory data release yet has not attracted enough attention.

What makes a location sensitive to re-identification is not the position itself but how the user visits there. In other words, the most personally-identifying places should be both representative and distinctive, i.e., frequently visited by a specific user yet rarely by others. Therefore, individual privacy can be successfully protected by distorting the *frequency* of such signature points, where the differential privacy model fits well. In this work, we propose a novel DP model with randomization mechanisms that perturb the frequency distributions of a limited number of signature points from both global and local perspectives instead of altering the entire trajectories. It intrinsically defends against re-identification attacks, as signatures are the most identifying information in a trajectory. More importantly, it balances privacy guarantee and data utility and ensures that the perturbed frequency distributions cannot be easily recovered. Technically, we leverage a non-trivial Laplace mechanism to achieve differential privacy during global and local frequency perturbation with a theoretically proven privacy guarantee. To edit trajectories for obeying new frequency distributions and meanwhile minimize the utility loss, we implement several trajectory edit operations, define their utility costs, and formalize trajectory modification as an optimization problem. A greedy solution accompanied by novel hierarchical spatial indices is proposed to speed up trajectory modification. To sum up, the major contributions of this work are listed below:

- We introduce a novel differential privacy model for trajectory data based on the frequency perturbation of personal signatures and the non-trivial Laplace mechanism. It not only balances privacy protection and utility preservation but also confidently prevents recovery attacks.
- We accomplish efficient trajectory modification to reflect the frequency changes on the trajectory level with the minimum utility loss. A hierarchical grid-based local index associated with a novel search strategy works well for intra-trajectory modification. Upon it, a global index equipped with a specialized encoding design helps further improve the efficiency of inter-trajectory modification.
- Extensive experiments on a real-world trajectory dataset verify the superiority of our DP model over existing trajectory protection solutions.

Section 2 summarizes existing trajectory privacy models; Section 3 presents our DP model with a focus on the frequency-based randomization mechanisms. We briefly introduce trajectory modification in Section 4 and explain the local intra-trajectory/global inter-trajectory modification in Section 5 and Section 6, respectively. Experiments are reported in Section 7, and we conclude in Section 8.

2 Related work

Existing trajectory privacy models can be classified into two types [11]. Ad-hoc models (e.g., *Mixzone* [5], [13] and *Dummy* [14]–[17]) specialize for the characteristics of trajectories yet fail to provide quantitative privacy guarantee. Formal models extend the privacy principles originally defined in relational databases to trajectories, detailed as follows.

K-anonymity Family (i.e., k-anonymity, l-diversity, and t-closeness) aims to make each object "indistinguishable" within a group of anonymous objects. *NWA* [18] was almost the first to adopt k-anonymity to trajectory data by pursuing the (k,δ)-anonymity for trajectories where each anonymized trajectory is enforced to co-locate with other $k-1$ trajectories within a cylinder of radius δ. *W4M* [6] further applied the spatiotemporal edit distance to measure the trajectory similarity. Differently, *GLOVE* [7] designs another framework by iteratively merging similar trajectory pairs with minimum cost in spatiotemporal dimensions until all resulting trips are k-anonymous. *KLT* [8] improves over *GLOVE* and protects semantics (i.e., the categories of POIs) from exposure by further introducing the l-diversity and t-closeness, such that each movement in a published trajectory can express various semantic information, and so is hard to distinguish. These methods define the anonymization specific to trajectory data yet fail to provide desirable privacy guarantees, especially when encountering reidentification.

Differential Privacy (DP) attempts to make the noisy result derived from a sanitized dataset sufficiently similar to the real answer, such that the adversary cannot obtain extra personal information from the query result while the data is still useful. Existing models achieving DP for trajectories can be classified into two categories:

1) Spatial perturbation to original trajectories: [12] designs three basic methods for producing noisy trajectories by adding various noises to the entire trajectory, to each sampling point, or to every coordinate, respectively. To solve the issue of too many crossings appearing in the anonymized trajectories, they further propose the *SDD* mechanism which imposes more constraints when transiting to the next position. Noises are separately sampled for distance and direction in the angular coordinate system with the help of the *exponential noise*, which has also been used in [19], [20].

2) Noisy patterns for synthetic generation: Instead of injecting noises and distorting trajectories geographically, *DPT* [9] models the trajectories via hierarchical reference systems and encodes the transition information among grid cells via prefix trees. By injecting *Laplace noise* into the prefix trees, the transition probabilities are distorted following DP yet still somehow able to preserve the movement patterns of the original data. Synthetic trajectories are generated from noisy trees rather than perturbing any real trajectory. Other *DPT*-based approaches [21]–[23] enhanced with trajectory semantics, temporal information and other utility features, respectively. *AdaTrace* [23], [24] combines DP with attack resilience constraints and a utility-aware synthesizer, ensuring more private and useful output than *DPT*.

Again, the major bottleneck of existing DP models is the huge utility loss caused by the excessive changes of trajectories or even synthetic results without record-level truthfulness. This work aims to balance privacy and utility.

3 Differential Privacy Model

In this section, we first introduce some basic concepts of differential privacy theory and then detail our privacy model, which provides a desirable DP guarantee by adding non-trivial Laplace noises to global/local frequency distributions of trajectories. It is worth noting that these two randomization mechanisms are independent and can be applied individually or collectively, supported by the composition property of DP [25] (as detailed in Theorem 1). Table 1 presents some major notations in this work.

TABLE 1
Summary of notations

Notation	Definition
$\mathcal{M}$	a differentially private mechanism
$D = \{\tau_1, \ldots, \tau_{\lvert D\rvert}\}$	a dataset with $\lvert D\rvert$ trajectories generated by $\lvert D\rvert$ moving objects
$D^* = \mathcal{M}(D)$	the randomized trajectory dataset
$\mathcal{S}_m = \{s_m(\tau_1), \ldots, s_m(\tau_{\lvert D\rvert})\}$	top-m signatures for each trajectory in D (m is a hyper-parameter)
$\mathcal{P} = \{p_1, p_2, \ldots, p_d\}$	the set of all distinct points composing all signatures in $\mathcal{S}_m$
$\mathcal{F}(\tau) = \langle f_1, f_2, \ldots, f_{\lvert\tau\rvert}\rangle$	the PF distribution over a trajectory τ: each point p_i has its PF f_i
$\mathcal{F}^*(\tau) = \langle f_1^*, f_2^*, \ldots, f_{\lvert\tau\rvert}^*\rangle$	the perturbed PF distribution of τ
$\mathcal{L} = \langle l_1, l_2, \ldots, l_d\rangle$	the TF distribution over $\mathcal{P}$ where l_i denotes the number of trajectories in D passing through point p_i
$\mathcal{L}^* = \langle l_1^*, l_2^*, \ldots, l_d^*\rangle$	the perturbed TF distribution over the global signature point set $\mathcal{P}$

3.1 Differential Privacy

Definition 1 (ϵ-differential privacy)**.** *A randomized mechanism $\mathcal{M}$ provides ϵ-differential privacy if for any two adjacent databases D and D' differing in at most one record, and for every possible output $S \in Range(\mathcal{M})$,*

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in S] \tag{1}$$

where ϵ is called the privacy budget.

Definition 2 (Sensitivity)**.** *Given a query function ϕ, the sensitivity of ϕ is defined as the maximum of the difference in the output value of ϕ if D and D' differ in at most one record, i.e.,*

$$\Delta\phi = \max_{D,D'} \|\phi(D) - \phi(D')\|_1 \tag{2}$$

Laplace mechanism was first proposed in [10] to achieve ϵ-differential privacy by adding Laplacian noise to the query results, where the scale of the Laplace distribution λ is determined by the sensitivity of a query function $\Delta\phi$ and the privacy budget ϵ together:

Definition 3 (Laplace mechanism)**.** *Given a query function ϕ and privacy budget ϵ, a randomized mechanism $\mathcal{M}$ provides the ϵ-differential privacy by sampling i.i.d variables from the Laplace distribution $Lap(\frac{\Delta\phi}{\epsilon})$[1] and adding them to the query answers.*

Theorem 1 (Sequential composition theorem [25])**.** *Assume there are multiple randomized mechanisms $\mathcal{M}_i$, each of which can achieve ϵ_i-differential privacy. Thus, the sequential combination of these mechanisms will become an ϵ-differentially private model with the privacy budget $\epsilon = \sum_i \epsilon_i$.*

1. In general, a Laplace distribution is denoted by $Lap(\mu, \lambda)$ where μ is the mean and λ is the scale. A simplified version is $Lap(\lambda)$, telling the distribution is centered by 0 with the scale λ, as the μ is omitted.

3.2 DP-based Frequency Randomization Mechanism

Definition 4 (Trajectory)**.** *A trajectory is denoted by a sequence of spatial points ordered chronologically, i.e., $\tau = \{p_1, \ldots, p_{\lvert\tau\rvert}\}$, and each individual is associated with a single trajectory representing its entire moving history.*

The dataset contains $|D|$ trajectories generated by $|D|$ moving objects, denoted as $D = \{\tau_1, \tau_2, ..., \tau_{|D|}\}$. We say D and D' are adjacent only if they differ in at most one trajectory. In the following, we first explain some fundamental concepts of trajectory signature and then introduce the randomization mechanisms for perturbing global and local frequency distributions of trajectories, respectively.

3.2.1 Trajectory Signature and Frequency Distributions

As mentioned, existing methods achieving differential privacy for trajectory data usually add a large amount of noise to each trajectory or the entire dataset (e.g., [9], [12]), making the anonymized data useless in practice due to enormous utility damage. Inspired by [3], we observe that *signature* points carry the majority of identifying information in a user's trajectories, indicating that it suffices to anonymize only the signature points for trajectory protection against re-identification attacks. Hence, our DP model is designed based on the selected signature points, which significantly preserves the utility of other non-signature points while providing DP guarantees as well.

Ideally, signatures should be both representative and distinctive in a user's traces, i.e., they should be frequently visited by a specific user but others rarely go there. Accordingly, our privacy model consists of two independent components, each achieving differential privacy for the trajectory dataset by blurring the representativeness and distinctiveness of top-ranked signature points, respectively. In particular, we utilize two types of frequencies to identify signature points for each trajectory in dataset D:

- *Point Frequency (PF)*: A signature point should be ubiquitous in a user's trajectory. We define PF f_p as the total number of times a point p in a trajectory τ, and the representativeness of p in τ is measured by $\frac{f_p}{|\tau|}$ where $|\tau|$ is the total number of points in τ. The higher PF the point has, the more representative it is for the user;
- *Trajectory Frequency (TF)*: The signature point should be unique (distinctive) for a specific user. We define TF l_p of a point p as the number of trajectories in D passing through p at least once, and the distinctiveness of p in a dataset D is computed as $\log(\frac{|D|}{l_p})$ where $|D|$ is the total number of trajectories (objects) in D. Hence, the lower TF the point has, the more distinctive it is within D.

The importance of each point in a trajectory can be measured as the product of its representativeness and distinctiveness. We extract the top-m points with the largest weights as the signatures of every individual. All points occurring in at least one signature are collected in a candidate point set $\mathcal{P} = \{p_1, p_2, \ldots, p_d\}$, where d represents the dimensionality of these distinct signature points. In our DP model, we only distort the selected signature points by injecting novel Laplace noises to the local PF distribution of every single trajectory $\mathcal{F}(\tau) = \langle f_1, f_2, \ldots, f_{|\tau|}\rangle$ with the privacy budget ϵ_L, and to the global TF distribution

$\mathcal{L} = \langle l_1, l_2, \ldots, l_d \rangle$ with the privacy budget ϵ_G (rather than altering the entire trajectory data in geographical space). Note that, based on the sequential composition property of DP (Theorem 1), combining local/global randomization mechanisms (with an exchangeable ordering) can also achieve ϵ-differential privacy, where $\epsilon = \epsilon_L + \epsilon_G$.

3.2.2 Global TF Randomization Mechanism

Global randomization applies to "Trajectory Frequency" (TF) distribution of all top-m signature points, i.e., $\mathcal{P} = \{p_1, p_2, \ldots, p_d\}$. Intuitively, a smaller TF value implies that this location is very unique to some specific users as fewer individuals have visited there. At the same time, a point with a larger TF is more likely to be a hotspot (e.g., a shopping mall or bus station) embedding less personal information. Hence, globally perturbing the TF distribution of signature points is vital for blurring those super distinctive locations and preventing the exposure of personal features.

Algorithm 1: Global TF Randomization Mechanism

Input: D: A set of trajectories; m: the reserved signature size; $\mathcal{P}$: the set of top-m signature points; ϵ_G: privacy budget
Output: D_G^*: The TF-perturbed dataset

1 $\mathcal{L} \leftarrow$ Build-TF-Distribution $(D, \mathcal{P}, m)$; ▷ *over* $\mathcal{P}$
2 **for** $\forall l_j \in \mathcal{L}$ **do**
3 $\quad \eta \sim Lap(\frac{1}{\epsilon_G})$;
4 $\quad l_j^* \leftarrow l_j + \eta$;
5 $\quad l_j^* \leftarrow$ Round$(l_j^*, [0, |D|])$; ▷ *post-processing: round the noisy TF value to a proper integer range*
6 $\mathcal{L}^* \leftarrow \langle l_1^*, l_2^*, \ldots, l_d^* \rangle$;
7 $D_G^* \leftarrow$ GlobalModify$(D, \mathcal{L}^*)$;
8 **return** D_G^*;

Algorithm Description: In Algorithm 1, we generate a global TF list $\mathcal{L}$ over the set of distinct top-m signature points $\mathcal{P}$, where each element l_j represents the TF value for a point p_j (line 1). Consider a point counting query $\phi(D, p_j)$ which is exactly asking for the TF value of the given point p_j. The difference of the query answers on two adjacent databases D and D' differing in one single trajectory is at most ± 1 depending on the existence or non-existence of that trajectory, resulting in the sensitivity as $\Delta\phi = 1$. Thus, we follow the classic Laplace mechanism (Definition 3) to obtain ϵ_G-DP guarantee by adding noise drawn from $Lap(1/\epsilon_G)$ to each l_j (lines 3-5). We round the noisy frequency values to zero or an integer range if needed (line 5), and such post-processing operations will not damage the DP guarantee [25]. After noise injection, we modify trajectories to reflect the perturbed TF distribution $\mathcal{L}^*$ on trajectory level (line 7, detailed in Section 6).

3.2.3 Local PF Randomization Mechanism

The purpose of perturbing global TF distribution is to distort the distinctiveness of top-ranked signature points in a global view, while some locations repeatedly appearing in a user's trajectory can also expose the personal identity. So, our local mechanism aims to perturb the "Point Frequency" (PF) distribution for each trajectory, more specifically, reducing the occurrence of personally-identifying locations to dilute their representativeness and meanwhile increasing the occurrences of other non-sensitive points to inject randomness.

However, a Laplace distribution $Lap(\mu = 0, \lambda = \frac{\Delta\phi}{\epsilon})$ implies that the probability of sampling positive noise is equal to that of negative noise as the distribution is centered at zero. To achieve the goal of reducing/increasing PF for specific points with much higher probability, we design a novel local randomization mechanism by using the Laplace distribution with non-zero mean, i.e., $Lap(\mu \neq 0, \lambda = \frac{\Delta\phi}{\epsilon})$. We will theoretically prove that using such a non-trivial Laplace mechanism still enjoys the ϵ-DP guarantee.

Algorithm 2: Local PF Randomization Mechanism

Input: D: A dataset with $|D|$ trajectories; $\mathcal{F}$: the original PF distribution of each trajectory; m: the number of signature points to be perturbed; PL: the selected point list for each trajectory; ϵ_L: the privacy budget for local perturbation
Output: D_L^*: The PF-perturbed dataset

1 **for** $\forall \tau_i \in D$ **do**
$\quad$ /* *Stage-1: perturb PF for top-m ranked points* */
2 $\quad \bar{\mu} \leftarrow 0$;
3 $\quad$ **for** $k \in [1, m]$ **do**
4 $\quad\quad p_k \leftarrow$ the k-th point in $PL(\tau_i)$;
5 $\quad\quad f_k \leftarrow \mathcal{F}(\tau_i, p_k)$; ▷ *the original PF*
6 $\quad\quad \eta \sim Lap(-f_k, \frac{1}{\epsilon_L})$;
7 $\quad\quad f_k^* \leftarrow f_k + \eta$; ▷ *noise injection*
8 $\quad\quad f_k^* \leftarrow$ RoundInt(f_k^*); ▷ *round to an integer*
9 $\quad\quad f_k^* \leftarrow \max(f_k^*, 0)$; ▷ *round negative to zero*
10 $\quad\quad \bar{\mu} \leftarrow \bar{\mu} + (f_k^* - f_k)$; ▷ *sum up the actual noise*
11 $\quad \bar{\mu} \leftarrow \bar{\mu}/m$; ▷ *the avg. truly added noises in Stage-1*
$\quad$ /* *Stage-2: perturb PF for the remaining m points* */
12 $\quad$ **for** $k \in [m+1, 2m]$ **do**
13 $\quad\quad p_k \leftarrow$ the k-th point in $PL(\tau_i)$; $f_k \leftarrow \mathcal{F}(\tau_i, p_k)$;
14 $\quad\quad \eta \sim Lap(-\bar{\mu}, \frac{1}{\epsilon_L})$; $f_k^* \leftarrow f_k + \eta$;
15 $\quad\quad f_k^* \leftarrow$ RoundInt(f_k^*); $f_k^* \leftarrow \max(f_k^*, 0)$;
16 $\quad \mathcal{F}^*(\tau_i) \leftarrow \langle f_1^*, f_2^* \ldots f_{2m}^* \rangle$;
17 $D_L^* \leftarrow$ LocalModify$(D, \mathcal{F}^*)$;
18 **return** D_L^*;

Algorithm Description: The local perturbation covers two stages of noise injection, which can not only blur the personally-sensitive information in a user's moving history but also ensure the total injected noises will not significantly affect the trajectory length. Initially, each trajectory is assigned to a list of $2m$ selected points where m denotes the reserved signature size. These points are selected by sequentially picking from the intersection of its top-ranked signature and the global point set $\mathcal{P}$ and randomly sampling from its remaining points until reaching $2m$. We next probabilistically decrease the frequency of top-m points in Stage-1 for diluting their representativeness, and meanwhile increase the occurrence of other m points in Stage-2 for keeping trajectory cardinality to a large extent.

In Algorithm 2, we inject the noise $\eta \sim Lap(-f_k, \frac{1}{\epsilon_L})$ into its original frequency f_k for each point p_k in Stage-1 (lines 4-7). The reason for using a Laplace distribution with the negative mean (i.e., $-f_k$) is to sample negative noises with a higher probability so as to reduce the occurrence of top-m signature points to the greatest extent. Regarding the scale λ, a point-counting query $\phi(p, \tau)$, which estimates the total number of occurrences in τ for the point p, would naturally result in the sensitivity $\Delta\phi = 1$. We use a post-processing operation to round the noisy frequency to the closest integer (line 8) or to zero if it is negative (line 9) [25]. Similar operations are conducted on the remaining

m points (lines 12-15). The main difference is the Laplace distribution used in Stage-2 is $-\bar{\mu}$, which is the average of the truly added noises in Stage-1. Regarding all the other points that are not as useful as signature points in terms of reidentification ability, they remain unchanged to preserve more data utility. Finally, we make the trajectories satisfying the perturbed PF distribution $\mathcal{F}^*$ using the intra-trajectory modification (line 17), which will be detailed in Section 5.

Recall that the most crucial goal of the local randomization mechanism is to reduce the occurrence of top-ranked signature points, as expected to be achieved in Stage-1. However, purely conducting Stage-1 would dramatically affect the cardinality of the output trajectory. More specifically, it would result in a huge drop in the total number of points, leading to poor utility of the anonymized data. Our mechanism avoids this issue with the help of Stage-2 and induces both raise and drop in trajectory length in a relatively even way. The prior choice for the frequency-increasing task is its other top-ranked signature points appearing in the global set $\mathcal{P}$ as well. It is more convincing to increase the occurrence of these points due to their important weights. Plus, a point appears in $\mathcal{P}$ implying its significance to other objects, raising its frequency would bring confusing messages to the global as additional benefits.

Privacy Analysis: We first prove that a generalized Laplace mechanism using Laplace distribution $Lap(\mu, \lambda)$ with a non-zero mean μ can still guarantee the ϵ-differential privacy, same as that of the commonly-used Laplace distribution centered by zero. Then, we leverage it to illustrate that our local mechanism can strictly provide the ϵ_L-differential privacy guarantee to the trajectory data.

Theorem 2. *A randomized mechanism $\mathcal{M}$ sampling noises from the Laplace distribution $Lap(\mu \neq 0, \lambda = \frac{\Delta\phi}{\epsilon})$ still preserves ϵ-differential privacy.*

Proof. Recall that the adopted Laplace distribution has a probability density function: $Lap(x|\mu, \lambda) = \frac{1}{2\lambda}\exp(\frac{-|x-\mu|}{\lambda})$. Assume $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ and $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)$ are two adjacent input variables differing at most one dimension, such that $\|x-y\| \leq 1$. Let $\phi(\cdot)$ be some query functions $\phi : \mathbb{N}^n \to \mathbb{R}^m$. We compare the probability of outputting the same arbitrary result $\boldsymbol{z} \in \mathbb{R}^m$ for them:

$$
\begin{aligned}
\frac{\Pr(\mathcal{M}(\boldsymbol{x}) = \boldsymbol{z})}{\Pr(\mathcal{M}(\boldsymbol{y}) = \boldsymbol{z})} &= \prod_{i=1}^{m}\left(\frac{\exp(-\frac{\epsilon\left||z_i - \phi(x)_i| - \mu\right|}{\Delta\phi})}{\exp(-\frac{\epsilon\left||z_i - \phi(y)_i| - \mu\right|}{\Delta\phi})}\right) \\
&= \prod_{i=1}^{m}\exp\left(\frac{\epsilon(\left||z_i - \phi(y)_i| - \mu\right| - \left||z_i - \phi(x)_i| - \mu\right|)}{\Delta\phi}\right) \\
&\leq \prod_{i=1}^{m}\exp\left(\frac{\epsilon\left||z_i - \phi(y)_i| - |z_i - \phi(x)_i|\right|}{\Delta\phi}\right) \\
&\leq \prod_{i=1}^{m}\exp\left(\frac{\epsilon|\phi(x)_i - \phi(y)_i|}{\Delta\phi}\right) \\
&= \exp\left(\frac{\epsilon\|\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})\|_1}{\Delta\phi}\right) \leq \exp(\epsilon)
\end{aligned}
$$

□

Theorem 3. *The local perturbation mechanism described in Algorithm 2 provides ϵ_L-differential privacy to a trajectory τ.*

Proof. Let τ and τ' be adjacent trajectories differing at most one point. Let $\phi(\cdot)$ be a point-counting query function, which returns the frequency of p_k in τ as the query answer, denoted as $\phi(\tau)_k$ for short. Apparently, $\|\phi(\tau) - \phi(\tau')\|_1 \leq 1$. According to the local perturbation mechanism, each point p_k belonging to the top-m signature of a specific trajectory is distorted in Stage-1 by injecting noises drawn from $Lap(-f_k, \frac{1}{\epsilon_L})$ into its original frequency f_k, while the noises sampled for the remaining m points are from $Lap(-\bar{\mu}, \frac{1}{\epsilon_L})$ where $\bar{\mu}$ is the average noises truly added in the previous stage. In other words, two Laplace distributions share the same scale $1/\epsilon_L$ but have various mean values. So, based on Theorem 2, we can prove that:

$$
\begin{aligned}
\frac{\Pr(\mathcal{M}(\tau) = \tilde{\tau})}{\Pr(\mathcal{M}(\tau') = \tilde{\tau})} &= \prod_{k=1}^{m}\left(\frac{\exp(-\epsilon_L\left||\tilde{\tau}_k - \phi(\tau)_k| - f_k\right|)}{\exp(-\epsilon_L\left||\tilde{\tau}_k - \phi(\tau')_k| - f_k\right|)}\right) \\
&\quad \times \prod_{k=m+1}^{2m}\left(\frac{\exp(-\epsilon_L\left||\tilde{\tau}_k - \phi(\tau)_k| + \bar{\mu}\right|)}{\exp(-\epsilon_L\left||\tilde{\tau}_k - \phi(\tau')_k| + \bar{\mu}\right|)}\right) \\
&\leq \prod_{k=1}^{m}\exp\left(\epsilon_L\left||\tilde{\tau}_k - \phi(\tau')_k| - |\tilde{\tau}_k - \phi(\tau)_k|\right|\right) \\
&\quad \times \prod_{k=m+1}^{2m}\exp\left(\epsilon_L(|\tilde{\tau}_k - \phi(\tau')_k| - |\tilde{\tau}_k - \phi(\tau)_k|)\right) \\
&\leq \prod_{k=1}^{2m}\exp\left(\epsilon_L|\phi(\tau)_k - \phi(\tau')_k|\right) \\
&= \exp(\epsilon_L\|\phi(\tau) - \phi(\tau')\|_1) \leq \exp(\epsilon_L)
\end{aligned}
$$

□

4 Trajectory Modification and Utility Loss

After DP-based noises have been injected into frequency distributions for identity privacy, the next goal is to modify the original trajectories such that: 1) the output trajectories satisfy the perturbed distributions, and 2) the overall utility loss is minimized. So, we first introduce the trajectory edit operations with their utility loss in the following.

Naturally, the change of point q's frequency (either local PF or global TF) only includes two cases: 1) frequency increasing requires inserting more q into a trajectory (or more), and 2) frequency decreasing requires deleting the existing q from a trajectory (or more). Here, the injected noise determines how many times point q should be inserted or deleted. Hence, we introduce two basic operations for trajectory modification: $\mathcal{OP}_i$ and $\mathcal{OP}_d$ denoting the insertion and deletion, respectively. Figure 1(a) illustrates an example where we delete point p_4 and insert a new occurrence of point p_2 (to its closest trajectory segment $\langle p_5, p_6\rangle$). The trajectory will be modified from $\tau = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$ to $\tau^* = \{p_1, p_2, p_3, p_5, p_2, p_6, p_7\}$.

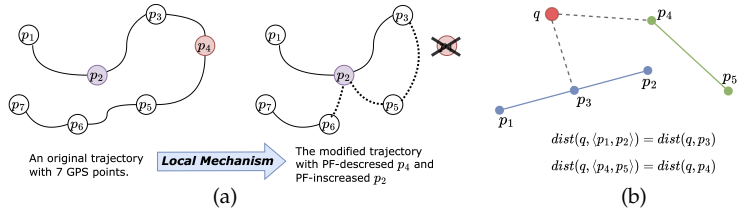

Fig. 1. (a) An example of local intra-trajectory modification via a possible detour. (b) An illustration of point-segment distance measure.

Another critical task is how to compute the utility loss caused by edit operations since the goal of privacy-preserving trajectory publishing requires that the output

data should be able to serve generic query/mining tasks with a controlled amount of information leakage. Thus, our utility loss is defined in a distance-based manner rather than a task-specific way. So, we first define the distance between a point and a line segment as follows:

Definition 5 (Point-Segment Distance)**.** *The distance from a point q to a line segment $s = \langle p_x, p_y \rangle$ is defined as the minimum distance between q and the closest point $\bar{p}$ on s, i.e.,*

$$dist(q, s) = \min_{\bar{p} \in s} dist(q, \bar{p}) \tag{3}$$

As a simple illustration in Figure 1(b), given the query point q (colored in red), the distance between q and $\langle p_1, p_2 \rangle$ (colored in blue) is computed as $dist(q, p_3)$ since p_3 is the closest point to q on the segment. Similarly, the distance from q to $\langle p_4, p_5 \rangle$ (colored in green) is $dist(q, p_4)$.

Definition 6 (Utility Loss of Point Insertion)**.** *Inserting a point q into a line segment s will cause a utility loss of: $\mathbb{L}[\mathcal{OP}_i(q, s)] = dist(q, s)$, where $s = \langle p_x, p_y \rangle$ is an arbitrary line segment composed of two endpoints p_x, p_y in τ.*

Hence, the cost of inserting q into a trajectory τ once with minimal utility loss is defined as $\mathbb{L}[\mathcal{OP}_i(q, \tau)] = \min_{s \in \tau} \mathbb{L}[\mathcal{OP}_i(q, s)]$. Naturally, if the insertion happens Δ times, the utility loss will be accumulated: $\mathbb{L}[\mathcal{OP}_i(q, \tau, \Delta)] = \sum_{i=1}^{\Delta} \mathbb{L}[\mathcal{OP}_i(q, s_i)]$, where s_i is the trajectory segment selected from τ for the i-th insertion.

Definition 7 (Utility Loss of Point Deletion)**.** *The utility loss of deleting a point q from an existing trajectory segment $s = \langle p_x, q, p_y \rangle$ in τ is calculated as: $\mathbb{L}[\mathcal{OP}_d(q, s)] = dist(q, s')$, where $s' = \langle p_x, p_y \rangle$ is the line segment by removing q and reconnecting two endpoints p_x, p_y of s.*

Similarly, the cost of deleting q from τ for Δ times is accumulated as: $\mathbb{L}[\mathcal{OP}_d(q, \tau, \Delta)] = \sum_{i=1}^{\Delta} \mathbb{L}[\mathcal{OP}_d(q, s_i)]$, where s_i is the trajectory segment selected from τ for the i-th deletion. If a point is forced to disappear from a trajectory, we simply remove all its occurrences with the overall utility loss as: $\mathbb{L}[\mathcal{OP}_d(q, \tau)] = \sum_{s \in \tau} \mathbb{L}[\mathcal{OP}_d(q, s)]$.

Based on the above edit operations, we next define the problems of intra- and inter-trajectory modification, respectively. Then, we explain how to solve them by reducing to K-nearest segment/trajectory search problems.

5 Local Intra-trajectory Modification

Given a trajectory τ and its original/perturbed PF distribution, denoted by $\mathcal{F}(\tau)$ and $\mathcal{F}^*(\tau)$ respectively, the goal of intra-trajectory modification is to insert (resp. delete) a point q into (resp. from) τ for Δf_q times, where q denotes every point whose frequency in τ increases (resp. decreases) after local PF perturbation and $\Delta f_q = |f_q^* - f_q|$, while minimizing the utility loss of τ simultaneously.

To achieve it, the top-ranked nearest segments for a point q should be found when editing it for its perturbed frequency, enabling us to reduce the intra-trajectory modification to a K**-nearest segment search** problem:

Definition 8 (K-Nearest Segment Search)**.** *Given a point q and a trajectory τ, the task of inserting q into τ for Δf_q times can be regarded as finding a list of top-Δf_q nearest trajectory segments with the minimum insertion utility loss for q, where $\Delta f_q = |f_q^* - f_q|$, such that:*

$$\mathbb{L}[\mathcal{OP}_i(q, s)] \leq \mathbb{L}[\mathcal{OP}_i(q, s')]$$
$$\forall s \in NS_q^+ \ , \ \forall s' \in \{S(\tau) \setminus NS_q^+\}$$

where NS_q^+ contains the Δf_q-nearest trajectory segments to q, and $S(\tau)$ represents the set of all possible trajectory segments composed of any two consecutive points in τ.

Similarly, deleting q from τ for Δf_q times is equivalent to finding a list of top-Δf_q trajectory segments passing q, namely, $NS_q^- = \{s = \langle p_x, q, p_y \rangle | s \in \tau\}$, such that:

$$\mathbb{L}[\mathcal{OP}_d(q, s)] \leq \mathbb{L}[\mathcal{OP}_d(q, s')]$$
$$\forall s \in NS_q^- , \ \forall s' \in \{S(\tau, q) \setminus NS_q^-\}$$

where NS_q^- records the selected trajectory segments with the size of $|NS_q^+| = \Delta f_q$, and $S(\tau, q)$ represents the set of all trajectory segments composed of any three consecutive points in τ and the middle one is the target point q.

5.1 Hierarchical Grid-based Local Index (HGL)

A straightforward solution for the above K-nearest search problem is linearly scanning the whole dataset for every frequency-perturbed point, which is quite time-consuming since the computation complexity of such naive solution reaches $O(mn\bar{l})$ for the intra-trajectory modification. Here, m is the signature size; n and $\bar{l}$ are the cardinality of the dataset and the average trajectory length respectively.

Considering the locality property of geographic data, adopting a spatial index would potentially speed up the nearest neighbor search and a grid index is suitable for line segment organization. Besides, the hierarchical structure can better handle the segments with variable lengths and, more importantly, prune unpromising branches as early as possible if adopting an appropriate search algorithm. Hence, we first design a ***H**ierarchical **G**rid-based **L**ocal* index (HGL in short) with multiple resolutions to organize all segments constituting a single trajectory, such that the intra-trajectory modification can be efficiently accomplished on the HGL for every noisy point. We will first introduce the basic index structure as well as the *point-grid cell distance* used for pruning unpromising segments earlier, and then present a non-trivial search algorithm to find the K-nearest segments for finishing the intra-trajectory modification efficiently.

5.1.1 Hierarchical Structure

Let $\mathcal{G}_r$ denote a set of grid cells in a uniform grid of granularity r (e.g., 512×512 cells). The hierarchical grids $\mathcal{HG} = \{\mathcal{G}_{r_1}, \ldots, \mathcal{G}_{r_H}\}$ consist of multiple-level grids with different granularity, where $r_1 < \cdots < r_H$. Usually, $\mathcal{G}_{r_1}$ is the coarsest-grained grid representing the entire spatial area with the granularity of 1×1, and the finest granularity r_H relates to the distribution of trajectory points.

Regarding every single grid cell g_r^i in $\mathcal{G}_r$, three pieces of information will be recorded: 1) the geographic coverage; 2) all trajectory segments that entirely stay in g_r^i but cannot further fit in any finer grid cell $g_{r'}^j$ in our hierarchical grids $\mathcal{HG}$, where $r' > r$; and 3) the parent/children pointers encoding the hierarchical relationship (as exemplified in Figure 2). Specifically, we define the parent and children of an arbitrary grid cell g_r^i in $\mathcal{G}_r$ as follows:

- Its parent should be a coarser grid cell in $\mathcal{G}_{r'}$ (i.e., $r' < r$) and be the smallest one that can completely enclose g_r^i.
- Its children should be a set of finer grid cells in $\mathcal{G}_{r''}$ where $r'' > r$, each of which is fully located in g_r^i but cannot be covered by any other cell finer than g_r^i.

Meanwhile, we assign every trajectory segment to a specific grid cell based on the following criteria:

Definition 9 (Best-fit Grid Cell)**.** *Given a trajectory segment $s = \langle p_x, p_y \rangle$, a grid cell g_r^i in the grid $\mathcal{G}_r$ is called the "best-fit" grid cell of s, if and only if:*

- *the two endpoints p_x and p_y locate in the same grid cell g_r^i in $\mathcal{G}_r$, namely, $p_x \in g_r^i \wedge p_y \in g_r^i$ where $g_r^i \in \mathcal{G}_r$;*
- *they locate in two different grid cells in a finer granularity $\mathcal{G}_{r'}$ with $r' > r$, i.e., $p_x \in g_{r'}^j \wedge p_y \in g_{r'}^k$ where $j \neq k$.*

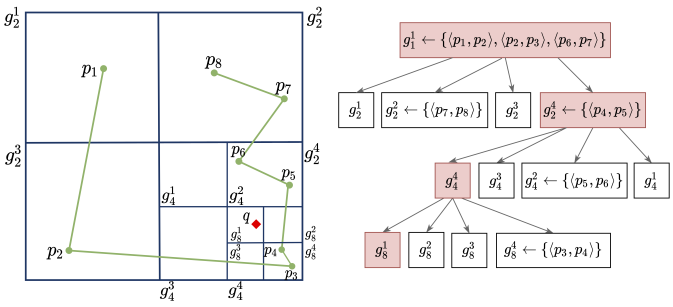

Fig. 2. An example of HGL index for a single trajectory.

For instance, a trajectory consists of eight points and is indexed by the HGL as shown in Figure 2. The best-fit grid cell for trajectory segments $\langle p_1, p_2 \rangle, \langle p_2, p_3 \rangle$ and $\langle p_6, p_7 \rangle$ is the coarsest cell g_1^1, while other segments including $\langle p_3, p_4 \rangle$, $\langle p_4, p_5 \rangle$, $\langle p_5, p_6 \rangle$, and $\langle p_7, p_8 \rangle$ perfectly locate in finer grid cells g_8^4, g_2^4, g_4^2 and g_2^2, respectively.

5.1.2 Pruning based on Best-fit Grid Cell

Next, we introduce a distance metric between a point and a grid cell that will be flexibly used for pruning during search and prove the correctness of the pruning strategy.

Definition 10 (Point-Grid Cell Distance)**.** *To compute the minimum distance from a point q to a grid cell g that is a rectangle with four connected edges, we first check whether q is located inside g. If so, then $MINdist(q, g) = 0$; otherwise, it depends on the edge $\bar{s}$ of g which is the closest to q:*

$$MINdist(q, g) = \begin{cases} \min_{\bar{s} \in g} dist(q, \bar{s}) & \text{if } q \notin g \\ 0 & \text{if } q \in g \end{cases} \quad (4)$$

The minimum distance between q and a grid cell can help us to prune its unpromising children cells and the inside trajectory segments based on the following theorem:

Theorem 4. *Given a grid cell g in $\mathcal{G}_r$, a point q and its current K-th closest segment s_K, if the minimum distance from q to g is greater than $\theta_K = dist(q, s_K)$, namely, $MINdist(q, g) > \theta_K$, then all trajectory segments within g and its children (i.e., all finer-grained grid cells inside g in $\mathcal{G}_{r'}$ with $r' > r$) will not be the promising candidates for q's K-nearest neighbors.*

Proof. Any segment s inside g enjoys a lower bound distance to a query point q, i.e., $dist(q, s) \geq MINdist(q, g)$ based on the Point-Segment Distance in Equation (3) and Point-Grid Cell Distance in Equation (4). Therefore, $\nexists s \in g$, s.t. $dist(q, s) \leq \theta_K$. Similarly, no finer-grained cell in $\mathcal{G}_{r'}$ with $r' > r$ can contribute a segment whose distance to q is smaller than θ_K. Hence, the grid cell g and all its children can be pruned safely during the K-NN search. □

5.2 Bottom-Up-Down Search on HGL

When searching on the index for the frequency change of a point, we expect to prune unpromising branches earlier and obtain the final results as directly as possible. Usually, a top-down or best-first algorithm is utilized when using a hierarchical index structure. However, in our K-nearest segment search for a point q, we notably observe that the most possible candidates are usually located in finer resolutions while some undesirable segments with longer lengths that stay in a promising grid cell will be checked earlier. Thus, searching on the hierarchical grids in a straightforward top-down manner would result in the late coming of a tight pruning threshold as well as some unnecessary computation for those unpromising segments.

Therefore, we design a novel *Bottom-Up-Down* search algorithm to efficiently obtain the K-nearest segments on the hierarchical grid index. In short, it initially starts from the finest grid cell on the bottom and gradually climbs to the root following the path of grid cells with zero $MINdist$. After accessing the root, it transfers to a top-down search by checking the waiting grid cells with the priority of non-zero $MINdist$. Without loss of generality, we explain in detail the K-nearest segment search for point insertion in a single trajectory (as depicted in Algorithm 3).

Starting from the finest-grained grid (i.e., $\mathcal{G}_{r_H}$) can help to quickly obtain a relatively smaller distance threshold θ_K in an earlier stage, which is admittedly capable of providing more powerful pruning without damaging the search correctness. Thus, we first conduct a "bottom-up" search starting from the finest grid cell where the query point q exactly stays (line 5). When checking all segments within g_{candi}, the qualified segments are maintained by a priority queue (lines 12-15). If possible, we tighten the distance threshold θ_K to enhance the pruning (line 17). Meanwhile, the parent of g_{candi} and its unvisited children will be considered (lines 18-26). In particular, after the root has been reached, we move to a "top-down" search manner and another priority queue of grid cells Q_g is enabled hereafter(line 24), implying all grid cells with zero $MINdist$ to q have been visited already. The search can be early terminated as long as the currently top-1 candidate grid cell is worse than the lower bound θ_K (line 8-10), which means it is impossible to find any other segment or grid cell with a smaller distance to q (proven by Theorem 4). Finally, we insert point q into these returned candidate segments as the anonymized trace τ^* and update the hierarchical grids $\mathcal{HGL}_\tau$ accordingly.

6 Global Inter-trajectory Modification

Given a set of trajectories D and the original/perturbed TF distribution over the selected point set $\mathcal{P}$, two types of points with noisy TF should be processed during inter-trajectory modification: 1) insert a point q into Δl_q selected trajectories **at least once** to make its TF increasing Δl_q times, and 2) **completely** delete a point q from Δl_q trajectories to

Algorithm 3: K-Nearest Segment Search for Insertion

Input: τ: the target trajectory; $\mathcal{HGL}_\tau$: the local index for τ; q: target point to be inserted; Δf_q: # insertion
Output: τ^*: the modified trajectory

$Q_s \leftarrow \emptyset$; ▷ *priority queue for candidate segments*
$Q_g \leftarrow \emptyset$; ▷ *priority queue for grid cells in top-down*
$V \leftarrow \emptyset$; ▷ *record all visited grid cells*
$\theta_K \leftarrow +\infty$; $rootAccess \leftarrow false$;
$g_{candi} \leftarrow$ LocatePoint$(q, \mathcal{HGL}_\tau, r_H)$; ▷ *bottom-up starts*
while $rootAccess \neq true \vee Q_g \neq \emptyset$ **do**
  if $rootAccess = true$ **then**
    $\langle g_{candi}, dist\rangle \leftarrow Q_g.pop()$; ▷ *pop from queue*
    if $|Q_s| \geq \Delta f_q \wedge dist > \theta_K$ **then**
      break; ▷ *earlier termination*
  $V \leftarrow V \cup g_{candi}$; ▷ *mark it as visited*
  for $\forall s \in g_{candi}$ **do**
    $dist \leftarrow$ ComputeDistance(q, s);
    if $|Q_s| < \Delta f_q \vee dist < \theta_K$ **then**
      $Q_s.push(\langle s, dist\rangle)$;
  if $|Q_s| \geq \Delta f_q$ **then**
    $\theta_K \leftarrow Q_s.top().dist$; ▷ *update the threshold*
  for $g_c \in g_{candi}.children \wedge g_c \notin V$ **do**
    $mindist \leftarrow$ ComputeMINdist(q, g_c);
    $Q_g.push(\langle g_c, mindist\rangle)$;
  if $rootAccess = false \wedge g_{candi}.parent \notin V$ **then**
    if $g_{candi}.parent \in \mathcal{G}_{r_1}$ **then**
      $rootAccess = true$; ▷ *bottom-up is over*
      $Q_g.push(\langle g_{candi}.parent, 0\rangle)$; ▷ *top-down starts*
    else
      $g_{candi} \leftarrow g_{candi}.parent$;
$\tau^* \leftarrow$ ModifyAndUpdate$(\tau, \mathcal{HGL}_\tau, Q_s, q)$;
return τ^*;

make its TF decreasing Δl_q times. Here, $\Delta l_q = |l_q^* - l_q|$ represents the growth/drop of q's TF value after global perturbation. Meanwhile, the minimum utility loss of the dataset is expected when altering trajectories.

Naturally, we need to select the most appropriate trajectories for the insertion/deletion, such that the utility loss caused by modifying them can be minimized. That is, we can solve it as a K**-nearest trajectory search** problem where the "nearest" can be defined as below:

Definition 11 (K-Nearest Trajectory Search)**.** *Given a point q and a trajectory set D, the TF-increasing task, i.e., inserting q into Δl_q selected trajectories of D at least once, aims to find a list of top-Δl_q trajectories with the minimum insertion utility loss for q, where $\Delta l_q = |l_q^* - l_q|$, such that:*

$$\mathbb{L}[\mathcal{OP}_i(q,\tau)] \leq \mathbb{L}[\mathcal{OP}_i(q,\tau')]$$
$$\forall \tau \in NT_q^+, \forall \tau' \in \{D \setminus NT_q^+\}$$

where NT_q^+ contains the Δl_q-nearest trajectory selected from D for the increasing TF of q.

By contrast, the TF-decreasing task requires completely deleting q from Δl_q trajectories in D, which is equivalent to finding a list of top-Δl_q trajectories with the minimum complete deletion loss for q, such that:

$$\mathbb{L}[\mathcal{OP}_d(q,\tau)] \leq \mathbb{L}[\mathcal{OP}_d(q,\tau')]$$
$$\forall \tau \in NT_q^-,\ \forall \tau' \in \{D \setminus NT_q^-\}$$

where NT_q^- records the selected trajectories for point deletion with the size of $|NT_q^+| = \Delta l_q$.

6.1 Hierarchical Grid-based Global Index (HGG)

The computational complexity of linear search reaches $O(dn\bar{l})$ for the inter-trajectory modification. Here, d denotes the total dimensionality of $\mathcal{P}$ (reaching $O(mn)$ in the worst case where m is the signature size); n and $\bar{l}$ are the cardinality of the dataset and average trajectory length respectively. Naturally, a naive alternative solution for the global inter-trajectory modification is the *"multi-HGL"* method, namely, independently constructing multiple HGL indices (one HGL for each trajectory), performing the "nearest segment search" on each HGL index one by one, and finally aggregating the results to answer the K-nearest trajectory search. Nevertheless, it is obviously quite time-consuming. In fact, the natural locality of spatial trajectories results in the true top-K answers being located/gathered around the given point, while such a multi-HGL solution cannot avoid the excessive unnecessary computation on those trajectories that are far away from the query point.

Therefore, we construct another hierarchical grid-based index to further speed up the global inter-trajectory modification by putting all segments from all trajectories in the dataset into a single index structure (associated with extra information like which trajectory a segment belongs to) and performing the top-K search similarly as in the local intra-trajectory modification. It indeed works, however, when massive segments are gathered into a single grid cell and organized in a disordered manner, the search within such a dense grid cell becomes much slower – the bottleneck of efficiency. To address this issue, we design the ***H**ierarchical **G**rid-based **G**lobal* index, denoted by HGG hereafter, which is built on top of HGL yet enhanced by a four-bit encoding strategy so as to structurally organize a grid cell in a more accurate way and tighten the lower bound of the distance between a query point and a segment (bounded by Definition 10 in the HGL), further improving the efficiency of trajectory modification to a large extent.

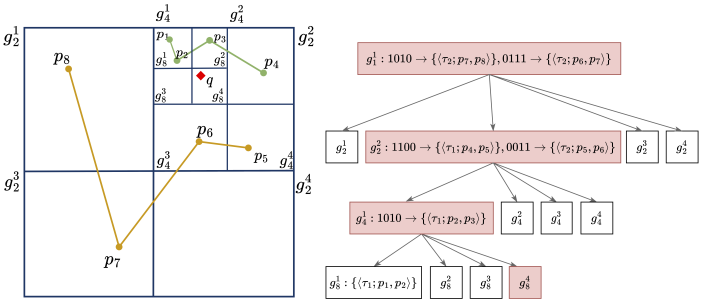

Fig. 3. An example of HGG index for the entire set of trajectories.

6.1.1 Four-bit Encoding

Indeed, a coarser-grained grid cell gives space to those longer segments that cannot stay at its finer-grained child nodes, while linearly storing a large number of segments within a single grid cell cannot avoid some unnecessary computation that may happen frequently. For example in Figure 3, the segment $\langle p_7, p_8\rangle$ of the trajectory τ_2 (in yellow) has to be checked when the query point q is looking for its nearest trajectory since it locates in g_1^1 whose $MINdist$ to q is zero, as defined in Equation (4). However, it is obvious

that $\langle p_7, p_8\rangle$ cannot be the top-1 result and such unnecessary computation is caused by the loose bound between q and $\langle p_7, p_8\rangle$, i.e., $dist(q, \langle p_7, p_8\rangle) \geq MINdist(q, g_1^1) = 0$. Inspired by this, our four-bit encoding helps to organize all segments within the same grid cell in the following way:

Definition 12 (Four-bit Encoding)**.** *An arbitrary grid cell g_r^i is quad-divided into 2×2 rectangles, each of which is simply labeled by one bit, i.e., $E = \{0001, 0010, 0100, 1000\}$. Given a segment $s = \langle p_x, p_y\rangle$ whose best-fit grid cell is g_r^i, its encoding e_s is determined by two endpoints p_x and p_y. In specific, we denote $e_x, e_y \in E$ as the encoding of p_x and p_y, respectively, indicating a specific finer rectangle of g_r^i where the point is located more accurately. Then, we encode the segment as $e_s = (e_x|e_y)|\sigma_{xy}$, where $\sigma_{xy} = 0000$ if the segment does not touch any other finer rectangle; otherwise $\sigma_{xy} \in E \backslash \{e_x, e_y\}$ encodes the third crossing rectangle. Besides, $|$ represents the bit-wise OR operator.*

In other words, the encoding is like a combination of (at least) two or (at most) three finer grid cells, depicting the position of a segment in a more detailed way. As a simple illustration, the segment $\langle p_7, p_8\rangle$ in Figure 3 can be further encoded by "1010", implying that it passes the first (i.e., top-left) and third (i.e., bottom-left) cells in g_1^1.

6.1.2 Pruning based on Four-bit Encoding

After applying the four-bit encoding to describe the more accurate position of each segment in a grid cell, we can bound the distance between a point and a segment in a more strict way to obtain a stronger pruning power. The distance between a point to multiple cells is defined as follows:

Definition 13 (Point-Multi Cells Distance)**.** *Given a point q and a set of grid cells $S_g = \{g_1, g_2, \dots\}$, the minimum distance from q to these cells is computed as:*

$$MINdist(q, S_g) = \min_{g \in S_g} MINdist(q, g)$$

In this way, the lower bound distance between a point and a segment can be smoothly tightened as the minimum distance from the point to the segment's encoding cells. Still using the query point q and the segment $s = \langle p_7, p_8\rangle$ in Figure 3 as an example: without encoding, the minimum distance is estimated as zero, namely $dist(q, s) \geq MINdist(q, g_1^1) = 0$; by making use of the encoding (i.e., $e_s = 1010$), the bound can be tightened as $dist(q, s) \geq MINdist(q, S(e_s, g_1^1)) > 0$ where $S(e, g_r^i)$ denotes the set of quad-cells encoded by e in a grid cell g_r^i. In some cases when a non-zero threshold has appeared, it can help to filter out a set of segments encoded by the same representation and avoid some unnecessary computation (easily proved similar to Theorem 4). For instance, assume we already obtain the $\langle p_2, p_3\rangle$ as q's nearest segment and hold a pruning threshold $\theta = dist(q, \langle p_2, p_3\rangle)$. When it comes to g_2^2 along with two segments, namely $\langle p_4, p_5\rangle$ in trajectory τ_1 encoded by "1100" and $\langle p_5, p_6\rangle$ in τ_2 encoded by "0011" respectively, there is no need to compute the exact utility loss of $\langle p_5, p_6\rangle$ for q since $dist(q, \langle p_5, p_6\rangle) \geq MINdist(q, S(0011, g_2^2)) > \theta$. Similarly, $\langle p_7, p_8\rangle$ will not bother the search either.

6.2 Bottom-Up-Down Search on HGG

The major difference between searching on HGG and that on the local index HGL (as detailed in Section 5.2) is that, when checking all segments within a single grid cell, we do not need to check the segments one by one. Instead, we could still utilize the encoding-based distance computation to prune some unpromising segments further. Technically, we replace the line 12-15 in Algorithm 3 with the new Algorithm 4. We first sort all encoding representations in g_{candi} based on the overlapping with q's encoding e_q (line 1 in Algorithm 4), since those with more overlapping bits should be more likely to cover the promising answers. Segments represented by the same encoding e_s with the unqualified distance bound $MINdist > \theta_K$ should not be considered anymore, otherwise, they are checked as usual (line 4-8). By employing the encoding strategy and the tightened distance bound, more segments can be pruned without unnecessary computation and the efficiency can be improved greatly when processing the global inter-trajectory modification with excessive segments gathered in a grid cell.

Algorithm 4: Check segments in a grid cell g_{candi}

```
1 E_s ← SortEncoding(e_q, g_candi);
2 for ∀e_s ∈ E_s do
3     mindist ← ComputeMINdist(q, S(e_s, g_candi));
4     if |Q_s| < Δf_q ∨ mindist < θ_K then
5         for ∀s ∈ e_s do
6             dist ← ComputeDistance(q, s);
7             if |Q_s| < Δf_q ∨ mindist < θ_K then
8                 Q_s.push(⟨s, dist⟩);
```

7 Experiments

We have conducted extensive experiments on a real-world trajectory dataset to evaluate the effectiveness and efficiency of our proposed methods. All the algorithms are implemented in C++, and all the experiments run on a server with two Intel(R) Xeon(R) CPU E5-2630, 10 cores/20 threads at 2.2GHz each, 378GB memory, and Ubuntu 16.04 OS.

7.1 Experimental Setting

Dataset: We evaluate our proposed models on a public one-week taxi dataset, *T-Drive* [26], generated by 10,357 Beijing taxis with more than 15 million GPS points. The average sampling rate is 3.1 minutes per point. Each taxi is associated with a single trajectory (i.e., its entire moving history), and each trajectory consists of 1,813 points on average.

Compared Methods and Parameter Setting: We compare our algorithms with several leading privacy models, and the parameters are set based on the original work:

- *K-anonymity-based models:* 1) W4M [6] ensures each trajectory is indistinguishable from other k-1 traces in a group; 2) GLOVE [7] achieves k-anonymity for region-based trajectories via spatiotemporal generalization; 3) KLT [8] extends GLOVE by ensuring not only k-anonymity but also l-diversity and t-closeness ($k = 5$ for all models and $l = 3, t = 0.1$ for KLT);
- *Signature closure* (SC) [4] is an effective trajectory protection method by discarding all top-m signature points ($m = 10$); *Radius-based signature closure* (RSC-α) is an enhanced variant of SC dropping extra points within the radius of α centered at any signature points ($\alpha \in [0.1, 0.5, 1, 3, 5]$, unit: km);

- *Generation-based DP models:* 1) DPT [9] is a pioneering model generating differentially private synthetic trajectories; 2) AdaTrace [23], [24] outperforms DPT by combing differential privacy with attack resilience and utility-aware generator ($\epsilon = 1.0$);
- *Frequency-based randomized DP models:* 1) *PureG* denotes our global mechanism with ϵ_G privacy; 2) *PureL* represents our local mechanism with ϵ_L privacy; 3) *GL* combines the global/local mechanisms (with exchangeable composition ordering) providing ϵ-differential privacy. ($\epsilon_G = \epsilon_L = 0.5$, $\epsilon = 1.0$). Top-m signature points are employed for frequency perturbation (10 by default).

Effectiveness Metrics: We compare these models in terms of privacy protection, utility preservation, and data recovery.

- *Privacy protection*: linking accuracy (LA_s) by a state-of-the-art user re-identification model [3]; mutual information (MI) [20], [27] generally measures the dependency between the original/anonymized data. Smaller LA_s/MI means better privacy protection.
- *Utility preservation*: point-level information loss (INF) [28] at the statistical level; the divergence of diameter distribution (DE) and trip distribution (TE) [23] from the spatial aspect; and the F-measure of frequent pattern mining (FFP) [29]. Smaller INF, DE, TE, and larger FFP indicate better utility preservation.
- *Trajectory recovery*: the success rate of recovering trajectories via an HMM-based map-matching technique measured from various aspects [30], [31].
- *Signature recovery*: the possibility of recovering the original signature points from the anonymized data.

Efficiency: We also evaluate model efficiency by competing with three baselines: the *Linear* comparison, the single-level uniform grid index (UG), and the basic hierarchical grid-based index in [32] (denoted as HG). Our new proposal in this paper is denoted by $HGLG$ representing the combination of the local index HGL and the newly designed global index HGG. Regarding other variants, hierarchical grid indices can flexibly integrate with various search strategies: top-down, bottom-up, and bottom-up-down. The granularity of the uniform grid and the finest level in hierarchical grid indices is set to 512×512 cells by default.

7.2 Effectiveness Evaluation

Table 2 reports the empirical results of the effectiveness comparison among the anonymization models.

1) Privacy Protection. Overall, k-anonymity-based models are ineffective for preventing the powerful linkage attack, although the mutual information between the original and the anonymized trajectories is acceptable. On the contrary, *DPT* and *AdaTrace*, as remarkable generative differential privacy models, are the most effective in privacy protection but with a huge sacrifice in data utility. *SC* also achieves relatively low linkage accuracy via all types of signatures except the spatiotemporal one. A similar trend happens in *RSC* which extends the point removal to a region centered at the signature points. This verifies the importance of signature points to prevent re-identification. Our DP model introduces frequency-based randomization based on the signatures, while its capability against reidentification is still desirable. We observe that the *PureG* model performs unsatisfactorily overall, despite globally perturbing trajectory frequency distribution is capable of damaging some mutual information. Fortunately, simply implementing the local point frequency perturbation and intra-trajectory modification (i.e., *PureL*) can largely reduce the linkage accuracy already ($LA_s < 12\%$). Moreover, the effect of protection can be further magnified when both local and global mechanisms are integrated based on Theorem 1, especially when dealing with the spatial-based linking attack ($LA_s = 1.6\%$). This undoubtedly verifies the effectiveness of our proposed DP model in privacy protection.

2) Utility Preservation. Privacy protection should not sacrifice data utility too much. *DPT* performs the worst in utility preservation among all the compared DP-based methods. Although DE and TE are relatively low ($\approx 30\%$) as *DPT* fully captures the objects' spatial transition behavior, it ends up with a low FFP and an extremely high INF which measures to what extent original points are lost after anonymization. *DPT* destroys almost 99% of the points due to the transmit-based synthetic generation framework, which impacts the performance of frequent pattern mining. *AdaTrace* shows a better ability to reserve more data utility thanks to its specially designed utility-aware synthesizer superior to *DPT*. In comparison, k-anonymity-based approaches perform differently: *W4M* is the best k-anonymity model in terms of utility preservation and, as mentioned, *GLOVE* and *KLT* outperform *W4M* in privacy protection which is at the price of huge utility damage. On the contrary, the signature-based methods (*SC*, *RSC*, and ours) distort only a limited number of signature points while keeping most of the other points unchanged, which naturally achieves satisfactory performance in utility preservation. The *RSC* variants with different radius α are surpassed by *SC*, with more points being deleted from the original trajectories. We can observe that all three versions of our proposed DP models reserve sufficient data utility after frequency perturbation and trajectory modification ($INF = 60\%$, $TE = 30\%$, and $FFP = 96\%$). Moreover, our model is extraordinarily powerful in retaining the diameter information in the anonymized trajectories, as demonstrated by a negligible divergence of diameter distribution ($DE < 1.5\%$). Overall, randomizing frequency distributions of trajectory points instead of brutally discarding them can provide not only the strong capability of resisting various privacy threats but also the controlled data utility, outperforming the generation-based DP models a lot.

3) Trajectory Recovery. Few works have paid attention to the recovery risks, i.e., the capability of reconstructing the original traces from the anonymized one using technologies like map-matching. In our experiments, we use the well-known HMM-based map-matching [30] to simulate the recovery attack on the protected dataset and apply various metrics to evaluate the success rate of data recovery. Specifically, the *F-score* and the length-based *Route Mismatch Fraction* (RMF) evaluate from the route-based view [33], whilst the *Accuracy* shows the performance of point-based matching [31]. Since our frequency-based DP mechanisms might insert extra points into the original data, making the output trajectories longer, RMF can exceed 1. The higher the RMF, the more erroneous the recovery and the better protection the model offers.

TABLE 2
A summary of effectiveness evaluation results ($|D| = 1000$ and $\epsilon = 1.0$)

	Metric	SC	RSC-0.1	RSC-0.5	RSC-1.0	RSC-3.0	RSC-5.0	W4M	GLOVE	KLT	DPT	AdaTrace	PureG	PureL	GL
Privacy	LA_s	0.188	0.146	0.098	0.081	0.049	0.033	0.847	0.370	0.269	0.006	0.000	0.922	0.119	0.016
	MI	0.163	0.162	0.158	0.156	0.150	0.148	0.244	0.385	0.312	0.097	0.057	0.184	0.098	0.095
Utility	INF	0.656	0.663	0.682	0.704	0.784	0.853	0.285	0.912	0.929	0.986	0.603	0.495	0.635	0.642
	DE	0.047	0.050	0.064	0.082	0.138	0.243	0.057	0.587	0.617	0.297	0.291	0.004	0.015	0.014
	TE	0.385	0.380	0.373	0.370	0.333	0.308	0.330	0.657	0.686	0.375	0.064	0.225	0.365	0.331
	FFP	0.990	0.988	0.983	0.977	0.943	0.929	0.994	0.416	0.330	0.373	0.908	0.980	0.953	0.956
Recovery	Precision	0.610	0.610	0.606	0.598	0.561	0.535	0.276	0.498	0.435	-	-	0.378	0.307	0.309
	Recall	0.611	0.605	0.578	0.540	0.384	0.251	0.157	0.307	0.276	-	-	0.481	0.331	0.339
	F-score	0.611	0.608	0.591	0.568	0.456	0.342	0.179	0.380	0.338	-	-	0.423	0.318	0.324
	RMF	0.779	0.782	0.798	0.823	0.916	0.967	0.693	0.516	0.592	-	-	1.310	1.342	1.420
	Accuracy	0.162	0.162	0.162	0.162	0.160	0.152	0.035	0.250	0.218	-	-	0.395	0.189	0.008

[1] We ignore the data recovery experiments for *DPT* and *AdaTrace* since the generated synthetic trajectories are no longer aligned to the road network.

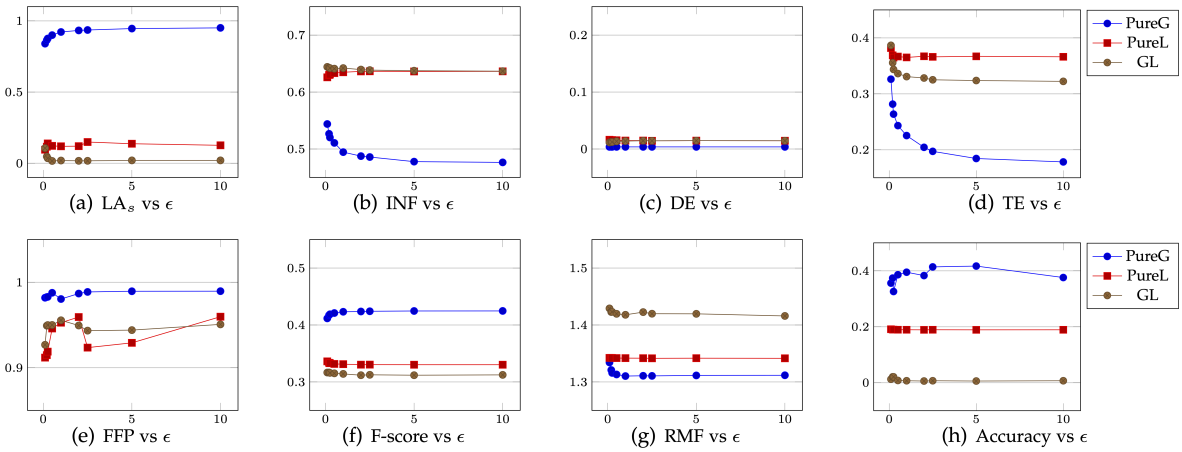

Fig. 4. The impact of ϵ (with $|D| = 1000$). The result of MI is neglected since it demonstrates a similar trend with LA_s.

First, simply removing these personally-identifying signature points (*SC*) and their nearby neighbors (*RSC*) is insufficient for trajectory privacy, since many raw points ($> 60\%$ in Table 2) can be recovered by the HMM-based map-matching. Interestingly, recovering the trajectories anonymized by *W4M* is much more difficult, because *W4M* enforces that each trajectory should be spatially close to its pivot trajectory, making itself deviate from the real paths. The region-based generalized trajectories from *GLOVE* and *KLT* are highly possible to hit one of its k-components and thus being successfully recovered via map-matching. In comparison, our frequency-based DP models (i.e., *PureG*, *PureL*, and *GL*) are capable of resisting recovery attacks, as evidenced by the poor map-matching results ($\approx 33\%$ route-based F-score and $\approx 10\%$ point-based Accuracy). It again proves the effectiveness of our main idea that perturbing the frequency distributions of signature points can help to protect the most sensitive geo-information. In other words, introducing probabilistic noise to the distributions leads to an unpredictable rise or drop in the frequency, making it harder to recover from the anonymized data.

4) Signature Recovery. Notably, guaranteeing that it is impossible to infer the original signature points from the modified trajectories is indeed indispensable, since the adversary might possess various prior knowledge like the trajectory modification procedure. In particular, we explore the possibility of signature recovery from two aspects: 1) the similarity between the original signatures and the new ones extracted from the anonymized data; and 2) the spatial distance between the original signatures and the top-ranked frequent segments in the anonymized trajectories.

From the view of *pointwise comparison*, we first conduct an experiment with 1000 trajectories to double-check the common points appearing in both signatures extracted from each original trajectory and the modified one respectively. It turns out that more than 21% of taxis have zero overlapping between their original and anonymized signatures, and the majority of trajectories ($> 90\%$) share less than two common signature points after our randomization-modification procedure, implying that the original signature points are hardly possible to rank top in the anonymized data.

On the other hand, if the anonymized trajectory frequently passes around a signature point (i.e., nearby segments), it is still possible to infer the original signature. Hence, we conduct another experimental study for *segment-based comparison*. The top-ranked frequent trajectory segments are first extracted from the anonymized trajectories, and then the average distances between each segment and the original signature points are computed. As shown in Figure 5, after our differentially private randomization, most of the top-frequent segments are far away from the original signature points (e.g., around 85% segments with the average distance of more than $5km$), indicating the most representative segment-level information is almost irrelevant to the original signatures in geographic space. Therefore, it is still extremely hard to recover the original signatures even if the top-ranked frequent segments are exposed.

5) The Impact of ϵ and Mechanism Comparison. Two

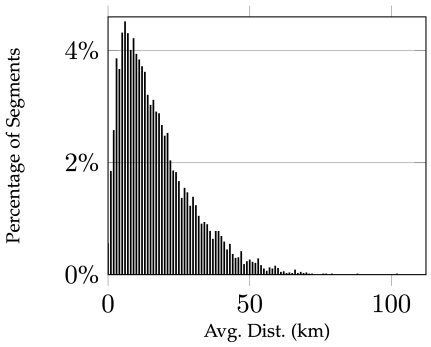

Fig. 5. The distribution of the average distance between the original signatures and the top frequent segments in the anonymized data.

independent DP mechanisms, namely, the global TF perturbation over the entire dataset and the local PF perturbation for each trajectory, were first proposed. We then combine them and evenly allocate the total privacy budget ϵ, i.e., $\epsilon_G = \epsilon_L = \frac{1}{2}\epsilon$. Here, we investigate the impact of the privacy budget on overall performance. We set ϵ_G and ϵ_L within $[0.1, 10.0]$ and report the effectiveness of the three variants (namely, *PureG*, *PureL*, and *GL*) in Figure 4.

It can be observed that simply utilizing the global mechanism (*PureG*) performs worst in balancing privacy and utility, and the gap of linking accuracy is gradually enlarged when the privacy budget ϵ grows. Indeed, *PureG* obscures the unique information in the whole dataset, i.e., some distinctive locations visited by fewer users. However, the sensitive points that are highly representative of a specific individual have not been distorted properly. On the other hand, the perturbation of local frequency distribution hides most of the personally-identifying information from the trajectories with a novel Laplace noise injection mechanism, achieving satisfactory protection. Whilst the inferiority of *PureG* in preventing attacks is compensated by the improved data utility, as verified by lower INF and TE as well as higher FFP. Furthermore, the global alteration is much easier to be recovered by map-matching than the local alteration as fewer changes are made to each trajectory during the inter-trajectory modification. Overall, utility preservation enhances while privacy protection degrades with the increase of privacy budget, mainly because fewer noises are injected into the trajectory data from Laplace distribution when ϵ raises. Our frequency-based randomization mechanisms (in particular the models that integrate local PF alteration) are not quite sensitive to the change of ϵ, as evidenced by the stable trend.

6) The Impact of Signature Size m. Since the protection benefit primarily comes from the personal implication of the top-ranked signature points, it is worth discussing to what extent the signature size would affect the performance. So an experiment is conducted to examine the performance when varying the signature size m. As depicted in Figure 6, an overall trend is that enlarging the signature size means distorting more points and more changes in frequency distributions, thus gaining better protection against both reidentification and recovery attacks, however, with an increasing cost of data utility. Another interesting observation is that the combined mechanism *GL* performs slightly better than *pureL* among all metrics and the privilege of local-involved mechanisms beats *pureG*, but the gap between *pureG* and the other two mechanisms narrows with the growth of m. The impact of m is fully exhibited as more points with personal interests participate in DP-based randomization which supplements the limitation of the global mechanism. Additionally, the HMM-based recovery becomes harder due to the larger amount of changing elements of the original trajectories. Nonetheless, the utility loss should not be neglected (e.g., *INF* increases more than half, and the decrease of *FFP* reaches 15% at most). Therefore, it is vital to select an appropriate signature size for the proposed mechanisms.

7.3 Efficiency Evaluation

We evaluate the efficiency of our proposed indices (denoted by $HGLG$, namely, the combination of hierarchical grid-based local/global index HGL and HGG) with different search strategies compared to three baselines. Two parameters are discussed further: data size and grid granularity.

1) The Impact of Dataset Size. We vary the dataset size $|D|$ (i.e., the total number of objects/trajectories) within {1000, 2000, 4000, 6000, 8000, 10000}, and report the overall efficiency. Note that the grid granularity of a uniform grid and that of the finest level in the hierarchical grid structures is set to 512×512 cells here. As shown in Figure 7(a), our hierarchical grid indices (both the enhanced $HGLG$ and the basic HG) are quite powerful in filtering unpromising candidates during K-nearest neighbor search, leading to dramatically higher efficiency than the *Linear* approach (reduce the time cost by more than 95 times at most). Compared to the single-level uniform grid index UG, the hierarchical structure helps to properly organize length-variable trajectory segments and drop misinformation, contributing to a five times improvement in efficiency. In particular, the $HGLG$ pays more attention to the special situation of global modification and employs the more powerful global index HGG which is equipped with the four-bit encoding and the enhanced pruning power, resulting in a huge improvement in efficiency (≈ 2.5 times) compared to the imperfect performance of HG [32].

2) The Superiority of Bottom-Up-Down Search. As shown in Figure 7(b), the index $HGLG$ combining with *bottom-up* search algorithm is faster than searching in the *top-down* manner, both of which are outperformed by the newly proposed *bottom-up-down* strategy (exceed to more than two times at most). And the difference in efficiency widens as the data size increases. Admittedly, the promising candidate segments usually stay at the same grid cells where the query point is located no matter at which resolution. Nevertheless, starting from the finest-grained grid cell covering the query point can significantly accelerate the process of threshold shrinking, which in turn enhances the pruning power of $HGLG$ and avoids more unnecessary calculations.

3) The Superiority of Global Index with Encoding. We also examine the usefulness of the newly proposed global index HGG along with its four-bit encoding scheme, which is particularly good at the global inter-trajectory modification task. In Figure 7(c), the multi-HGL as a naive solution consumes excessive unnecessary costs that waste on constructing the individual local index HGL for every object and checking those unpromising trajectories for adapting the perturbed global TF distribution, mainly because of ignoring the locality property of trajectories and the human

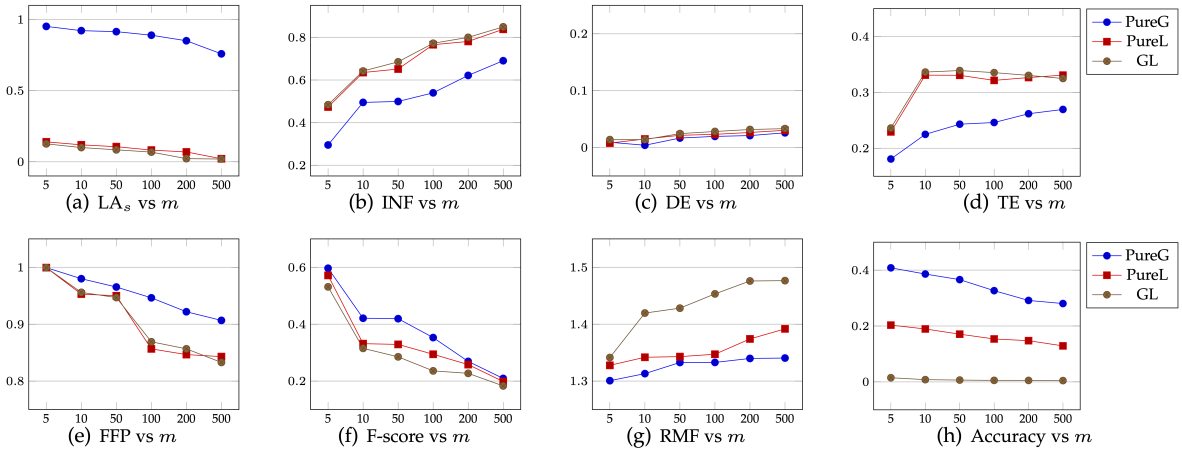

Fig. 6. The impact of signature size m (with $|D| = 1000$ and $\epsilon_G = \epsilon_L = 0.5$).

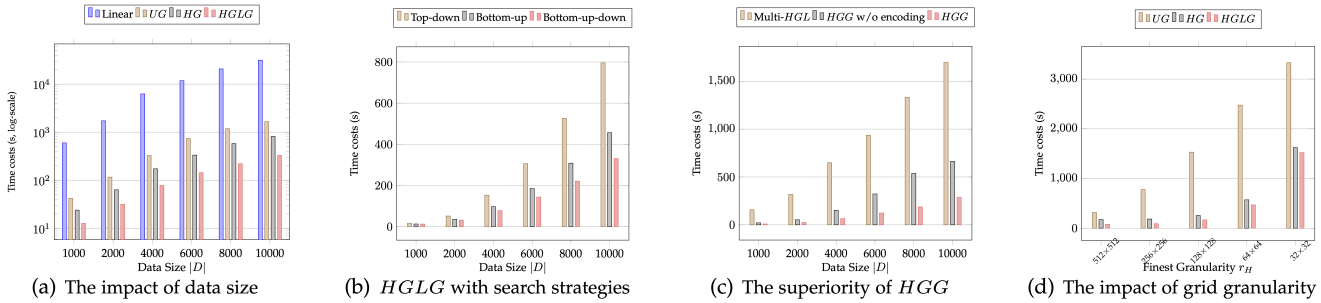

Fig. 7. Efficiency comparison ($\epsilon_G = \epsilon_L = 0.5$). Particularly, in (a)-(c), the grid granularity is fixed as 512×512 for grid-based solutions with the data size varying; in (d), the impact of grid granularity on the efficiency is fully discussed with the maximum data size $|D| = 4,000$.

daily patterns in real life. By contrast, the HGG without the four-bit encoding strategy relatively mitigates the efficiency load, outperforming the multi-HGL more than two times. However, it still suffers from comparing massive segments in a single grid cell and shows undesirable performance when tackling large-scale data. The enhanced global index HGG performs best for the inter-trajectory modification, leading to around three times improvement further, thanks to the well-encoded segments and the tightened bound between a point and the candidate segments.

4) The Impact of Grid Granularity. We also vary the grid granularity r_H (i.e., # of grid cells on the finest level in hierarchical indices, or the total grid cells in UG) within $\{512 \times 512, 256 \times 256, 128 \times 128, 64 \times 64, 32 \times 32\}$, and present the overall efficiency of grid-based approaches in Figure 7(d). Note that we do not consider the finer granularity like 1024×1024 anymore, since the area of a cell is nearly $0.8\ km^2$ under the resolution 512×512 in Beijing which is reasonable in practice. As expected, the grid-based indices work much better when the space is partitioned in a sufficiently finer way but the performance gets worse as the granularity becomes coarser. We observe that the global inter-trajectory modification is relatively more affected by the change in grid granularity. Regarding the single-level uniform grid UG, the larger each grid cell, the more crowded the line segments gather and the more expensive the search regardless of either insertion or deletion. Similarly, a fine-grained grid partition helps to arrange variable-length segments in a proper resolution and express accurate information when searching on the hierarchical index. Furthermore, holding a threshold for pruning unpromising cells/segments will be practically effective only if the hierarchical structure makes sense (i.e., with enough levels). As a result, it is recommended to determine the grid granularity based on the preliminary observation of the datasets like the average/min/max length of trajectory segments and their distribution in the geographic space.

8 Conclusion

In this work, we propose a frequency-based randomization model with differential privacy guarantees to protect identity privacy that might be exposed from spatial trajectories. Two independent perturbation mechanisms are provided by adding non-trivial Laplace noises to the local point frequency and global trajectory frequency distributions over top-ranked private signature points, respectively. Moreover, the task of trajectory modification for reflecting distorted frequency distributions with minimum utility loss is formalized as K-nearest trajectory (segment) search problems. To support efficient search, we leverage the locality property of trajectories and design two hierarchical grid indices combined with an encoding-based structure and a novel bottom-up-down search algorithm, obtaining superior performance in a large-scale trajectory dataset. Empirically, our privacy model outperforms most existing competitors, achieving a satisfactory balance between privacy protection and utility preservation. Besides, the frequency-perturbed trajectories can hardly be recovered by map-matching techniques.

ACKNOWLEDGMENT

This work was supported in part by the Australian Research Council under Grants DP200103650 and LP180100018 and in part by the Natural Science Foundation of China under Grants 62072125 and 61902134.

REFERENCES

[1] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific Reports*, vol. 3, p. 1376, 2013.
[2] W. Chen, H. Yin, W. Wang, L. Zhao, W. Hua, and X. Zhou, "Exploiting spatio-temporal user behaviors for user linkage," in *ACM CIKM*, 2017, pp. 517–526.
[3] F. Jin, W. Hua, J. Xu, and X. Zhou, "Moving object linking based on historical trace," in *IEEE ICDE*, 2019, pp. 1058–1069.
[4] F. Jin, W. Hua, T. Zhou, J. Xu, M. Francia, M. Orowska, and X. Zhou, "Trajectory-based spatiotemporal entity linking," *IEEE TKDE*, 2020.
[5] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang, "Traffic-aware multiple mix zone placement for protecting location privacy," in *IEEE INFOCOM*, 2012, pp. 972–980.
[6] O. Abul, F. Bonchi, and M. Nanni, "Anonymization of moving objects databases by clustering and perturbation," *Information Systems*, vol. 35, no. 8, pp. 884–910, 2010.
[7] M. Gramaglia and M. Fiore, "Hiding mobile traffic fingerprints with GLOVE," in *ACM CoNEXT*, 2015, pp. 1–13.
[8] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Protecting trajectory from semantic attack considering k-anonymity, l-diversity, and t-closeness," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 264–278, 2019.
[9] X. He, G. Cormode, A. Machanavajjhala, C. M. Procopiuc, and D. Srivastava, "Dpt: Differentially private trajectory synthesis using hierarchical reference systems," *VLDB*, 2015.
[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*, 2006, pp. 265–284.
[11] F. Jin, W. Hua, M. Francia, P. Chao, M. Orowska, and X. Zhou, "A survey and experimental study on privacy-preserving trajectory data publishing," *IEEE TKDE*, 2022.
[12] K. Jiang, D. Shao, S. Bressan, T. Kister, and K.-L. Tan, "Publishing trajectories with differential privacy guarantees," in *SSDBM*, 2013.
[13] B. Palanisamy and L. Liu, "Mobimix: Protecting location privacy with mix-zones over road networks," in *IEEE ICDE*, 2011.
[14] P.-R. Lei, W.-C. Peng, I.-J. Su, C.-P. Chang *et al.*, "Dummy-based schemes for protecting movement trajectories," *Journal of Information Science and Engineering*, vol. 28, no. 2, pp. 335–350, 2012.
[15] R. Kato, M. Iwata, T. Hara, A. Suzuki, X. Xie, Y. Arase, and S. Nishio, "A dummy-based anonymization method based on user trajectory with pauses," in *ACM SIGSPATIAL*, 2012, pp. 249–258.
[16] X. Wu and G. Sun, "A novel dummy-based mechanism to protect privacy on trajectories," in *IEEE ICDM Workshop*, 2014.
[17] X. Liu, J. Chen, X. Xia, C. Zong, R. Zhu, and J. Li, "Dummy-based trajectory privacy protection against exposure location attacks," in *WISA*, 2019, pp. 368–381.
[18] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," in *IEEE ICDE*, 2008.
[19] J. Hua, Y. Gao, and S. Zhong, "Differentially private publication of general time-serial trajectory data," in *IEEE INFOCOM*, 2015.
[20] M. Li, L. Zhu, Z. Zhang, and R. Xu, "Achieving differential privacy of trajectory data publishing in participatory sensing," *Information Sciences*, vol. 400, pp. 1–13, 2017.
[21] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *IEEE Symposium on Security and Privacy*, 2016, pp. 546–563.
[22] K. Al-Hussaeni, B. C. Fung, F. Iqbal, G. G. Dagher, and E. G. Park, "Safepath: Differentially-private publishing of passenger trajectories in transportation systems," *Computer Networks*, 2018.
[23] M. E. Gursoy, L. Liu, S. Truex, and L. Yu, "Differentially private and utility preserving publication of trajectory data," *IEEE Trans. Mob. Comput.*, vol. 18, no. 10, pp. 2315–2329, 2019.
[24] M. E. Gursoy, L. Liu, S. Truex, L. Yu, and W. Wei, "Utility-aware synthesis of differentially private and attack-resilient location traces," in *ACM CCS*, 2018, pp. 196–211.
[25] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy." *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.
[26] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in *ACM SIGSPATIAL*, 2010, pp. 99–108.
[27] W. Yang, N. Li, Y. Qi, W. H. Qardaji, S. E. McLaughlin, and P. D. McDaniel, "Minimizing private data disclosures in the smart grid," in *ACM CCS*, 2012, pp. 415–427.
[28] P.-I. Han and H.-P. Tsai, "Sst: Privacy preserving for semantic trajectories," in *IEEE MDM*, vol. 2, 2015, pp. 80–85.
[29] S. Gurung, D. Lin, W. Jiang, A. Hurson, and R. Zhang, "Traffic information publication with privacy preservation," *ACM TIST*, vol. 5, no. 3, pp. 1–26, 2014.
[30] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *ACM SIGSPATIAL*, 2009.
[31] P. Chao, Y. Xu, W. Hua, and X. Zhou, "A survey on map-matching algorithms," in *Australasian Database Conference*. Springer, 2020.
[32] F. Jin, W. Hua, B. Ruan, and X. Zhou, "Frequency-based randomization for guaranteeing differential privacy in spatial trajectories," in *IEEE ICDE*, 2022, pp. 1727–1739.
[33] G. Wang and R. Zimmermann, "Eddy: An error-bounded delay-bounded real-time map matching algorithm using hmm and online viterbi decoder," in *ACM SIGSPATIAL*, 2014, pp. 33–42.

Fengmei Jin is a Research Assistant Professor in the Department of Computing at The Hong Kong Polytechnic University. She obtained her PhD from The University of Queensland in 2023. Prior to that, she received her Bachelor and Master degrees from Sun Yat-Sen University in 2016 and Renmin University of China in 2019, respectively. Her research interests include spatiotemporal databases, trajectory data privacy, efficient indexing, and trajectory-user linking.

Wen Hua is an Associate Professor under the Presidential Young Scholar scheme in the Department of Computing at the Hong Kong Polytechnic University. She received her Bachelor's and PhD degrees in Computer Science from the Renmin University of China in 2010 and 2015, respectively. Her research interests include information extraction, knowledge graph, and spatiotemporal data management.

Lei Li is an Assistant Professor at the Hong Kong University of Science and Technology (Guangzhou). He obtained his PhD in 2018 from the University of Queensland under Prof. Xiaofang Zhou's supervision. He obtained his Bachelor and Master degrees from Harbin Institute of Technology in 2012 and 2014, respectively. His research interests include spatial and temporal data, graph, and distributed databases.

Boyu Ruan is a postdoctoral fellow at The Hong Kong University of Science and Technology. He received his Bachelor of Engineering degree from Tsinghua University in 2015 and PhD degree from The University of Queensland in 2022. His research interests include graph theory, data quality and incremental algorithms.

Xiaofang Zhou is Otto Poon Professor of Engineering and Chair Professor at The Hong Kong University of Science and Technology. Before joining HKUST, he was a Professor of Computer Science at The University of Queensland from 1999 to 2020. His research interests include spatial and multimedia databases, high-performance query processing, data mining, data quality management, and machine learning. He is a Fellow of IEEE.