An Efficient Local Differential Privacy Approach for Trajectory Publishing with High Utility

Haolong Yang[1], Dingyuan Shi[1], Yuanyuan Zhang[2(✉)], Yi Xu[1(✉)], and Ke Xu[1]

[1] State Key Laboratory of Complex & Critical Software Environment, Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, School of Computer Science and Engineering and Institute of Artificial Intelligence, Beihang University, Beijing, China
[2] North China Institute of Computing Technology, Beijing, China
{yanghaolong,chnsdy,zyy-buaa,xuy,kexu}@buaa.edu.cn

Abstract. Trajectory data underpins many data-driven spatiotemporal applications, such as navigation or ride-hailing. However, privacy concerns hinder the collection and utilization of high-quality, large-scale trajectory data. Local Differential Privacy (LDP), injecting noise to trajectories for perturbation, has been proposed to protect its privacy. Yet, existing LDP-based mechanisms overlook widely adopted denoising pre-processings of trajectories (*e.g.*, filtering, outlier detection and trajectory similarity measure), rendering noise injection in the whole domain unnecessary and incurring low utility and poor efficiency in practice. In this paper, we observe that various denoising pre-processings all lead to bringing thresholds to perturbation domain, which indicates the noise injected by LDP mechanisms may be useless. This provides an opportunity for enhancing utility of LDP mechanism by eliminating unnecessary noise injection. We propose t-LDP, a novel LDP-based mechanism for trajectory publishing. It integrates threshold into noise injection, eliminating redundant noise that could be denoised by trajectory pre-processing. Additionally, we devise an automaton-based algorithm for efficient perturbation. Experiments on real datasets demonstrate the effectiveness and efficiency of our approach. Especially in extensive perturbation domains, our method shows a 20% improvement of utility and a 600-fold increase in speed compared to existing methods while maintaining robust privacy protection.

Keywords: Local differential privacy · Trajectory publishing · Privacy preservation.

1 Introduction

Trajectory data has largely promoted the development of spatiotemporal applications. Modern spatiotemporal applications tend to devise data-driven approaches for smarter services, such as transportation recommendation [21,22], ride-hailing [29,26,25] and route planning [27]. Despite its significant benefits, gathering and using trajectory data raises serious privacy concerns. The leakage

of trajectory data will incur exposure of sensitive or personal information, risking individual privacy and potentially threatening national security [15].

To overcome this, Differential privacy(DP) has been proposed and become the *de facto* standard for publishing sensitive data with provable guarantees of individual privacy. Local differential privacy (LDP), one important variant of (centralized) DP, provides stronger protection by removing the need for trusted data collectors. These DP/LDP methods achieve indistinguishability among different users' trajectories via injecting noise into the original trajectory for perturbation. Although LDP offers practical privacy solutions for trajectory publishing, its mechanism often suffers lower utility and poor scalability due to its strong privacy requirements and large perturbation domain where the raw input is obfuscated. This issue primarily arises from the exponential growth of the perturbation domain in spatiotemporal data contexts [9,31].

However, existing privacy mechanisms all overlook the fact that denoising pre-processings are widely adopted when utilizing trajectories. Typically, these pre-processings aim to reduce noise or detect outliers, making the noise injection across the entire domain unnecessary. Over-injected noise, often treated as pure noise, is either ignored or smoothed out, contributing nothing to privacy preservation. This observation provides us an opportunity to reduce the noise injection for enhancing data utility. Motivated by this observation, we review several commonly-used denoising pre-processings of trajectory such as filtering, outlier detection, and anti-noise trajectory similarity measures. We observe that these denoising methods introduce thresholds to the perturbation domain, resulting in noise injections beyond these thresholds being considered as pure noise and subsequently disregarded. Based on this observation, we propose a threshold-integrated local differential privacy mechanism (t-LDP), which enhances trajectory utilization by perturbing the trajectory within the threshold, aligning with denoising pre-processings, thus minimizing redundant noise injection.

To better accommodate the mechanism for large-scale applications, we propose an automaton-based data structure to streamline the sampling process. Based on it, we devise an efficient two-step perturbation algorithm to sample the sanitized trajectory, taking the sampling process as the running of the automaton. Initially, it confines perturbations to a symmetric subdomain by determining the perturbation distance through sampling, narrowing the scope of potentially sanitized trajectories. Subsequently, it employs the automaton, which efficiently encodes the sampling states, to select a sanitized trajectory within the subdomain. We further extend this algorithm to include reachability constraints, catering to scenarios requiring more realistic trajectories.

Experiments on real-world datasets validate the effectiveness and scalability of our approach. Our approach produces perturbed trajectory data that have high utility in much less time. It notably boosts data utility by 20% and achieves a 600-fold increase in processing speed within large perturbation domains, all while maintaining the same level of privacy protection as the existing method.

The contributions of this paper can be summarized as below:

- We propose threshold-integrated LDP mechanism (t-LDP), a variant of LDP, which integrates threshold-constraints into perturbations, eliminating redundant noise that could be denoised by trajectory pre-processing;
- We devise an efficient two-step automaton-based algorithm for perturbing spatiotemporal trajectory under the guarantee of t-LDP, accommodating reachability constraints for realistic trajectory publishing, and addressing computational challenges in local settings. Furthermore, we provide theoretical proof of the utility bound for our solution.
- Extensive experiments conducted on real-world datasets demonstrate the effectiveness and scalability of our approach.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 details our observations on LDP application in trajectory data and in Section 4 we elaborate our t-LDP and problem statement. Section 5 elaborates on the proposed automaton-based mechanisms. Experimental results are discussed in Section 6, and the paper concludes in Section 7.

2 Related Work

Centralized DP [12] and its variant LDP [11] have become the *de facto* privacy standards and have been widely used for spatiotemporal data publishing. Due to its assumption-free approach regarding attackers' background knowledge and the provision of stringent privacy guarantees, centralized DP has been widely used in many privacy-preserving spatiotemporal data publishing tasks, such as range counting [13] and density-based query [18]. However, centralized DP relies on trusted data aggregators to collect the raw data from users, which can be unrealistic for real-world scenarios. To avoid this limitation, LDP was proposed, enabling data owners to perturb their data locally. LDP now is widely adopted in spatiotemporal data publishing like distribution queries [7] and frequency estimation [8]. However, most of the LDP mechanisms assume a uniform perturbation probability and privacy-preserving level across the domain, which can lead to low utility, especially for spatiotemporal data.

To better accommodate the property of spatiotemporal data, a number of relaxations of (L)DP have been proposed to allow non-uniform perturbation across the domain. For example, the d_χ-privacy [4], and its location-specific adaption geo-indistinguishability [2], are widely adopted. These mechanisms guarantee the indistinguishability level between any two inputs is proportional to their distance. In trajectory privacy, aggregated domain perturbation with a trusted collector, such as transition probability [6] and length distribution [16], has been popular, aiming to approximate trajectory domain perturbation with lower-dimensional aggregated data. Though attempts made, its innate fitness for aggregation remains the bottleneck for direct data publishing.

Recently, a trend towards direct privacy-preserving trajectory publishing in local settings has emerged. This shift presents significant challenges due to the vastness of the domain and the complexity involved in adding noise effectively with a limited privacy budget. The leading discussion was proposed by Cunningham *et al.* [9]. They propose an n-gram-based methodology leveraging external

knowledge to ensure the trajectory level indistinguishability. Further advancing this field, Zhang *et al.* [31] integrate directional information between points and sample restricted perturbation domain of trajectory within the perturbation process to ensure privacy while preserving utility.

However, these methods suffer bad performances when trajectories cover a wider range. This issue stems from their outputs being excessively perturbed with noise. Although intended for privacy preservation, such excessive noise has minimal impact on actual privacy protection, as denoising pre-processings like outlier detection tend to ignore these perturbations. In this paper, we introduce threshold to LDP mechanism to eliminate the drawbacks of existing methods.

3 Motivation

In this section, we introduce our motivation for our new LDP mechanism. We observe that commonly-used trajectory pre-processings will eliminate excessive noise, which can be equivalently viewed as bringing a threshold in perturbation domain. Below we reviewed three kinds of commonly used denoising pre-processings: filtering, outlier detection, and anti-noise trajectory similarity measure, and discuss how these pre-processings will influence the perturbation of LDP.

Filtering. Filtering (*e.g.*, Kalman Filter [1] and Particle Filter [17]) is a critical process used to smooth trajectory data. it helps maintain trajectory's overall structure and refine the trajectory by removing random fluctuations or anomalies introduced by noise. Once the points are perturbed beyond a certain threshold, filtering techniques would place significantly less belief on the points, leading to their smoothing out and consequently failing to adequately reveal the original trajectory information.

Outlier Detection. Outlier detection aims to identify points that substantially deviate from an expected route or pattern. Prevalent methods adopt either distance-based [19,20] or density-based manners [3] to identify whether a point is an outlier. In other words, only when a point's distance to others or local density within a given threshold will it contribute to the final trajectory utility. Perturbing points beyond these thresholds is unnecessary, similar to filtering.

Anti-noise Trajectory Similarity Measure. Trajectory similarity measures are normally adopted to measure the spatiotemporal similarity of trajectories. Techniques like Longest Common Subsequence (LCSS) [28] and Edit Distance on Real Sequences (EDR) [5] are particularly effective in this regard, as they are designed to be noise-robust. Once again, their noise-robust property is achieved by incorporating distance threshold when measuring similarity. Therefore, the excessive perturbed points will be regarded as noise.

These various pre-processings applied to different aspects of trajectory data share a common underlying principle: the use of a threshold to determine whether a point sufficiently reveals the spatiotemporal characteristics of the original trajectory. This approach effectively renders any excessive perturbation as mere noise, contributing nothing to the data utility. Based on this observation, we devise a threshold-integrated LDP mechanism by limiting the noise injections, as we discuss in the next section.

4 Problem Statement

In this section, we introduce our threshold-integrated LDP privacy mechanism and problem statement.

For a given spatiotemporal domain, we use $\mathcal{S}_{st}$ to denote the spatiotemporal points in this domain. Hence, we define a trajectory in this domain, τ, as a sequence of spatiotemporal points that $\tau = \{s_1..., s_{|\tau|}\} = \{(p_1, t_1)..., (p_{|\tau|}, t_{|\tau|})\}$, where $|\tau|$ denotes the length of the trajectory and each $s_i \in \mathcal{S}_{st}$.

We define the spatiotemporal correlated δ-trajectory set of τ, denoted as $\Delta\mathcal{T}_\tau(\delta)$, and formulate the concept of local differential privacy within the δ-trajectory set. Our central premise is to ensure that the sanitized trajectory denoted as z does not facilitate an adversary in obtaining an undue amount of sensitive information from the δ-trajectory set.

Essentially, the δ-trajectory set represents a group of trajectories spatiotemporally related to the original one, effectively obfuscating it.

Definition 1 (δ-Trajectory Set). *Let $\tau = \{s_1, s_2..., ..., s_n\}$ be the trajectory subject to obfuscation. δ-trajectory set, denoted by $\Delta\mathcal{T}_\tau(\delta)$, is a set containing all trajectories $\tau' = \{s'_1, s'_2, ..., ..., s'_n\}$ of length n, where each point s'_i is at most distance δ from the corresponding point s_i in τ.*

$$\Delta\mathcal{T}_\tau(\delta) = \{\tau' | \forall i \in \{1, ..., n\}, d(s_i, s'_i) \leq \delta\}$$

When δ is set to domain's diameter, the δ-trajectory set includes trajectories spanning the full spatiotemporal domain, aligning with the standard LDP model.

Definition 2 (Threshold-integrated Local Differential Privacy, t-LDP). *Given any trajectory τ, threshold δ, a randomized mechanism, $\mathcal{A}$, adheres to ϵ threshold-integrated local differential privacy on δ-trajectory set $\Delta\mathcal{T}_\tau(\delta)$, if for any output trajectory z that belongs to $\Delta\mathcal{T}_\tau(d_2)$, and any pair of trajectories τ_1 and τ_2 within $\Delta\mathcal{T}_\tau(d_1)$ such that $d_1 + d_2 = \delta$, the output from mechanism $\mathcal{A}$ satisfies the inequality*

$$\frac{\Pr[\mathcal{A}(\tau_1) = z]}{\Pr[\mathcal{A}(\tau_2) = z]} \leq \exp(\epsilon) \tag{1}$$

This definition employs a graded privacy protection approach based on the proximity of output trajectories to the input. Outputs closer to the input trajectory are afforded stronger privacy safeguards, with a variety of trajectories in $\Delta\mathcal{T}_\tau(\delta)$ producing z. The output probability similarity remains within $\exp(\epsilon)$ difference. Conversely, maintaining the same privacy level for more distant outputs may not be advantageous, as it could reduce data utility without significantly improving privacy. This method aims for a balanced privacy-utility trade-off, considering the spatiotemporal nature of trajectory data.

According to [24], threshold-integrated local differential privacy is able to protect privacy by limiting the information gained in δ-trajectory set for adversaries with specific prior knowledge.

Theorem 1 (Adversarial Privacy[24]). *For trajectory $s \in \Delta\mathcal{T}_\tau(d_1)$, output $z \in \Delta\mathcal{T}_\tau(d_2)$ such that $d_1 + d_2 = \delta$ and adversaries knowing the true trajectory is $s \in \Delta\mathcal{T}_\tau(d_1)$, it holds that:* $\Pr(x = s|z)/\Pr(x = s) \leq \exp(\epsilon)$, *where $Pr(*)$ and $Pr(*|z)$ are the prior and posterior probabilities of adversaries.*

Problem Statement. A trajectory is defined as a sequence of spatiotemporal points. Each individual has a sensitive trajectory $\tau = \{s_1..., s_{|\tau|}\}$. To preserve privacy and minimize redundant noise, each user locally perturbs their trajectory into $\hat{\tau} = \{\hat{s}_1..., \hat{s}_{|\tau|}\}$ within a δ-trajectory set, conforming to t-LDP. This perturbation ensures that adversaries cannot accurately deduce any specific point in the trajectory. After receiving these perturbed trajectories, various trajectory data analyses, like range queries, can be conducted. The objective is to retain maximal trajectory information, ensure an efficient and scalable perturbation process, and comply with the t-LDP privacy standard.

5 Our Trajectory Publishing Algorithm

The new privacy mechanism necessitates new LDP perturbation algorithms. This section introduces the exponential mechanism for t-LDP (Sec. 5.1), then details an efficient version of the perturbation mechanism based on automaton (Sec. 5.2), and extends it for producing more realistic trajectories (Sec. 5.3).

5.1 Exponential Mechanism for t-LDP

We adapt the exponential mechanism [14], a key perturbation method for LDP, to underpin our privacy mechanism. The exponential mechanism's output depends on a utility function $u(\cdot, \cdot)$ and sensitivity Δu. The utility function evaluates the spatiotemporal correlation between perturbed and the original trajectory, while sensitivity confines the variability in utility scores.

Mechanism 1. ***Exponential Mechanism for t-LDP.*** *Given the spatiotemporal domain $\mathcal{R}_{st}$, for an n-length trajectory $\tau \in \mathcal{R}_{st}^n$, the exponential mechanism $\mathcal{A}_e$ outputs $\tau' \in \Delta\mathcal{T}_\tau(\delta)$ with probability*

$$\Pr(\tau') = \frac{\exp(\frac{\epsilon u(\tau', \tau)}{\Delta u})}{\sum_{\tau'' \in \Delta\mathcal{T}_\tau(\delta)} \exp(\frac{\epsilon u(\tau'', \tau)}{\Delta u})} \tag{2}$$

Lemma 1. *For any input trajectory τ, mechanism 1 satisfies ϵ t-LDP with respect to δ-trajectory set of τ.*

Proof. Let $\mathcal{A}_e$ denotes exponential mechanism for t-LDP on δ-trajectory set, τ be any trajectory to be perturbed, $\tau_1, \tau_2 \in \Delta\mathcal{T}_\tau(d_1)$ and $z \in \Delta\mathcal{T}_\tau(d_2)$ that $d_1 + d_2 = \delta$. We get

$$\begin{aligned} \frac{\Pr[\mathcal{A}_e(\tau_1) = z]}{\Pr[\mathcal{A}_e(\tau_2) = z]} &= \frac{\exp(\frac{\epsilon u(z, \tau_1)}{\Delta u})}{\sum_{\tau \in \Delta\mathcal{T}_{\tau_1}(\delta)} \exp(\frac{\epsilon u(\tau, \tau_1)}{\Delta u})} \Big/ \frac{\exp(\frac{\epsilon u(z, \tau_2)}{\Delta u})}{\sum_{\tau \in \Delta\mathcal{T}_{\tau_2}(\delta)} \exp(\frac{\epsilon u(\tau, \tau_2)}{\Delta u})} \\ &= \exp(\epsilon \frac{u(z, \tau_1) - u(z, \tau_2)}{\Delta u}) \leq \exp(\epsilon) \end{aligned} \tag{3}$$

□

The second step derives from the fact that each trajectory in $\Delta\mathcal{T}_{\tau_1}(\delta)$ corresponds uniquely to a trajectory in $\Delta\mathcal{T}_{\tau_2}(\delta)$ with an identical relative location to τ_2, thus sharing the utility, and vice versa. Given that these paired trajectories possess equivalent probabilities, their aggregated probabilities remain consistent.

In this paper, we use Manhattan distance as the distance measure, suitable for grid-based city layouts, *i.e.*, for two trajectories $\tau_1 = \{s_1, ..., s_n\}, \tau_2 = \{s'_1, ..., s'_n\}$, the distance is $d(\tau_1, \tau_2) = \sum_{i=1}^{n} d(r_i, r'_i)$, where r_i, r'_i correspond to the region where s_i, s'_i in. The **utility function** u measuring the utility of the private output $\tau' \in \mathcal{R}_{st}^n$ and the **sensitivity** Δu of utility function on δ-trajectory set are: $u(\tau', \tau) = -d(\tau', \tau), \Delta u = \max\limits_{\tau \in \mathcal{R}_{st}^n} \max\limits_{\tau_1, \tau_2 \in \Delta\mathcal{T}_\tau(\delta)} |u(\tau_1, \tau) - u(\tau_2, \tau)| \leq n\delta$

However, Direct implementation of the exponential mechanism on $\Delta\mathcal{T}_\tau(\delta)$ is impractical due to the computation cost reaching $O(|\tau|\delta^{3|\tau|})$. Thus, an efficient mechanism is required for practical implementation.

5.2 Efficient Automaton-based Algorithm

To efficiently perturb the trajectory while conforming to t-LDP, we propose an automaton-based data structure and introduce a two-step automaton-based algorithm, using the automaton to streamline the perturbation process.

Automaton-based Data Structure. To develop an efficient mechanism, we propose the automaton-based data structure, called *Manhattan Distance Nondeterministic Finite State Automaton*(MHNFA), which identifies all the trajectories that have a specific Manhattan distance to an input trajectory and encodes the random states to sample one of them.

Definition 3 (Manhattan Distance NFA). *For spatiotemporal domain $\mathcal{R}_{st}$, a length $n \in \mathbb{N}^+$, a trajectory $\tau \in \mathcal{R}_{st}^n$, a perturbation region threshold δ and a distance $\ell \in \mathbb{N}$, the Manhattan Distance NFA is an automaton $\mathcal{M}_{\tau,\ell} = (Q_{\tau,\ell}, \Delta\mathcal{R}_\tau(\delta), tr, q^0, F_{n,\ell})$ where: (1)$Q_{\tau,\ell}$ is the set of states. (2)$\Delta\mathcal{R}_\tau(\delta)$ is the input alphabet representing the regions within δ distance from the regions occupied by τ. (3)$tr : Q_{\tau,\ell} \times \Delta\mathcal{R}_\tau(\delta) \to Q_{\tau,\ell}$ is the transition function between states. (4)$q^0 \in Q_{\tau,\ell}$ is the initial state. (5)$F_{n,\ell}$ is the set of accepting states.*

Besides, we use policy $\mu(\cdot, \cdot|q_i)$: $Q_{\tau,\ell} \times \Delta\mathcal{R}_\tau(\delta) \to [0, 1]$ denoting the probability function of transferring between states. It holds that $\sum_{q_{i+1}, \sigma_{i+1}} \mu(q_{i+1}, \sigma_{i+1}|q_i) = 1$, where $\mu(q_{i+1}, \sigma_{i+1}|q_i)$ represents the probability of transitioning from state q_i to state q_{i+1} given input symbol σ_{i+1}.

Based on the above data structure, we propose a two-step framework, and the overall procedure is outlined in Algorithm 2:

Main Idea of our Automaton-based algorithm (Mechanism 2). *(1)Distance Sampling*: We sample a distance ℓ for input trajectory τ to confine the perturbation domain, narrowing the scope of potential trajectories. *(2)Trajectory Selection*: By leveraging an automaton that effectively encodes the sampling states, we swiftly sample a sanitized trajectory within this limited subdomain.

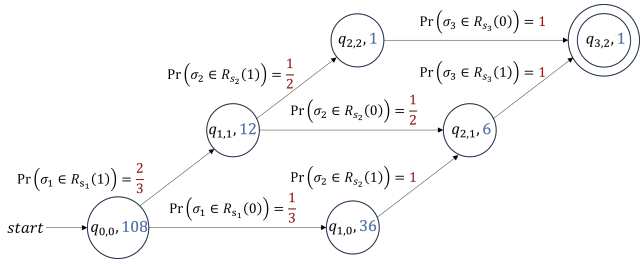


Fig. 1. Illustration of Algorithm 1 via $\mathcal{M}_{\tau,2}$, sampling trajectories 2 units from τ. States $q \in Q_{\tau,\ell}$ are circles, transitions are arrows, and the double-circled state is the accepting state. The value $V(q_i)$ at any state q_i is in blue, and transition probability $\mu(q_{i+1}, \sigma | q_i)$ is in red. Sampling begins at $q_{0,0}$, leading to the perturbed trajectory $\{\sigma_1, \sigma_2, \sigma_3\}$ at the accepting state.

Distance Sampling. According to mechanism 1, all the trajectories that share the same distance from the input n-length trajectory τ have the same probability of being chosen. Thus, the chance to select the subdomain with distance ℓ to τ is

$$\Pr(\ell; \tau, \delta) = \frac{count(\ell) \exp(-\frac{\epsilon \ell}{n\delta})}{\sum_{i=0}^{n\delta} count(i) \exp(-\frac{\epsilon i}{n\delta})} \tag{4}$$

where $count(\ell)$ is the number of trajectories that are ℓ away from τ.

Generating function, a key combinatorial tool, is used to calculate aggregate $count(*)$. The generating function $G(x)$ denotes the composition of possible regions which are represented by $g(x)$ for each step,and $c(i)$ stands for the number of regions with distance i from one region:

$$G(x) = g(x)^n = (c(0)x^0 + ... + c(\delta)x^\delta)^n = (1 + \sum_{\delta \geq d \geq 1} (4d^2 + 2)x^d)^n \tag{5}$$

where the coefficient of x^i is the number of trajectories at distance i away from τ.

Trajectory Selection. The MHNFA efficiently encodes the sampling states and samples evenly in the trajectory subdomain with the determined distance selected in the previous step by taking the symbol list between the transition of states. States $q_{i,j} \in Q_{\tau,\ell}$ represent the scenario where i regions of the output trajectory, with a cumulative distance j from τ, are determined. Each state $q_{i,j}$ can transition to $q_{i+1,j+k}$, with k being the distance to the corresponding region in τ and represented by symbol $\sigma_{i+1,k} \in \Delta\mathcal{R}_{st}$. The transition function is $tr(q_{i,j}, \sigma_{i+1,k}) = q_{i+1,j+k}$. The automaton starts at $q^0 = q_{0,0}$ and accepts at $q_{n,\ell}$.

Given a distance ℓ and a trajectory τ, we construct an MHNFA $\mathcal{M}_{\tau,\ell}$ by Algorithm 1. It takes inputs τ and $\{q_{n,\ell}\}$, maintaining a function $V : Q_{\tau,\ell} \to \mathbb{N}$ for the count of unique paths to $q_{n,\ell}$, and outputs the transition policy μ. The probability of selecting a region σ at state q_{i,d_i} is equal to $V(q_{i+1,d_{i+1}}) * c(d_{i+1} - d_i)/V(q_{i,d_i})$. The algorithm iterates backward from the end state to q^0 for

Algorithm 1: Manhattan Distance NFA Construction

Input: n-length input trajectory $\tau = \{r_1, r_2, ..., r_n\}$, accepting set $\{q_{n,\ell}\}$
Output: policy μ

$n, i \leftarrow |\tau|, 1$
$V(q_{n,\ell}) \leftarrow 1$
$CurrQueue, ActiveQueue \leftarrow \{q_{n,\ell}\}, \{\}$
while $i \leq n$ **do**
  // q'', q', q correspond to $q_{i+1,d''_{i+1}}, q_{i+1,d'_{i+1}}, q_{i,d_i}$
  for $q' \in CurrQueue$ **do**
    for $(q, \sigma)\ s.t.\ q' \in tr(q, \sigma)$ **do**
      $V(q) \leftarrow \sum_{\{q'' | \exists \sigma', q'' \in tr(q,\sigma')\}} V(q'') * c(d''_{i+1} - d_i)$
      $\mu(q', \sigma | q) \leftarrow \frac{V(q') * c(d'_{i+1} - d_i)}{V(q)}$
      $ActiveQueue \leftarrow ActiveQueue + \{q\}$
  $CurrQueue \leftarrow ActiveQueue$
  $ActiveQueue \leftarrow \{\}$
  $i \leftarrow i + 1$
return μ

Algorithm 2: Automaton-based Perturbation Algorithm

Input: n-length input trajectory $\tau = \{r_1, r_2, ..., r_n\}$, threshold δ
Output: n-length output trajectory $\hat{\tau}$

$\{count(0), count(1), ..., count(n\delta)\} \leftarrow$ Coefficients of $G(x)$ from Eq.(5)
$\ell \leftarrow$ Sample distance from distribution in Eq.(4)
$\mu \leftarrow$ Manhattan Distance NFA Construction with parameters τ,$q_{n,\ell}$
$\{\sigma_1, ..., \sigma_n\} \leftarrow$ Sample the trajectory in the NFA with policy μ
return $\{\sigma_1, ..., \sigma_n\}$

constructing transition policy μ, and progresses forward to select one trajectory based on μ. Fig. 1 shows an example MHNFA.

After the region-level trajectory is sampled, the point-level trajectory is constructed with uniform sampling in the corresponding regions.

Privacy Analysis. Here, we prove that the algorithm provides the same level of t-LDP to input trajectory τ as mechanism 1.

Lemma 2. *Given threshold δ, privacy budget ϵ, and n-length input trajectory $\tau \in \mathcal{R}^n_{st}$, mechanism 2 provides ϵ t-LDP to τ w.r.t δ-trajectory set.*

Proof. For Mechanism 2 $\mathcal{A}_{ea}$, the probability of sampling trajectory z is equal to the probability in Mechanism 1.

$$\Pr[\mathcal{A}_{ea}(\tau) = z] = \frac{count(d(z,\tau)) \exp(-\frac{\epsilon d(z,\tau)}{n\delta})}{count(d(z,\tau)) \sum_{i=0}^{n\delta} count(i) \exp(-\frac{\epsilon i}{n\delta})} = \Pr[\mathcal{A}_e(\tau) = z] \quad (6)$$

Time Complexity Analysis. Using mechanism 2 and a distance selected as per Eq.(4), the time complexity for generating a private output within threshold δ is $O(|\tau|\delta)$. This is a marked improvement over the direct implementation of the exponential mechanism 1, thanks to two main factors: (1) initial distance sampling that reduces the sampling domain, and (2) domain symmetry utilization that minimizes automaton states, speeding up the generation process.

Utility Analysis. Analyzing the accuracy of perturbed trajectories is crucial for calibrating privacy, aiming to balance privacy strength against induced errors. We compute the expectation and variance of the distance between input and output trajectories as functions of ϵ and δ. Letting $x = e^{\frac{-\epsilon}{|\tau|\delta}}$, the expectation and variance are defined as:

$$E(\ell) = \frac{\sum_{|\tau|\delta \geq d \geq 0} d \cdot count(d) x^d}{\sum_{|\tau|\delta \geq d \geq 0} count(d) x^d} = \frac{|\tau| \sum_{|\delta \geq d \geq 1} (4d^2 + 2) d x^d}{1 + \sum_{\delta \geq d \geq 1} (4d^2 + 2) x^d} \tag{7}$$

$$var(\ell) = \frac{\sum_{|\tau|\delta \geq d \geq 0} d^2 \cdot count(d) x^d}{\sum_{|\tau|\delta \geq d \geq 0} count(d) x^d} - E(\ell)^2 \tag{8}$$

In the limit as $\epsilon \to \infty$ and $x \to 0$, both $E(\ell)$ and $Var(\ell)$ approaches zero, indicating that when privacy diminishes, the mechanism outputs the input trajectory unchanged.

5.3 Efficient Automaton-based Algorithm with Reachability Constraint

Imposing reachability constraints is essential for ensuring the realism of perturbed trajectories adhering to spatiotemporal constraints, thereby enhancing practical utility.

Definition 4 (Reachability). *For a threshold θ, representing maximum travel distance over time, and spatial distance $d_s(\cdot, \cdot)$, a region r_a is reachable from r_b at time t if $d_s(r_a, r_b) \leq \theta(t)$.*

For a trajectory $\tau = \{s_1, s_2, ..., s_n\}$ in the spatiotemporal domain, where $s_i = (p_i, t_i)$, reachability demands that each p_{i+1} be reachable from p_i at t_i for $1 \leq i < n$. The trajectories in δ-trajectory set conforming to reachability constraint constitute $\Delta\mathcal{T}_{\tau,\theta}(\delta)$.

Implementing reachability across complete trajectories, as described in [9], entails traversing an exponentially growing trajectory domain. This complexity can be mitigated by limiting traversal to an n-gram domain, thus partitioning the problem to lower time complexity. Nevertheless, such an approximation might not accurately capture the actual reachability-constrained domain and can suffer from scalability issues owing to its sampling and post-processing methods.

Automaton-based Data Structure with Reachability Constraint. To accommodate our algorithm to the scenario where reachability constraint is required, we introduce our revised automaton-based data structure, *Reachability Constrained MHNFA (RC-MHNFA)*, designed for scalable sampling within trajectory domain $\Delta\mathcal{T}_{\tau,\theta}(\delta)$ under reachability constraints θ.

Definition 5 (Reachability Constrained MHNFA). *Consider a spatiotemporal domain $\mathcal{R}_{st}$. The reachability constraint is defined by a matrix P_θ, where $P_{i,j} = \mathbb{P}[r_j|r_i] = 1$ if region r_j is reachable from region r_i, and $P_{i,j} = 0$ otherwise.*

Let $\mathcal{M}_{\tau,\ell} = (Q_{\tau,\ell}, \Delta\mathcal{R}_\tau(\delta), tr, q^0, F_{n,\ell})$ be a MHNFA. The Reachability Constrained MHNFA, denoted $\mathcal{M}_{\tau,\ell,\theta}$, is defined as $\mathcal{M}_{\tau,\ell,\theta} = (Q_{\tau,\ell,\mathcal{R}_{st}}, \Delta\mathcal{R}_\tau(\delta), tr_\theta, q^0_{\mathcal{R}_{st}}, F_{n,\ell,\theta})$, where:

$$Q_{\tau,\ell,\mathcal{R}_{st}} = Q_{\tau,\ell} \times \mathcal{R}_{st}, \quad tr_\theta : Q_{\tau,\ell} \times \mathcal{R}_{st} \times \Delta\mathcal{R}_\tau(\delta) \to Q_{\tau,\ell,\mathcal{R}_{st}},$$

$$q^0_{\mathcal{R}_{st}} = (q^0, r_0), \quad F_{n,\ell,\theta} = \{(q_f, r) \in Q_{\tau,\ell,\mathcal{R}_{st}} | q_f \in F_{n,\ell}, r \in \mathcal{R}_{st}\}$$

For any $(q', r') \in tr_\theta(q, r, \sigma)$, we have $tr(q, \sigma) = q'$ and $\mathbb{P}[r'|r] = 1$. A state $(q, r) \in Q_{\tau,\ell,\mathcal{R}_{st}}$ can transfer to another state by a policy $\mu_\theta(\cdot, \cdot|q, r) : Q_{\tau,\ell} \times \Delta\mathcal{R}_\tau(\delta) \to [0, 1]$.

The $\mathcal{M}_{\tau,\ell,\theta}$ is the synchronous product of $\mathcal{M}_{\tau,\ell}$ and the reachability constraint θ. The transition function tr_θ of $\mathcal{M}_{\tau,\ell,\theta}$ has to satisfy both the transition function tr and be a feasible transition under the reachability constraint. Thus the output of the automaton is one of the trajectories in trajectory set $\Delta\mathcal{T}_{\tau,\theta}(\delta)$.

Main Idea of Automaton-based Algorithm with Reachability Constraint (Mechanism 3). *(1)Dynamic Programming based-Distance Sampling*: We sample a distance ℓ for input τ with dynamic programming, restricting the range of possible trajectories within $\Delta\mathcal{T}_{\tau,\theta}(\delta)$; *(2)Trajectory Selection*: By leveraging an automaton that effectively encodes the sampling states under reachability constraint, we swiftly sample a sanitized trajectory within the limited subdomain.

Dynamic Programming based-Distance Sampling. In $\Delta\mathcal{T}_{\tau,\theta}$, trajectories equidistant from input τ are equally likely (Mechanism 1). The probability of choosing distance ℓ is:

$$\Pr_\theta(\ell; \tau, \delta) = \frac{count_\theta(\ell) \exp(-\frac{\epsilon\ell}{n\delta})}{\sum_{i=0}^{n\delta} count_\theta(i) \exp(-\frac{\epsilon i}{n\delta})} \tag{9}$$

where $count_\theta(\ell)$ counts trajectories ℓ away from τ in $\Delta\mathcal{T}_{\tau,\theta}(\delta)$.

However, calculating $count_\theta(\cdot)$for the entire domain $\Delta\mathcal{T}_{\tau,\theta}(\delta)$ proves to be less straightforward than in the trajectory domain $\Delta\mathcal{T}_\tau(\delta)$ used in Mechanism 2, due to the imposition of a reachability constraint. The symmetry of the trajectory domain and independence of perturbing each point is no longer maintained, thus an efficient method to calculate $count_\theta(\cdot)$ would be needed.

Here, we propose a method based on dynamic programming to efficiently calculate the sampling distance under the reachability constraint for an n-length trajectory represented as discrete regions $\tau = \{r_{\tau,1}, ..., r_{\tau,n}\} = \{(r^s_{\tau,1}, t_{\tau,1})...(r^s_{\tau,n}, t_{\tau,n})\}$.

We define $count_{i,r} = count_{i,(r^s,t)}(\ell)$ as the number of trajectories that start at the states $(q_{i,\ell}, r) = (q_{i,\ell}, (r^s, t))$ and end at accepting states. Due to the fact that $count_{i,r}(\ell)$ can be calculated by all its succeeding reachable states, for each succeeding state r' in time region t', we can formalize its transition function as

$$count_{i,r}(\ell) = \sum_{t'} \sum_{\mathbb{P}[r'|r]=1} count_{i+1,r'}(\ell - d(r, r')) \tag{10}$$

Having calculated the $count_{1,*}$, the $count_\theta$ for the whole trajectory domain under reachability constraint θ is the sum of the $count_{1,*}$ for each corresponding distance. The output trajectory distance from the input trajectory is selected according to Eq.(9).

Trajectory Selection. The construction process of the automaton and trajectory selection in Mechanism 2 parallels that of Mechanism 3. The transition function, considering reachability tr_θ, retains its previous form but integrates a reachability constraint, that is, $tr_\theta(q_{i,j}, r, \sigma) = (q_{i+1,j+k}, \sigma)\ , \forall\ \mathbb{P}[\sigma|r] = 1$. Moreover, the accepting states combine the accepting state in Mechanism 2 with the perturbed regions of r_n, expressed as $\{(q_{n,\ell}, r)|\forall r \in \Delta\mathcal{R}_{r_n}(\delta)\}$. The construction iterates backward from these accepting states to the initial state. Throughout this process, the function V is updated as $V_\theta(q, r) = \sum_{(q'',r''|tr_\theta(q,r,r'')=(q'',r''))} V_\theta(q'', r'')$, effectively tracking the count of unique paths to each state. The transition policy $\mu_\theta(q', r'|q, r)$ is then determined as the ratio $\frac{V_\theta(q',r')}{V_\theta(q,r)}$, guiding the selection of the next state and region in the trajectory.

6 Experimental Evaluation

6.1 Experimental Setup

Experimental Environment. The experiments were conducted on a server equipped with an Intel Platinum 8361HC CPU and 32GB of RAM. The implementation was carried out using Python 3.8.

Datasets. Our experiments employ two real-world datasets: Porto [23] and Tdrive [30]. Porto dataset[1] comprises 53,442 taxi trajectories collected over 8 months in a 20 km square area of Porto, each lasting 30 minutes and sampled every 5 minutes, typically featuring shorter, faster movements. In contrast, Tdrive[2] includes 10,878 longer taxi trajectories over a larger area, each with a duration of 60 minutes and a 10-minute sampling interval.

Baseline. NGRAM [9], noted for its excellence in LDP trajectory publishing, is well-suited for our spatiotemporal study needs and serves as our chosen baseline, as also seen in [10,31]. Unlike other methods that perturb aggregated properties and neglect the temporal aspect, thus struggling with direct trajectory-level privacy, NGRAM effectively addresses both spatial and temporal dimensions by perturbing each n-gram separately to form the overall trajectory. Here, we adapt NGRAM to our LDP mechanism with the same reachability constraint as t-LDP-RC to ensure the same level of privacy protection.

Due to the prohibitive computational complexity of the direct exponential mechanism (refer to Mechanism 1), which is exponential and thus unfeasible for experimentation, we opt to compare **t-LDP** (refer to Mechanism 2) and **t-LDP-RC** (refer to Mechanism 3) alongside **NGRAM** in our analysis.

[1] https://www.kaggle.com/datasets/crailtap/taxi-trajectory
[2] https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

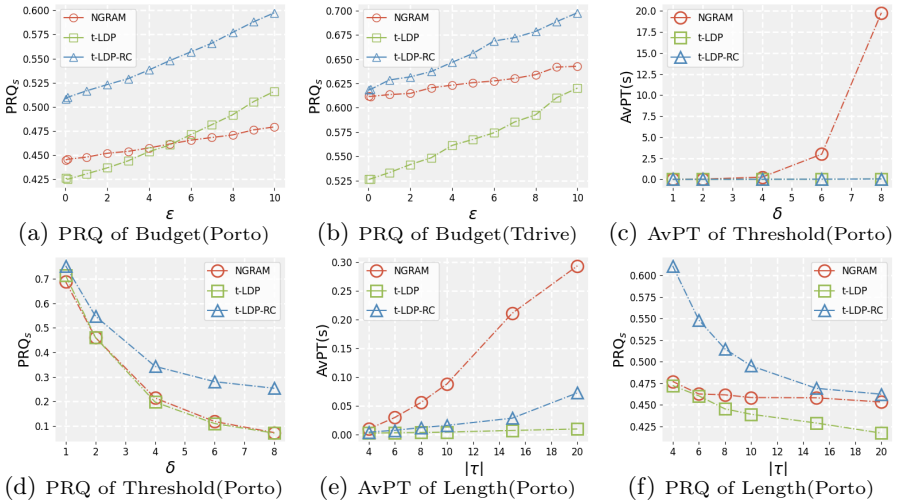
Parameter Settings. In our experiments, we use the following default parameters, with a more comprehensive analysis of parameter effects in 6.2. The reachability constraint is set to 10 m/s, representing taxi speed. As per [5,28], we set the spatial threshold to a quarter of the maximum standard deviation of trajectories and the temporal threshold to 20% of the trajectory's total time. spatiotemporal regions are formed by partitioning the spatial domain into $g_s \times g_s$ grids (500m side lengths for both Porto and Tdrive) and dividing the temporal domain into g_t grids (3min for Porto, 2.5 min for Tdrive).

Utility Measures. Utility is evaluated using two metrics from prior studies [9,31]: Mean Normalized Error (NE) and Preservation Range Query (PRQ). NE measures the normalized distance between points in the output and original trajectories: $NE = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\frac{1}{|\tau_i|}\sum_{j=1}^{|\tau_i|} d_\chi(\hat{s}_j, s_j)$ where $|\mathcal{T}|$ is the number of trajectories, and $d_\chi(\cdot,\cdot)$ is the distance in dimension χ, normalized by the threshold distance. PRQ assesses the proximity of points in the perturbed trajectory to those in the original within range ρ_χ, with $\pi_\chi(\hat{s}_j, s_j, \rho_\chi)$ indicating whether the distance is within ρ_χ: $PRQ_\chi = \frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\frac{1}{|\tau_i|}\sum_{j=1}^{|\tau_i|}\pi_\chi(\hat{s}_j, s_j, \rho_\chi)$.

6.2 Experimental Results

Table 1. Comparison of utility performance. The subscript indices x, y, t, and s represent the x-dimension, y-dimension, time dimension, and the combined spatial dimension (x and y), respectively. The optimal result for each utility is highlighted in **bold**. For both NE and AvPT, lower values indicate better performance, whereas for PRQ, a higher value is preferable.

		Porto				**Tdrive**			
		$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 0.1$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$
$\boldsymbol{NE_x}$	NGRAM	0.2387	0.2375	0.2334	0.2289	0.2237	0.2233	0.2192	0.2157
	t-LDP	0.2498	0.2475	0.2376	0.2200	0.2422	0.2399	0.2315	0.2174
	t-LDP-RC	**0.2211**	**0.2193**	**0.2108**	**0.1978**	**0.2097**	**0.2083**	**0.2011**	**0.1890**
$\boldsymbol{NE_y}$	NGRAM	0.2383	0.2376	0.2348	0.2291	0.2244	0.2241	0.2206	0.2145
	t-LDP	0.2495	0.2483	0.2373	0.2195	0.2438	0.2406	0.2300	0.2159
	t-LDP-RC	**0.2207**	**0.2192**	**0.2110**	**0.1970**	**0.2103**	**0.2086**	**0.2003**	**0.1889**
$\boldsymbol{NE_t}$	NGRAM	0.2125	0.2113	0.2092	0.2056	0.2296	0.2281	0.2256	0.2231
	t-LDP	0.2175	0.2161	0.2083	0.1972	0.2410	0.2415	0.2345	0.2228
	t-LDP-RC	**0.1840**	**0.1826**	**0.1761**	**0.1653**	**0.2181**	**0.2159**	**0.2097**	**0.2006**
$\boldsymbol{PRQ_s}$	NGRAM	0.4440	0.4472	0.4593	0.4779	0.6116	0.6136	0.6259	0.6428
	t-LDP	0.3986	0.4046	0.4441	0.5116	0.5256	0.5349	0.5673	0.6165
	t-LDP-RC	**0.5108**	**0.5156**	**0.5477**	**0.5990**	**0.6214**	**0.6255**	**0.6527**	**0.6979**
$\boldsymbol{PRQ_t}$	NGRAM	0.7808	0.7836	0.7877	0.7957	0.8378	0.8420	0.8518	0.8568
	t-LDP	0.7710	0.7749	0.7901	0.8137	0.8183	0.8170	0.8287	0.8460
	t-LDP-RC	**0.8371**	**0.8405**	**0.8524**	**0.8728**	**0.8380**	**0.8427**	**0.8544**	**0.8727**
$\boldsymbol{AvPT}$	NGRAM	0.0155	0.0157	0.0090	0.0090	0.1528	0.1538	0.1545	0.1208
	t-LDP	**0.0021**	**0.0022**	**0.0018**	**0.0010**	**0.0017**	**0.0017**	**0.0017**	**0.0014**
	t-LDP-RC	0.0031	0.0032	0.0026	0.0015	0.0256	0.0256	0.0248	0.0220



(a) PRQ of Budget(Porto) (b) PRQ of Budget(Tdrive) (c) AvPT of Threshold(Porto)

(d) PRQ of Threshold(Porto) (e) AvPT of Length(Porto) (f) PRQ of Length(Porto)

Fig. 2. Preservation range query (PRQ) an Average perturbation time (AvPT) under different privacy budget, threshold and trajectory length.

Evaluation of Utility and Efficiency. We set ρ_χ as half of the threshold in the corresponding dimension. We assess the efficacy of our proposed mechanisms, t-LDP and t-LDP-RC, alongside NGRAM across three metrics: Mean Normalized Error (NE), Preservation Range Query (PRQ), and Average Perturbation Time in seconds (AvPT) with various privacy budget ϵ, as shown in Table 1.

In utility comparison, t-LDP-RC consistently surpasses other mechanisms on all datasets, showcasing its robust utility preservation. This superior performance stems from t-LDP-RC sampling trajectories directly within the full reachability-constrained domain, unlike NGRAM, which samples within segmented n-gram domains under reachability constraint, and t-LDP, which does not apply a reachability constraint. This approach not only conserves the privacy budget by minimizing the need for repeated sampling but also solve the key issue in NGRAM's method, where sampling individual n-grams under reachability constraints may not accurately reflect the constraints of the full trajectory.

Effects of Parameters. Here, we examine how utility and efficiency is influenced by single mechanism or trajectory parameter with other parameters set to default.

Effects of Privacy Budget. The accuracy trends of preservation range queries with increasing privacy budget ϵ are captured in Fig. 2(a) and Fig. 2(b). Both t-LDP and t-LDP-RC display a notable improvement, with t-LDP-RC maintaining superior performance. The enhancement in query accuracy is notable, achieving up to **10% and 12% improvements** over t-LDP and NGRAM in Porto, and **7% and 5%** in Tdrive, respectively. In contrast, the utility growth in NGRAM is slower due to the per n-gram privacy budget distribution.

Effects of Threshold. Fig. 2(c) and Fig. 2(d) demonstrate the effects of an increasing threshold δ. A rise in the threshold corresponds to a broader

perturbation domain and consequently introduces greater noise, leading to lower perturbation efficiency. Automaton-based mechanisms, however, prove to be more effective and scalable under these conditions. Both t-LDP and t-LDP-RC strike a delicate balance between efficiency and utility, especially t-LDP-RC, which achieves significantly **20% better utility preservation** despite a marginal increase in perturbation time when δ is high. On the other hand, NGRAM, despite maintaining comparable utility to t-LDP, suffers from a perturbation process that is over **600 times longer**, limiting its practicality for extensive perturbation domains.

Effects of Trajectory Length. The variation in error and perturbation time with increasing trajectory lengths $|\tau|$ is shown in Fig. 2(e) and Fig. 2(f). t-LDP-RC's utility begins to align with NGRAM's as trajectory length grows. The convergence arises for the impact of reachability constraints diminishes over longer trajectories, causing the per-point privacy budget in t-LDP-RC to approach that of NGRAM, thus yielding similar utility profiles. However, t-LDP and t-LDP-RC still perform efficiently and are scalable with longer trajectory length.

7 Conclusion

In this paper, we address the challenge of maintaining a balance between data utility and privacy in the use of trajectory data, a critical concern in spatiotemporal applications. Based on the observation that the trajectory data utilization often adopts denoising pre-processing, we devise a variant of Local Differential Privacy that significantly reduces the need for excessive noise injection originally required. Together with adaptable automaton-based mechanisms for efficient trajectory sampling, our methods can achieve privacy, utility, and efficiency.

Acknowledgement. This work is partially supported by National Science Foundation of China (NSFC) under Grant No. U21A20516, 62336003, and 62076017, Beijing Natural Science Foundation No. Z230001, CCF-Huawei Populus Grove Fund, Didi Collaborative Research Program NO2231122-00047, and the Beihang University Basic Research Funding No. YWF-22-L-531. Yuanyuan Zhang, Yi Xu are the corresponding authors of this paper.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. TKDE **17**(6), 734–749 (2005)
2. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., et al.: Geo-indistinguishability: Differential privacy for location-based systems. In: CCS. pp. 901–914 (2013)
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., et al.: Lof: identifying density-based local outliers. In: SIGMOD. pp. 93–104 (2000)
4. Chatzikokolakis, K., Andrés, M.E., Bordenabe, N.E., et al.: Broadening the scope of differential privacy using metrics. In: PETS. pp. 82–102 (2013)
5. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: SIGMOD. pp. 491–502 (2005)
6. Chen, R., Acs, G., Castelluccia, C.: Differentially private sequential data publication via variable-length n-grams. In: CCS. pp. 638–649 (2012)

7. Chen, R., Li, H., Qin, A.K., et al.: Private spatial data aggregation in the local setting. In: ICDE. pp. 289–300 (2016)
8. Cormode, G., Maddock, S., Maple, C.: Frequency estimation under local differential privacy. Proc. VLDB Endow. **14**(11), 2046–2058 (2021)
9. Cunningham, T., Cormode, G., et al.: Real-world trajectory sharing with local differential privacy. Proc. VLDB Endow. **14**(11), 2283–2295 (2021)
10. Du, Y., Hu, Y., Zhang, Z., et al.: Ldptrace: Locally differentially private trajectory synthesis. Proc. VLDB Endow. **16**(8), 1897–1909 (2023)
11. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: FOCS. pp. 429–438 (2013)
12. Dwork, C.: Differential privacy. In: ICALP. pp. 1–12 (2006)
13. Dwork, C., Naor, M., Pitassi, T., et al.: Differential privacy under continual observation. In: STOC (2010)
14. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
15. Gritten, D.: Strava app flaw revealed runs of israeli officials at secret bases. BBC (June 2022), https://www.bbc.com/news/world-middle-east-61879383
16. Gursoy, M.E., Liu, L., Truex, S., et al.: Utility-aware synthesis of differentially private and attack-resilient location traces. In: CCS. pp. 196–211 (2018)
17. Hightower, J., Borriello, G.: Particle filters for location estimation in ubiquitous computing: A case study. In: UBICOMP. pp. 88–106 (2004)
18. Kellaris, G., Papadopoulos, S.: Practical differential privacy via grouping and smoothing. Proc. VLDB Endow. **6**(5), 301–312 (2013)
19. Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: VLDB. vol. 99, pp. 211–222 (1999)
20. Knox, E.M., Ng, R.T.: Algorithms for mining distancebased outliers in large datasets. In: VLDB. pp. 392–403 (1998)
21. Liu, H., Li, T., Hu, R., et al.: Joint representation learning for multi-modal transportation recommendation. In: AAAI. pp. 1036–1043 (2019)
22. Liu, H., Tong, Y., Han, J., et al.: Incorporating multi-source urban data for personalized and context-aware multi-modal transportation recommendation. TKDE **34**(2), 723–735 (2022)
23. Moreira-Matias, L., Gama, J., Ferreira, M., et al.: Predicting taxi–passenger demand using streaming data. ITSC **14**(3), 1393–1402 (2013)
24. Rastogi, V., Hay, M., Miklau, G., et al.: Relationship privacy: output perturbation for queries with joins. In: PODS. pp. 107–116 (2009)
25. Shi, D., Tong, Y., Zhou, Z., et al.: Learning to assign: Towards fair task assignment in large-scale ride hailing. In: KDD. pp. 3549–3557 (2021)
26. Tong, Y., Shi, D., Xu, Y., et al.: Combinatorial optimization meets reinforcement learning: Effective taxi order dispatching at large-scale. TKDE pp. 1–1 (2023)
27. Tong, Y., Zeng, Y., Zhou, Z., Chen, L., et al.: A unified approach to route planning for shared mobility. VLDB **11**(11), 1633–1646 (2018)
28. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: ICDE. pp. 673–684 (2002)
29. Xu, Z., Li, Z., Guan, Q., et al.: Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In: KDD. pp. 905–913 (2018)
30. Yuan, J., et al.: Driving with knowledge from the physical world. In: KDD (2011)
31. Zhang, Y., Ye, Q., Chen, R., et al.: Trajectory data collection with local differential privacy. Proc. VLDB Endow. **16**(10), 2591–2604 (2023)